Data Mining and Knowledge Discovery, 13, 11–40, 2006

DOI: 10.1007/s10618-005-0028-0

A Bit Level Representation for Time Series Data Mining with Shape Based Similarity

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

CHOTIRAT "ANN" RATANAMAHATANA ann@cp.eng.chula.ac.th
Department of Computer Engineering, Chulalongkorn University, Thailand

EAMONN KEOGH eamonn@cs.ucr.edu
STEFANO LONARDI stelo@cs.ucr.edu
Department of Computer Science and Engineering, University of California, Riverside, USA

GARETH JANACEK G.Janacek@uea.ac.uk
School of Computing Sciences, University of East Anglia, Norwich, UK

Received May 12, 2005; Accepted October 31, 2005

Published online: 12 May 2006

Abstract. Clipping is the process of transforming a real valued series into a sequence of bits representing whether each data is above or below the average. In this paper, we argue that clipping is a useful and flexible transformation for the exploratory analysis of large time dependent data sets. We demonstrate how time series stored as bits can be very efficiently compressed and manipulated and that, under some assumptions, the discriminatory power with clipped series is asymptotically equivalent to that achieved with the raw data. Unlike other transformations, clipped series can be compared directly to the raw data series. We show that this means we can form a tight lower bounding metric for Euclidean and Dynamic Time Warping distance and hence efficiently query by content. Clipped data can be used in conjunction with a host of algorithms and statistical tests that naturally follow from the binary nature of the data. A series of experiments illustrate how clipped series can be used in increasingly complex ways to achieve better results than other popular representations. The usefulness of the proposed representation is demonstrated by the fact that the results with clipped data are consistently better than those achieved with a Wavelet or Discrete Fourier Transformation at the same compression ratio for both clustering and query by content. The flexibility of the representation is shown by the fact that we can take advantage of a variable Run Length Encoding of clipped series to define an approximation of the Kolmogorov complexity and hence perform Kolmogorov based clustering.

Keywords: clipping, time series data mining, Kolmogorov complexity

1. Introduction

The increasing prevalence of longitudinal databases has led to a massive growth in the amount of research being conducted into time series data mining (TSDM) (Aach and Church, 2001; Berndt and Clifford, 1994; Chiu et al., 2003; Keogh, 2002; Yi and Faloutsos, 2000). A key feature of many time series databases is their enormous size. It is not uncommon in fields such as meteorology, medical imaging, or customer relationship modeling to collect terrabytes or even petabytes of time-dependent data. A proper analysis of these huge data sets is often impossible because of memory and time

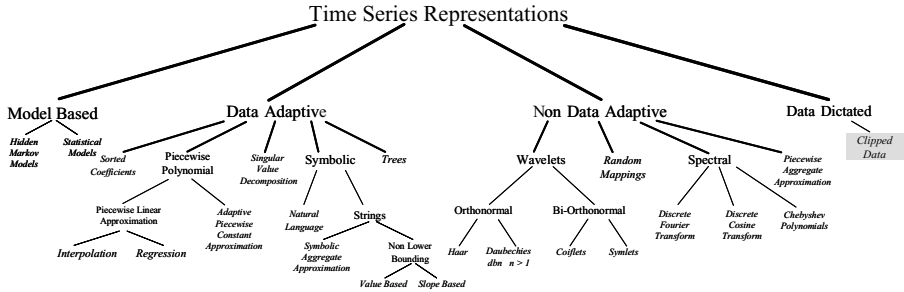

Figure 1. A hierarchy of time series representations used for data mining.

constraints. Thus, one of the fundamental issues in any TSDM task is how to compress the series while maintaining discriminatory power. Many transformations with associated compression schemes have been proposed, including Discrete Fourier Transforms (DFT) (Agrawal et al., 1993; Bagnall et al., 2005); Discrete Wavelet Transforms (DWT) (Chan and Fu, 1999); Principle Component Analysis (PCA) (Korn et al., 1997; Basak et al., 2004); Piecewise Constant (PAA) (Keogh and Pazzani, 2000) and Linear (PLA) (Morinaka et al., 2001) Approximations; Symbolic Aggregate Approximations (SAX) (Lin et al., 2003); and Chebyshev polynomials (CHEB) (Cai and Ng, 2004). An overview of the alternative representations is shown in Figure 1. This paper investigates the application of a simple transformation, *clipping*, to TSDM problems.

The choice of representation is strongly influenced by the data mining task and the similarity objective. The fundamental related issues in any TSDM task are: how to measure the similarity between time series; how to compress the series while maintaining discriminatory power; and what algorithm to use to perform a particular task. The choice of similarity measure is problem dependent, but the types of similarity can be categorized as follows:

- *Similarity in time*. If the similarity between series is strongly dependent on time, a correlation or Euclidean based distance metric is normally used.
- *Similarity in shape*. The common characteristics that are being searched may only be weakly dependent (or independent) of time. Recent research (Ratanamahatana and Keogh, 2005) indicates that weak time dependent similarity is most successfully measured using dynamic time warping.
- *Similarity in change*. Characterized by similarity in the autocorrelation structure (also called *structural similarity*).

Since similarity in time is just a special case of similarity in shape, the two are often grouped together as *shape based similarity.* We examine how clipping series effects TSDM tasks with a shape based similarity objective. Clipped series are simply binary series where each bit stores whether the raw data point is above or below the average. Clipping, or hard limiting, a time series is the process of transforming a real valued time series C into a binary series c, where 1 represents above the population average and 0 below, i.e., if μ is the population mean of series C, then

$$c_t = \begin{cases} 1 & \text{if } C_t > \mu \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

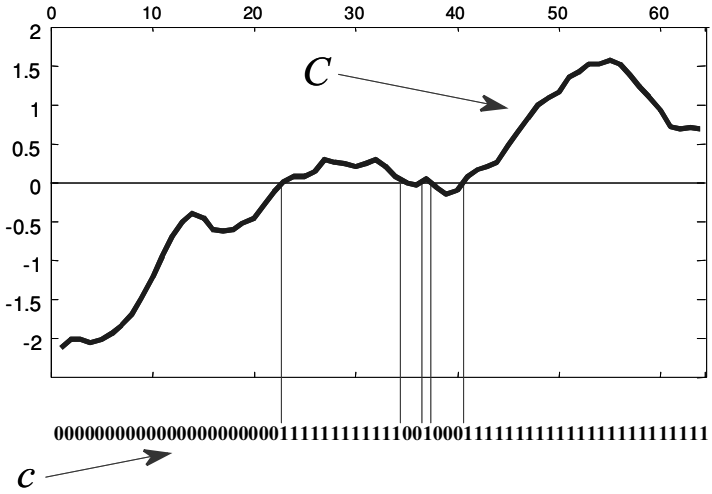

Figure 2. A time series of length 64, denoted C, is converted to the clipped representation, denoted c, simply by observing the elements of C that are strictly above zero, then setting the corresponding bits in c to 1, and to 0 otherwise.

Since we are not interested in detecting differences in population mean, we can assume without loss of generality that $\mu = 0$. As shown in Figure 2, clipping replaces each real valued data point with a single bit.

Clipping is a very simple transformation that has been used in many other research context (Kedem, 1980; Weld and de Kleer, 1990). It is also a transformation that data mining practitioners may have found useful through experience. Our contribution is to theoretically and experimentally examine the usefulness of the representation in the context of time series data mining and to demonstrate its flexibility by showing how clipped series can be used in conjunction with a variety of algorithms and statistical tests.

Clipping a series can yield compression ratios from 32:1 to better than 1,000:1, while still retaining a large amount of useful information, especially with long series. Clipping series allows for extensive exploratory analysis of databases that otherwise may be too large to analyze. Other compression schemes used in TSDM require the user to make fundamental choices concerning what type of transformation to use and what compression ratio to adopt. For example, an analyst wishing to compress a very large data set prior to building a classifier must choose whether to use local approximations, e.g., Wavelets, or a global method such as Discrete Fourier Transforms (DFT). Once a type of transformation is chosen, the user then needs to select from a choice of methods (such as Haar, Debuchies or Symlet Wavelets) and/or compression techniques (e.g., keep the first or largest DFT coefficients). There are then other parameters to set, not least of which is the compression ratio. Often, all of these choices need to be made prior to any real data mining, and the size of the data makes parameter tuning infeasible. In contrast, clipping is a data dictated compression scheme. It is simple to perform, and fundamental graphical characteristics of the original data are retained. Another benefit of using binary series is that there are many procedures applicable to binary data but

not useable for TSDM with other representations. This means that clipped series can often be employed to obtain results of higher quality than other techniques. Hence, the central message of this paper is that because of the simplicity, flexibility, and quality of the results obtained, clipping should be the first transformation used in an exploratory analysis of very large time dependent data sets. This paper seeks to verify this claim both theoretically and experimentally.

Bagnall and Janacek (2005) extend the work of Kedem (1980) to demonstrate the advantages of clipping for clustering based on similarity in change. Bagnall and Janacek (2005) prove that if the underlying series is stationary Autoregressive Moving Average (ARMA), then, if we assume that the unclipped series is both Gaussian and stationary to second order, the clipped series is asymptotically ARMA. They demonstrate experimentally what this property means, and that clipped series can be used to find clusters as accurate as those formed with the raw data but much faster. In this paper, we describe how clipping can be used for shape based similarity for the TSDM tasks of clustering, classification, and query by content (indexing). We show that:

1. Uniquely amongst the representations shown in Figure 1, clipped series can be compared directly to the raw data series, as shown in Section 2.1. Additionally, we show that we can form a tight lower bounding distance metric for Euclidean and Dynamic Time Warping distance.
2. Under some basic assumptions, the difference in discriminatory power between clipped and unclipped tends to zero as the series length tends to infinity (Section 2.2).
3. Clipped series can be very efficiently stored and manipulated, as described in Section 3. By using variable length prefix encoding (Huffman, 1952; Schwarz, 1964), we can get compression ratios of up to 1,000:1 with a lossless compression of clipped data.
4. Run length encoding is just one way we can exploit the binary representation. In Section 4, we describe some algorithms and statistics that can be adapted for use with clipped series.
5. In Section 5, we compare using a clipped representation with two popular alternatives, DFT and PAA (The PAA representation is equivalent to Haar wavelets when the length of time series is a power of two (Keogh and Pazzani, 2000; Yi and Faloutsos, 2000).

