Structure-based Statistical Features and Multivariate Time Series Clustering

Xiaozhe Wang Anthony Wirth Liang Wang
Department of Computer Science and Software Engineering
The University of Melbourne
{catwang,awirth,lwwang}@csse.unimelb.edu.au

Abstract

We propose a new method for clustering multivariate time series. A univariate time series can be represented by a fixed-length vector whose components are statistical features of the time series, capturing the global structure. These descriptive vectors, one for each component of the multivariate time series, are concatenated, before being clustered using a standard fast clustering algorithm such as k-means or hierarchical clustering. Such statistical feature extraction also serves as a dimension-reduction procedure for multivariate time series. We demonstrate the effectiveness and simplicity of our proposed method by clustering human motion sequences: dynamic and high-dimensional multivariate time series. The proposed method based on univariate time series structure and statistical metrics provides a novel, yet simple and flexible way to cluster multivariate time series data efficiently with promising accuracy. The success of our method on the case study suggests that clustering may be a valuable addition to the tools available for human motion pattern recognition research.

1. Introduction

The clustering of time series data has attracted great attention in the data mining community recently. Clustering is a descriptive task that seeks to identify homogeneous groups of objects based on the values of their attributes [24]. The clustering or classification of univariate time series has been recognized as an essential tool in process control, intrusion detection and character recognition, etc. [12]. As in many other real-world applications, the volumes of data collected in the form of time series are growing rapidly. Challenges are raised by the growth of data in three different directions:

- The length of time series, or the number of data points in each univariate time series, increases as time progresses;
- The number of objects in the data set could increase dramatically; and
- The dimension of the objects observed could increase, or their representation change from univariate to multivariate.

Multivariate time series datasets have appeared in both practical industry domains (for example, telecommunication and network) and scientific research fields (neural imaging and pattern recognition). In a later section, we include a case study in human motion pattern recognition.

Recent research has proposed many approaches for dealing with the considerable lengths and large number of objects now appearing in (univariate) time series datasets. Some popular methods include applying Fourier and wavelet transformations, as well as statistical parameter extraction for models such as the AutoRegressive Moving Average. Typically, the transformed series or extracted parameters are clustered using conventional clustering algorithms such as k-means clustering [44]. However, there is evidence that some of these methods are inappropriate for massive multivariate time series. They are either undefined and very expensive to compute on high-dimensional data, or restricted to data that satisfies strong linearity assumptions. The challenge of clustering large datasets of multivariate time series remains. Our intention was to produce a *simple*, *flexible* and *accurate* method for clustering multivariate time series.

In this paper, we propose a new method based on extracting structure-based statistical features for clustering multivariate time series. We wish to extract the most informative features or characteristics to represent the multivariate time series in our datasets. Such metrics, extracted from univariate time series structure, are used to construct new vectors for fast clustering algorithms, like k-means.

The remainder of the paper is organized as follows. Section 2 presents the related work aligned with our research focus. Section 3 explains our proposed method on clustering with new vectors from feature extraction on multivariate time series data set. Section 4 describes the details of the

1550-4786/07 $25.00 © 2007 IEEE
DOI 10.1109/ICDM.2007.103

statistical metrics extracted. The case study on human motion recognition with experimental results are demonstrated in Section 5.

2. Related Work

Clustering time series and other sequences of data has become an important topic, motivated by several research challenges including similarity search of medical and astronomical sequences, as well as the challenge of developing methods to recognize dynamic changes in time series [40]. However, most of the literature deals with methods and applications on univariate time series data: only a few applications have been reported on clustering multivariate time series data. There are three main tracks in current multivariate time series clustering.

Principal Component Analysis (PCA) PCA has been most commonly used in the limited number of applications on clustering multivariate time series data [43, 46]. Multivariate time series data have been clustered according to features found using PCA as the dimension-reduction tool for the feature space. Huang used PCA to split large multivariate time series clusters into smaller clusters [23]. In general, the number of principal components should be known as a predetermined parameter, which may be difficult to select. In very recent research, Singhai and Seborg [42] demonstrated clustering multivariate time series by combining two similarity factors: one is based on a PCA of the series, the other one is based on Mahalanobis distance between datasets. The final step is the application of the k-means algorithm to cluster multivariate time series based on the similarity factors calculated.

Hidden Markov Models (HMMs) HMMs have been used to cluster multivariate time series [35] based upon their ability to capture both the dependencies between variables and the serial correlations in the measurements [37]. An assumed probability distribution is required for the HMM representation for multivariate time series data.

