
Machine Learning, 58, 151–178, 2005


# Clustering Time Series with Clipped Data

ANTHONY BAGNALL ajb@cmp.uea.ac.uk
GARETH JANACEK gj@cmp.uea.ac.uk
*School of Computing Sciences, University of East Anglia, Norwich, UK*

**Editor:** Eamonn Keogh

**Abstract.** Clustering time series is a problem that has applications in a wide variety of fields, and has recently attracted a large amount of research. Time series data are often large and may contain outliers. We show that the simple procedure of clipping the time series (discretising to above or below the median) reduces memory requirements and significantly speeds up clustering without decreasing clustering accuracy. We also demonstrate that clipping increases clustering accuracy when there are outliers in the data, thus serving as a means of outlier detection and a method of identifying model misspecification. We consider simulated data from polynomial, autoregressive moving average and hidden Markov models and show that the estimated parameters of the clipped data used in clustering tend, asymptotically, to those of the unclipped data. We also demonstrate experimentally that, if the series are long enough, the accuracy on clipped data is not significantly less than the accuracy on unclipped data, and if the series contain outliers then clipping results in significantly better clusterings. We then illustrate how using clipped series can be of practical benefit in detecting model misspecification and outliers on two real world data sets: an electricity generation bid data set and an ECG data set.

**Keywords:** clustering time series, clipping

## 1. Introduction

Clustering time series has a wide range of possible applications and has hence attracted researchers from a wide range of disciplines. A survey of time series data mining (including clustering) has recently been published (Keogh & Kasetty, 2003). Time series databases are often extremely large and frequently contain outliers. Much recent research has concentrated on means of speeding up the mining process, commonly through discretisation and/or dimensionality reduction (Lin et al., 2003; Keogh & Pazzani, 2000). This paper demonstrates that the simple process of *clipping* time series reduces memory requirements by a factor of 200 and can speed up fundamental operations on time series by up to a factor of 5. We go on to show that the information discarded by clipping does not significantly decrease the accuracy of clustering algorithms if the series are long enough, and can improve accuracy if the data contains outliers.

Clipping, or *hard limiting*, a time series is the process of transforming a real valued time series $Y$ into a binary series $C$ where 1 represents above the population mean and 0 below, i.e. if $\mu$ is the population mean of series $Y$ then

$$C(t) = \begin{cases} 1 & \text{if} \quad Y(t) > \mu \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Clipped time series require much less memory to store and can be manipulated faster, yet retain much of the underlying structure that characterises the real valued series. This means that if the series are long enough clustering with clipped data is not significantly less accurate than clustering with unclipped data. Clipping makes clustering more robust to outliers. We show that the clusters formed are significantly better with clipped data when there is at least a small probability of the data containing outliers. Additional advantages of using clipped data are that algorithms developed for discrete or categorical data can be employed and that clustering on clipped data can serve as a diagnostic method for outlier and model misspecification detection.

We show these benefits on data from classes of models used in the literature and on real world data. One method that has similarities to our approach is the Symbolic Aggregate Approximation (SAX) transformation described in Lin et al. (2003). SAX is a transformation from real valued series into symbolic series through a two stage process of dimension reduction and discretisation. Clipping could be considered a specific type of SAX transformation with two classes and no dimension reduction. SAX may be of more general use in time series mining than clipping. However, the properties of clustering with clipped series are worthy of study because, firstly, binary data can be more compactly represented and efficiently manipulated and secondly, it is possible to assess their theoretical behaviour. A review of other time series clustering research related to this work can be found in Bagnall et al. (2003).

The key decision in any clustering task is the choice of similarity measure. This choice is governed by the nature of the main purpose of the clustering exercise. There are three main objectives in clustering time series, each of which requires a different approach.

*Type 1 objective: Similarity in Time.*

The first possible objective is to group together series that vary in a similar way on each time step. For example, one may want to cluster share prices of companies to discover which shares change in price together. An obvious approach to measure similarity for this type of clustering is to use a correlation based distance metric, as used by Ormerod and Mounfield (2000) or Euclidean distance on normalised data. There is a large body of work on methods to represent series compactly so that these distance metrics can be applied to large data sets. Transformations used include discrete fourier transforms, wavelets and piecewise aggregate approximation. See Keogh and Kasetty (2003) for an overview.

*Type 2 objective: Similarity in Shape.*

The second possible objective is to cluster series with common shape features together. This may constitute identifying common trends occuring at different times or similar subpatterns in the data. For example, the chartist stock analyst may be interested in grouping shares that have exhibited similar patterns of change independent of when they occured. The two broad approaches to achieving this objective are to either transform the data using techniques such as dynamic time warping (e.g., Chu et al., 2002), or to develop specific algorithms for matching subsequence patterns (e.g., Agrawal, Faloutsos, & Swami, 1993; Aref, Elfeky, & Elmagarmid, 2004).

*Type 3 objective: Similarity in Change.*

The third objective is to cluster series by the similarity in how they vary from time step to time step. For example, a stock analyst may wish to cluster together shares that tend to follow a rise in share price with a fall the next day. The popular approach for this type of objective is to assume some form of underlying model such as a hidden Markov model (e.g., Smyth, 1997) or an ARMA process (e.g., Kalpakis, Gada, & Puttagunta, 2001; Xiong & Yeung, 2002), fit a model to the series and cluster based on similarity of fitted parameters.

In Sections 3 and 6.1 we consider data where the intuitive objective is to group based on similarity in time. In Section 6.2 we cluster an ECG data set where the objective is to identify common shape. In Sections 4 and 5 we use data from models which have a specified autocorellation structure and hence the objective is of the third type.

Our approach to demonstrating the benefits of clipping is to firstly specify a class of model from which the data may arise, secondly to investigate the theoretical properties of the clipped series from these models and thirdly to experimentally demonstrate that there is significant evidence of the benefits of clipping over the class of model. The rest of this paper is organised as follows: In Section 2 we describe the generalised data model and the clustering procedure adopted. We use $k$-means and $k$-medoids to cluster. In (Bagnall et al., 2003; Bagnall & Janacek, 2004); Sections 5 and 4 we demonstrate that that our model fitting and clustering techniques are not the cause of the variation between clipped and unclipped data by achieving results of equivalent accuracy to those published in Kedem (1980), Kalpakis, Gada, and Puttagunta (2001), Xiong and Yeung (2002), Maharaj (2000), Alon et al. (2003). In Section 3 we consider data from polynomial models, examples of which have been used in the clustering of time series in e.g., (Gaffney & Smyth, 2003). In Section 4 we cluster data from Autoregressive Moving Average (ARMA) models. Similar generating models have been used in Kalpakis, Gada, and Puttagunta (2001), Xiong and Yeung (2002), Maharaj (2000). In Section 5 we cluster data generated by Hidden Markov Models (HMM), which have probably been the most popular form of generating model for simulated data sets (e.g., Oates, Firoiu, & Cohen, 1999; Smyth, 1997; Alon et al., 2003; Zhong & Ghosh, 2003; Li & Biswas, 1999). In Section 6 we then demonstrate how using clipped series can be of practical benefit in detecting model misspecification and outliers on two real world data sets: an electricity generation bid data set and an ECG data set. Finally, in Section 7 we present our conclusions.

## 2. Model and clustering procedure

For the simulated data we assume that the series in each true underlying cluster is generated by an underlying model. The model without outliers for cluster $i$ is a random variable denoted $X_i$. One of the objectives of this paper is to demonstrate the robustness in the presence of outliers of using a clipped time series rather than the the raw data for clustering. Hence, we add a further term to model the effect of outliers. A time series is assumed to be generated by a sequence of observations from the model

$$Y_i(t) = X_i(t) + r \tag{2}$$

where $r = s \cdot a \cdot b$. $s$ is a constant, $a \in \{0, 1\}$ and $b \in \{-1, 1\}$ are observations of independent random variables, $A$ and $B$, where $A$ has density $f(a) = p^a(1-p)^{1-a}$ and $B$ has density $f(b) = \frac{1}{2}$. We model a *random shock* effect with $r$, which 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 time series $y$ is a sequence of observations from a model $Y$, $y \leq y(1), \ldots y(t), \ldots y(n) >$. A binary data series is generated by clipping above and below the sample median. If $\phi_y$ is the sample median of the data series $y(t)$, $t = 1, \ldots, n$, then the associated discretised time series, $c$, is defined as

$$c(t) = \begin{cases} 1 & \text{if} \quad y(t) > \phi_y \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

If several series are generated by the same model, we identify the generating process by a subscript $i$ and the series index with a superscript $j$. Thus series $y_i^j$ would be assumed to be the $j$th series from model $Y_i$.