The experiments in Section 5 demonstrate the benefits of clipping on a large number of data sets. We use clipped series in increasingly complex ways to illustrate the great flexibility of the representation. Section 5.1 shows that working directly with the clipped series lead to better clustering on simulated and real world data sets, assuming a compression ratio of 32:1. Section 5.2 demonstrates that variable run length encoded sliding windows of clipped series (described in Section 3) leads to better indexing results when the Euclidean distance lower bounding function defined in Section 2.1 is used. Section 5.3 illustrates that the Dynamic Time Warping of run length encoded sliding windows of clipped series also performs better on the same data sets as used in Section 5.2. In Section 5.4, we assess the performance of Kolmogorov complexity clustering (Li et al., 2003) of clipped series, one of the most promising extensions described in Section 4. Finally, in Section 6, we summarize our findings and highlight some future directions.

2. Similarity measures for clipped series

The distance between series $Q = Q_1, Q_2, \ldots, Q_n$ and $C = C_1, C_2, \ldots, C_n$ is most commonly measured by the squared Euclidean distance, given in Eq. (2).

$$D_E(Q, C) = \sum_{t=1}^{n} (Q_t - C_t)^2 \tag{2}$$

Equation (2) can be used to measure the Euclidean distance between two clipped series, $D_E(c, q)$, and the bitwise representation can be exploited to calculate D_E more efficiently (see Section 3). $D_E(c, q)$ is the natural approximation of $D_E(C, Q)$ to use with clustering. However, one of the advantages of clipping over other representations is that Equation (2) can also be used if only one series is clipped. This is beneficial for classification or query by content, particularly if it is prohibitively expensive to transform the test data or candidate series. It also allows us to form tighter lower bounds on the distance function for indexing (see Section 3.1).

For similarity in shape, Dynamic Time Warping (DTW) is commonly used to mitigate against distortions in the time axis (Ratanamahatana and Keogh, 2005). Suppose $M(Q, C)$ is the $n \times n$ pointwise distance matrix between Q and C, where $M_{i,j} = D_E(Q_i, C_j)$. A warping path $W = \langle (a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k) \rangle$ is a set of points that define a traversal of matrix M. A valid warping path must satisfy the conditions $(a_1, b_1) = (1, 1)$ and $(a_k, b_k) = (n, n)$, and that $0 \leq a_{k+1} - a_k \leq 1$ and $0 \leq b_k - b_{k+1} \leq 1$ for all $k < n$.

The DTW distance between series is the path through M that minimizes the total distance, subject to constraints on the amount of warping allowed which is determined by the warping window width parameter r. This places a constraint on the maximum difference between the warping indexes a_k and b_k. A full window width ($r \geq n$) means all paths are considered whereas the minimum window width ($r = 0$) results in DTW simplifying to Euclidean distance. Let $\mathcal{W}_r$ be the space of all feasible paths for warping window r and $w_j = M(Q_{a_j}, C_{b_j})$ be the distance between element a_j of Q and b_j of C for the j^{th} pair of points in a warping path.

The distance for any path X is the sum of the distances between the points in that path

$$D_X(Q, C) = \sum_{i=1}^{k} x_i.$$

The DTW path W_r is the path that has the minimum distance for a given warping window r, i.e.,

$$W_r = \min_{X \in \mathcal{W}_\nabla} (D_X(Q, C)),$$

and hence the DTW distance between series is

$$D_{W_r}(Q, C) = \sum_{i=1}^{k} w_i. \tag{3}$$

The warping window is kept constant in experiments, hence we denote the DTW distance between two series Q and C as $D_W(Q, C)$. As with Euclidean distance, the warped distance between series can be calculated directly with the clipped series ($D_W(q, c)$), or if just one series is clipped ($D_W(Q, c)$ or $D_W(q, C)$).

Note that this formulation only allows lower bounding where the two series Q and C are of the same length. However, recent work (Ratanamahatana and Keogh, 2005) has shown that we can re-interpolate two sequences of differing lengths to be equal length with no measurable effect on accuracy of classification or precision/recall of indexing.

2.1. *Lower bounding for indexing*

In order to achieve efficient indexing with compressed series, it is essential that the distance measure used lower bounds the distance for the uncompressed data (Agrawal et al., 1993). However, for a query Q and candidate match C, it is clear that neither $D_E(q, c)$ nor $D_E(Q, c)$ lower bound $D_E(Q, C)$. Equation (4) defines a distance function that does lower bound Euclidean distance between C and Q.

$$\text{LB\_clipped}(Q, c) = \sum_{i=1}^{n} \begin{cases} Q_i^2 & \text{if } Q_i > 0 \text{ and } c_i = 0 \\ Q_i^2 & \text{if } Q_i \leq 0 \text{ and } c_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Proposition 1. *For any two time series Q and C of length n, LB_clipped$(Q, c) \leq D_E(Q, C)$.*

Proof: Since the distance between any two points with either measure is non-negative, it is sufficient to show that for any length 1 series x and y with two real values x_1, y_1 we have LB_clipped$(x, y) \leq D_E(x, c)$, where c is the clipped series of y, and so c_1 is the clipped value of the single element of y, y_1 . The result for a series of n points then naturally follows. Firstly, we note that by the definition of Euclidean distance, given in Eq. (2), $D_E(x, y) \geq 0 \ \ \forall x_i, y_i \in \Re$. We then consider the four possible cases for single value series.

(1) $x_1 > 0$ and $y_1 > 0 \quad (c_1 = 1)$
(2) $x_1 \leq 0$ and $y_1 \leq 0 \quad (c_1 = 0)$
(3) $x_1 > 0$ and $y_1 \leq 0 \quad (c_1 = 0)$
(4) $x_1 \leq 0$ and $y_1 > 0 \quad (c_1 = 1)$

□

In cases (1) and (2), LB_clipped$(x, y) = 0$ by definition of Eq. (4), hence LB_clipped $(x, y) \leq D_E(x, c)$. In cases (3) and (4), LB_clipped$(x, y) = x_1^2$. $D_E(x, y)$ can be rewritten as $x_1^2 + y_1^2 - 2x_1y_1$. The fact that $y_1^2 \geq 0$ and $x_1y_1 \leq 0$ means that $y_1^2 - 2x_1y_1 \geq 0$. Hence

$$\text{LB\_clipped}(x, y) = x_1^2 \leq x_1^2 + y_1^2 - 2x_1y_1 = D(x, c).$$

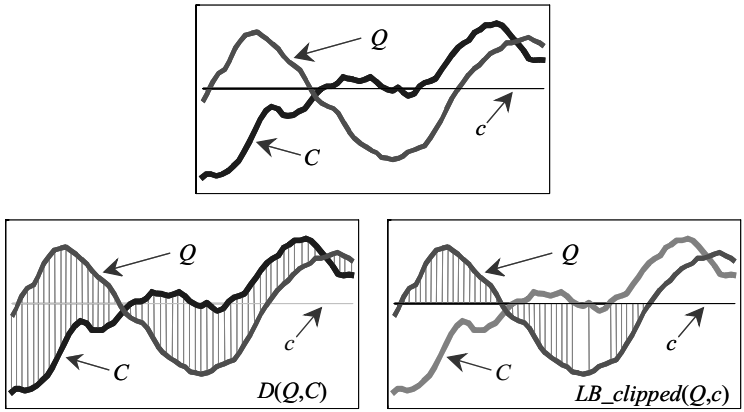

Figure 3. The distance returned by both LB_clipped(Q, c) and $D(Q, C)$ is the sum of squared lengths of the gray hatch lines. Because every hatch line for LB_clipped(Q, c) is matched with the corresponding line in $D(Q, C)$ which is at least as long, we must have LB_clipped$(Q, c) \leq D(Q, C)$.

Thus, for any two series with a single real valued element, the clipped distance is less than or equal to the Euclidean distance. Since both distance measures are metrics, the distance between two points is never negative, hence LB_clipped$(Q, c) \leq D(Q, C)$ for any series of real numbers Q and C.

An intuitive example of why Proposition 2.1 is true is shown in Figure 3. Note that if Q is also clipped, LB_clipped(q, c) is identical to $D_E(q, c)$.

It is also true that the dynamic time warping distance between clipped series, $D_W(q, c)$ does not lower bound the distance between the full series $D_W(Q, C)$. To find a distance measure that does lower bound $D_W(Q, C)$, we first define two new sequences, $U = \langle U_1, U_2, \ldots, U_n \rangle$ and $L = \langle L_1, L_2, \ldots, L_n \rangle$. Let $K = \{-r, -r + 1, \ldots, r - 1, r\}$. Then

$$U_i = \max_{k \in K}(Q_{i+k})$$

and

$$L_i = \min_{k \in K}(Q_{i+k}).$$

U and L form a bounding envelope that encloses Q from above and below (see Figure 4).

Given U and L, Eq. (5) defines a function LB_Keogh_clipped(Q, c) which lower bounds $D_W(Q, C)$.

$$\text{LB\_Keogh\_clipped}(Q, c) = \sum_{i=1}^{n} \begin{cases} L_i^2 & \text{if } Q_i > 0 \text{ and } c_i = 0 \\ U_i^2 & \text{if } Q_i \leq 0 \text{ and } c_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The proof that LB_Keogh_clipped $(Q, c) \leq D_W(Q, C)$ is a straightforward combination of the proof above and the proof in Keogh (2002); we omit it here for brevity.

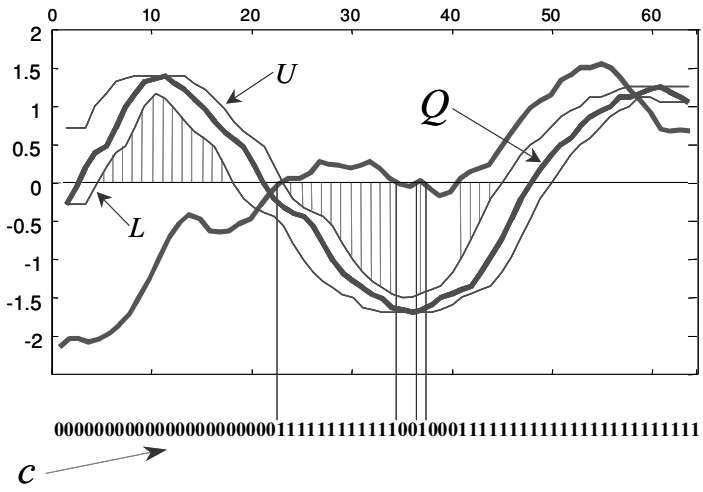

Figure 4. The intuition behind the lower bounding function LB_Keogh_clipped(Q, c), which lower bounds DTW(Q, C).

2.2. *Asymptotic properties of clipped series*

The fact we can lower bound both the Euclidean distance and the DTW distance without transforming the query series supports the use of clipping with query by content. However, it does not tell us anything about the ability of the clipped distance to approximate the true distance. We can relate the auto and cross correlations function to their clipped series versions by using the following result.

