A Scalable Method for Time Series Clustering

Xiaozhe Wang[1], Kate A. Smith[1], Rob Hyndman[2] and Damminda Alahakoon[1]

[1] School of Business Systems and [2] Department of Econometrics and Business Statistics
Monash University, Victoria, Australia

{catherine.wang, kate.smith, damminda.alahakoon}@ infotech.monash.edu.au
rob.hyndman@buseco.monash.edu.au

Abstract

Time series clustering has become an important topic, particularly for similarity search amongst long time series such as those arising in bioinformatics. Unfortunately, existing methods for time series clustering that rely on the actual time series point values can become impractical since the methods do not scale well for longer time series, and many clustering algorithms do not easily handle high dimensional data. In this paper we propose a scalable method for time series clustering that replaces the time series point values with some global measures of the characteristics of the time series. These global measures are then clustered using a self-organising map, which performs additional dimension reduction. The proposed approach has been tested using some benchmark time series previously reported for time series clustering, and is shown to yield useful and robust clustering. The resulting clusters are similar to those produced by other methods, with some interesting variations that can be intuitively explained with knowledge of the global characteristics of the time series.

Key words: Time series, clustering, chaos, self-similarity, self-organising map

1 Introduction

The mining of time series data has attracted great attention in the data mining community in recent years (Bradley & Fayyad, 1998; Halkidi, Batistakis, & Vazirgiannis, 2001; Kalpakis, Gada, & Puttagunta, 2001; Keogh, Lin, & Truppel, 2003; Wang & Wang, 2000). Clustering time series and other sequences of data has become an important topic, motivated by several research challenges including similarity search of bioinformatics sequences, as well as the challenge of developing methods to recognize dynamic change in time series. There are two main categories in time series clustering as summarized by Keogh et al. (2003). "Whole clustering" is the clustering performed on many individual time series to group similar series into clusters. "Subsequence clustering" is based on sliding window extractions of a single time series and aims to find similarity and differences among different time windows of a single time series.

This paper focuses on whole clustering of time series using a variety of statistical measures to capture the global characteristics of the time series. This approach is a departure from the more common method of clustering time series based on distance measures within the space determined by the actual point values of the time series'. Many clustering algorithms have been applied to time series data based on the actual value of each time point. K-means clustering is the most commonly used clustering algorithm (Bradley & Fayyad, 1998; Halkidi et al., 2001), with the number of clusters K specified by the user. Hierarchical clustering generates a nested hierarchy of similar groups of time series according to a pairwise distance matrix of the series' (Keogh et al., 2003). One advantage of hierarchical clustering is that the number of clusters is not required to be provided as a parameter. Both of these clustering approaches however require that the length of each time series be identical due to the Euclidean distance calculation, and are unable to deal effectively with long time series due to poor scalability. Euclidean distance, while the most common metric (Agrawal, Faloutsos, & Swami, 1993; Chan & Fu, 1999; Chu & Wong, 1999; Faloutsos, Ranganathan, & Manolopoulos, 1994; Keogh, Chakrabarti, Pazzani, & Mehrotra, 2001; Popivanov & Miller, 2002) is certainly not the only metric available for measuring the similarity between data series. Autocorrelation has been proposed (Wang & Wang, 2000), along with a variety of other metrics in recent years including cepstrum (Kalpakis et al., 2001), piecewise normalization (Indyk, Koudas, & Muthukrishnan, 2000), cosine wavelets (Huntala, Karkkainen, & Toivonen, 1999), and piecewise probabilistic metrics (Keogh & Smyth, 1997). But the survey and empirical comparison in Keogh & Kasetty (2002) revealed that the Euclidean distance metric still performs best compared to others when tested on the same datasets. Dynamic Time Warping (DTW) has been applied in time series mining to resolve the difficulty caused when clustering time series of varying lengths in Euclidean space or containing possible out-of-phase similarities (Berndt & Clifford, 1994; Keogh, 2002; Ratanamahatana & Keogh, 2004).

While these methods have been quite effective at clustering moderate length time series, there are some well-appreciated drawbacks of the existing approaches. Euclidean distance is the most common method for discerning similarity in time series clustering, and it requires the time series being compared are of exactly the same dimensionality (length). Hierarchical clustering is one of the most widely used approaches, but is restricted to small datasets due to its quadratic computational complexity (Keogh et al., 2003). K-means is a faster method (Bradley & Fayyad, 1998) compared to hierarchical clustering, but the number of clusters has be pre-assigned which does not help to obtain natural clustering results. Dynamic Time Warping (DTW) (Keogh, 2002) can assist with clustering of different length time series, but is not defined if a single data point is missing (Ratanamahatana & Keogh, 2004).

This paper addresses the limitations of existing whole clustering approaches, and seeks to provide a scalable method for clustering time series of varying lengths, that is robust and invariant to missing data. Regardless of the length of the time series, a finite set of statistical measures will be used to capture the global nature of the time series. Certainly the idea of using feature selection to summarise time series data has been used before for different purposes, and using different features. For example, when classifying speech signals a variety of different features including statistical and speech data specific features were used as inputs to a classification model (Dellaert, Polzin, & Waibel, 1996).

The experimental results revealed the advantage of classification based on speech signal features. Following on from this idea, Nanopoulos et al. (2001) also proposed classification based on series features. Four different statistical time series measures were suggested for first-order and second-order decompositions, thus a total of eight features were used to form the input vector for a neural network classifier for a control chart classification application.