A data set $D$ is parameterised as follows: there are $k$ models of the form given in Eq. (2), each of which generates $l$ time series, thus

$$D = \left\{y_1^1, \ldots, y_1^l, y_2^1, \ldots, y_2^l, \ldots, y_k^l, \ldots, y_k^l\right\}$$

each of the $l \cdot k$ time series is of length $n$ and is sampled at the same points $t = 1, 2, \ldots, n$; $\sigma$ defines the variability of static noise, $s$ the level of random shocks and $p$ the probability of shocks. Given a data set $D$, the problem is then to split the data into distinct sets or clusters, $D_1, D_2, \ldots, D_k$ such that,

$$\forall y_i^j \in D, y_i^j \in D_m \quad \text{if and only if} \quad i = m.$$

### 2.1. *Experimental procedure*

An experiment consists of the following steps: set the parameters; define a model space $\mathcal{M}$; randomly sample the model space to determine the $k$ cluster models; for each cluster model, generate $l$ series of length $n$; cluster the series $u$ times, taking as the best the clustering with the lowest within cluster distance from the centroid; evaluate the accuracy of the best clustering. 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 we 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 given in Halkidi, Batistakis, and Vazirgiannis (2001) 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.

Our interest lies in testing whether the average accuracy over a class of models is significantly different with clipped data. We adopt the following procedure to test this hypothesis. Let $\mathcal{M}$ be the set of all models considered in the experiment, with an instance denoted $m_i$. Let $C$ be the set $\mathcal{M} \times \mathcal{M} \times \cdots \times \mathcal{M}$ of generators of the $k$ cluster model. $\phi_y$ is the population median of the average classification accuracy of the restart $k$-means algorithm ($k$ known, random initial centroids, restart $u$ times) over the space of underlying models $C$ and $\phi_c$ denotes the population median when using the clipped data. $\mu_y$ and $\mu_c$ are the associated population means. Given a random sample of size $v$ from $C$ we wish to test $H_0 : \phi_c = \phi_y$ against the alternative $H_1 : \phi_c < \phi_y$ and $H_0 : \mu_c = \mu_y$ against the alternative $H_1 : \mu_c < \mu_y$.

### *2.2. Data storage*

One of the benefits of using clipped series is that the data can be packed efficiently in integer arrays and manipulated using bit operators. A series of doubles of length $n$ can be stored in an array of $n/64$ integers. This means a clipped series will require over 200 times less memory, a worthwhile reduction particularly in applications that involve very long or very many series. For very large data sets packing the series may make the difference between being able to store the series in main memory and having to access the data from disk. Hence packing clipped series could provide a significant time improvement for clustering algorithms that require the recalculation of models directly from the data at each iteration of the clustering.

## 3. Polynomial models

We generate time series data from polynomial models of the form

$$Y(t) = p(t) + \epsilon$$

where $\epsilon$ is $N(0, \sigma^2)$ and $\sigma$ is constant. We assume the polynomial is of a maximum order $m$, i.e.

$$p(t) = a_0 + a_1 \cdot t + a_1 \cdot t^2 + \cdots + a_m \cdot t^m.$$

$k$ models are generated through sampling a predetermined space of possible models. $l$ series of length $n$ are generated from each model for $t = 0, 0.1, 0.2, \ldots, n/10$. $u$ clusterings are performed on a particular data set and accuracy is measured on the best clustering found. The process of generation and clustering is repeated $v$ times.

Our clustering objective is to group together series with similar variation in time (defined in Section 1 as a type 1 objective), hence we use a correlation based distance metric. The distance between two observed series $y_i$ and $y_j$ is defined as

$$d(y_i, y_j) = \sqrt{2(1 - \rho(y_i, y_j))} \tag{4}$$

where $\rho(y_i, y_j)$ is the correlation between the two series. It was shown by Ormerod and Mounfield (2000) that this distance measure is a metric. In a previous paper (Bagnall et al., 2003) we clustered clipped and unclipped data from two randomly generated linear models (i.e. $k = 2$ and $m = 1$) with the $k$-means algorithm. We showed that the clustering accuracy with clipped data was not significantly worse than clustering with unclipped data and was significantly better if the data contained outliers. In this paper we extend these results by showing that:

- correlation calculations are approximately three times faster with clipped series (Section 3.1);
- the accuracy results hold for $k$-medoids clustering and for larger values of $k$ (Section 3.2); and
- the accuracy results hold for a wider class of polynomial model ($m = 4$) (Section 3.3).

### *3.1. Improved time complexity*

The time complexity for the distance calculation for clipped data can be improved by taking advantage of the efficient storage of binary data described in Section 2.2. The correlation between two clipped series $c_i$ and $c_j$, denoted $r_{ij}$, is simply the sum of the bitwise XOR of the two binary sequences divided by the series length,

$$r_{ij} = \frac{\sum_{t=1}^{n} c_i(t) \oplus c_j(t)}{n}.$$

If binary series are packed into integers we can find the terms in the summation very quickly using bit operators. We can also speed up the operation to sum the bits. Any algorithm to count the bits is $\Omega(n)$. We can however improve the constant terms in the time complexity function by using shift operators to evaluate the integer value of each 8 bit sequence, then using a lookup table to find the number of bits in that integer. Figure 1 demonstrates that calculating the correlations is approximately three times faster with the clipped data even when the series are stored in main memory. Each point in figure 1 gives the time taken (averaged over 100 repetitions) to find all the correlations between 100 series of length $n$. The times include the procedure to clip the data. The median of each series is found using the quick select algorithm, so clipping takes $O(n)$ time on average. This overhead is the major factor in the timing and usually incurred only once. Calculating just the correlations is 5–10 times faster with clipped data.

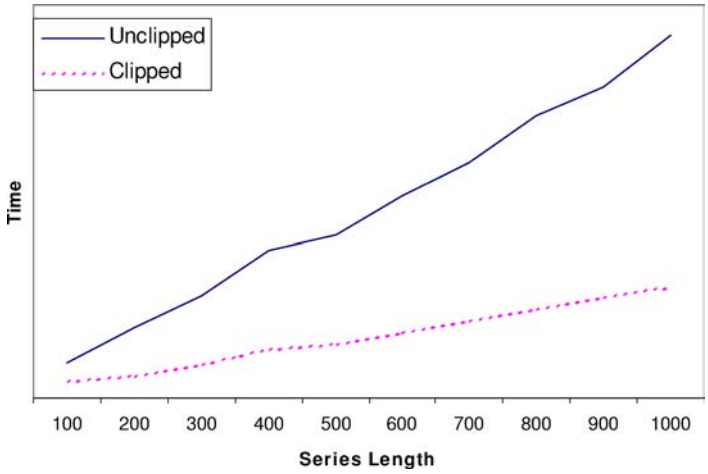


*Figure 1.* Time taken to calculate 100 × 100 correlation matrix with clipped and unclipped data.

### 3.2. *Linear models*

Linear models of the form

$$Y(t) = a_0 + a_1 \cdot t + \epsilon,$$

were randomly sampled for each cluster, where $a_0 \in [-1, 1]$ and $a_1 \in [-0.5, 0.5]$. $l$ series of length 50 were generated from each of the $k$ models selected ($n = 50$). A data set, then, consists of $k \cdot l$ time series, and a sample of 100 data sets were generated ($v = 100$). For each data set, $k$-means and $k$-medoids were run 100 times and the clustering with the lowest within-cluster variation was taken as the best ($u = 100$). The results for $k = 2$, $k = 4$ and $k = 10$, with $k$-means and $k$-medoids are presented in Table 1. The probability of an outlier was set to 0.05.

*Table 1.* Clustering accuracy summary for linear generating models.

| | | Unclipped | Clipped |
|---|---|---|---|
| | | *k-medoids clustering* | |
| $k = 2$, | No outliers | (70.19%/**72.50%**/*16.7*) | (66.81%/**62.50%**/*16.1*) |
| $l = 10$ | Outliers | (64.31%/**62.50%**/*17.4*) | (65.75%/**62.50%**/*15.3*) |
| | | *k-means clustering* | |
| $k = 2$, | No outliers | (83.35%/**85.00%**/*16.7*) | (83.60%/**84.50%**/*16.5*) |
| $l = 10$ | Outliers | (75.00%/**72.50%**/*17.4*) | (82.40%/**82.50%**/*17.2*) |
| $k = 4$, | No outliers | (56.55%/**60.00%**/*11.3*) | (57.85%/**60.00%**/*11.5*) |
| $l = 5$ | Outliers | (53.40%/**52.50%**/*8.0*) | (58.00%/**55.00%**/*10.3*) |

Each cell gives the Mean/**Median**/*Std Dev* over 100 runs.

When $k = 2$ and $l = 10$, with $k$-means we can observe the following.