Unfolding Data Wang and McGreavy [49] proposed unfolding each multivariate time series into a long row vector and supplying it to the *Autoclass* algorithm [7]. When the length of the multivariate time series increases or varies, this method could become computationally infeasible.

Therefore, from the above brief discussion on three related works, we find some drawbacks of these approaches, for instance, lack of flexibility, running time overheads,and computational restrictions. However, these approaches have paved a direction for our research in seeking a more flexible, simple and less complex means to deal with multivariate time series data for clustering or classification.

A number of authors have clustered time series based on structure-based similarity measures. Nanopoulos extracted four basic statistical features from Control Chart Pattern data and used them as input in a multi-layer perception neural network for time series classification [34]. Their experimental results showed the robustness of the method against noise and time series length compared to other methods that used every data point. By using two popular feature extraction techniques, the Discrete Wavelet Transform and the Discrete Fourier Transform, Mörchen has demonstrated the advantages of feature extraction for time series clustering in terms of computational efficiency and clustering quality on a benchmark dataset [33]. In a classification setting, parameters of a AutoRegression Moving Average (ARMA) model can be estimated and used as a limited dimensional representation for the original time series [11]. However, using ARMA parameters is not a reliable method because different sets of parameters can be obtained from time series with similar structure that could affect the clustering results dramatically. Ge and Smyth [13] used an approach for time series pattern matching based on segmental semi-Markov models: this proved to be flexible and accurate on real datasets. The time series is modeled as k distinct segments—with constraints on how the segments are *linked*—before the authors apply a Viterbi-like algorithm to compute the similarity measures. Compression-based Dissimilarity Measures are proposed by Keogh and others to compare long time series structure using co-compressibility as a dissimilarity measure [27]. This measure can be directly used in data mining algorithms, such as hierarchical clustering. Their extensive experiments have demonstrated the ability in handling different-length and missing-value time series. While the above works have shown utility in certain domains, most of them have high computational complexity and require that the data satisfy a number of conditions.

3. Clustering Using Extracted Features

We start with some notation. Let $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_Q\}$ represent a collection of Q multivariate time series. The series Y_i consists of n observations of a d-dimensional variable and will often be written as

$$Y_i = \{Y_{ijt}\}, \quad \text{for } j = 1, \ldots, d; t = 1, \ldots, n\,,$$

indicating dnQ observations in total.

3.1 Vector Construction based on Characteristic Metrics

Step I: Treat the j-th component of the i-th time series, that is $Y_{ij} = \{Y_{ij1}, \ldots, Y_{ijt}\}$, as a univariate time series. Then, for each Y_{ij}, produce a finite vector of L metrics $M = (m_1, m_2, \ldots, m_L)$ where each $m.$ is some statistical feature extracted from the time series. As such, each time series Y_{ij} is transformed into a new vector, M_{ij}.

Step II: The number of features (or metrics) that are actually used, L, can be based on a more generalized study of univariate time series structure-based characteristics. If the dataset comes from a particular domain with certain background knowledge, some sort of learning procedure such a feed-forward algorithm can select either a subset of the features or a convex combination of them.

Step III: Each multivariate time series therefore has d M-vectors: concatenating these into a single vector produces a simple dL-dimensional sketch of the Y_i. The data is now ready for clustering.

3.2 Clustering Based on Extracted Features

k-means clustering [31] is one of the simplest unsupervised learning clustering algorithms and is widely used for classification and clustering problems. Hierarchical clustering algorithms [25] also have a long and successful history. There are three major variants of hierarchical clustering: single link, complete link, and minimum variance. Of these three, the single-link and complete-link algorithms are most popular. In time series clustering research, k-means and hierarchical clustering have been commonly used with many measures of distance between series. However, there are obvious drawbacks or limitations for all of these clustering algorithms in handling time series data. Either they require the number of clusters, k, to be predefined as a parameter, or they require the time series length to be identical due to the distance calculation, or they are unable to deal effectively with lengthy time series due to poor scalability when some common used distance measure (for instance, Euclidean distance) is used in the clustering algorithm. However, when multivariate time series are transformed into representative vectors with our proposed structure-based statistical feature extraction, the latter drawbacks are not so apparent. Comparing these two basic algorithms, k-means is faster than hierarchical clustering [5], but the number of clusters has to be pre-assigned, which could be impractical in obtaining natural clustering results. In this paper, we test our method on a dataset which the number of clusters is known from prior classification work on the (training) data. Therefore, both clustering algorithms are adopted in the case study (details in Section 5) to demonstrate the robustness and reliability of the features extracted for clustering.