Proposition 2.2. *Suppose X and Y are bivariate normal with zero means and common unit variances. Define the indicators I_x and I_y as*

$$I_x = 1 \quad when\ X > 0\ and\ 0\ otherwise$$
$$I_y = 1 \quad when\ Y > 0\ and\ 0\ otherwise$$

then

$$E[I_x I_y] = \frac{1}{2\pi}\arcsin(\rho)$$

where ρ is the correlation between X and Y.

Proof: The (X, Y) density function is

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}(x^2 + y^2 - 2\rho xy)\right\}$$

so

$$P[X > 0] = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_{-\infty}^\infty f(xy)\, dy\, dx = 1/2$$

and $P[Y > 0] = 1/2$ from symmetry.

In addition,

$$\begin{aligned} P[X > 0, Y > 0] &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^\infty f(xy)\, dx\, dy \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^\infty \int_0^{\pi/2} r \exp\left\{-\frac{1}{2(1-\rho^2)}(1-\rho\sin 2\theta)r^2\right\} d\theta dr \end{aligned}$$

using polars. This gives

$$\begin{aligned} P[X > 0, Y > 0] &= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_0^{\pi/2} \frac{1}{1-\rho\sin 2\theta} d\theta \\ &= \frac{1}{2\pi}\left\{\frac{\pi}{2} + \arctan\left(\frac{\rho}{\sqrt{1-\rho^2}}\right)\right\} \end{aligned}$$

It then follows that

$$E[I_x I_y] = \frac{1}{2\pi} \arctan\left(\frac{\rho}{\sqrt{1-\rho^2}}\right)$$

Since

$$\arcsin(\rho) = \arctan\left(\frac{\rho}{\sqrt{1-\rho^2}}\right)$$

we have

$$E[I_x I_y] = \frac{1}{2\pi}\arcsin(\rho)$$

and hence

$$\text{correlation}[I_x I_y] = \frac{2}{\pi}\arcsin(\rho) \tag{6}$$

□

This result establishes a one to one monotonic function between the correlation of clipped and unclipped series. Suppose X_t and Y_t are Gaussian time series, where each observation is normally distributed, and that the clipped series are C_t and D_t. From Eq. (6), we deduce that the cross correlation function, $\rho_{xy}(k)$ is

$$\rho_{cd}(k) = \frac{1}{2\pi} \arcsin(\rho_{xy}(k)) \qquad k = \cdots, -2, -1, -0, 1, \cdots \tag{7}$$

For clipped series, the Euclidean distance between two series is a linear function of the correlation, if we assume a zero mean. This means that if we knew the correct correlation between clipped series, we could detect differences between the series as accurately with clipped data as with unclipped data. Of course, we never know the true correlation. Instead, we estimate it from the data. Hence, to show that Euclidean distance on clipped data can discriminate as accurately as on unclipped data, we have to show that the estimate of the correlation tends asymptotically to the true value, i.e., the estimator is unbiased and that the variance tends to zero as n increases, i.e., the estimator is consistent. The common estimate cross-covariance $\gamma_{xy}(y)$ between series is

$$\hat{\gamma}_{xy}(k) = \frac{1}{N} \sum_{t=1}^{N-|k|} (X_t - \bar{X})(Y_t - \bar{Y})$$

with a cross correlation estimate of

$$\hat{\rho}_{xy}(k) = \hat{\gamma}_{xy}(k)/\sqrt{\hat{\gamma}_{xx}(0)\hat{\gamma}_{yy}(0)}.$$

However, in the clipped case, we know that $E[C_t] = 1/2$ and that $\text{Var}(C_t) = 1/4$, so our estimate will be

$$\hat{\gamma}_{CD}(k) = \frac{1}{N} \sum_{t=1}^{N-|k|} \left(C_t - \frac{1}{2}\right)\left(D_t - \frac{1}{2}\right)$$

while

$$\hat{\rho}_{CD}(k) = 4\hat{\gamma}_{CD}(k)$$

Now, to show that our estimates are unbiased and consistent, we need to take expectations

$$\begin{aligned} E[\hat{\gamma}_{CD}(k)] &= E\frac{1}{N} \sum_{t=1}^{N-|k|} \left(C_t - \frac{1}{2}\right)\left(D_t - \frac{1}{2}\right) \\ &= \frac{1}{N} \sum_{t=1}^{N-|k|} \gamma_{CD}(k) = \left(1 - \frac{|k|}{N}\right)\gamma_{CD}(k) \end{aligned}$$

which is asymptotically unbiased.

To prove consistency, we need to look at the variance of $\hat{\gamma}_{CD}(k)$. This will involve at some point the evaluation of terms of the form

$$E[C_t D_{t+k} C_s D_{s+k}]$$

and orthant probabilities $P[X_t \geq 0, Y_{t+k} \geq 0, X_s \geq 0, Y_{s+k} \geq 0]$. The analytical evaluation of these has been an unsolved problem for many years. We can demonstrate the consistency experimentally by repeatedly measuring the correlation between randomly generated series of differing lengths. Figure 5 shows the variation of the estimate of the

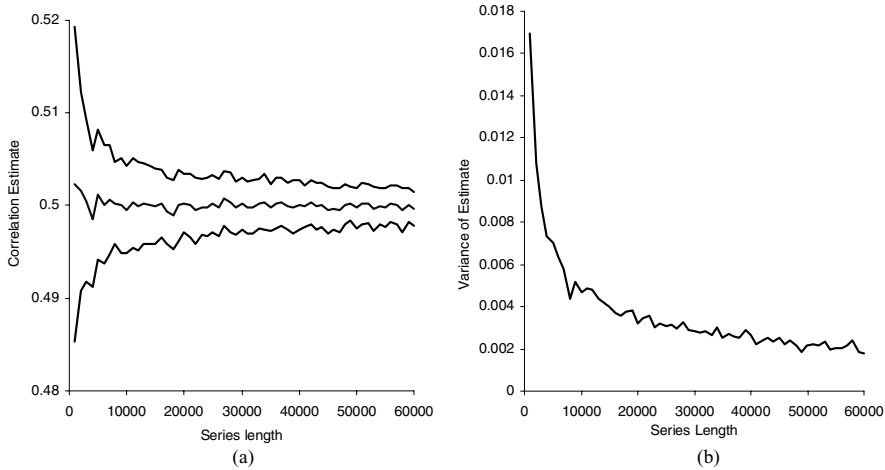

Figure 5. Variation in the estimate of the correlation between series. Figure (a) shows the mean estimate and one standard deviation above and below the mean (the true correlation between series is 0.5). Figure (b) shows the variance of the estimate.

correlation over 100 repetitions of varying length series. It demonstrates clearly that, firstly, the estimate is unbiased, and secondly that it is consistent.

The clipped series maintain the broad shape of the unclipped data, although of course fine detail is lost. We can demonstrate this by considering the spectra. The clipped spectrum is defined as

$$f_y(\omega) = \frac{1}{2\pi} \sum_{s=-\infty}^{\infty} \rho_y(s)\cos(s\omega)$$

so, from Eq. (7) and using the expansion for the arcsin, we have for the clipped series

$$\begin{aligned} f_c(\omega) &= \frac{1}{2\pi} \sum_{s=-\infty}^{\infty} \rho_c(s)\cos(s\omega) \\ &= \left(\frac{2}{\pi}\right) \frac{1}{2\pi} \sum_{s=-\infty}^{\infty} \left[\rho_Y(k) + \frac{1}{6}\rho_Y(k)^3 + \cdots\right] \cos(s\omega) \end{aligned}$$

so

$$f_c(\omega) = \left(\frac{2}{\pi}\right) \left[f_y(\omega) + \frac{1}{6} f_y(\omega) * f_y(\omega) * f_y(\omega) + \cdots\right]$$

where $*$ indicates convolution. Thus, we see that the clipped series spectrum is the unclipped spectrum smeared with the convolutions terms.

We have assumed normality, which commonly will be an invalid assumption. However, these results provide a theoretical basis for using clipped data for shape based mining, because we know that under ideal conditions, mining with clipped data will produce the same results as mining with the uncompressed data. We have concentrated on the correlation between series, which is a measure of similarity in time. However,

the results generalize to the cross correlation and hence extend to the more general case of mining based on similarity of shape. So, for example, dynamic time warping essentially involves rescaling the time axes of the series to maximize the correlation. For any particular pair of points considered in a warping path, we can relate the correlations of the unclipped data to the clipped using Eq. (7). Hence, for any complete warping path, we can theoretically link the correlations between the warped series and hence draw the same conclusions as to the asymptotic properties of the clipped data.

3. Manipulation and storage of clipped series

One of the benefits of using the clipped representation is that binary series can be manipulated and stored efficiently. The compression benefits of clipping are obvious. If we assume that each data point in the raw time series requires 4 bytes (a conservative estimate), then clipping achieves a 32:1 compression ratio. However, we can use various techniques to achieve further compression. In Section 3.2, we describe how Run Length Encoding (RLE) can improve the compression. We demonstrate that sliding window representations of RLE time series can be quickly and efficiently generated, then show how variable length prefix encoding of run lengths can be used to exploit regularity in the data sets.

3.1. Bit comparison

Using clipped data can provide time improvements for many commonly used algorithms. This is because the bit level representation allows for the utilization of bitwise operators. For example, when clustering, it is necessary to measure the the distance between two clipped series q and c using Eq. (2). $D_E(q, c)$ can be efficiently calculated by finding the XOR of q and c then summing the number of 1s in the result, i.e.,

$$D_E(q, c) = \sum_{i=1}^{n} q_i \bigoplus c_i.$$

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 $O(n)$. However, we can improve the constant terms in the time complexity function by using shift operators to evaluate the integer value of each eight or sixteen bit sequence, then using a lookup table to find the number of bits in that integer. This mechanism makes the distance calculation approximately five to ten times faster even when the series are loaded into main memory.

3.2. Compression

Run Length Encoding (RLE) (Golomb, 1966) is a lossless compression scheme for binary series that can achieve significant compression for periodic or seasonal data. For example, consider the clipped series

00000000000000000000000111111111111001000111111111111111111111111

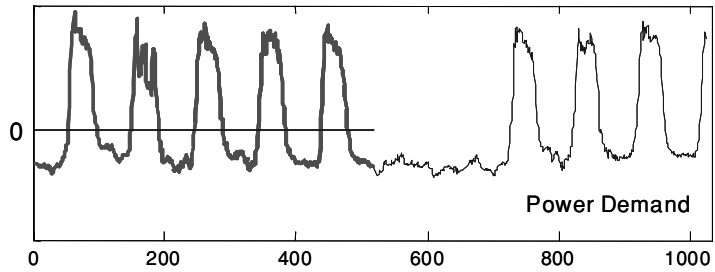

Figure 6. The first 512 data points of the Power Demand dataset prior to clipping.