- We *cannot* reject the null hypothesis that the population means for unclipped and clipped are the same with a paired sample $t$-test at the 1% level (sample means 83.35 and 83.60 respectively). Also, using Wilcoxon's test for matched pairs, we cannot reject the null hypothesis that the population medians are the same (sample medians 85.00 and 84.50).
- We *can* reject the null hypothesis that the population means and medians for unclipped data without outliers and unclipped data with outliers are the same (sample means 83.35 and 75.00 and medians 85.00 and 72.50).
- We *cannot* reject the null hypothesis that the population means and medians for clipped data without outliers and clipped data with outliers are the same (sample means 83.60 and 82.40 and medians 84.50 and 82.50).

Identical conclusions can be drawn for the results from when $k = 4, l = 5$. $k$-medoids clustering displays a similar pattern of results, although the differences are less conclusive. With unclipped data there is still a significant drop in mean and median accuracy when outliers are introduced, and there is no associated decrease with clipped data. We can conclude that clipping data from the class of linear models considered does not decrease the average accuracy of clustering with $k$-means. We can further conclude that clipping increases the average accuracy if the data contains outliers when $k$-means or $k$-medoids are used to cluster.

### *3.3. Higher order models*

The experiments were repeated with data from a more general class of polynomial,

$$Y(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + a_4 \cdot t^4 + \epsilon.$$

A generating model was sampled by first randomly selecting the order, then randomly sampling $a_i$, where $a_i \in [-1, 1]$. Table 2 displays the results. The pattern of performance is identical to that of linear models. The accuracy on clipped data is not significantly less than that on unclipped data, and is significantly better if there are outliers in the data.

*Table 2.* Clustering accuracy summary for higher order polynomial generating models.

| | | Unclipped | Clipped |
|---|---|---|---|
| $k = 2, l = 10$ | No outliers | (82.05%/**86.50%**/*15.3*) | (82.05%/**90.00%**/*15.7*) |
| | Outliers | (75.75%/**75.00%**/*15.9*) | (81.70%/**85.00%**/*15.6*) |
| $k = 4, l = 5$ | No outliers | (60.90%/**60.00%**/*12.2*) | (60.80%/**62.50%**/*13.5*) |
| | Outliers | (56.10%/**55.00%**/*9.8*) | (60.60%/**60.00%**/*11.6*) |

Each cell gives the Mean/**Median**/*Std Dev* over 100 runs.

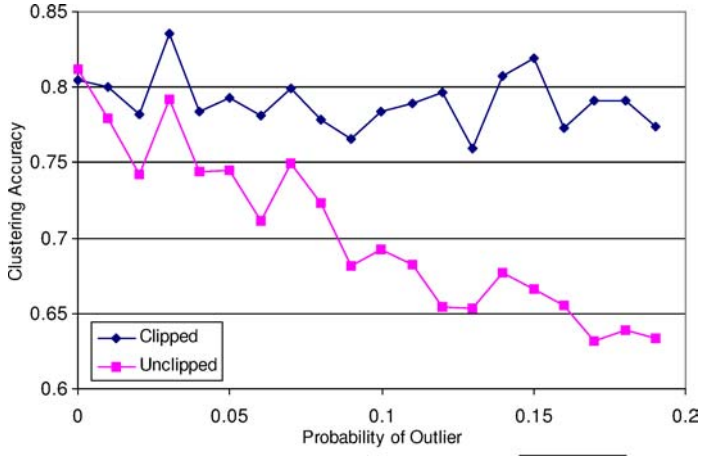


*Figure 2*. Accuracy for clipped and unclipped clustering averaged over 100 runs on random polynomials.

Figure 2 demonstrates how robust clustering on clipped data is to the presence of outliers. As the probability of an outlier increases the clipped accuracy does not decrease from around the 80% mark. This contrasts with the accuracy on the unclipped data, which steadily decreases as the outliers become more common.

In addition to demonstrating the time benefits that may be accrued when using clipped data, the results presented in this Section confirm the conclusions made in Bagnall et al. (2003) and extend them to a wider class of model.

## 4. ARMA models

An ARMA($p, q$) model has the form

$$Y(t) = \phi_1 \cdot Y(t-1) + \cdots \phi_p \cdot Y(t-p) + \varepsilon(t) + \theta_1 \cdot \varepsilon(t-1) + \cdots + \theta_q \cdot \varepsilon(t-q)$$

where $\varepsilon(t)$ are normally distributed random variables with variance $\sigma^2$. ARMA parameters can be estimated from data using the autocorrelations. The data generation procedure is identical to that used in Section 3, except an ARMA model rather than a polynomial model is used. The clustering objective is to group together series which vary from time step to time step in a similar way (defined in Section 1 as a type 3 objective). Because of this we adopt a model based approach and cluster on similarity of fitted model. An AR model is fitted to each series, then clustering is based on the Euclidean distance between parameters. Any invertible ARMA model can be represented as an infinite AR model,

$$\varepsilon(t) = \sum_{j=0}^{\infty} \pi_j Y(t-j).$$

We estimate the AR model for each series by using a standard three stage procedure. Firstly the autocorrelations $\rho_1, \rho_2, \ldots$ up to a fixed maximum lag are calculated (set to 50 in experiments reported). Secondly the partial fitted models and hence the partial autocorrelations are found by solving the Durbin-Levinson recursion. These use the fact that the correlation matrix is Toeplitz to create a computationally efficient iteration. If the $i$th autoregressive term of the model fit to $j$ parameters is denoted $\phi_{i,j}$ and $\phi_{1,1} = \rho_1$ then the Durbin-Levinson recursion is

$$\phi_{k,k} = \frac{\rho_k - \rho_{k-1}\phi_{1,k-1} - \rho_{k-2}\phi_{2,k-1} - \cdots - \rho_1\phi_{k-1,k-1}}{1 - \rho_{k-1}\phi_{k-1,k-1} - \rho_{k-2}\phi_{k-2,k-1} - \cdots - \rho_1\phi_{1,k-1}}$$

Finally the model that minimizes the Akaike information criteria (AIC) is chosen (Akaike, 1981). The AIC for a model with $k$ AR terms fitted to a series of length $n$ is defined as

$$AIC = \ln \hat{\sigma}^2 + 2 \cdot \frac{k}{n}$$

In Bagnall & Janacek (2004) we demonstrate that using this procedure does not adversely affect the efficiency of the estimates of the model parameters by repeating the experiments described in Section 4 of Kedem (1980).

### *4.1. Asymptotic properties of clipped series*

Kedem (1980) and Kedem and Slud (1991) derived several useful links between the original and the clipped series. We use their results to demonstrate that, under certain assumptions, if the unclipped data is from an ARMA model then the clipped series is also from an ARMA model.

If we assume that the unclipped series $y$ is both Gaussian and stationary to second order, we can use the bivariate normal distribution to compute probabilities. We assume there are no outliers in the data ($p = 0$), and we denote $\rho_Y(r)$ the autocorrelation of lag $r$ for an unclipped model $Y$ and $\rho_C(r)$ the autocorrelation of lag $r$ for a clipped model $C$. It is not difficult to show that

$$P[C(r+s) = 1 \quad \text{and} \quad C(s) = 1] = \frac{1}{4} + \frac{1}{2\pi}\sin^{-1}\{\rho_Y(r)\}.$$

It follows that the autocorrelations of the clipped series, $\rho_C(k)$, are given by

$$\rho_C(k) = \frac{2}{\pi}\sin^{-1}\rho_Y(k) \qquad k = \cdots -2, -1, 0, 1, 2, \ldots \tag{5}$$

Since the binary series gives the runs in the time series, we have, from Eq. (5), a link between the probability of a run and the correlation structure of the original process. Using

a Taylor expansion we have

$$\rho_C(k) = \frac{2}{\pi}\left\{\rho_Y(k) + \frac{1}{6}\rho_Y^3(k) + \frac{3}{40}\rho_Y^5(k) + O\left(\rho_Y^7(k)\right)\right\}$$

which implies that the two series will have similar correlations but that those from the clipped series will have smaller absolute values. In fact we can go further and can show that with the Gaussian assumptions

$$E[C(t)Y(t+k)] = \frac{1}{2}E[Y(t+k)|C(t)] = \frac{1}{\sqrt{2\pi}}\rho_Y(k).$$

Now define

$$e(t) = C(t) - \frac{1}{\sqrt{2\pi}}Y(t)$$

since $E[e(t)Y(t+k)] = 0$ we have a process

$$C(t) = \frac{1}{\sqrt{2\pi}}Y(t) + e(t)$$

where $E[e(t)] = 0$. This implies that if the original series can be written in autoregressive form, say

$$Y(t) + \phi_1 Y(t-1) + \phi_2 Y(t-2) + \cdots + \phi_p Y(t-p) = \varepsilon(t)$$

then the clipped series has the form

$$\begin{aligned} &C(t) + \phi_1 C(t-1) + \cdots + \phi_p C(t-p) \\ &\quad = \frac{\varepsilon(t)}{\sqrt{2\pi}} + e(t) - \phi_1 e(t-1) - \cdots - \phi_p e(t-p). \end{aligned}$$

Here of course the errors are correlated. We can deduce that, given the covariance matrix of the errors $e(t)$, we can produce a linear transformation which will give us uncorrelated errors. Hence we deduce *a linear ARMA model for original series implies a linear ARMA model for the clipped series*.