4. Structure-based Statistical Features Extraction

In this study, we investigated various data characteristics from diverse perspectives related to univariate time series structure-based characteristic identification and feature extraction. We selected the nine most informative, representative and easily-measurable characteristics to summarize the time series structure. Based on these identified characteristics, corresponding metrics are calculated for constructing the structure-based feature vectors.

4.1 Identified Structure-based Statistical Features

A univariate time series can be represented as an ordered set of n real-valued variables Y_1, ...,Y_n. Time series can be described using a variety of adjectives such as seasonal, trending, noisy, non-linear, chaotic, etc.

Three common data characterization methods are: (i) statistical and information-theoretic characterization, (ii) model-based characterization, and (iii) landmarking concepts [36] . We take the path of statistical feature extraction in this study. The extracted statistical features should carry summarized information of time series data, capturing the *global picture* based on the structure of the entire time series. After a thorough literature review, we propose a novel set of characteristic metrics to represent univariate time series and their structure-based features. This set of metrics not only includes conventional features (for example, trend) [1], but also cover many advanced features (for example, chaos) which are derived from research on new phenomena [26]. The corresponding metrics for the following structure-based statistical features form a rich portrait of the nature of a time series: Trend, Seasonality, Serial Correlation, Non-linearity, Skewness, Kurtosis, Self-similarity, Chaotic, and Periodicity. We now explain these in detail.

4.1.1 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 detrend and de-seasonalize the time series to enable additional features such as noise or chaos to be more easily detectable. A trend pattern exists when there is a long-term change in the (local) mean value [32]. To estimate the trend, we can

use a smooth nonparametric method, such as the penalized regression spline [53].

Seasonal factors, such as month of the year or day of the week can often affect time series data. The seasonality of a time series is defined as the presence of a pattern that repeats itself over fixed interval of time [32]. In general, the seasonality can be found by identifying a large autocorrelation coefficient or a large partial autocorrelation coefficient at the seasonal lag.

There are three main reasons for making a transformation after plotting the data: i) to stabilize the variance, ii) to make the seasonal effect additive, and iii) to make the data normally distributed [6]. The two most popularly used transformations, logarithms and square-roots, are special cases of the class of Box-Cox transformations [3], these are used to make the data appear normally distributed. Given a time series, Y_t, and a transformation parameter, λ, the transformed series is defined thus:

$$Y_t^* = (Y_t^\lambda - 1)/\lambda, \ \ (\lambda \neq 0) \qquad Y_t^* = \log Y_t, \ \ (\lambda = 0)\,.$$

This transformation applies to situations in which the dependent variable is known to positive. We have used the basic decomposition model in Chapter 3 of Makridakis's text [32]:

$$Y_t^* = T_t + S_t + E_t\,,$$

where Y_t^* denotes the series after Box-Cox transformation. At time t, T_t denotes the trend, S_t denotes the seasonal component, and E_t is the irregular (or remainder) component. For a given transformation parameter, λ, if the data are seasonal—that is, a frequency of periodicity parameter, generated from the data, is greater than 1—the decomposition is carried out using a Seasonal-Trend decomposition procedure based on the Loess (STL) procedure [8]. This is a filtering procedure for decomposing a time series into trend, seasonal, and remainder components, assuming fixed seasonality. The amount of smoothing for the trend is taken to be the default in the **R** implementation of the `stl` function. Otherwise, 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 [53] with the smoothing parameter chosen using cross validation. The transformation parameter λ is chosen to make the residuals from the decomposition as normal as possible in distribution. We choose $\lambda \in (-1, 1)$ to minimize the Sahpiro-Wilk statistic [39]. We only consider a transformation if the minimum of Y_t is non-negative. If the minimum of Y_t is 0, we add a small positive constant (equal to $1/1000$ of the maximum of Y_t) to all values to avoid undefined results.

In summary:

Y_t	original data
$X_t = Y_t^* - T_t$	de-trended data after Box-Cox transformation
$Z_t = Y_t^* - S_t$	de-seasonalized data after Box-Cox transformation
$Y_t^{'} = Y_t^* - T_t - S_t$	time series after trend and seasonality adjustment
$1 - \mathrm{Var}(Y_t^{'})/\mathrm{Var}(Z_t)$	a suitable measure of trend
$1 - \mathrm{Var}(Y_t^{'})/\mathrm{Var}(X_t)$	a suitable measure of seasonality

4.1.2 Periodicity

Since the periodicity is very important for determining the seasonality and examining the cyclic pattern of the time series, periodicity feature extraction is essential. Unfortunately, time series from some domains do not come with known frequencies or regular periodicities. Therefore, we propose a new algorithm to measure the periodicity in univariate time series. A time series is called *cyclic* if there is some fixed period after which a pattern repeats itself. Seasonal time series are a subset of cyclic time series in which the cycle time must belong to a special family such as one day, one week, one month or one year. For time series with no seasonal pattern, the frequency is set to 1. We measure the periodicity using following algorithm:

Algorithm: Periodicity measure extraction

1. Detrend time series using a regression spline with 3 knots
2. Find $r_k = corr(Y_t, Y_{t-k})$ (auto-correlation function) for all lags up to 1/3 of series length, then look for peaks and troughs in auto-correlation function
3. Frequency is the first peak satisfying the following conditions
   (a) there is also a trough before it
   (b) the difference between peak and trough is at least 0.01
   (c) the peak corresponds to positive correlation
4. If no such peak is found, frequency is set to 1 (equivalent to non-seasonal).

4.1.3 Serial Correlation

We use Box-Pierce statistics in our study to estimate the serial correlation measure, and to extract measures from both *raw* and *TSA* (*Trend and Seasonally Adjusted*) data. The Box-Pierce statistic [32] was introduced in 1970 to test

residuals from a forecast model [4]. It is a common portmanteau test for computing the measure. The Box-Pierce statistic is

$$Q_h = n \sum_{k=1}^{h} r_k^2 \,,$$

where n is the length of the time series, and h is the maximum lag being considered, usually 20.

4.1.4 Non-linear Autoregressive Structure

Non-linear time series models have been used extensively in recent years to model dynamics not adequately represented by linear models [19]. For example, the well-known *sunspot* data set [9] and *lynx* data set [16] have non-linear structure. In times of recession, many economic time series appear non-linear [14].

There are many approaches to test the non-linearity in time series models such as nonparametric kernel and neural network tests. The Neural Network test has been reported to have better reliability [29]. In our study, we used Terasvirta's neural network test [45] for measuring time series data nonlinearity, which can correctly model the nonlinear structure of time series data [28]. It is a test for neglected nonlinearity, likely to have power against a range of alternative based on the neural network model. The test is based on a function chosen as the activations of *phantom* hidden units. Refer to Terasvirta [45] for a detailed discussion on the testing procedures and formulas.

4.1.5 Skewness

Skewness is a measure of symmetry, or more precisely, the lack of symmetry in a distribution, or a data set. For univariate time series, Y_t, the skewness coefficient is

$$\frac{1}{n\sigma^3} \sum_{t=1}^{n} (Y_t - \overline{Y})^3 \,,$$

where $\overline{Y}$ is the mean, σ is the standard deviation, and n is the number of data points in the series. The skewness for a normal distribution is zero, and any symmetric data should have skewness near zero. Negative values for the skewness indicate that the data distribution is skewed left, and positive values indicate a right-skewed distribution.

4.1.6 Kurtosis

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, declines rather rapidly, and has heavy tails. A data set with low kurtosis tends to have a flat top near the mean rather than a sharp peak. For a univariate time series, Y_t, the kurtosis coefficient is

$$\frac{1}{n\sigma^4} \sum_{t=1}^{n} (Y_t - \overline{Y})^4 \,,$$

where $\overline{Y}$ is the mean, σ is the standard deviation, and n is the number of data points in the series. The kurtosis for the standard Normal distribution is 3. Therefore, the excess kurtosis is defined as

$$\frac{1}{n\sigma^4} \sum_{t=1}^{n} (Y_t - \overline{Y})^4 - 3 \,.$$

Positive excess kurtosis indicates a *peaked* distribution and negative excess kurtosis indicates a *flat* distribution.

4.1.7 Self-similarity

Processes with long-range dependence have attracted a good deal of attention from probabilist and theoretical physicists. In 1984, Cox [10] first presented a review of second-order statistical time series analysis. The subject of self-similarity (or *long-range dependence*) 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 [38]. Given this, we decided to include this feature into our feature selection set; so far this has been paid little attention in time series feature identification.

The definition of self-similarity most related to the properties of time series is the self-similarity, *Hurst exponent* (H) parameter [51]. The class of AutoRegressive Fractionally Integrated Moving Average (ARFIMA) processes has been recommended as a suitable estimation method for computing H [22]. We fit a *ARFIMA(0,d,0)* to maximum likelihood which is approximated by using the fast and accurate Haslett and Raftery method [20]. The Hurst parameter is estimated using the relation as $H = d + 0.5$ and this self-similarity feature can only be detected on the *raw* data of the time series.