This paper differs in both the type of features extracted, and their use as inputs to a self-organised clustering process. Our proposed scalable method uses the idea of dimension reduction by feature extraction, and applies a statistical treatment to the analysis of time series data to arrive at a set of global measures of trend, seasonality, serial correlation, non-linear autoregressive structure, skewness, kurtosis (heavy-tailed distributions), and some other more specialized time series features such as self-similarity (long-range dependence), and measures of chaotic dynamics. These features concisely represent the relevant characteristics of each time series as a finite set of inputs to a clustering algorithm, that can then discern similarity and differences between the time series. For additional dimension reduction, we have used a self-organising map (Kohonen, 1995) to cluster the features, but any clustering technique could readily be applied, given that the inputs are now low-dimensional features of the original (potentially) high dimensional time series'. SOMs have been used before for time series clustering (Van Laerhoven, 2001; Debregeas & Hebrail, 1998) to project the time series onto a 2-dimensional space to visualise the clustering result. However the clusters in the SOM were generated in these studies using the actual time series data as inputs rather than global features as we propose in our scalable method. It appears that the SOM algorithm has not received much attention from the time series clustering community as yet.

The proposed approach is summarized in Figure 1. The measures are calculated on both the raw time series data Y_t, as well as the remaining time series after de-trending and de-seasonalising Y'_t (which we will refer to as "decomposed"). A total of 15 features are extracted from each time series (9 on the raw data and 6 on the decomposed data), which become inputs to the clustering process. The 15 features are a finite set used to measure the global characteristics of any time series, regardless of its length, enabling the proposed approach depicted in Figure 1 to be considered a scalable method.

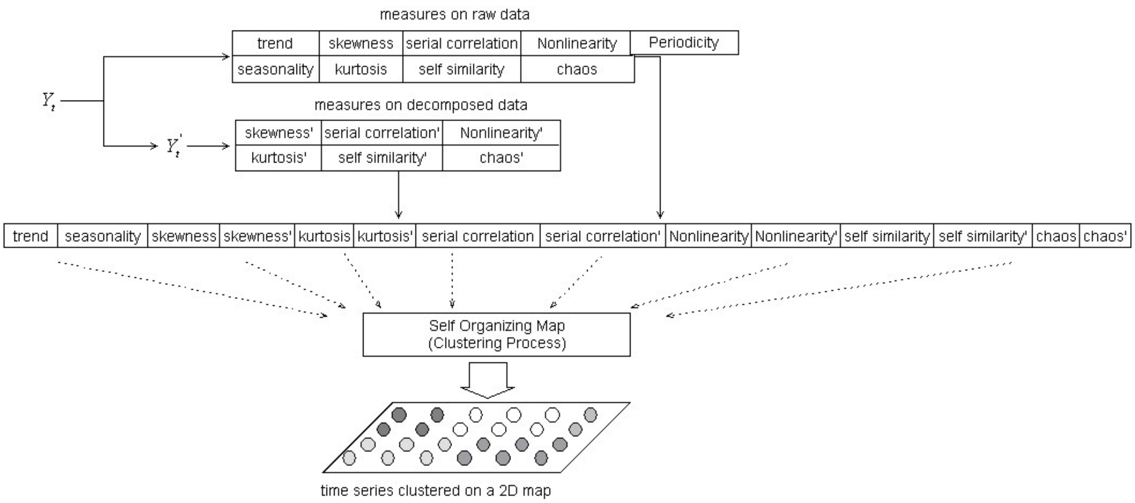


Figure 1: Proposed scalable method

The remainder of the paper is organized as follows. Section 2 describes the various measures used to extract the features of each time series. Section 3 describes the clustering algorithm – the self-organizing map (SOM) – and how it is applied to the global measures. Section 4 presents our experimental results based on some benchmarked time series data sets, and provides the evidence of the suitability of our proposed clustering method. Finally, Section 5 concludes the paper and outlines some directions for future research.

2 Measuring Characteristics of Time Series

A time series is the simplest form of temporal data and is a sequence of real numbers collected regularly in time, where each number represents a value. We represent a time series as an ordered set of m real-valued variables $Y_t = x_{t1}, ..., x_{tm}$. Time series can be described using a variety of qualitative terms such as seasonal, trending, noisy, non-linear, chaotic, etc. This section presents a collection of measures that seek to quantify these descriptors. In addition to the standard statistical measures of a time series used in (Nanopoulos et al., 2001), we have extended the scope to include a collection of special features such as long-range dependence and chaotic measures such as Lyapunov and Hurst exponents. These help to provide a rich portrait of the nature of a time series.

For each of the features described below, we have attempted to find the most appropriate way to measure the presence of the feature, ultimately scaling the metric to (0,1) to indicate the degree of presence of the feature, so that a measure near 0 for a certain time series indicates an absence of the feature, while a measure near 1 indicates a strong presence of the feature. The calculation of the measures and scaling transformations has been coded using R (see www.r-project.org).