### *4.2. Improved time complexity*

The autocorrelations can be calculated much faster with clipped data than with unclipped. An autocorrelation requires the calculation of the sum of the product of a lagged series with

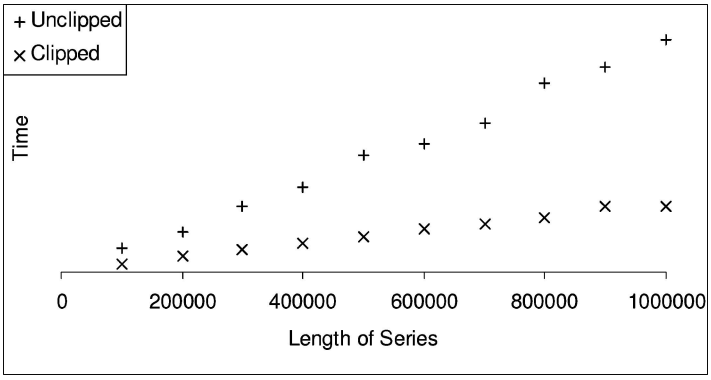


*Figure 3*. Time taken to find autocorrelations upto lag 50, averaged over 100 runs.

the original, i.e.

$$R_i = \sum_{t=i}^{n} C(t) \cdot C(t-i).$$

The autocorrelation calculations for a clipped series represented as arrays of binaries requires fewer operations than that of an unclipped series, and as with the correlation function, the autocorrelations can be calculated approximately three times faster with clipped data. This is demonstrated in figure 3. Each data represents the average time to find the 50 autocorrelations for clipped and unclipped data. The times include the time to normalise and to clip the data.

Packing the data also offers the potential for improving the performance of the clustering algorithm. For example, the mean calculation for $k$-means becomes the problem of counting bits in subseries, which can be done more efficiently than the equivalent calculation with the unclipped data using masks and bit count lookup tables. Calculating distance measures can be made more efficient by similar mechanisms.

### *4.3. AR(1) models*

AR(1) models of the form

$$Y(t) = \phi \cdot Y(t-1) + \varepsilon(t)$$

were sampled for each cluster, where $\phi_i \in [-1, 1]$. Experiments with fixed AR(1) models were reported in Kalpakis, Gada, and Puttagunta (2001) and Xiong and Yeung (2002). We reproduced equivalent experiments, the results of which are presented in Table 3. We observe the following: both $k$-means and $k$-medoids achieve an accuracy on unclipped data with no outliers within the range of results reported in Kalpakis, Gada, and Puttagunta (2001) and Xiong and Yeung (2002); there is no significant difference between $k$-means

*Table 3.* Two cluster AR(1) model with $\phi_1 = 0.30$ and $\phi_2 = 0.50$. Accuracy using $k$-means and $k$-medoids, $l = 10, n = 256$.

| | $k$-Means | | $k$-Medoids | |
|---|---|---|---|---|
| | Unclipped (%) | Clipped (%) | Unclipped (%) | Clipped (%) |
| No Outliers | 95 | 81 | 93 | 80 |
| Outliers ($p = 0.01$) | 57 | 81 | 58 | 81 |

and $k$-medoids; outliers significantly decreases the accuracy for unclipped data but not for clipped series; and accuracy on clipped data is significantly worse with no outliers. Apart from the last observation, all these points support our general hypothesis about the benefit of clipping.

The results given in Section 4.1 imply that clustering accuracy should improve as the series get longer. To investigate whether this is in fact observable, we sample the space of all possible stationary AR models. A model for each cluster was randomly sampled from the space of possible AR(1) models where $\phi_i \in (-0.5, 0.5)$. Figure 4 shows the average accuracy with clipped and unclipped series for $k$-means clustering with $k = 2, l = 10, u = 100, v = 100$ for increasing values of $n$. Some information is of course lost with clipping, hence for lower values of $n$ the clipped accuracy is lower. However, as $n$ increases the gap reduces, until the length of the series is such that both methods solve the problem perfectly. This contrasts to the situation where there are outliers in the data. Figure 5 shows the results for the same experiment, but with outliers added to the data. It clearly shows that the difference in accuracy between clipped and unclipped is maintained as $n$ increases. This implies that if the series are long enough, clipping will not adversely effect clustering accuracy, and will actually improve accuracy if there is a small probability of the data containing outliers.

With no outliers, the mean level of accuracy on clipped data is lower but the median is the same. This is indicative of a general trend we have observed with clipped series: with

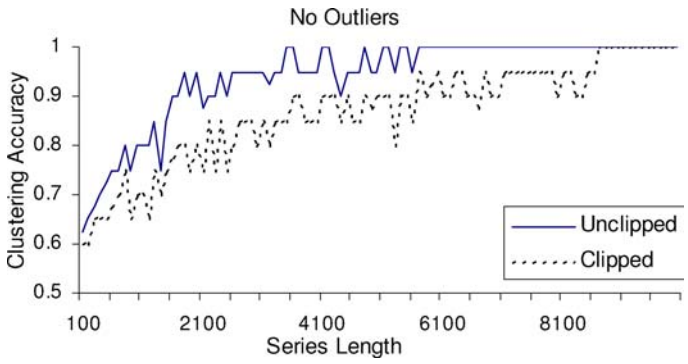


*Figure 4.* Clustering accuracy for increasing $n$ with no outliers in the data.

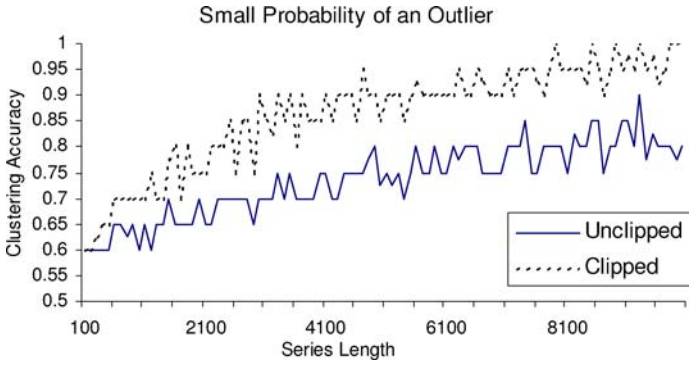


*Figure 5*. Clustering accuracy for increasing $n$ with outliers in the data.

relatively short data clipping occasionally results in a very bad clustering that lowers the average. However, on most problems clipped data does as well. When the probability of outliers is low, the average clustering accuracy using unclipped data is significantly worse. As the probability of a data being an outlier increases the accuracy on the unclipped data decreases more rapidly than the accuracy on the clipped data, demonstrated by the positive slope of both graphs in figure 6. Thus for two classes of data from AR(1) models, $k$-means clustering based on the Euclidean distance between fitted parameters and $n = 256$, clipping the data does not significantly reduce the median accuracy of clustering when there are no outliers and significantly increases the accuracy even when the probability of an outlier is small. To demonstrate that this result is not an artifact of the clustering algorithm we repeat the experiments using the $k$-medoids algorithm. Table 4 shows the results for clipped and unclipped data. Although $k$-medoids performs marginally better on this problem, the pattern of performance is the same: when there are no outliers the mean for unclipped data is higher but the medians are approximately the same; when the probability of a data being an outlier is 0.01 the average (both mean and median) clustering is significantly better with the clipped data.

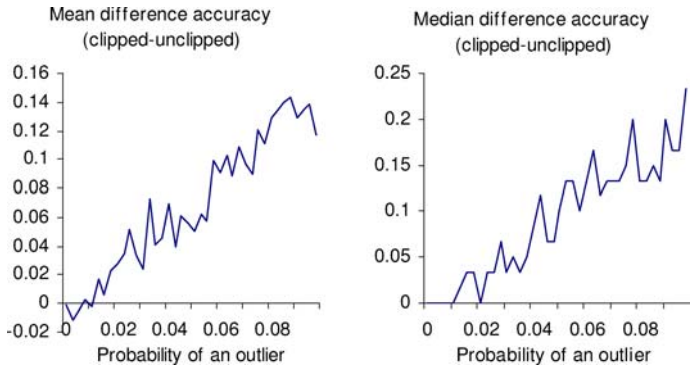


*Figure 6*. Difference in clustering accuracy between the clipped and unclipped series for varying values of $p$.