4.1.8 Chaos

Many systems in nature that were previously considered as random processes are now categorized as chaotic systems. Nonlinear dynamic systems often exhibit chaos, which is characterized by sensitive dependence on initial values, or more precisely by a positive Lyapunov Exponent (*LE*). The *LE* is a measure of the divergence of nearby trajectories which can be used to qualify the notion of chaos. Recognizing and quantifying chaos in time series are important steps toward understanding the nature of random behavior, and reveal the dynamic feature of time series [30]. The first algorithm for computing the *LE*

of a time series was proposed by Wolf [52]. It applies to continuous dynamic systems in a n-dimensional phase space. For a one-dimensional discrete time series, we used the method demonstrated by Hilborn [21] to calculate *LE* from the *raw* time series data.

Algorithm: Lyapunov Exponent measure extraction for univariate time series

- Let Y_t denote the univariate time series
- Consider the rate of divergence of nearby points in the series by looking at the trajectories of n periods (time lag) ahead
- Suppose Y_i and Y_j are two points in Y_t, such that $|Y_j - Y_i|$ is small, then define

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

- Them, average these values over all i values, where N denotes the total number of Y_i, and choose Y_i^* as the closest point to Y_i, where $i \neq j$.

$$M = \frac{1}{N} \sum_{i=1}^{N} \lambda(Y_i, Y_i^*)$$

- Estimate *LE* of the series as

$$LE = \frac{e^M}{(1 + e^M)}$$

4.1.9 Decomposition and Scaling Transformation

In time series analysis, decomposition is a critical step for transforming the series into a format for statistical measurement [15]. Therefore, to obtain a precise and comprehensive calibration, some measures need to be calculated on both the raw time series data, Y_t, (referred to as *raw* data), as well as the remaining time series, $Y_t^{'}$, 'Trend and Seasonally Adjusted' (*TSA*) data. Note that some features such as periodicity can only be calculated on *raw* data.

For the nine selected features, four of them are calibrated on both *raw* and *TSA* data. Serial-correlation, Non-linearity, Skewness, and Kurtosis each contribute two metrics to our family. The remaining five features are calibrated only on *raw* data, leading to a total of thirteen.

The ranges of each the metrics extracted can vary significantly without the scaling transformation process. Each of the metrics is ultimately normalized to have a range of $[0, 1]$. A measurement near 0 for a certain time series indicates an absence of the particular feature, while a measurement near 1 indicates a strong presence of the feature identified. The data scaling transformation is required for clustering process in order to avoid certain measures dominating the clustering due to the data range itself.

There are many data transformation methods aviaible to normalize the metrics onto a certain required span, for instance, $[0, 1]$ in our work. We perform a statistical transformation of the data because it is convenient to normalize variable ranges across a new span from orignial data, while preserving underlying statistical properties. Compared to simple min-max transformation method (a linear transformation method), the statistical method also has a better control over the data distribution to obtain a reliable outcome, because if there are outliers in the original data, they can dominate the transformation results. Three transformations f_1, f_2 and f_3 are used to rescale a raw measure, Q, of various ranges to a new value q in the $[0, 1]$ range. (See detailed parameter settings and analysis in [50]).

$$f_1: \quad when \quad Q \in [0, \infty), \qquad q = \frac{(e^{aQ} - 1)}{(b + e^{aQ})}$$

$$f_2: \quad when \quad Q \in [0, 1], \qquad q = \frac{(e^{aQ} - 1)(b + e^{a})}{(b + e^{aQ})(e^{a} - 1)}$$

$$f_3: \quad when \quad Q \in (1, \infty), \qquad q = \frac{(e^{(Q-a)/b} - 1)}{(b + e^{(Q-a)/b})}$$

4.1.10 Computational Consideration

The computational time for calculating all statistical metrics is in general very low: most have linear running times. A few that revolve around long-range dependence have quadratic-time requirements, though we omit details in this paper.

5. A Case Study on Human Motion Time Series

We demonstrate our method by applying it to a real-world application on human activities and motion recognition.

5.1 Multivariate Time Series Representation for Motion Sequences

Various cues have been used in human motion recognition. These include key poses, optical flow, local descriptors, trajectories and joint angles from tracking. The features should be simple, intuitive and reliable to extract without manual labour. Human activities can be regarded as temporal variations of human silhouettes. Silhouette extraction from video is relatively easy for current imperfect vision techniques. So the method that we present here prefers to use (probably imperfect) space-time silhouettes for human activity representation. The silhouette images are centered and normalized on the basis of preserving the aspect

ratio of the silhouette so that the resulting images contain as much foreground as possible, do not distort the motion shape, and are of equal dimension for all input frames. To obtain a compact description and efficient computation, we use the Kernel Principal Component Analysis algorithm (KPCA) [41] to perform nonlinear dimension reduction. After obtaining the embedding space including the first d principal components, any one video can be projected into an associated trajectory in d-dimensional feature space as shown in Figure 1.