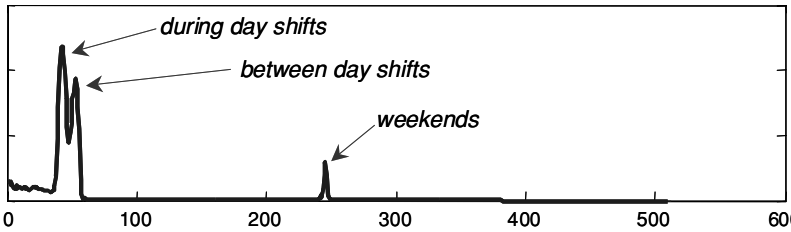

Figure 7. The relative frequency of run lengths for the Power Demand dataset for a single year.

shown in Figure 4. This series could more compactly be written as 22#0, 11#1, 2#0, 1#1, 3#0, 24#1, (or equivalently as @22,11,2,1,3,24). Regularity in a series can be exploited to gain greater compression benefits through encoding run lengths efficiently. For example, Figure 6 shows a subsequence of the highly seasonal power-demand data.

We can usefully encode the run lengths with variable length encoding (prefix encoding) (Huffman, 1952; Schwarz, 1964). The distribution of run lengths is shown in Figure 7. The most commonly occurring are runs of 43, 42 and 53, and these can be encoded with a shorter length key. We use simple Huffman encoding (we found by experimentation that the more complex arithmetic encoding in Rissanen and Langdon (1979) yields only slightly more compact encodings). With an on-line implementation of Huffman encoding (Knuth, 1985), only a single pass over the data is required. The average Huffman code length obtained for the entire PowerDemand dataset is only 5.8 bits, giving a compression ratio of 281:1.

Sliding windows are commonly used in TSDM, particularly for streaming data (Faloutsos et al., 1994; Keogh et al., 2000). With RLE clipped data, time series in consecutive sliding windows are frequently identical except for differences in the first and last run. This can be exploited to further compress the windows. For example, consider the run length encoding @22,11,2,1,3,24 for the data shown in Figure 4. Suppose the encoding of the next five sliding windows is:

@22, 11, 2, 1, 3, 24
@*21*, 11, 2, 1, 3, *25*
@*20*, 11, 2, 1, 3, *26*
@*19*, 11, 2, 1, 3, *27*
@*18*, 11, 2, 1, 3, 27, *1*
@*17*, 11, 2, 1, 3, 27, *2*

The first four windows are very similar (changes are italicized for clarity) and can be compressed. By introducing a new symbol $ to represent incrementing the last run and decrementing first run by 1, this form of *numerosity reduction* results in the final encoding

@22, 11, 2, 1, 3, 24$3@18, 11, 2, 1, 3, 27, 1$1.

With the Power Demand dataset which has 10,000 data points, numerosity reduction together with Huffman coding yields a huge compression ratio of 1,057:1.

In Section 4, we describe various algorithms that can manipulate RLE clipped time series.

4. Using clipped series in TSDM

In Section 5, we show that clipping often produces better results than other techniques. This is not, however, the only benefit of clipping. Further advantages of using a clipped representation include:

1. *Many data mining algorithms are faster with binary attributes*. For example, many machine learning classification algorithms are designed specifically for categorical data; splitting algorithms used with classification/decision trees such as Gini, Information Gain, Chi-statistic, and Twoing all require discretization prior to splitting.
2. *Clipped data can be used by algorithms designed to work on the raw data without further parameterization*. Clustering, classification, rule discovery, motif discovery, visualization, and novelty detection with any shape based distance measure can be performed with clipped data just as one would do with the raw data. Query by content can be efficiently performed with the lower bounded distance metrics given by Eqs. (4) and (5).
3. *Algorithms designed specifically for binary series can be employed*. Many of these algorithms can work directly with RLE clipped data with no major time overhead. For example, Kaufman and Rousseeuw (1990) describe a hierarchical clustering algorithm called MONA (Monothetic Analyis), and Ordonez (2003) describes an online, scalable and incremental version of *k*-means, both of which are designed specifically for binary series and can work on RLE data. Another example is a form of neural network called a Correlation Matrix Memory (CMM) Austin and Lees (1998) used as part of the advanced uncertainty reasoning architecture (AURA) (Austin, 1996). AURA has been used successfully in a variety of fields (Hodge and Austin, 2003; Austin et al., 2005) but requires binary input vectors. A natural way of applying CMM and AURA to time series would be to use it in conjunction with RLE clipped data, with, for example, the *k*-NN binary classifier described in Austin and Zhou (1998). More generally, there are numerous algorithms for clustering categorical data that can be applied to clipped series (For example, Ganti et al., 1999; Huang, 1998.)
4. *Clipped series can be run length encoded*. The fact that we can lossless compress clipped data with a model based technique means that algorithms that are not applicable to any other representation shown in Figure 1 can be used with clipped data. This advantage is discussed in more detail in Sections 4.1 and 4.2.

4.1. Kolmogorov complexity for RLE clipped series

Recently, Li et al. (2003) have proposed a clustering algorithm inspired by the concept of Kolmogorov complexity. The Kolmogorov complexity $K(x)$ of a string x is defined as the length of the shortest program capable of producing x on a universal computer—such as a Turing machine. Intuitively, $K(x)$ is the minimal quantity of information required to generate x by an algorithm. The conditional Kolmogorov complexity $K(x \mid y)$ of x given y is defined as the length of the shortest program that computes x when y is known. If x and y are similar and y is known, then information present in y will mean that a very short program can be used to generate x. Equation (8) shows that a distance metric derived from the Kolmogorov complexity is based on the difference between $K(x \mid y)$, $K(y \mid x)$ and $K(xy)$, where $K(xy)$ is the complexity of string x concatenated with string y.

$$D_K(x, y) = \frac{K(x \mid y) + K(y \mid x)}{K(xy)} \tag{8}$$

The Kolmogorov complexity is commonly estimated by standard off-the-shelf compression algorithms, such as WinZip or Stuffit. If $C(x)$ is the size of compressed string x with the chosen compression algorithm and $C(x \mid y)$ is the size of file x after compressing it with the compression model built for y, then the distance between two strings is given by Eq. (9).

$$D_C(x, y) = \frac{C(x \mid y) + C(y \mid x)}{C(xy)} \tag{9}$$

Li and Vitanyi have shown that this distance measure works well for clustering data from a wide range of domains, such as DNA strings, MIDI files, natural language text, and computer programs. However, a key feature of using the estimate of the Kolmogorov complexity given in Eq. (9) is that the compression algorithm must be lossless and use a model learned from the data (typically a substitution dictionary). While there are a host of compression algorithms for real-valued time series (DFT, DWT, SVD, etc.), virtually all of them are lossy. The handful of lossless techniques (delta encoding, for example) do not produce a compression model. However, compression of clipped data with variable length RLE model described in Section 3.2 is both lossless and uses a model. We can, therefore, define $C(x \mid y)$ as the size of time series x when compressed with the run length dictionary learned for time series y. In Section 5.4, we show how this measure produces good clusters on a hard problem.

4.2. Statistical tests for RLE clipped series

The RLE gives us the runs of 1's and 0's together with their lengths. The frequency distributions derived to form the variable length RLE can be the basis for statistical non parametric procedures for run data. The most obvious way of using RLE clipped data is for a run test for randomness of a single series (Bradley, 1968). Highly structured data will have a regularity of pattern in runs of 1's or 0's not evident in random data. Rice's well known zero crossing result states that if D is the number of crossings of the mean then the autocorrelation of order 1, $\rho(1)$ is related to the expected value of D by

$$\rho(1) = \cos\left(\frac{\pi E[D]}{N-1}\right)$$

(Rice, 1944) Figure 7 shows an example of a highly structured run length distribution indicating extreme deviation from randomness. Figure 8 shows the distribution of run length for a clipped AR1 process with parameter 0, 0.5, −0.5, and −0.9. A run test would detect that Figures 8(b), (c), and (d) are significantly different from Figure 8(a). We can also detect that, for example, Figure 8(b) is significantly different from Figures 8(d). Since we have the entire frequency distribution of run lengths, we can compare two series by using a measure of discrepancy between their two run distributions. Any test of distribution can be used to discriminate between series based on the run frequency. For example, we currently use a Chi-squared criterion, but alternatives such as the Kolmogorov-Smirnov distance may prove useful. Chi-squared statistic is a simple and well used measure, and we are currently investigating the possibility of decomposing it into orthogonal components to examine the moments of the observed frequencies (Rayner and Best, 2001).

A recent statistical research direction directly applicable to query by content and anomaly detection has been moved toward scan statistics (Glaz and Balakrishnan, 1999; Glaz et al., 2001). Scan statistics relate to distributions over sliding windows. One of the problems with scan statistics is that they can become difficult to calculate for real valued series. For binary series, however, the estimates are much easier to calculate. Furthermore, these statistics can be calculated directly from the numerosity reduced variable run length encoded time series. This means it is feasible to apply this promising technique to very long clipped time series.

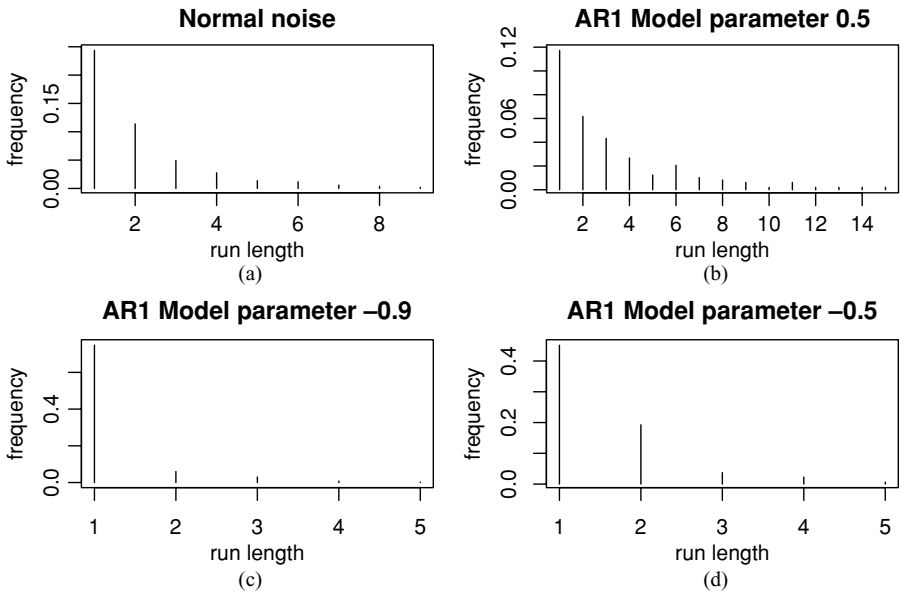

Figure 8. Variation in the run lengths for different AR 1 models.