*Table 4*. Accuracy using $k$-medoids to cluster clipped data from random AR(1).

| | Unclipped | Clipped |
|---|---|---|
| No outliers | (91%/**100%**/*16.4*) | (87%/**97%**/*17.6*) |
| Outlier $p = 0.01$ | (80%/**83%**/*21*) | (86%/**97%**/*18.2*) |

In roman—mean. In bold—median. In italic—Standard deviation.

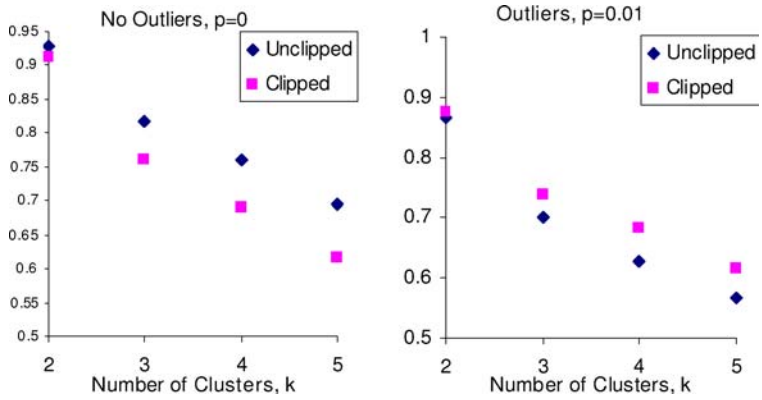


*Figure 7*. Clustering accuracy for alternative values of $k$, with no outliers (left) and a small probability of outlier (right).

To demonstrate that the result is observable in harder clustering problems, the experiments were repeated with larger values of $k$. Figure 7 shows the median accuracy for clipped and unclipped data for $p = 0$ on the left hand graph and $p = 0.01$ on the right. For each cluster we are generating 15 series of length 256 (i.e. $l = 15$ and $n = 256$). Since we keep $l$ and $n$ constant, the accuracy of clustering decreases for both clipped and unclipped as $k$ increases. This achieves the desired result of presenting progressively harder clustering problems. Figure 7 demonstrates that clustering on the clipped data improves the accuracy in the presence of outliers. With no outliers, clustering with clipped data is less accurate, but this gap in accuracy reduces as $n$ increases.

### *4.4. ARMA models*

Data series for each cluster were generated from a randomly selected ARMA model. The possible model structures were in the range ARMA(1, 0) and ARMA(0, 1) to ARMA(2, 2). The parameters were randomly selected in the range $[-1, 1]$. We are primarily interested in clustering stationary ARMA models because for long series non-stationary models quickly increase or decrease to a point where numeric errors can occur. To make sure the random model generated is stationary we perform a simple test where a series is discarded if a value above or below a certain threshold is observed.

*Table 5.* Clustering accuracy and counts on random ARMA models, $k = 2$. Each count is the number of times one technique outperformed the other.

| | Accuracy | | Counts | |
|---|---|---|---|---|
| | Unclipped (%) | Clipped (%) | Unclipped | Clipped |
| No Outliers | 96.75 | 96.68 | 25 | 21 |
| Outlier $p = 0.01$ | 94.23 | 95.70 | 12 | 29 |

10 series of length 1000 were generated from each class. The mean classification accuracy for clipped and unclipped data with and without outliers is shown in Table 5, averaged over 200 experiments.

Tests on the data show that firstly, the average accuracy with clipped data is not less than the average accuracy with unclipped data and secondly, when there are outliers in the data the average accuracy with clipped data is significantly higher (see figure 8).

To test whether these benefits of clipping are still present on harder problems we generated data with the number of clusters, $k$ in the range 2 to 9. Figure 9 shows the accuracy for clipped and unclipped data with $n = 400$ and no outliers. Figure 10 shows the results from the same experiment with the probability of an outliers equal to 0.01. $n$ was set to 400 as it was found to be a level at which the clipped accuracy was not signifcantly less until the number of clusters exceeded 6. Figures 9 and 10 demonstrate that: the clustering problem becomes harder with the number of clusters; with no outliers clipping does not decrease the accuracy compared to unclipped data even for harder problems up to six clusters; and if outliers are present, clipping increases the accuracy.

In this section we have demonstrated that for data from ARMA models, clustering with clipped data means that:

– less memory and processor time are required to cluster the time series;
– if the series are long enough, clustering with clipped series is at least as accurate as clustering with the raw data; and

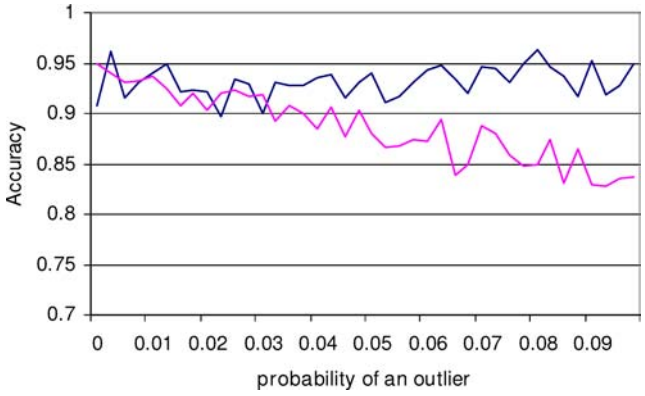


*Figure 8.* Accuracy for increasing probability of an outlier.

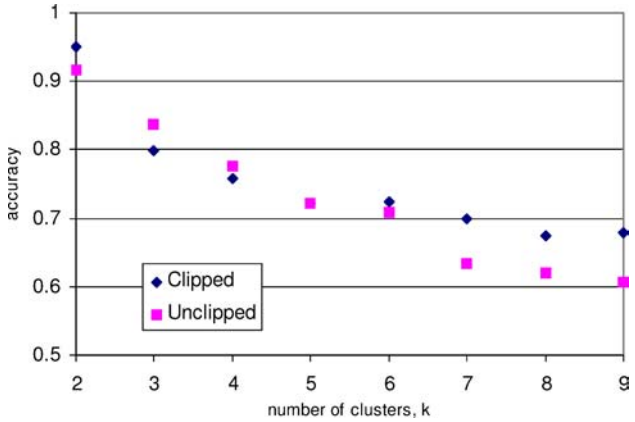


*Figure 9.* Mean accuracy for data with no outliers.

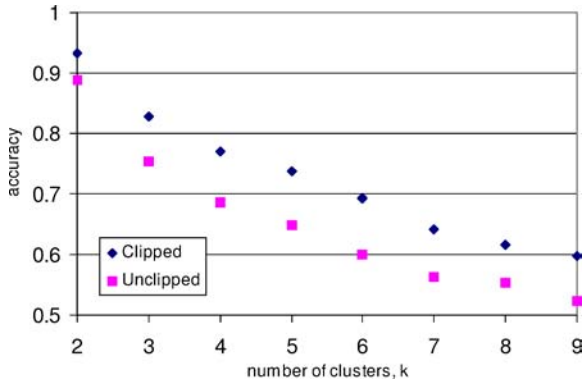


*Figure 10.* Mean accuracy for data with probability of an outlier 0.01.

– if outliers are present in the data, clustering with clipped series is more accurate than clustering with the raw data.

## 5. HMM models

Using the standard notation, we assume that any data is a series of observations from one of $k$ first order HMM. A first order HMM consists of $N$ states, $S = \{S_1, S_2, \ldots, S_N\}$, independent generating densities for each state $f_1, f_2, \ldots, f_N$, a transition distribution $A$ where $a_{ij}$ is the probability of moving from state $i$ to state $j$ and initial state distribution $\pi$. The generating densities are often assumed to be normal, hence a parameterised model is often expressed with the notation

$$\lambda = \{\pi, A, \mu_j, \Sigma_j\} \quad 1 \leq j \leq N.$$

$m$ series are generated from $k$ different HMM models. The clustering problem is to group together the series that are most likely to have been generated from the same model. Because the objective is to group together series that vary from time step to time step in similar ways, we adopt a model based approach. The most common approach to clustering data from HMM is to view the overall mixture model as a single "composite" HMM and use an iterative clustering method such as $k$-means (Alon et al., 2003) or EM (Smyth, 1997) to assign cluster membership and a fitting technique such as the Baum-Welch to find the best composite model given a membership. Our approach is to follow the same procedure as used in Section 4.4, i.e. we fit an ARMA model to each series and cluster on the ARMA parameters. Our justification for doing this is that firstly, perhaps rather surprisingly, it works. In Section 5.2 we show that $k$-means on the ARMA parameters clusters as accurately as the $k$-means algorithm used by Alon et al. (2003). Secondly, we can deduce some properties of the autocorrelation function of clipped data from HMM (see Section 5.1), and this result is applicable only if using a technique based on the autocorrelations. Our aim is to examine the effects of clipping rather than find the best method of clustering HMM models. The fact that ARMA fitting works on data generated from the class of models considered is interesting and requires further investigation, but we make no claims as to its suitability to a wider class of HMM. In Section 5.3 we show that for data from randomly generated HMM, as with data from polynomial and ARMA models, clipping does not significantly decrease the accuracy of clustering if the series are long enough and significantly increases accuracy if the data contains outliers.