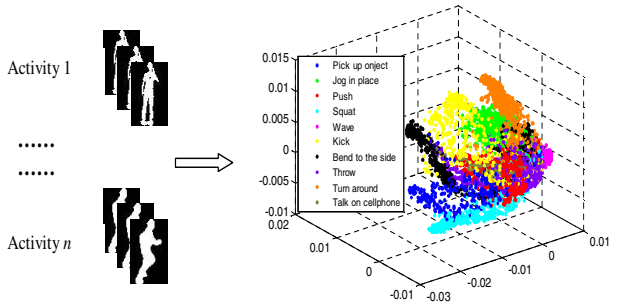

Figure 1. Transformation from silhouette data to multi-dimensional time series sequences (only 3 dimensions are shown)

The aim is to classify the observed silhouette sequences into known actions. To date, nearest-neighbour classification has been commonplace, rather than clustering algorithms.

5.2. Data Set Description

There is no common evaluation database in the domain of human activity recognition, so we use a recently collated database [47]. The dataset consists of 10 different activities performed by one person: pick up an object, jog in place, push, squash, wave, kick, bend to the side, throw, turn around, and talk on a cell phone. There are 10 different instances collected for each activity, hence 100 sequences in total. Different instances of the same activity may represent different rates of motion execution. This dataset is thus used to systematically examine the effect of the rate on activity recognition both between different activities and between different instances of the same person carrying out the same activity. Each activity has been represented in a multivariate time series format, with 25 indexed sequences recorded with 70 time intervals after the KPCA pre-processing procedure. An example of using multivariate time series to represent the activity *pick up an object* is demonstrated in Figure 2.

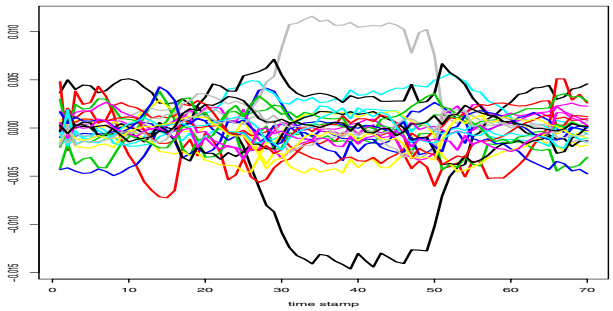

Figure 2. The activity *pick up an object* represented in multivariate time series format

5.3 Vector Construction and Feature Selection

As discussed in Section 4, a finite set of structure-based features are identified and their corresponding metrics are then calculated using the most suitable algorithms. Thirteen measures are summarized for each single univariate time series in the data set. Then, 100 new vectors are built for the 100 multivariate time series data sets provided for experiments. Before the measures for univariate time series data are combined and transformed into new vectors for multivariate time series data, a feature subset search is employed on a subset of the large data set. We decided to train on half the data (for feature selection); the training examples were equally spread between all of the activities.

In practice, we need to consider the generality of the method and the selection of the features for particular data sets or in certain application domains. In this section, a greedy forward search (FS) algorithm is employed as a searching mechanism to optimize the feature set (see Section 3.1). Forward search is a powerful general method for detecting multiple influences in a model [2]. It is only optimal for models which that have independent observations, such as linear and non linear regression, generalized linear models and multivariate analysis. However even in cases where the operators are not independent, it has been shown to be very robust in practice [14].

As illustrated in Table 1, classification accuracy reaches its peak after adding the top 10 metrics based on FS using the training dataset with known class labels. Then, a subset of these 10 metrics are selected to form the new vector before feeding them into clustering algorithms. Because the classification accuracy using all the metrics did provide a high level of accuracy and the difference between this and of top 10 features is not great, we decided to take both sets of metrics for new vector construction in our experiments.

Table 1. Classification accuracy for feature selection (* indicates the metrics extracted based on the TSA data). Each feature is added to the family of features above it.

Feature (metric)	Accuracy (%)
Hurst	17.6
Frequency	24.0
Kurtosis	32.8
Seasonal	31.2
Trend	29.6
Serial Correlation	38.4
Serial Correlation*	47.2
Nonlinearity	52.8
Skewness	54.4
Lyapunov	56.8
Skewness*	56.0
Monlinear*	51.2
Kurtosis*	45.6

5.4. Clustering Experiments

In k-means clustering, we used two methods: one by Hartigan and Wong [18], the other (the most commonly-used method) given by MacQueen [31]. Background knowledge of the dataset in our experiments suggested that $k = 10$ be assumed by the algorithm. Since the initial start of the cluster centers can affect the clustering result, we employed multiple runs with different random restarts for the clustering process in the experiments in order to achieve a more reliable outcome.