2.1 Measuring trend and seasonality

Trend and seasonality are common features of time series, and it is natural to characterize a time series by its degree of trend and seasonality. In addition, once the trend and seasonality of a time series has been measured, we can de-trend and de-seasonalise the time series to enable additional features such as noise or chaos to be more easily detected. We have used the basic decomposition model in chapter 3 in (Makridakis, Wheelwright, & Hyndman, 1998):

- $Y_t^* = T_t + S_t + e_t$, where $Y_t^* = f_\lambda(Y_t)$, $f_\lambda(u) = (u^\lambda - 1)/\lambda$ denotes a Box-Cox transformation, T_t denotes the trend at time t and S_t denotes the seasonal component at time t.
- For a given transformation parameter λ, if the data are seasonal, the decomposition is carried out using the STL procedure (Cleveland, Cleveland, McRae, & Terpenning, 1990) (a filtering procedure for decomposing a time series into trend, seasonal, and remainder components) with fixed seasonality. The amount of smoothing for the trend is taken to be the default in the R implementation. If the data is nonseasonal, the S_t term is set to 0 and the estimation of T_t is carried out using a penalized regression spline (Wood, 2000) with smoothing parameter chosen using crossvalidation.
- The transformation parameter λ is chosen to make the residuals from the decomposition as normal as possible. We choose $\lambda \in (-1,1)$ to minimize the Shapiro-Wilk statistic (Royston, 1982). We only consider a transformation if the minimum of $\{Y_t\}$ is non-negative. If the minimum of Y_t is zero, we add a small positive constant (equal to 0.001 of the maximum of Y_t) to all values to avoid undefined results.

A trend pattern exists when there is a long-term change in the mean level (Makridakis et al., 1998). To estimate the trend, we can use smooth nonparametric method, for instance, penalized regression spline. Let Y_t be original data and $Y_t^{'}$ be detrended data. Then the measure of trend is $1-\frac{Var(Y_t^{'})}{Var(Y_t)}$.

A seasonal pattern exists when a time series is influenced by seasonal factors, such as month of the year and day of the week. The seasonality of a time series is defined as a pattern that repeats itself over fixed intervals of time (Makridakis et al., 1998). In general, the seasonality can be found by identifying a large autocorrelation coefficient or a large partial autocorrelation coefficient at the seasonal lag. Let Y_t be original data and $Y_t^{'}$ be de-seaonalised data. Then the measure of seasonality is $1-\frac{Var(Y_t^{'})}{Var(Y_t)}$.

2.2 Measure of Periodicity

If there is a seasonal pattern of a certain period, the length of that time period can be used as an additional measure. For time series with no seasonal pattern the period is set to 1. We measure the period using the following algorithm:

- Detrend time series using a regression spline with 3 knots
- Find $r_k = Corr(Y_t, Y_{t-k})$ (autocorrelation function) for all lags up to 1/3 of series length. And look for peaks and troughs in autocorrelation function.
- Frequency is first peak provided:
  a) there is also a trough before it.
  b) the difference between peak and trough is at least 0.1
  c) the peak corresponds to positive correlation.
- If no such peak is found, frequency is set to 1 (equivalent to non-seasonal).

2.3 Measure of serial correlation

To measure the degree of serial correlation of the data set, two possible measures could be used. $r_k = Corr(Y_t, Y_{t-k})$, where r_1 is the first-order autocorrelation, and Q_h the Box-Pierce statistic (Makridakis et al., 1998), where $Q_h = n\sum_{k=1}^{h} r_k^2$ and n is the length of the time series. We have used Box-Pierce statistics in our approach.

2.4 Measure of non-linear autoregressive structure

Nonlinear time series models have been used extensively in recent years to model complex dynamics not adequately represented using linear models (Harvill, Ray, & Harvill, 1999). We have used Teräsvirta's neural network test for nonlinearity (Blake & Kapetanios, 2003).

2.5 Measure of skewness

Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point. It is used to characterize the degree of asymmetry of values around the mean value. For univariate data $y(t)$, the skewness coefficient is $S = \frac{\sum_{t=1}^{n}(y(t)-\mu)^3}{n\sigma^3}$, where μ is the mean and σ is the standard deviation. We replace μ and σ^3 by their sample equivalents in computing skewness.

2.6 Measure of kurtosis (heavy-tails)

Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution. A data set with high kurtosis tends to have a distinct peak near the mean, decline rather rapidly, and have heavy tails. Data sets with low kurtosis tends to have a flat top near the mean rather than a sharp peak. A uniform distribution would be the extreme case. The kurtosis for a standard normal distribution is 3. The sample kurtosis coefficient is defined as $K = \frac{\sum_{t=1}^{n}(y(t)-\mu)^4}{n\sigma^4} - 3$, where μ is the mean and σ is the standard deviation. We replace μ and σ^4 by their sample equivalents in computing kurtosis.

2.7 Measuring self-similarity (long-range dependence)

Processes with long-rang dependence have attracted a good deal of attention from probabilists and theoretical physicists and Cox (1984) presented a review of second-order statistical time series analysis. The subject of self-similarity and the estimation of statistical parameters of time series in the presence of long-range dependence are becoming more common in several fields of science (Rose, 1996). The definition of self-similarity most related to the properties of time series is the self-similarity parameter Hurst exponent (*H*) (Willinger, Taqqu, Sherman, & Wilson, 1997) and is used as given in (Rose, 1996) as:

- Let Y_t be a covariance stationary stochastic process with mean μ, variance σ^2, and autocorrelation function r_k.
- For each $m = 1, 2, ...,$ let $\mathrm{Y}^{(m)} = (Y_k^{(m)}, k = 1, 2, 3, ...)$ denote the time series obtained by averaging the original series X over non-overlapping blocks of size, i.e. $Y^{(m)}$ is given by $Y_k^{(m)} = \frac{1}{m}(Y_{(k-1)m} + ... + Y_{km-1}), k \ge 1$
- A process Y is called (exactly) second-order self-similarity with self-similarity parameter $H = 1 - \beta/2$ if, for all $m = 1, 2, ..., VAR[Y^{(m)}] = \sigma^2 m^{-\beta}$ and $r^{(m)}(k) = r(k) = \frac{1}{2}((k+1)^{2H} - 2k^{2H} + |k-1|^{2H}), k \ge 0$, where $r^{(m)}$ denotes the autocorrelation function of $Y^{(m)}$
- Or, a process Y is called (asymptotically) second-order self-similarity, $r^{(m)}(k) \to r(k),$ as $m \to \infty$ (k large enough)