### *5.1. Properties of data from HMM*

Our approach to demonstrating that we will be able to cluster as well with clipped series from HMM is to demonstrate that for the cases we consider we shall see that they are the same as AR models, at least at the correlation level. Suppose we have a time series $Y_t$ which is generated by an underlying Markov chain $M_t$. We can think of a probability

$$\pi(y_i) = p[Y_t = y \mid M_t = i]$$

In many cases $Y(t)$ takes fixed values, as above but we can just as easily think of the random variable taking the identity of another random variable at time $t$. Thus $Y_t$ could be equivalent to $U_i$ depending on $M_t$.

As we have a Markov chain we can consider the transition probabilities

$$p_{ij} = p[M_t = j \mid M_{t-1} = i]$$

which are elements of the transition matrix $A$. We can now condition on the possible states to give

$$E[Y_t] = \sum_{i=1}^{m} E[Y_t \mid M_t = i]p[M_t = i].$$

It is convienient to have a symbol for the unconditional probabilities $p_{\bullet i} = p[M_t = i]$. Also we can condition to find the expectations of cross terms

$$E[Y_t Y_{t+k}] = \sum_i \sum_j E[Y_t Y_{t+k} \mid M_t = i, M_{t+k} = j] p_{\bullet i} p_{ij}(k) \tag{6}$$

where $p_{ij}(k)$ is the $k$ step transition probability.

For convenience we assume that

- $E[Y_t]$ takes values of the form $\mu_i$.
- var$[Y_t]$ takes values of the form $\sigma_i^2$.
- $E[Y_t^2]$ takes values of the form $d_i$.

Thus

$$E[Y_t] = \sum_{i=1}^{m} E[Y_t \mid M_t = i] p[M_t = i] = \sum_{i=1}^{m} \mu_i p_{\bullet i}$$

while

$$E\left[Y_t^2\right] = \sum_{i=1}^{m} E\left[Y_t^2 \mid M_t = i\right] p[M_t = i] = \sum_{i=1}^{m} d_i p_{\bullet i} \sum_{i=1}^{m} \left( \sum_{i=1}^{m} \mu_i p_{\bullet i} \right)^2 (d_i) p_{\bullet i}$$

We now assume that

$$E[Y_t Y_{t+k} \mid M_t = i, M_{t+k} = j] = 0 \quad \text{when } i \neq j$$

which follows from the structure of our process. Thus

$$\begin{aligned} E[Y_t Y_{t+k}] &= \sum_i \sum_j E[Y_t Y_{t+k} \mid M_t = i, M_{t+k} = j] p_{\bullet i} p_{ij}(k) \\ &= \sum_j \left(\sigma_j^2 + \mu_j^2\right) p_{\bullet i} p_{ij}(k) \end{aligned}$$

and hence the covariances are

$$\operatorname{cov}[Y_t Y_{t+k}] = \sum_j \left(\sigma_j^2 + \mu_j^2\right) p_{\bullet i} p_{ij}(k) - \left( \sum_{i=1}^{m} \mu_i p_{\bullet i} \right)^2$$

Now we have in this equation the $k$-step transition probabilities $p_{ij}(k)$ which, because we have a Markov chain, we get by powering the transition matrix. Using some linear algebra

we can hence deduce these $k$ step transitions are made up of linear combinations of powers of the eigenvalues of the transition matrix $A$. Thus

$$p_{ij}(k) = \sum_i \alpha_i w_i^k$$

where the $\alpha_i$ are constants and the $w_i$ are the eigenvalues.

A consequence of this is that we can write the autocovariances of our series as

$$\text{cov}[Y_t Y_{t+k}] = \alpha_0 + \sum_i \alpha_i w_i^k$$

Now we also know that an AR time series model

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

has autocovariances which satisfy

$$\gamma(k) = \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2) + \cdots + \phi_p \gamma(k-p)$$

and in consequence

$$\gamma(k) = \sum_{i=1}^{p} \beta_i \lambda_i^k$$

where the $\beta_i$ are constants and the $\lambda_i$ are the roots of

$$z^p = \phi_1 z^{p-1} + \phi_2 z^{p-2} + \cdots + \phi_p$$

Of course some of these roots may be a root of unity. We thus conclude that the Hidden Markov model will have autocovariances which are the same as for an AR(p) model, and hence we would expect that clustering methods based on fitting AR models would perform well on data generated by hidden markov models.

### 5.2. *Fixed HMM models*

In Smyth (1997) and Alon et al. (2003) data from two HMM which differ only in the transition matrix is clustered. For both HMM $N = 2$, both generating densities are Gaussian with $\sigma = 1$, with $f_1 = N(0, 1)$ and $f_2 = N(\mu, 1)$ where $\mu \in [1, 3]$. The two transition matrices are

$$A^{(1)} = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}$$

*Table 6*. Clustering accuracy of $k$-means on clipped and unclipped HMM over 50 runs, no outliers. Integer entries are the counts of the number of times clustering accuracy exceeded 90%.

| Model mean $\mu_2^{(2)}$ | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 | 2.5 | 2.75 |
|---|---|---|---|---|---|---|---|---|
| | | | | *Counts of above 90% accuracy (high count)* | | | | |
| $k$-means with B-W | 0 | 0 | 0 | 0 | 21 | 29 | 30 | 44 |
| EM with B-W | 0 | 0 | 0 | 1 | 22 | 46 | 49 | 47 |
| ARMA (Y) | 0 | 1 | 4 | 13 | 22 | 38 | 44 | 49 |
| ARMA (C) | 0 | 0 | 3 | 6 | 17 | 32 | 46 | 48 |
| | | | | *Average accuracy* | | | | |
| ARMA (Y) | 70.3% | 78.5% | 84.5% | 88.9% | 90.8% | 94.0% | 95.8% | 96.8% |
| ARMA (C) | 65.6% | 69.4% | 78.3% | 84.5% | 89.0% | 92.7% | 96.0% | 97.3% |

50 series of length 200 are generated from each HMM. The first model has generating means $\mu_1^{(1)} = 0$ and $\mu_1^{(2)} = 1$. The second model has generating means $\mu_2^{(1)} = 0$ and $\mu_2^{(2)} \in [1, 3]$. In Alon et al. (2003) the authors conclude that $k$-means performs similarly to EM, but that EM does better when there is more overlap in the models. Table 6 shows the results presented in Alon et al. (2003) along with the results for clustering from ARMA data with $k$-means. Firstly, $k$-means ARMA clustering on unclipped data is approximately the same as $k$-means with Baum-Welch. Although the ARMA clustering finds more over 90% when $\mu = 1.75$, we cannot reject the hypothesis that the distribution of number correct is the same with a Chi-test for independence at the 0.05 level (after grouping the results for $\mu \leq 2$ together to get an adequate number of cells).

The fact that the ARMA clustering does as well as Baum-Welch based clustering on this data reinforces the result from Section 5.1 that data from HMM can be well aproximated by fitted AR models. We continue to use it to demonstrate the benefits of clipping on these models and other, randomly generated, models in Section 5.3.

Table 6 also shows the number above 90% accuracy found with clipped data. The clipped high counts are not significantly different to the unclipped ARMA results or those reported in Alon et al. (2003). The evidence to suggest that clipping does not decrease the clustering accuracy is reinforced by the mean accuracies also given in Table 6. Table 7 demonstrates that if there are outliers in the data then clipping results in a better clustering.