5. Results

In this section, we will provide extensive empirical evidence of the benefits of clipping. Our sequence of experiments is designed to show how the clipped representation can be used to good effect in increasingly complex ways. The first experiments described in Section 5.1 involve clustering clipped series directly with Eq. (2). Section 5.2 reports a set of experiments assessing clipped representations with sliding windows and run length encoding for query by content. The third set of experiments in Section 5.3 demonstrate that clipped series give better indexing results when used with sliding windows, run length encoding, and dynamic time warping. Section 5.4 summarizes the final set of experiments which show how the variable length encoding methods can be employed in conjunction with Kologorov based clustering to achieve better results.

Over 50 different data sets were used in the experiments, all freely available at (Keogh and Folias, 2002). The data sets range from 66 kilobytes to 2 gigabytes in size.

The experiments are designed to remove the possibility for bias against either DFT or PAA. For example, in PAA representation, if m (the number of constant approximations) is less than two or does not evenly divide the length of the sliding window, we increase m to two or to the next smallest integer that evenly divides the size of sliding window. This is advantageous to PAA, as it will result in a lower compression ratio, but is justified by the observation that since the data is normalized, having a single constant approximation is not informative.

5.1. *Clustering clipped series*

We examine a class of problems where a DFT approach should produce good results and show that clipping does better than the most commonly used DFT approach described in Agrawal et al. (1993). The class of model we consider is generated from a mixture of Sine waves. A generating model is of the form

$$M(t) = \sum_{j=1}^{r} a_j \cdot \sin(b_j + c_j \cdot t) \tag{10}$$

The parameters a, b, and c control the amplitude, offset, and frequency of the curves, respectively. Each can take a value on the range [0,1], but they are rescaled so that any particular sine wave has a maximum amplitude of 2 and a maximum offset of $n/2$ (where n is the series length). A frequency parameter of 0 means the sine wave completes a single oscillation over the data length, and a frequency parameter of 1 means the curve will complete $n/4$ cycles over the n data points. For a clustering experiment, we generate k different models, $M_1, M_2, \ldots, M_k$, then each time series from cluster c is found by adding Gaussian noise to M_c. We wish to assess whether compressed data can be clustered as accurately as the raw data. Our aim is to evaluate which representation provides clusters as accurate as the baseline results on the class of model described. Following the algorithm described in Agrawal et al. (1993), we cluster with DFT by performing an $O(n \log n)$ FFT, retain the first f_c coefficients (set to $n/64$ to achieve a compression ratio of 32:1), then use Euclidean distance between the difference of the coefficients as the clustering distance metric. We cluster data from eight clusters, with

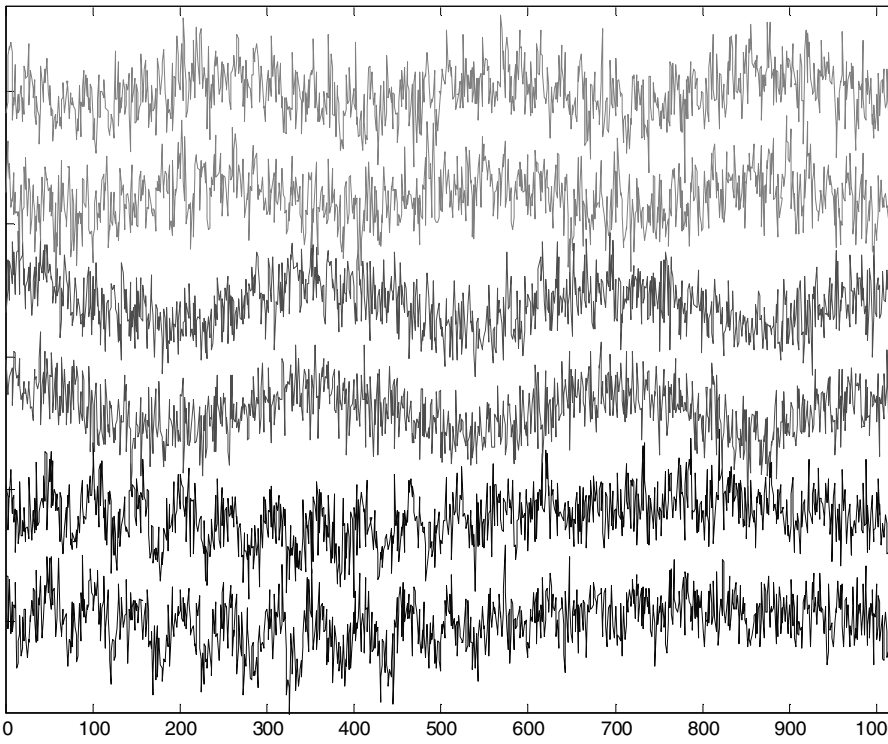

Figure 9. Example of sine wave data series from three separate cluster models.

five series in each cluster, generated from functions of the form given in Eq. (10) with parameters randomly selected in the range [0,1], each model being a combination of three sine waves (i.e., $r = 3$). We cluster with k-means restarted 100 times at random initial centroids. Each series is 1,024 data points long. An example of the data from 6 series is shown in Figure 9.

The results for 20 randomly generated sets of models are shown in Table 1. Accuracies are measured by enumerating all possible cluster labelings and comparing against the known correct clustering. The clipped clusters are often identical to those found with the raw data and the difference in mean accuracy is very small. On 12 out of the 20 runs, the clipped series found completely correct clusters, whereas the DFT approach, in addition to having a significantly lower average accuracy, found the correct clustering on a single occasion only.

Table 1. The mean accuracies for 20 randomly generated sets of models.

	Mean accuracy	Number of perfect clusterings
Raw	96.87%	15
Clipped	94.22%	12
DFT	81.56%	1

More recent DFT approaches such as those described in Morchen (2003) and Vlachos et al. (2005) may perform better on the data used in these experiments. The choice of how to compress the DFT coefficients is problem dependent. This demonstrates one of the strengths of clipping, in that it is a very simple and robust procedure with no need for parameterization. It clusters well in relation to using the raw data for all parameter settings attempted, and has been shown to do well on data where a DFT approach is not appropriate (see Bagnall and Janacek, 2004, 2005). To demonstrate how clipping can help with a real world problem, especially in large data sets, we clustered optical recording data from a bee's olfactory system. Analysis of this data can help us understand the mechanisms of the olfactory code and the temporal evolution of activity patterns in the antennal lobe. Further information can be found in Galan et al. (2004). The data consists of 980 images, each image containing 688 × 520 measurements. If we

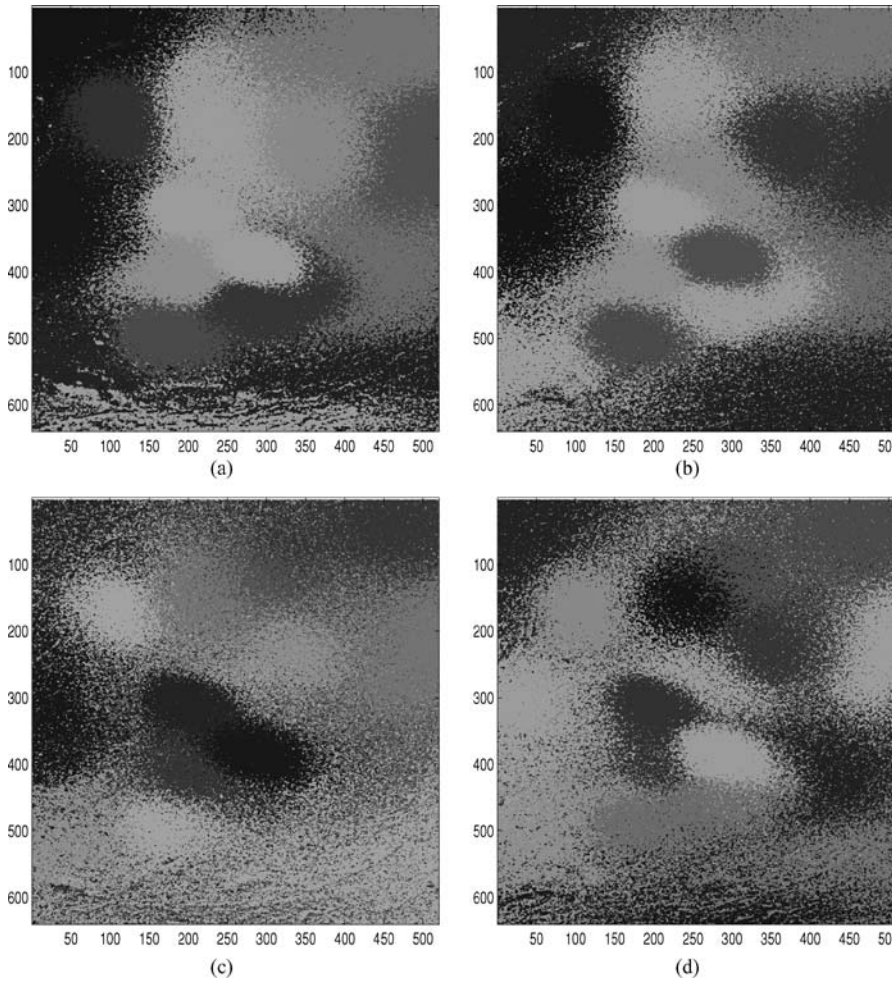

Figure 10. Sixteen clusters produced using (a) the whole 2-gigabyte raw data (b) clipped series (c) DFT series (d) PAA series. These pictures can be seen at http://www.cmp.uea.ac.uk/Research/kdd/.

consider each position in the image as a time series, the data consists of 357,760 time series of length 980. Preliminary analysis has shown that clustering the series based on similarity in time produces results that have a sensible physiological interpretation (Galan et al., 2004). We cluster with k-means (with k set to 16) restarted 50 times from random initial centroids, and take as the best clustering the one with the lowest within-cluster variation. The dataset size is around 2 gigabytes, so clustering the raw data may be prohibitively demanding of time and space resources. A single run of k-means takes 50 – 150 iterations to converge (taking hours to complete), and most machines do not have 2 gigabytes of main memory. Hence, some form of data reduction is appropriate for this problem. We use this data set to demonstrate that compressing the data through clipping can produce clusters more similar to those produced with the whole data than those found using DFT and PAA. We assume that clipping provides a compression ratio of 32:1 (in practice, it may be much higher than this), and we set the parameters of DFT and PAA to achieve a similar or smaller ratio. For PAA, we compress each series into 49 mean values (since the number of PAA coefficients must evenly divide the length of each time series), giving a compression ratio of 20:1. For DFT, we pad the series to length 1,024 and retain the first 17 DFT coefficients, giving 29.7:1 compression ratio.