In hierarchical clustering, we choose three different clustering methods in the experiments. Ward's minimum variance method aims at finding compact, spherical clusters. The complete linkage method finds similar clusters. The Single Linkage (SL) method (which is closely related to minimal spanning trees) adopts a *friends of friends* clustering strategy [17]. The clustering process is performed on both new vectors based on 13 features and new vectors based on a selected subset of 10 features. Since visualization and dendrogram plots are not the main focus in this paper, we first used the two sets of derived vectors as clustering inputs to obtain the natural clustering from three methods under hierarchical clustering algorithm. Then, for evaluation, given the expected ten clusters as the best solution, the dendrogram trees obtained are cut arbitrarily into ten clusters and reconstruct the upper part of the tree from the cluster centers.

The quality of each clustering algorithm is measured by cluster purity, CP, which is a percentage count: $CP = (\mathrm{Count}_D \,/\, \mathrm{Count}_E) * 100\%$, where Count_D is the number of elements of the dominant class within that cluster, and Count_E is the expected number of objects in that cluster.

Table 2. Hierarchical clustering accuracy (%*CP*) using all metrics (without feature selection)

Activity	Complete	Ward	SL	*Mean*
pick	100	80	100	93.3
jog	100	100	100	100.0
push	60	90	60	70.0
squash	90	80	90	86.7
wave	50	50	50	50.0
kick	90	100	100	96.7
bend	100	90	100	96.7
throw	70	100	80	83.3
turn	100	100	100	100.0
talk	50	90	100	80.0
Mean	**81**	**88**	**88**	**85.7**

Hierarchical clustering results are shown in Tables 2 and 4, and k-means clustering algorithms results are shown in Tables 3 and 5. If we focus on the average performance of a particular algorithm over all clusters, Hierarchical and k-means perform similarly. The results demonstrate that our features are not model- or algorithm-dependent, which shows their generality and adaptability. We note also that the response to feature selection is not uniform. The Ward and SL variants of hierachical clustering are improved, whereas the worse-performing Complete-link approach becomes even worse.

Clearly *jog in place* is the best-recognized activity, whereas *wave* is the worst-recognized activity. It appears that the extracted structure-based features tend to be inadequate for describing the *wave* data set, which caused poor clustering results. On the other hand, our proposed features are suitable for summarizing activities like jogging, which was clustered with 100% accuracy. This study of human motion has provided us with an understanding of the strengths and weaknesses of our approach, both of which we hope to address in future.

6. Conclusions

In this paper, we proposed a new method to convert multivariate time series into vectors consisting of statistical measures extracted based on univariate time series structure or global characteristics. The proposed method is applied on a real-world dataset of human activity pattern recognition. The empirical results demonstrated that our method is able to cluster multivariate time series data with high accuracy efficiently.

Table 3. k-means clustering accuracy (%*CP*) using all metrics (without feature selection)

Activity	Hartigan-Wong	MacQueen	*Mean*
pick	80	70	75.0
jog	100	100	100.0
push	90	70	80.0
squash	90	70	80.0
wave	60	70	65.0
kick	80	90	85.0
bend	90	90	90.0
throw	100	80	90.0
turn	100	100	100.0
talk	100	100	100.0
Mean	**89**	**84**	**86.5**

Table 4. Hierarchical clustering accuracy (%*CP*) using a subset of metrics (with feature selection)

Activity	Complete	Ward	SL	*Mean*
pick	100	80	100	93.3
jog	100	100	100	100.0
push	50	90	90	76.7
squash	80	80	90	83.3
wave	60	60	50	56.7
kick	60	100	100	86.7
bend	100	100	100	100.0
throw	40	90	90	73.3
turn	60	100	100	86.7
talk	100	100	100	100.0
Mean	**75**	**90**	**92**	**85.7**

Table 5. k-means clustering accuracy (%*CP*) using a subset of metrics (with feature selection)

Activity	Hartigan-Wong	MacQueen	*Mean*
pick	80	80	80.0
jog	100	100	100.0
push	90	90	90.0
squash	90	70	80.0
wave	60	80	70.0
kick	90	80	85.0
bend	100	100	100.0
throw	90	90	90.0
turn	100	100	95.0
talk	100	100	100.0
Mean	**90**	**88**	**89.0**

References

[1] J. Armstrong. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Kluwer Academic Pub, 2001.