To estimate the Hurst parameter, traditional models such as autoregressive moving-average (ARMA) and autoregressive integrated moving-average (ARIMA) are only capable of exploring short-range correlations of data sets. Therefore, the class of fractional autoregressive integrated moving-average (FARIMA) processes (Hosking, 1984) lead from Brownian motion is a good estimation method for computing H. The random walk is defined as an FARIMA (0,1,0) process, where for a FARIMA (p,d,q) process the orders p and q are the classical ARMA parameters and $d = H - 0.5$ is the fractional difference parameter. We fit a FARIMA (0,d,0) by maximum likelihood which is approximated using the fast and accurate method of Haslett and Raftery (Haslett & Raftery, 1989). We then estimate the Hurst parameter using the relation H=d+0.5.

2.8 Measure of chaos

Many systems in the natural world that were previously considered random processes are now shown to be chaotic systems. Nonlinear dynamical systems often exhibit chaos, which is characterized by sensitive dependence on initial values, or more precisely by a positive Lyapunov Exponent (LE). Recognizing and quantifying chaos in time series represents an important step toward understanding the nature of random behavior and revealing the extent to which short-term forecasts may be improved (Lu, 1994).

LE as a measure of the divergence of nearby trajectories has been used to qualifying chaos by giving a quantitative value. The first algorithm of computing LE from time series was proposed by Wolf et al. (1985). It applies to continuous dynamical systems in an n-dimensional phase space. For a one-dimensional discrete time series, we used the method demonstrated by Hilborn (1994) to calculate LE of a one-dimensional time series:

- Let $Y_t = \{Y_1,...,Y_N\}$ denote the time series.
- We consider the rate of divergence of nearby points in the series by looking at the trajectories n periods ahead. Suppose Y_j and Y_i are two points such that $|\tilde{Y}_jY_i|$ is small. Then we define

$$\lambda(Y_i, Y_j) = \frac{1}{n}\log\frac{|Y_{j+n} - Y_{i+n}|}{|Y_j - Y_i|}$$

- We estimate the Lyapunov exponent of the series by averaging these values over all i, choosing Y_j as the closest point to Y_i where $i \neq j$.

  Thus, $\lambda = \frac{1}{N}\sum_{i=1}^{N}\lambda(Y_i, Y_i^*)$ where Y_i^* is the nearest point to Y_i.

2.9 Scaling transformations

The ranges of each of the above measures are quite dissimilar. In order to present the clustering algorithm with scaled data in the (0,1) range (so that certain features do not dominate the clustering), we perform a statistical transformation of the data. It is convenient to normalize variable ranges and using anything less than the most convenient methods hardly contributes to easy and efficient completion of a task (Pyle, 1999). While we have experimented with linear and logistic transformations of the measures, we prefer the following more statistically based approach.

In order to map the raw measures Q in the range $(0,\infty)$ to a scaled value q in the range (0,1) we use the transformation: $q = \frac{(e^{aQ}-1)}{(b+e^{aQ})}$, where a and b are constants to be chosen. We choose a and b such that q satisfies the following conditions: q has 90th percentile of 0.10 when Y_t is standard normal white noise and q has value of 0.9 for a well-known benchmark data set with the required feature: for example, for measuring serial correlation we use the Canadian Lynx data set.

Now that the global measures have been defined, we have a means of extracting the basic features of a time series. Using this finite set of measures to characterize the time series, regardless of their length or missing data, these features can be clustered using any appropriate clustering algorithm. In the following section we describe the self-organising map for clustering.

3 Clustering using a Self-Organising Map (SOM)

The SOM is a class of neural network algorithms in the unsupervised learning category, originally proposed by Kohonen in 1981-1982. The central property of SOM is that it forms a nonlinear projection of a high-dimensional data manifold on a regular, low-dimensional (usually 2D) grid (Kohonen, 1995). The clustered results can show the data clustering and metric-topological relations of the data items. It has a very powerful visualization output and is useful to understand the mutual dependencies between the variables and data set structure.

Like other neural network models, the learning algorithm for the SOM follows the basic steps of presenting input patterns, calculating neuron output, and updating weights. The only difference between the SOM and the more well-known (supervised) neural network algorithms lies in the method used to calculate the neuron output (a similarity measure), and the concept of a neighborhood of weight updates (Smith, 1999). The neural architecture of the SOM is shown in Figure 2, and the details of the SOM learning algorithm are presented below:

1. Initialize weights w to small random values, neighborhood size $N_m(0)$ to be large (but less than the number of neurons in one dimension of the array), and parameter function $\alpha(t)$ and $\sigma^2(t)$ to be between 0-1.
2. Present an input pattern x through the input layer and calculate the similarity (distance) of this input to the weight of each neuron j:

   $$d_j = \| x - w_j \| = \sqrt{\sum_{i=1}^{n} (x_i - w_{ij})^2}$$
3. Select the neuron with minimum distance as the winner m
4. Update the weights connecting the input layer to the winning neuron and its neighboring neurons according to the learning rule:

   $w_{ij}(t+1) = w_{ij}(t) + c[x_i - w_{ij}(t)]$,

   where $c = \alpha(t)\exp(-\| r_i - r_m \| / \sigma^2(t))$ for all neurons j in $N_m(t)$ and r_i is the position of the i^th^ neuron in the array
5. Continue from Step 2 for Ω epochs, then decrease neighborhood size, $\alpha(t)$ and $\sigma^2(t)$: repeat until weights have stabilized.