*Table 7*. ARMA $k$-means clustering on clipped and unclipped data with probability of an outlier 0.01.

| Model mean $\mu_2^{(2)}$ | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 | 2.5 | 2.75 |
|---|---|---|---|---|---|---|---|---|
| | | | | *Counts of above 90% accuracy (high count)* | | | | |
| ARMA (Y) | 0 | 0 | 0 | 0 | 5 | 14 | 15 | 30 |
| ARMA (C) | 0 | 0 | 0 | 4 | 21 | 34 | 40 | 48 |
| | | | | *Average accuracy* | | | | |
| ARMA (Y) | 63.7% | 68.35% | 74.20% | 81.05% | 82.65% | 86.90% | 88.55% | 91.95% |
| ARMA (C) | 64.15% | 70.15% | 77.25% | 83.05% | 89.90% | 92.30% | 94.30% | 96.65% |

### 5.3. *Random HMM models*

To test the effects of clipping on a general class of HMM, we generated random HMM of the same form to that used in Section 5.2 by randomly generating transition matrices for each cluster. The means of the two normal distributions were set to 0 and 2. For each parameter setting we randomly generated 200 models then ran $k$-means 100 times (i.e. $v = 200$, $u = 100$). The results for $k = 2$ are summarised in Table 8. For random models, when $n = 200$ unclipped clustering gives significantly better average accuracy. When $n$ is increased to 1000 the mean accuracy for unclipped data is still higher (89.4% for unclipped against 86.4% for clipped). However, the medians are much closer, and the median of the series of differences is now 0. Unclipped data is still performing better, achieving 100% accuracy on 95 cases as opposed to the 83 instances when the problem was solved with clipped data. This small, but significant, discrepancy may be a result of using ARMA fitting rather than Baum-Welch. The benefits of clipping are clear, however, when outliers are introduced into the data. For the cases when $n = 200$ and $n = 1000$ the median and mean accuracy is significantly better for clipped data than for unclipped.

Figure 11 shows the accuracy using clipped and unclipped series on data with and without outliers for multiple cluster problems. The pattern of difference is the same as observed with data from other models, in that as the problem gets harder the difference between clipped and unclipped increases, with unclipped doing better on outlier free data and clipped doing better on contaminated data sets.

*Table 8.* Clustering accuracy of $k$-means on clipped and unclipped data from randomly generated HMM over 200 runs.

| Parameters | Clipped | Unclipped | Clipped-unclipped |
|---|---|---|---|
| $N = 200$, $p = 0$ | (87.5%/**82.7%**/*16.5*) | (92.5%/**86.2%**/*15.9*) | (−2.5%/−**3.5%**/*8.5*) |
| $N = 200$, $p = 0.02$ | (85.0%/**81.7%**/*16.1*) | (80.0%/**79.2%**/*15.2*) | (2.5% /**2.5%**/*9.5*) |
| $N = 1000$, $p = 0$ | (96.3%/**86.3%**/*16.4*) | (97.5%/**89.4%**/*14.9*) | (0.0%/−**3.1%**/*7.5*) |
| $N = 1000$, $p = 0.02$ | (95.0%/**88.3%**/*15.2*) | (87.5%/**82.4%**/*16.0*) | (5.9%/**5.0%**/*8.6*) |

In roman—median. In bold—mean. In italic—st.dev.

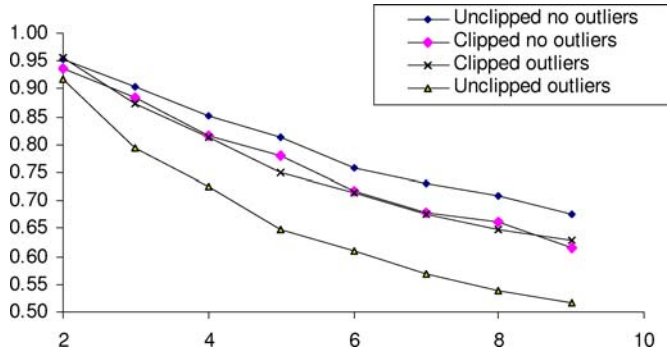


*Figure 11.* Accuracy on multi cluster HMM problems.

## 6. Real world data

### *6.1. Electricity generation bid data*

The UK market in electricity involves a daily market process of bids and offers for the right to generate and supply electricity. Electricity markets cannot operate like traditional markets as there is a necessity to match generation to demand. The UK market structure fundamentally changed in 2001 with the introduction of the New Electricity Trade Agreement (NETA). The structures and mechanisms used to make sure the system remains stable are based on the percieved effect on bidding behaviour. See Bagnall and Smith (2000) for an overview. The degree to which behaviour has changed under the new system is of great interest. However, in this paper we cluster using bid data generated prior to NETA to demonstrate the benefits of clipping. A comprehensive case study of clustering bid data is an area for future research.

Prior to NETA, the market involved generating units placing daily bids for the right to generate electricity. Each unit belongs to a generating station, and each station is owned by a particular company. Our aim is to group together units with bidding series that appear to have followed the same bidding strategy. We are interested in clustering bidding series that vary on the days included in the study period in the same way, thus either a correlation distance metric or a Euclidean measure on normalised data is appropriate. If our objective were to group together units that altered their bid based on the previous days bidding patterns then then an ARMA or HMM model approach may be appropriate. If instead we wished to group together units that had exhibited similar global strategies a DTW approach would be more useful. For example, a general strategy of "*Bid high after a unit has made a large profit*" would probably only be detected if the objective were to cluster based on similarity of shape. However, detecting that a generator was using its market strength to manipulate the market by making some units bid low while others bid high (a strategy known to have been used and commonly called *straddling*) would be more likely with an objective of clustering based on similarity in time.

Our premise is that generating units from the same power station will follow a similar strategy. To demonstrate how clustering can be used to test this premise, bid data from three power stations with 5, 6 and 3 units each respectively were taken from bids for the period 1995–1996 (547 days of data). Figure 12 shows the time series of bids for three units from each station.

*k*-means clustering with Eulcidean, Correlation and ARMA parameter distances was performed on this data. Accuracy was based on the number of units assigned to the correct generating station. Table 9 presents the results on the raw and normalised data. Several observations can be made from this table.

1. Euclidean distance on the raw data and ARMA modelling perform poorly on this problem. This was expected and emphasizes the importance of using the correct metric for the clustering objective.
2. For the three correct procedures (correlation and Euclidean on normalised data), unclipped data performs better than clipped data, but the difference is small.

*Table 9.* Clustering accuracy of $k$-means on electricity data set.

| | Raw data | | Normalised data | |
|---|---|---|---|---|
| | Unclipped (%) | Clipped (%) | Unclipped (%) | Clipped (%) |
| Euclidean | 57 | 86 | 100 | 93 |
| ARMA | 51 | 71 | 54 | 50 |
| Correlation | 100 | 86 | 100 | 93 |

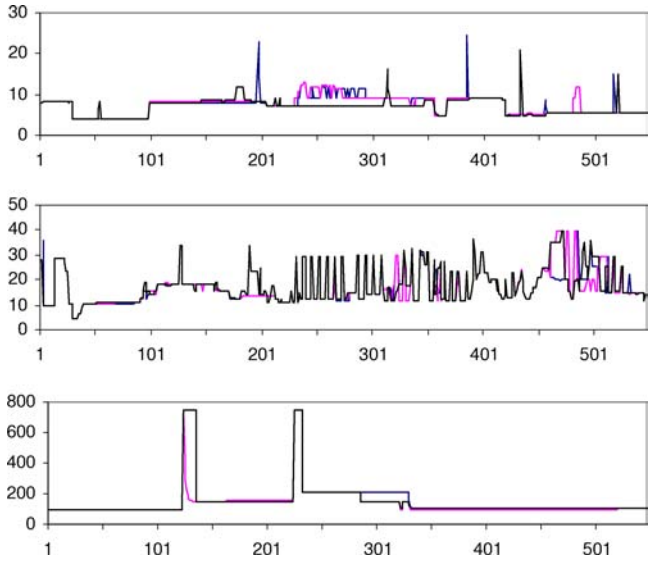


*Figure 12.* Bid series from units from three power stations. The top is a coal powered station with 5 units, the middle is a gas powered station with 6 units and the bottom is a gas turbine station with 3 units.

3. The accuracy of clipped data for Euclidean and ARMA were higher when clustering was performed on the raw data. This demonstrates the potential for using clustering with clipped data to detect model misspecificiation.

Generating units do not have to bid every day. For example the unit may be out of service or the company may decide it is simply not economic to generate on a given day. Our data was collected from the National Grid Company, and the data policy was to record a missing value as $-9999$. We have preprocessed the data shown in figure 12 by smoothing to the 20 day average. However, it would be easy for an inexperienced analyst to not notice these outliers. Clipping the data offers the potential to detect anomalous values. To illustrate this, we added three outlier values of $-9999$ to the data and repeated the clustering. The results shown in Table 10 clearly demonstrate that clustering is better with clipping with these outliers. We also note that the clipped results are very similar to those with the clean data given in Table 9, but accuracy on unclipped data is much lower for the correlation and normalised Euclidean distance.

*Table 10.* Clustering accuracy of $k$-means on on electricity data set with three outliers.

| | Raw data | | Normalised data | |
|---|---|---|---|---|
| | Unclipped (%) | Clipped (%) | Unclipped (%) | Clipped (%) |
| Euclidean | 57 | 86 | 57 | 93 |
| ARMA | 57 | 71 | 50 | 50 |
| Correlation | 71 | 86 | 57 | 93 |

### 6.2. *ECG data set*

In a previous study (Bagnall & Janacek, 2004) we clustered an ECG data set used in Kalpakis, Gada, and Puttagunta (2001) and Xiong and Yeung (2002). A further ECG data set is available from (UCR TSDM). It consists of 20 series of 2000 measurements from the MIT-BIH Arrhythmia Database. 10 series are of healthy patients and 10 from unhealthy. An example of two series from each cluster is shown in figure 13.