The results for all four techniques are shown in Figure 10. It is clear from these plots that the clusters with clipped series are much closer to those found with the raw data than those formed with DFT and PAA. There are structural similarities between all four clusterings, but the clipped clusterings are much more clearly defined than those found with PAA and DFT. The only major difference between the unclipped and the clipped clusters is an additional central cluster formed around position (325,450) with the unclipped data. To measure the similarity between the clusterings, we use three well-known measures based on the count of coincidence of common cluster membership: the Jaccard coefficient (Milligan et al., 1983); the Randstatistic (Rand, 1971); and the Folkes and Mallows index (Fowlkes and Mallows, 1983). For all three measures, higher values indicate a greater degree of similarity. We randomly selected 10 clusterings from the run of 50 for each experiment with clipped, DFT, and PAA compression. For each of these 10 clusterings, we measure the similarity to 10 randomly selected clusterings formed with the complete data, giving us 100 sets of comparisons for each technique. The mean Jaccard, Rand, and Folkes and Mallows statistics, along with the standard deviations, are shown in Table 2.

For each of the three measures, we can, at the 1% level, reject the null hypothesis that (clipped, DFT), and (clipped, PAA) average similarity are the same (using a t-test for the mean and a Mann-Whittley test for the median). This clearly demonstrates that, when using restarted k-means with an objective to cluster based on similarity in time, clipping is a more appropriate compression than DFT or PAA for this data set. This

Table 2. Average similarity and standard deviation between the clusters formed with the raw and compressed data, comparing among10 clusterings for each technique.

	Jaccard	Rand	Folkes and Mallows
Clipped	0.4456 ± 0.042	0.9469 ± 0.006	0.6156 ± 0.04
DFT	0.2108 ± 0.017	0.9118 ± 0.012	0.3425 ± 0.093
PAA	0.2761 ± 0.017	0.9227 ± 0.003	0.4327 ± 0.021

is true even though we have assumed the compression ratio is 32:1. In practice, as we show in the following sections, much better compression can normally be achieved with RLE.

5.2. *Indexing sliding windows*

We tested our approach for indexing on twelve datasets with a wide range of shape properties, all available from the UCR Time Series Data Mining Archive (Keogh and Folias, 2002). The sizes of the data sets range from 6,875 data points to 198,400 data points. The problem is to assess which of the clipped, PAA and DFT techniques will find the nearest neighbor to a query series in the shortest time.

Since disk I/O is known to dominate the query time and provide a good indication of the tightness of the lower bound (Keogh and Kasetty, 2003), performance is measured by the number of I/O disk accesses required. This also avoids any possibility of implementation bias.

For realism, we construct our experiment so that there is never an exact or a trivial match in the database. We do this by removing a randomly selected section of size query length plus 100, then forming a query by taking away the first and last 50 points. The remaining data is then clipped, and the database to be queried is formed by using a sliding window over the clipped series. So, for example, if the original series is of length 1,000 and the query is size 200, the database will contain 500 binary series of

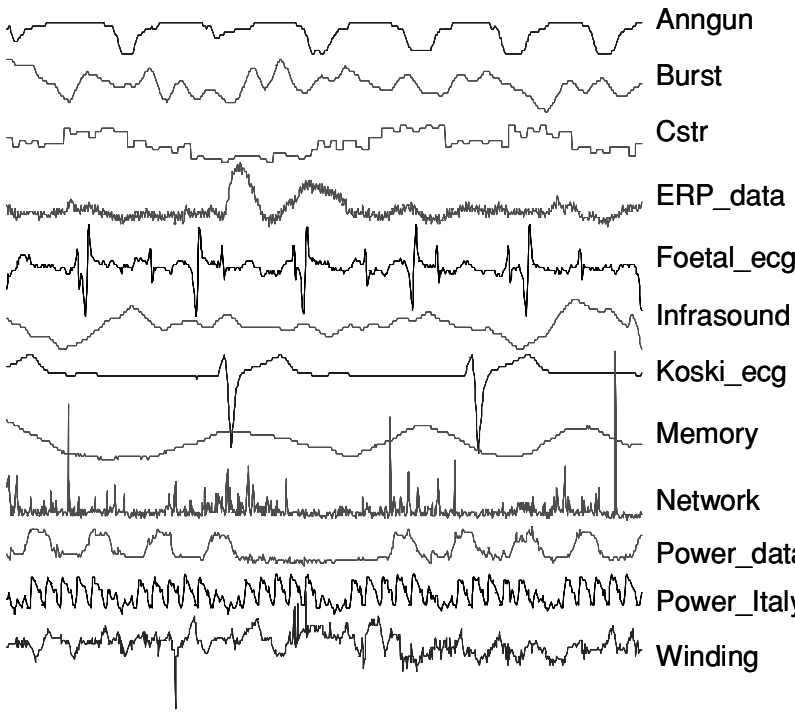

Figure 11. Sections of length 1,000 from each of the twelve data sets used in the experiments.

Table 3. Number of disk accesses required with Euclidean distance lower bounding.

		Query size 256				Query size 512			
Dataset	Size	Ratio	Clipped	PAA	DFT	Ratio	Clipped	PAA	DFT
Anngun	10, 001	691.2	**982**	5478	1433	677.6	**1053**	5614	4656
Burst	9, 382	381.7	**2298**	4785	2644	498.9	5481	**4086**	4424
Cstr	22, 500	464	**1885**	6704	2433	608.8	**2445**	7533	2978
ERP_data	198, 400	239.8	**4897**	17438	17701	321	8763	33210	**1860**
Foetal_ecg	20, 000	121.5	**3034**	7802	10141	157.3	**3507**	15641	16843
Infrasound	8, 192	390.8	1826	3778	**1722**	379.1	**2474**	3119	3263
Koski_ecg	144, 002	419.2	3024	10098	**2314**	628.5	**2931**	28829	9386
Memory	6, 875	396.9	**973**	1882	439	715	1513	2739	**1134**
Network	18, 000	60.2	**13421**	16174	17084	62.4	**12305**	17074	17384
Power_data	35, 040	437.3	**1636**	9071	4207	1089.5	**708**	9949	4214
Power_Italy	29, 931	209.9	**651**	2919	3666	286	**771**	13430	16765
Winding	17,500	152.3	**2090**	5433	4214	176.1	**2684**	5257	6101

length 200. These series are run-length variable size encoded with numerosity reduction (as described in Section 3.2). The data dictated compression ratio resulting from this process is then used to determine the parameters for PAA and DFT. For the twelve data sets considered, the compression ratios for clipped data range between 60.2:1 to 1,089.5:1 (the complete list is given in Table 3).

The nearest neighbor is found by first computing the lower bound distances between a query and all the sequences in the database and then determining in ascending order of lower bounded distance which sequence is the true nearest neighbor solution.

The lower bound distances for clipped data were calculated using Eq. (4); the lower bounded distances for PAA and DFT were found by first transforming the query series then calculating the Euclidean distance between retained parameters.

Table 3 gives the number of disk accesses for all the data sets. To allow a visual comparison, Figures 12 and 13 present the results from Table 3 normalized by the worst performing algorithm on each data set.

With query lengths of 256 and 512, clipped outperforms DFT and PAA on 10 out of 12 data sets. When considered in conjunction, clipped is always better with either query length 256 or 512, and is better with both on 7 out of 12 data sets. It is worth emphasizing that these experiments are designed to favor PAA and DFT; we have assumed PAA and DFT variables can be stored in two bytes. In addition, we have rounded up the number of coefficients to retain to the nearest whole integer and insisted on at least two per series irrespective of the compression ratio. These results show that the shape information retained by clipping and the ability to further lossless compress clipped series mean that indexing on Eulcidean distance is more efficient with clipped data than with DFT or PAA on these twelve series. The wide range of shapes displayed by the data sets used (see Figure 11) indicates that this result should generalize to a wide class of data distributions.

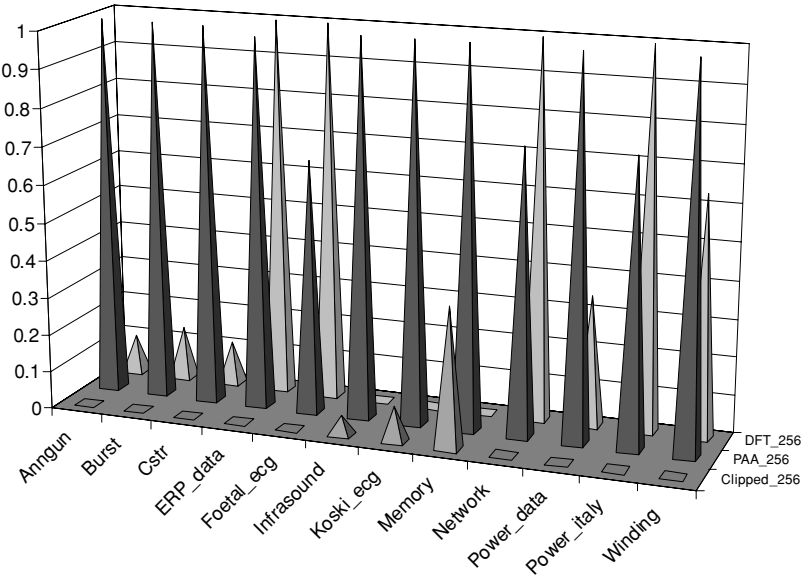

Figure 12. Number of disk accesses with lower bounding of Euclidean distance, normalized by the worst performing approach, using the three representations for the query length of 256 data points.