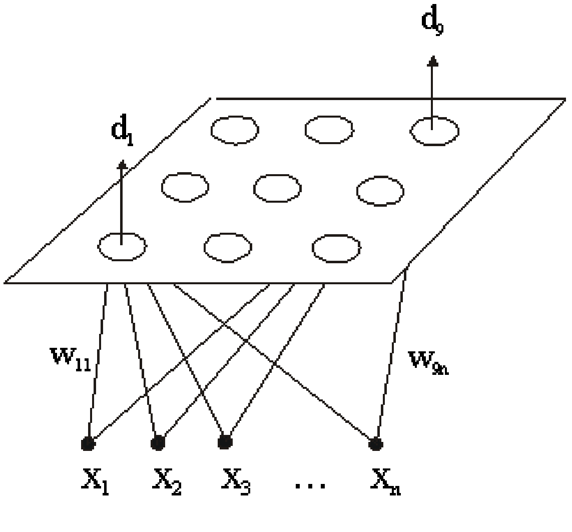


Figure 2: Neural architecture of SOM showing input vector (**x**), weights (**w** – not all weights are shown), and distance calculation as output

In our approach, a data set containing summarized features of many time series has been mapped onto a 2D map, with each time series (originally described by a vector of inputs $x(t) \in R^n$ where t is the index of the data set) described as a set of 15 inputs using the features shown in Table 1 (see Appendix) and discussed in the previous section. The output from the training process is the clustering of the time series data into groups visualized on a 2D map.

The SOM is both a projection method which maps high-dimensional data space into low-dimensional space, and a clustering method so that similar data samples tend to be mapped to nearby neurons. Since its originally introduction in the early 1980s, the SOM has evolved in purpose to be used as a statistical tool for multivariate analysis (Honkela, 1997). We have chosen to use a SOM for clustering in our approach due to its advantage of being quite robust to parameter selection, producing a natural clustering results, and providing superior visualization compared to other clustering methods, such as hierarchical and k-means. A comparison between different clustering algorithms however is not the focus of this paper.

4 Experimental Evaluation

While the scalability of the proposed approach should be apparent from the fact that we are extracting a finite set of features, it is most important to determine if the selected measures of global characteristics enable adequate clustering. To determine the effectiveness of the measures and the proposed approach, we have used the time series clustering benchmark datasets (Keogh, 2003) which consist of data from Space Shuttle telemetry, Exchange Rates and artificial sequences. The data is normalized so that the minimum value is zero and the maximum is one. There are 14 time series and each contains 1000 data points.

Appendix 1 shows the features extracted from the 14 time series, after applying the statistical transformation described in Section 2. Recall that a normalized value of near 1 indicates the strong presence of a certain feature, while near 0 indicates an absence of the feature. Using the data presented in Appendix 1 as inputs to the SOM, we have used the Viscovery SOMine software package (Eudaptics, 2002) to generate a 2D map as shown in Figure 3. A trial and error process was used to determine a few parameter settings by comparing the normalized distortion error and quantization error. The map shown in Figure 3 was generated after achieving the minimum errors for both normalized distortion and quantization. The map shown in Figure 3 shows that the 14 datasets have been clustered into 5 groups. The distance between datasets on the map indicates the similarity between them.

Of perhaps more interest however, when using benchmark datasets, is how the results compare to those obtained by other researchers. In order to compare with the benchmarking clusters generated using hierarchical clustering of the time series data point values (Keogh, 2003), we have re-interpreted the clusters generated by the SOM into a hierarchical structure as shown in Figure 4(b). The numbers across the top of the tree show the number of clusters, where 5 is our best performance based on quantization error. Compared to the hierarchical clustering structure (Figure 4(a)) from Keogh (2003), we can see that similar clusters have been obtained from our approach. But our clustering results are arguable better, or at least more intuitive. For example, series 1&4 and 9&10 have been grouped far from each other based on the hierarchical clustering using actual data point values, but a visual inspection of these series shows that they are actually quite similar in character. Using the global measures of our proposed approach, the clustering algorithm is aware of the "whole picture" and recognizes the similarity of these four time series.