We use this data set to demonstrate the benefits of clipping and to reinforce our observations about the relationship between clustering objective and the correct technique. Similar ECG are not defined by how they vary together on a given time step (objective 1) or necessarily how a series changes from one step to another (objective 3). It is clear that the objective of clustering ECG data is to group together series of similar shape (objective 2). A technique based on clustering on subsequence similarity or some transformation such as Dynamic Time Warping will probably give the best results. This is demonstrated by the results for clustering with $k$-means and $k$-medoids with three inappropriate measures shown in Table 11. Note that for two of the methods the clipped accuracy is higher than the unclipped (Euclidean with $k$-medoids and correlation with $k$-means) even though there are no outliers in the data. This demonstrates how clipping can help in the detection of an incorrect problem specification.

To show that clipping can achieve equivalent accuracy, we performed a crude form of transformation of the data by shifting each series so it starts with a peak. This transformation

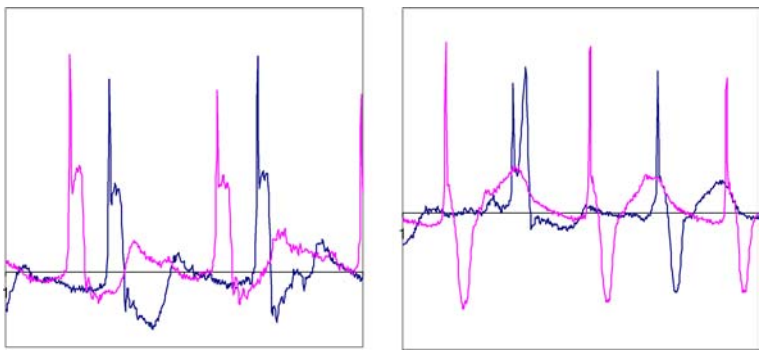

*Figure 13.* Segments from ECG data. The series on the left are from healthy patients.

*Table 11*. Clustering accuracy on the ECG data.

| | Unclipped | | Clipped | |
|---|---|---|---|---|
| | $k$-means (%) | $k$-medoids (%) | $k$-means (%) | $k$-medoids (%) |
| Euclidean | 60.0 | 60.0 | 60.0 | 67 |
| ARMA | 67.8 | 69.2 | 60.2 | 60.0 |
| Correlation | 60.0 | 59.8 | 65.0 | 60.3 |

*Table 12*. Clustering accuracy on the Shifted ECG data.

| | Unclipped | | Clipped | |
|---|---|---|---|---|
| | $k$-means (%) | $k$-medoids (%) | $k$-means (%) | $k$-medoids (%) |
| Euclidean | 100.0 | 100.0 | 100.0 | 100 |
| ARMA | 55.0 | 58.7 | 60.0 | 60.2 |
| Correlation | 100.0 | 85.3 | 100.0 | 66.3 |

works for this particular data and is hence sufficient for our purposes, however we are not recommending it as a means of clustering ECG data generally. Table 12 shows that clustering with Euclidean distance and correlation is now 100% accurate for unclipped data with $k$-means, and there is no decrease in accuracy when the data is clipped.

## 7. Conclusions

This paper demonstrates the benefits of clipping data when clustering time series. With most real world time series clustering problems space and time will be a genuine constraint on the mining process. Clipping time series will allow for more series to be stored in main memory, which in itself will increase the speed of mining. Specific algorithms for binary series can lead to improved time complexity. When clustering, the information discarded by clipping does not decrease the accuracy if the series are long enough. For data from the class of ARMA and hidden Markov models we show that, theoretically, the models fitted to clipped data asymptotically approach the model fitted from the unclipped data. Over the class of polynomoial, ARMA and hidden Markov models we demonstrate the application of this result through a series of clustering experiments with $k$-means and $k$-medoids to show that: calculating distances and correlations is faster with clipped data; if the data series are long enough then clipping does not significantly decrease clustering accuracy; and if the data contains outliers, the clustering accuracy on clipped data is significantly better. Thus our advice when clustering time series is to start with clipped data, then examine any results from more sophisticated transformations in relation to the results obtained after clipping, particularly if the series are long and time and space are important considerations.

We describe a new form of clustering algorithm based on fitting AR parameters chosen to minimize the AIC which works well when the objective is to group series by how they change over time. The method is fast and as accurate as other techniques based on autocorrelations. We demonstrate that this clustering procedure also works for data from HMM, a perhaps surprising result explained by the relationship between HMM and ARMA models described in Section 5.1.

We have also introduced a new data set for time series clustering derived from electricity bid data. The next steps in this research will be to evaluate the effects of clipping in the context of other time series data mining tasks such as indexing, segmentation and classification, and to apply the technique to data mining projects on electricity and auction bid data sets.

## References


Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of Econometrics, 16*, 3–14.

Alon, J., Sclaroff, S., Kollios, G., & Pavlovic, V. (2003). Discovering clusters in motion time-series data. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*.

Aref, W., Elfeky, M., & Elmagarmid, A. (2004). Incremental, online and merge mining of partial periodic patterns in time-series databases. *IEEE Transactions on Knowledge and Data Engineering, 16:3*, 332–342.

Agrawal, R., Faloutsos, C., & Swami., A. N. (1993). Efficient similarity search In sequence databases. In *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)* (pp. 69–84).

Bagnall, A. J., & Smith, G. D. (2000). Game playing with autonomous adaptive agents in a simplified economic model of the UK market in electricity generation. In *Proceedings of IEEE-PES / CSEE International Conference on Power System Technology POWERCON 2000* (pp. 891–896).

Bagnall, A. J., Janacek, G., de la Iglesia, B., & Zhang, M. (2003). Clustering time series from mixture polynomial models with discretised data. In *Proceedings of the Second Australasian Data Mining Workshop* (pp. 105–120).

Bagnall, A. J., & Janacek., G. (2004). Clustering time series from ARMA models with clipped data. Technical Report CMP-C04-01, School of Computing Sciences, University of East Anglia.

Chu, S., Keogh, E., Hart, D., & Pazzani. M. (2002). Iterative deepening dynamic time warping for time series. In *Proceedings of the 2nd SIAM International Conference on Data Mining*.

Gaffney, S., & Smyth., P. (2003). Curve clustering with random effects regression mixtures. In C. M. Bishop B. J. Frey (Eds.), *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17/2/3*, 107–145.

Kalpakis, K., Gada, D., & Puttagunta, V. (2001). Distance measures for effective clustering of ARIMA time-series. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01)* (pp. 273–280).

Kedem, B. (1980). Estimation of the parameters in stationary autoregressive processes after hard limiting. *Journal of the American Statistical Association, 75*, 146–153.

Kedem, B., & Slud, E. (1991). On goodness of fit of time series models: An application of higher order crossings. *Biometrika, 68*, 551–556.

Keogh, E., & Pazzani, M. (2000). A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PAKDD 2000* (pp. 122–133).

Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery, 7:4*, 349–371.

Keogh, E., & Folias, T. The UCR time series data mining archive. `http://www.cs.ucr.edu/ eamonn/TSDMA/`.

Li, C., & Biswas, G. (1999). Clustering sequence data using Hidden Markov model representation. In *SPIE'99 Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology* (pp. 14–21).

Lin, J., Keogh, E., Lonardi, S., & Chiu. B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (pp. 2–11). ACM Press.

Maharaj, E. A. (2000). Clusters of time series. *Journal of Classification, 17*, 297–314.

Oates, T., Firoiu, L., & Cohen, P. (1999). 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* (pp. 17–21).

Ormerod, P., & Mounfield, C. (2000). Localised structures in the temporal evolution of asset prices. In *New Approaches to Financial Economics*. Santa Fe Conference.

Smyth, P. (1997). Clustering sequences with hidden markov models. In Michael C. Mozer, Michael I. Jordan, & Thomas Petsche (Eds.), *Advances in Neural Information Processing Systems*, vol. 9 (p. 648). The MIT Press.

Xiong, Y., & Yeung, D.-Y. (2002). Mixtures of ARMA models for model-based time series clustering. In *IEEE International Conference on Data Mining (ICDM'02)*.

Zhong, S., & Ghosh, J. (2003). Scalable, balanced model-based clustering. In *Proceedings of SIAM Int. Conf. on Data Mining* (pp. 71–82).

Received March 25, 2004
Revised October 6, 2004
Accepted October 6, 2004
Final manuscript October 6, 2004