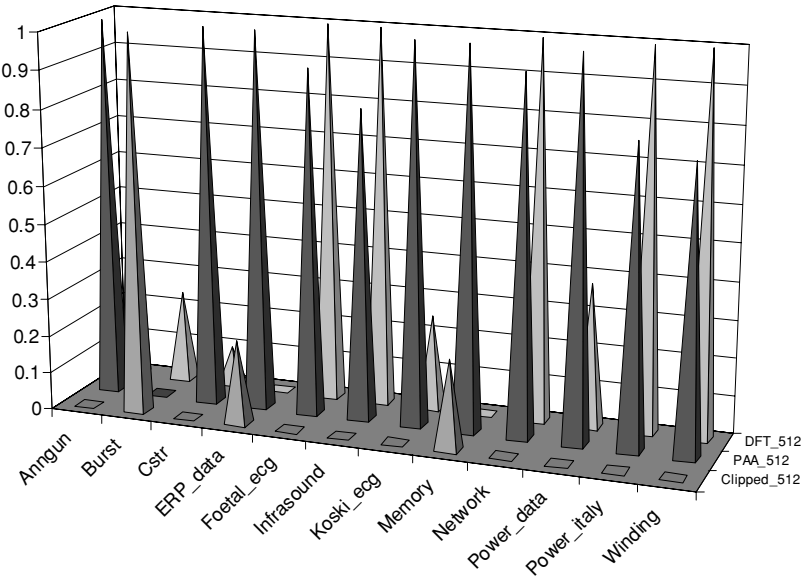

Figure 13. Number of disk accesses with lower bounding of Euclidean distance, normalized by the worst performing approach, using the three representations for the query length of 512 data points.

Table 4. Number of disk accesses required with DTW lower bounded distances.

		Query size 256				Query size 512			
Dataset	Size	Ratio	Clipped	PAA	DFT	Ratio	Clipped	PAA	DFT
Anngun	10, 001	714.2	**995**	9411	3074	694.6	**2**	5614	9164
Burst	9, 382	390.3	**1832**	6607	4269	511.8	**795**	4086	6572
Cstr	22, 500	474.4	**614**	12291	7904	612.7	**28**	7533	10573
ERP_data	198, 400	204.2	**6**	127520	122025	321.5	**1**	33210	92590
Foetal.ecg	20, 000	121.6	**2**	14394	11757	159.3	**2**	15641	6352
Infrasound	8, 192	401.1	**1314**	6262	4796	387.7	**72**	3119	968
Koski_ecg	144, 002	419.8	**370**	40955	3759	629.5	**711**	28829	27215
Memory	6, 875	396.9	1271	2528	**953**	715.8	**2293**	2739	2454
Network	18, 000	60.3	**434**	17583	17441	63	**91**	17074	17388
Power_data	35, 040	440.8	**42**	21260	15401	1103.9	**6**	9949	19174
Power_Italy	29, 931	210	**2**	29534	29575	286.2	**7**	28412	28946
Winding	17, 500	154.3	**1**	16464	14367	178.2	**1**	5257	16864

5.3. Indexing sliding windows with DTW

We extended the results described in Section 5.2 by conducting identical experiments, except lower bounded Dynamic Time Warping distance metrics were used. For clipped series, we use the lower bounded defined in Eq. (3). We used the DFT and PAA lower bounded distance defined in Zhu and Shasha (2003). Table 4 gives the number of disk accesses and Figures 14 and 15 show the normalized graphs. The results are even more conclusive than those with Euclidean distance. Clipping outperforms PAA and DFT on 11 out of 12 data sets with a query length of 256, and on all data sets with a query length of 512. Furthermore, on most data sets, DTW requires several orders of magnitude fewer disk accesses than both PAA and DFT. For example, Clipped outperforms the other two approaches by 127,000 disk accesses on the ERP_data.

5.4. Kolmogorov complexity distance

As discussed in Section 4, the ability to lossless compress clipped series means they can be used in ways not possible with other representations. We demonstrate this advantage by using a Kolmogorov estimate described in Section 4.1 to cluster time series. These experiments serve to demonstrate the flexibility of the clipped representation and highlight the usefulness of Kolmogorov complexity based clustering. To test the discriminatory power of the distance metric defined by Eq. (9), we took several data sets from the UCR Time Series Archive that have natural pairing. For example, in the Bouy sensor dataset, there are two time series, East and North, and in the Great Lakes dataset, there are two time series, Ontario and Erie. We would expect a clustering algorithm to group these series together. We identified twenty-four pairs, from a diverse collection of time series covering domains of finance, science, medicine and industry.

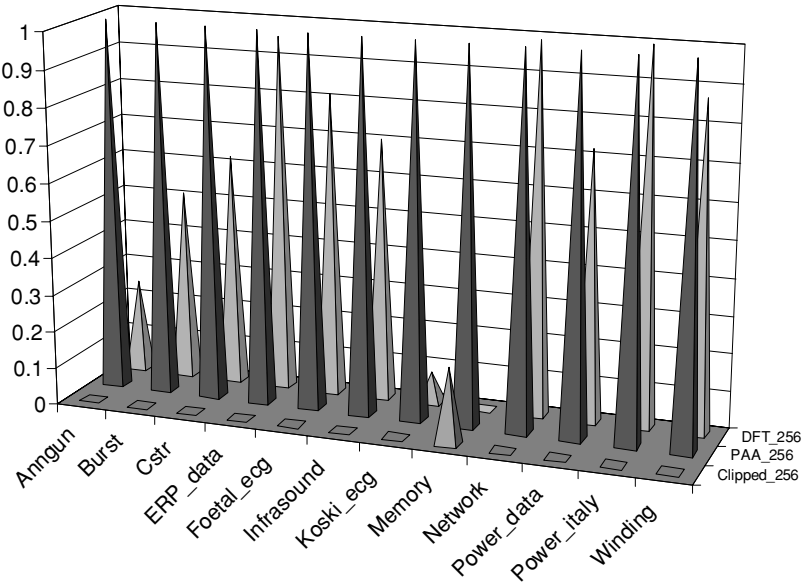

Figure 14. Number of disk accesses with lower bounding of DTW, normalized by the worst performing approach, using the three representations for the query length of 256 data points.

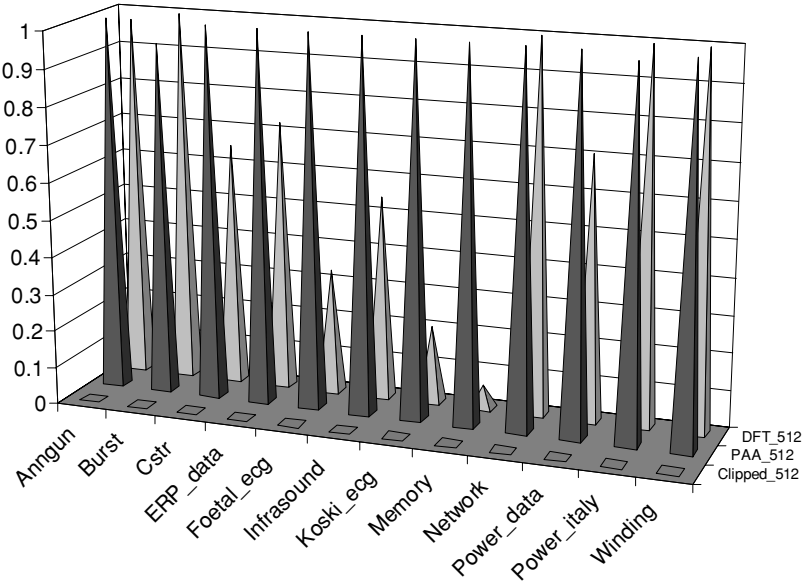

Figure 15. Number of disk accesses with lower bounding of DTW, normalized by the worst performing approach, using the three representations for the query length of 512 data points.

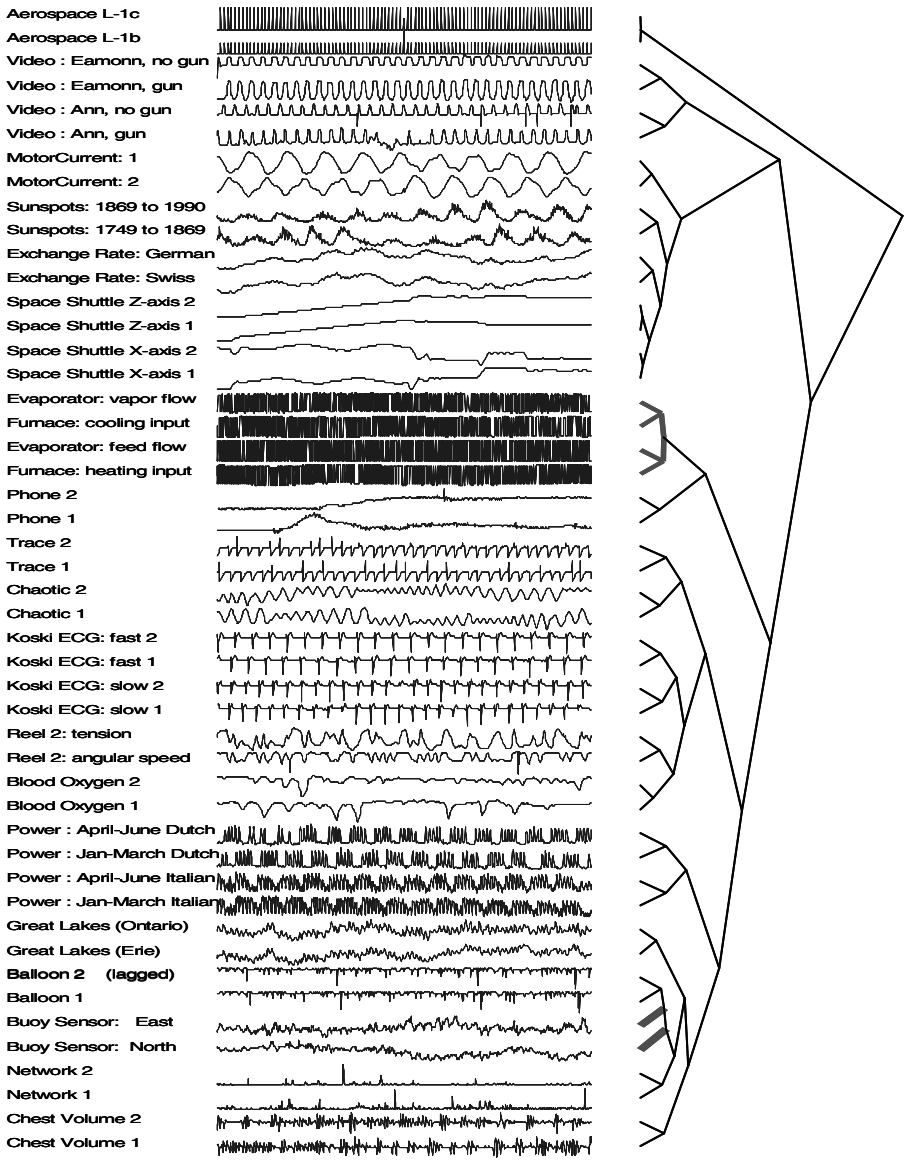

Figure 16. Forty-eight time series (in twenty-four pairs) clustered using the approach proposed in this paper. Bold lines denote incorrect subtrees.