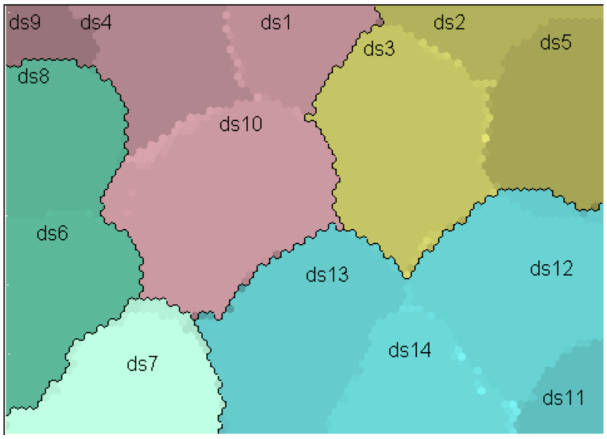
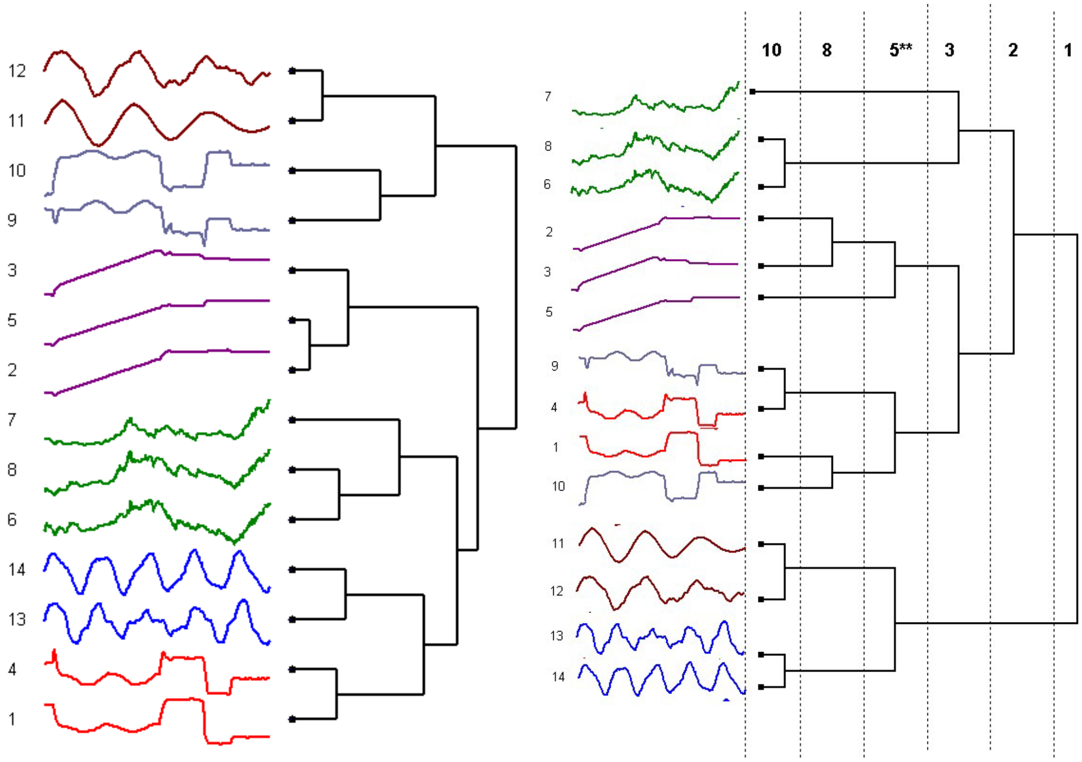


Figure 3 2D map from SOM process showing 5 clusters



(a) benchmark

(b) our result

Figure 4 Comparison of clustering results using hierarchical representation

We have also tested the robustness of the proposed approach to different scaling methods, and the significance of the different measures by generating numerous SOMs using different variations of inputs and data transformations. We have found that:

1. different scaling transformations of the features did not significantly affect the resulting map. We tested a linear transform method to map $(-\infty,\infty)$ to $[0,1]$: $v_n = \frac{v_i - \min(v_1 ... v_n)}{\max(v_1 ... v_n) - \min(v_1 ... v_n)}$, where v_n is normalized value and v_i is instance value, and the Softmax scaling (the logistic function) to map $(-\infty,\infty)$ to $(0,1)$: $v_n = \frac{1}{1+e^{-v_i}}$, where $e^{-v_i} = \frac{1}{e^{v_i}}$, v_n denotes the normalized value and v_i denotes the instance value.
2. repeating the experiments omitting some of the inputs (testing for example "raw data only", "decomposed data only", and omitting some measures such as nonlinearity and periodicity) showed that measuring nonlinearity is a necessary component for generate good clustering, but frequency is less important compared to the other measures. The best results were obtained when both raw and decomposed data were used.

5 Conclusions and Future Research

In this paper, we have proposed a new method for time series clustering and compared the results to a collection of benchmarked time series clustering data. Our empirical results demonstrate that our proposed clustering approach is able to cluster time series without using the actual data point values in the clustering algorithm. Using only a finite set of global measures, we can still achieve useful clustering. In fact, the knowledge provided to the clustering algorithm by the global measures appears to benefit the quality of the clustering results.

The advantage of our approach lies in its scalability and reduction of dimensionality, which overcomes a severe limitation of most of the existing approaches to time series clustering. The SOM has provided a nice visualization of the clustering results, although we have also represented the clusters using a hierarchical perspective for comparison with the benchmarked results. A further advantage of the proposed approach is that it is able to handle missing data since each of the measures we have used to extract features of the time series are still valid with missing data. The approach is parameter free, involving only the parameters of the selected clustering algorithm, with no new parameter introduced by the feature extraction process.

In future work we will be extending the ideas presented here to consider additional metrics to summarize time series data. Additional time series datasets will be analysed to evaluate the approach, including very long time series. We will also be exploring how the metrics change over time for chronological clustering, and identifying dynamic changes in time series.

References:

Agrawal, R., Faloutsos, C., & Swami, A. (1993, October 13-15, 1993). *Efficient Similarity Search in Sequence Databases.* Paper presented at the 4th international conference on foundations of data organization and algorithms, Chicago, IL, USA.

Berndt, D., & Clifford, J. (1994). *Using Dynamic Time Warping to Find Patterns in Time Series.* Paper presented at the AAAI'94 Workshop on Knowledge Discovery in Databases.

Blake, A. P., & Kapetanios, G. (2003). A Radial Basis Function Artificial Neural Network Test for Neglected Nonlinearity. *The Econometrics Journal, 6*(2), 357-373.

Bradley, P. S., & Fayyad, U. M. (1998, July 24-27). *Refining Initial Points for K-means Clustering.* Paper presented at the 15th international conference on machine learning, Madison, WI, USA.

Chan, K., & Fu, A. W. (1999, March 23-26). *Efficient Time Series Matching by Wavelets.* Paper presented at the 15th IEEE international conference on data engineering, Sydney, Australia.

Chu, K., & Wong, M. (1999, May 31-June 2, 1999). *Fast Time-series Searching with Scaling and Shifting.* Paper presented at the 18th ACM symposium on principles of database systems, Philadephia, PA, USA.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics, 6*, 3-73.

Cox, D. R. (1984). *Long-Range Dependence: A Review.* Paper presented at the Statistics: An Appraisal, 50th Anniversary Conference, Iowa State Statistical Laboratory.

Debregeas, A., & Hebrail, G. (1998, August 27-31). *Interactive Interpretation of Kohonen Maps Applied to Curves.* Paper presented at the 4th international conference of knowledge discovery and data mining, New York, NY, USA.

Dellaert, F. T., Polzin, T., & Waibel, A. (1996, October 3-6, 1996). *Recognizing Emotion in Speech.* Paper presented at the 4th International Conference on Spoken Language Processing (ICSLP'96), Philadelphia, PA, USA.

Eudaptics (2002). Viscovery SOMine, Eudaptics Software gmbh. www.eudaptics.com

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994, May 25-27). *Fast Subsequence Matching in Time-series Databases*. Paper presented at the ACM SIGMOD international conference on management of data, Minneapolis, MN, USA.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems (JIIS), 17*(2-3), 107-145.

Harvill, J. L., Ray, B. K., & Harvill, J. L. (1999). Testing for Nonlinearity in a Vector Time Series. *Biometrika, 86*, 728-734.

Haslett, J., & Raftery, A. E. (1989). Space-Time Modelling with Long-Memory Dependence: Assessing Ireland's Wind Power Resource (With Discussion). *Applied Statistics, 38*, 1-50.

Hilborn, R. C. (1994). *Chaos and Nonlinear Dynamics : an Introduction for Scientists and Engineers*. New York: Oxford University Press.

Honkela, T. (1997). *Self-Organizing Maps in Natural Language Processing*. Unpublished PhD thesis, Helsinki University of Technology, FINLAND.

Hosking, J. R. M. (1984). Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research, 20*(12), 1898-1908.

Huntala, Y., Karkkainen, J., & Toivonen, H. (1999, April, 1999). *Mining for Similarities in Aligned Time Series Using Wavelets*. Paper presented at the Data mining and knowledge discovery: theory, tools, and technology, SPIE proceedings series, Orlando, FL, USA.

Indyk, P., Koudas, N., & Muthukrishnan, S. (2000, September 10-14, 2000). *Identifying Representative Trends in Massive Time Series Data Sets Using Sketches*. Paper presented at the 26th International Conference on Very Large Data Bases, Cairo, Egypt.

Kalpakis, K., Gada, D., & Puttagunta, V. (2001, November 29-December 2, 2001). *Distance Measures for Effective Clustering of ARIMA Time-series*. Paper presented at the IEEE international conference on data mining, San Jose, CA, USA.

Keogh, E. (2002, Augeust 20-23). *Exact Indexing of Dynamic Time Warping*. Paper presented at the the 28th international conference on very large data bases, HongKong.

Keogh, E. (2003). *Clustering Data Set*. Retrieved, from the World Wide Web: http://www.cs.ucr.edu/~eamonn/TSDMA/index.html

Keogh, E., Chakrabarti, K., Pazzani, M. J., & Mehrotra, S. (2001, May 21-24). *Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases*. Paper presented at the ACM SIGMOD conference on managment of data, Santa Barbara, CA, USA.

Keogh, E., & Kasetty, S. (2002, July 23-26). *On the Need for Time Series Data Mining Benchmarks: a Survey and Empirical Demonstration*. Paper presented at the the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada.

Keogh, E., Lin, J., & Truppel, W. (2003, November 19-22, 2003). *Clustering of Time Series Subsequences is Meaningless: Implications for Past and Future Research*. Paper presented at the the 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA.

Keogh, E., & Smyth, P. (1997, August 14-17). *A Probabilistic Approach to Fast Pattern Matching in Time Series Databases*. Paper presented at the 3rd international conference on knowledge discovery and data mining, Newport Beach, CA, USA.

Kohonen, T. (1995). *Self-Organizing Maps* (Vol. 30): Springer Verlag.

Lu, Z.-Q. (1994). *Estimating Lyapunov Exponents In Chaotic Time Series With Locally Weighted Regression*. Ph.D. Dissertation, University of North Carolina, Chapel Hill.

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting Methods and Applications* (Third Edition ed.): John Wiley & Sons, Inc.

Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001). Feature-based Classification of Time-series Data, *International Journal of Computer Research* (pp. 49-61): Nona Science.

Popivanov, I., & Miller, R. J. (2002, February 26-March 1, 2002). *Similarity Search Over Time Series Data Using Wavelets*. Paper presented at the 18th international conference on data engineering, San Jose, CA, USA.

Pyle, D. (1999). *Data Preparation for Data Mining*. San Francisco, California: Morgan Kaufmann Publishers, Inc.