The dendrogram produced from clustering this data with single linkage hierarchical clustering is shown in Figure 16. We have used a different clustering procedure to that described in Section 5.1 to demonstrate that the superior results from clipping are not an artifact of the algorithm used. The correct hierarchical clustering at the top of the tree is somewhat subjective, but at the lower level of the tree, we would hope to find a single bifurcation separating each pair in the dataset. Our metric, Q, for the quality

of clustering is therefore the number of such correct pairwise groupings divided by the number of pairs. For a perfect clustering, $Q = 1$, and for a random clustering, we would expect $Q = 0$ (the number of possible dendrograms of forty-eight objects is greater than 2.2∗10,106). The clustering shown in Figure 16 has a performance measure of $Q = 0.875$. The minor mistakes made, particularly with the grouping of the Furnace and Evaporator sequences are reasonable and plausible. For comparison, Markov models (Ge and Smyth, 2000) scored $Q = 0.458$, and ARIMA/ARMA models (Xiong and Yeung, 2002) scored $Q = 0.625$. In the former case, it took considerable parameter tuning to achieve this score, whereas our approach has no parameters. The Kolmogorov clustering algorithm is a technique with great potential (see Keogh et al., 2004). However, our primary purpose of using it in this paper is not to promote it as a clustering technique. Instead, our objective is to demonstrate the flexibility of clipping. The fact that clipped series can be lossless compressed with a model based algorithm means that a wide range of algorithms that cannot be used with other algorithms can be swiftly and usefully employed.

6. Conclusions

This paper assesses the usefulness of clipping for time series data mining based on similarity in shape. The theoretical properties of clipped series are discussed in Section 2. We show that under normality assumptions, distance measures on clipped data are asymptotically equivalent to distances measured on the raw data. We also derive lower bounding distance metrics for indexing that have the unique property that they can be used without transforming the query series. The practical benefits of clipped series are highlighted in Sections 3 and 4. Clipped series can be efficiently stored and manipulated, further compressed with RLE, and can be used in conjunction with a wide range of tests and algorithms specifically for binary data. These benefits are demonstrated with a series of four experiments in Section 5. In Section 5.1, we demonstrate that clipping series can produce better results than PAA and DFT at compression ratios of 32:1. Clipped series can be directly compared to uncompressed series and efficiently compressed with a run-length encoding. In Sections 5.2 and 5.3, we show that Euclidean and DTW distance based indexing with clipped series can be performed with up to five orders of magnitude fewer disk accesses than with DFT and PAA based indexing. In Section 5.4, we show that the fact a model based lossless compression algorithm can be applied means that techniques based on estimating the Kolmogorov complexity can be applied to clipped series. In the interests of competitive scientific inquiry, all data sets used in this work are available by emailing author Keogh.

These results demonstrate unequivocally that clipping is a useful transformation for time series data mining based on similarity in structure.

Our advice to the data miner confronted with very large time dependent data set is to start their analysis by clipping their data. It is an intuitive and simple transformation that retains much of the fundamental shape and structural information in the data. It can yield better results than more complex methods and can be used in conjunction with a wide range of algorithms.

Acknowledgments

This research was partly funded by the National Science Foundation under grant IIS-0237918 and by the UK Engineering and Physical Sciences Research Council under grant GR/T18479/01. We would like to thank the anonymous reviewers for their helpful comments.

References

Aach, J. and Church, G. 2001. Aligning gene expression time series with time warping algorithms. Bioinformatics, 17:495–508.

Agrawal, R., Faloutsos, C., and 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).

Austin, J. 1996. Distributed associative memories for high speed symbolic reasoning. International Journal of Fuzzy Sets and Systems, 82:223–233.

Austin, J., Davis, R., Fletcher, M., Jackson, T., Jessop, M., Liang, B., and Pasley, A. 2005. DAME: Searching large data sets within a grid-enabled engineering application. In Proceedings of the IEEE—Special Issue on Grid Computing, 93(3):496–509.

Austin, J. and Lees, K. 1998. A novel search engine based on correlation matrix memories. In Proceedings of the British Machine Vision Conference.

Austin, J. and Zhou, P. 1998. A binary correlation matrix memory *k-nn* classifier with hardware implementation. In Proceedings of the British Machine Vision Conference.

Bagnall, A.J. and Janacek, G.J. 2004. Clustering time series from ARMA models with clipped data. In Tenth International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2004), pp. 49–58.

Bagnall, A.J. and Janacek, G.J. 2005. Clustering time series with clipped data. Machine Learning, 58(2):151–178.

Bagnall, A.J., Janacek, G.J., and Powell, M. 2005. A likelihood ratio distance measure for the similarity between the fourier transform of time series. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005).

Basak, J., Sudarshan, A., Trivedi, D., and Santhanam, M.S. 2004. Weather data mining using independent component analysis. Journal of Machine Learning Research, 5:239–253.

Berndt, D. and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In Proceedings of AAAI-94 Workshop on Knowledge Discovery in Databases, pp. 229–248.

Bradley, J.V. 1968. Distribution-Free Statistical Tests. Prentice Hall.

Cai, Y. and Ng, R.T. 2004. Indexing spatio-temporal trajectories with chebyshev polynomials. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 599–610.

Chan, K.P. and Fu, A.W. 1999. Efficient time series matching with wavelets. In Proceedings of the 15th IEEE Int'l Conference on Data Engineering.

Chiu, B., Keogh, E., and Lonardi, S. 2003. Probabilistic discovery of time series motifs. In Ninth International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2003).

Faloutsos, C. Ranganathan, M., and Manolopoulos, Y. 1994. Fast subsequence matching in time-series databases. In Proc. ACM SIGMOD Conference, pp. 419–429.

Fowlkes, E. and Mallows, C. 1983. A method for comparing two hierarchical clusterings. Journal of the American Statistical Association, 78:553–569.

Galan, R.F., Sachse, S., Galizia, C.G., and Herz, A.V.M. 2004. Odor-driven attractor dynamics in the antennal lobe allow for simple and rapid olfactory pattern classification. Neural Computation, 16(1):999–1012.

Ganti, V., Gehrke, J., and Ramakrishnan, R. 1999. CACTUS-clustering categorical data using summaries. In Fifth International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 1999), pp. 73–83.

Ge, X. and Smyth, P. 2000. Deformable markov model templates for time-series pattern matching. In Sixth International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2000), pp. 81–90.

Glaz, J. and Balakrishnan, N. 1999. Scan Statistics and Applications. Birkhäuser.

Glaz, J., Naus, J., and Wallenstein, S. 2001. Scan Statistics, Springer.

Golomb, S.W. 1966. Run-length encodings. IEEE Trans. on Information Theory, 12(3):399–401.

Hodge, V.J. and Austin, J. 2003. An evaluation of standard spell checking algorithms and a binary neural approach. IEEE Transactions on Knowledge and Data Engineering, 15(5).

Huang, Z. 1998. Extensions to the *k*-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 2(3).

Huffman, D.A. 1952. A method for the construction of minimum-redundancy codes. Inst. Radio Eng., 40:1098–1101.

Kaufman, L. and Rousseeuw, P.J. 1990. Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons.

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

Keogh, E. 2002. Exact indexing of dynamic time warping. In Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), pp. 406–417.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. 2000. Locally adaptive dimensionality reduction for indexing large time series databases. In Proc. ACM SIGMOD Conference on Management of Data, pp. 151–162.

Keogh, E. and Folias, T. 2002. The UCR time series data mining archive. http://www.cs.ucr.edu/eamonn/TSDMA.

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

Keogh, E., Lonardi, S., and Ratanamahatana, C.A. 2004. Towards parameter-free data mining. In Tenth International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2004), pp. 206 – 215.

Keogh, E. and Pazzani, M. 2000. A simple dimensionality reduction technique for fast similarity search in large time series databases. In 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2000.

Knuth, D.E. 1985. Dynamic huffman coding. J. of Algorithms, 6(2):163–180.

Korn, F., Jagadish, H., and Faloutsos, C. 1997. Efficiently supporting ad hoc queries in large data sets of time sequences. In Proceedings of the ACM SIGMOD Int'l Conference on Management of Data.

Li, M., Chen, X., Li, X., Ma, B., and Vitanyi, P. 2003. The similarity metric. In Proceedings of the 14th annual ACM-SIAM Symposium on Discrete Algorithms, pp. 863–872.

Lin, J., Keogh, E., Lonardi, S., and 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.

Milligan, G.W., Sokol, L.M., and Soon, S.C. 1983. The effect of cluster size, dimensionality and the number of clusters on recovery of true cluster structure. IEEE Trans. PAMI, 5(1):40–47.

Morchen, F. 2003. Time series feature extraction for data mining using DWT and DFT. Technical Report 3, Departement of Mathematics and Computer Science Philipps-University Marburg.

Morinaka, Y., Yoshikawa, M., Amagasa, T., and Uemura, S. 2001. The L-index: An indexing structure for efficient subsequence matching in time sequence databases. In Proceedings of 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2001).

Ordonez, C. 2003. Clustering binary data streams with *k*-means. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pp. 12–19.

Rand, W.M. 1971. Objective criterion for evaluation of clustering methods. Journal of American Statistical Association, 66:846–851.

Ratanamahatana, C.A. and Keogh, E. 2005. Three myths about dynamic time warping data mining. In Proceedings of SIAM International Conference on Data Mining (SDM '05).

Rayner, J.C. and Best, D.J. (eds.) 2001. A Contingency Table Approach to Non-parametric Testing. Chapman and Hall.

Rice, S.O. 1944. Mathematical analysis of random noise. Bell Syst. Tech. J., 23:292–332.

Rissanen, J. and Langdon, G.G. 1979. Arithmetic coding. IBM J. of Res. and Dev., 23(2):149–162.

Schwarz, E.S. 1964. An optimum encoding with minimum longest code and total number of digits. Inf. and Control, 7:37–44.

Vlachos, M., Yu, P., and Castelli, V. 2005. On periodicity detection and structural periodic similarity. In Proceedings of the Siam International conference on Data Mining (SDM 05).

Weld, D.S. and de Kleer, J. (eds.) 1990. Readings in qualitative reasoning about physical systems. Morgan Kaufmann Publishers Inc.,

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

Yi, B.K. and Faloutsos, C. 2000. Fast time sequence indexing for arbitrary Lp norms. In Proceedings of the 26th International Conference on Very Large Data Bases (VLDB), pp. 385–394.

Zhu, Y. and Shasha, D. 2003. Warping indexes with envelope transforms for query by humming. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 181–192.