Ratanamahatana, C. A., & Keogh, E. (2004, April 22-24, 2004). *Making Time-series Classification More Accurate Using Learned Constraints*. Paper presented at the SIAM International Conference on Data Mining (SDM'04), Lake Buena Vista, Florida, USA.

Rose, O. (1996). *Estimation of the Hurst Parameter of Long-Range Dependent Time Series* (Rsearch Report 137): Institute of Computer Science, University of Wurzburg, Am Hubland.

Royston, P. (1982). An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. *Applied Statistics, 31*, 115-124.

Smith, K. A. (1999). *Introduction to Neural Networks and Data Mining for Business Applications*. Melbourne: Eruditions Publishing.

Van Laerhoven, K. (2001). Combining the Knohonen Self-organizing Map and K-means for On-line Classification of Sensor Data. In G. Dorffner & H. Bischof & K. Hornik (Eds.), *Artificial neural networks, Lecture Notes in Artificial Intelligence* (Vol. 2130, pp. 464-470): Springer Verlag.

Wang, C., & Wang, X. S. (2000, July 26-28). *Supporting Content-based Searches on Time Series via Approximation*. Paper presented at the 12th international conference on scientific and statistical database management, Berlin, Germany.

Willinger, W., Taqqu, M. S., Sherman, R., & Wilson, D. V. (1997). Self-similarity through High-variability: Statistical Analysis of Ethernet LAN traffic at the Source Level. *IEEE/ACM Transactions on Networking (TON), 5*(1), 71-86.

Wolf, A., Swift, J. B., Swinney, H. L., & Vastano, J. A. (1985). Determining Lyapunov Exponents from a Time Series. *PHYSICA D, 16*, 285-317.

Wood, S. N. (2000). Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. *J.R.Statist.Soc.B, 62*(2), 413-428.

Appendix

Table 1 Normalized global measures (features) for 14 benchmarked datasets

	trend	seasonal	serial.correlation	serial.correlation	non.linear	non.linear	skewness	skewness	kurtosis	kurtosis	Hurst	Hurst	Lyapunov	Lyapunov	frequency
	raw	raw	decomp	raw	decomp	raw	decomp	raw	decomp	raw	decomp	raw	decomp	raw	
ds1	0.906631525	0.000000000	0.997184892	0.999768005	0.008407811	0.340838199	0.061191486	0.230906114	0.997228725	0.016206653	0.999773448	0.999859983	0.674007416	0.721486360	0.000000000
ds2	0.997189380	0.000000000	0.997472048	0.999886810	0.003871297	0.999855167	0.012292712	0.176023843	0.793890813	0.005565127	0.999814161	0.999954177	0.699693625	0.425619734	0.000000000
ds3	0.993118170	0.000000000	0.996897473	0.999769271	0.005018138	0.518460560	0.004464978	0.427279432	0.611853105	0.211919067	0.999773448	0.999954177	0.655798514	0.399611286	0.000000000
ds4	0.789909980	0.000000000	0.997104503	0.999708272	0.143947965	0.006989367	0.620556326	0.108235609	0.996985081	0.017190088	0.999773448	0.999859983	0.732524142	0.774660543	0.000000000
ds5	0.994841260	0.225035585	0.998834861	0.999844967	0.000806959	0.855028621	0.023780215	0.207737405	0.063284542	0.010444605	0.999814161	0.999954177	0.000000000	0.000000000	0.825688073
ds6	0.929479140	0.000000000	0.983070588	0.999650677	0.467266724	0.120213309	0.017943943	0.135299555	0.062420456	0.018806577	0.999592719	0.999819271	0.988518647	0.956072362	0.000000000
ds7	0.902564657	0.000000000	0.996301030	0.999374069	0.307601985	0.019918941	0.040940241	0.455700462	0.058678223	0.721055453	0.999714856	0.999859983	0.904964665	0.919883498	0.000000000
ds8	0.932221246	0.000000000	0.988194073	0.999564807	0.493830277	0.065326841	0.045458892	0.051313069	0.077390290	0.007511894	0.999592689	0.999859983	1.000042120	1.000000000	0.000000000
ds9	0.917576842	0.000000000	0.992922309	0.999748758	0.025717849	0.149869695	0.703580356	0.046997870	0.999978202	0.003181698	0.999773448	0.999859983	0.729041958	0.769003409	0.000000000
ds10	0.870242365	0.000000000	0.997807645	0.999663759	0.116444476	0.042770290	0.097080653	0.315849687	0.255192289	0.019544460	0.999814161	0.999859983	0.680951011	0.771315320	0.000000000
ds11	0.181484294	0.897963360	0.999620100	0.999931646	0.073886264	0.004481898	0.043210861	0.011025449	0.026023600	0.007432458	0.999859983	0.999913465	0.000282953	0.002803827	1.000000000
ds12	0.221399961	0.858064226	0.999276957	0.999874556	0.011395104	0.001597036	0.012511036	0.024197790	0.121662705	0.010435832	0.999819271	0.999913465	0.000127451	0.003256432	0.981651376
ds13	0.050560225	0.733897471	0.999115216	0.999570224	0.007756723	0.024586044	0.000133823	0.005789897	0.016555523	0.008676909	0.999819271	0.999859983	0.002229978	0.006208606	0.617737003
ds14	0.077200618	0.884746077	0.997111902	0.999685049	0.035807275	0.040004401	0.002060387	0.007170094	0.040053250	0.003738721	0.999773448	0.999859983	0.002627704	0.005725584	0.608562691