[2] A. Atkinson and M. Riani. *Robust diagnostic regression analysis*. Springer New York, 2000.

[3] G. Box and D. Cox. An Analysis of Transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[4] G. Box and D. Pierce. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970.

[5] P. Bradley and U. Fayyad. Refining Initial Points for K-Means Clustering. *Proc. 15th International Conf. on Machine Learning*, 727, 1998.

[6] C. Chatfield. *The Analysis of Time Series: An Introduction*. CRC Press, 2004.

[7] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, 180, 1996.

[8] R. Cleveland, W. Cleveland, J. McRae, and I. Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[9] W. Cleveland. *The elements of graphing data*. Wadsworth Publ. Co. Belmont, CA, USA, 1985.

[10] D. Cox. Long-range dependence: a review. Statistics: an appraisal. *Proc. 50th Anniversary Conf., Iowa State Statistical Laboratory, HA David and HT David, Eds., The Iowa State University Press*, pages 55–74, 1984.

[11] K. Deng, A. Moore, and M. Nechyba. Learning to Recognize Time Series: Combining ARMA models with memory-based learning. *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997.

[12] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. *ACM SIGMOD Record*, 23(2):419–429, 1994.

[13] X. Ge and P. Smyth. Deformable Markov model templates for time-series pattern matching. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, 2000.

[14] L. Grossi and M. Riani. Robust Time Series Analysis Through the Forward Search. *Proc. of the 15th Symposium of Computational Statistics, Berlin, Germany*, pages 521–526, 2002.

[15] J. Hamilton. *Time series analysis*. Princeton University Press Princeton, NJ, 1994.

[16] D. Hand. *A Handbook of Small Data Sets*. Chapman & Hall/CRC, 1994.

[17] J. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc. New York, NY, USA, 1975.

[18] J. Hartigan and M. Wong. A K-means clustering algorithm. *JR Stat. Soc., Ser. C*, 28:100–108, 1979.

[19] J. Harvill and B. Ray. Testing for Nonlinearity in a Vector Time Series. *Bionmetrika*, 86:728–734, 1999.

[20] J. Haslett and A. Raftery. Space-Time Modelling with Long-Memory Dependence: Assessing Ireland's Wind Power Resource. *Applied Statistics*, 38(1):1–50, 1989.

[21] R. Hilborn. *Chaos and nonlinear dynamics*. Oxford University Press New York, 1994.

[22] J. Hosking. Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research*, 20(12), 1984.

[23] Y. Huang, J. Gertler, and T. McAvoy. Fault isolation by partial PCA and partial NLPCA. *Preprints of the 14th IFAC World Congress (Beijing, China)*, pages 545–550, 1999.

[24] A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988.

[25] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[26] E. Joseph. Chaos Driven Futures. *Future Trends Newsletter*, 24(1):1, 1993.

[27] E. Keogh, S. Lonardi, and C. Ratanamahatana. Towards parameter-free data mining. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215, 2004.

[28] M. La Rocca and C. Perna. Subsampling Model Selection in Neural Networks for Nonlinear Time Series Analysis. 2004.

[29] T. Lee, H. White, and C. Granger. Testing for neglected nonlinearity in time series models: a comparison of neural network methods and alternative tests. *Essays in econometrics: Spectral analysis, seasonality, nonlinearity, methodology, and forecasting table of contents*, pages 208–229, 2001.

[30] Z. Lu. *Estimating Lyapunov Exponents in Chaotic Time Series with Locally Weighted Regression*. PhD thesis, University of North Carolina at Chapel Hill, 1994.

[31] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.

[32] S. Makridakis, S. Wheelwright, and R. Hyndman. Forecasting Methods and Applications: John Wiley & Sons. *Inc. New York*, 1998.

[33] F. Morchen. Time series feature extraction for data mining using DWT and DFT. Technical report, Technical Report, 2003.

[34] A. Nanopoulos, R. Alcock, and Y. Manolopoulos. Feature-based Classification of Time-series Data. *International Journal of Computer Research, Special Issue: Information processing and technology*, 10(3):49–61, 2001.

[35] L. Owsley, L. Atlas, and G. Bernard. Automatic clustering of vector time-series for manufacturing machine monitoring. *ICASSP IEEE INT CONF ACOUST SPEECH SIGNAL PROCESS PROC*, 4:3393–3396, 1997.

[36] B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. *Proceedings of the Seventeenth International Conference on Machine Learning*, 951(2000):743–750, 2000.

[37] L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

[38] O. Rose. Estimation of the Hurst Parameter of Long-Range Dependent Time Series. *Research Report*, 137, 1996.

[39] J. Royston. An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. *Applied Statistics*, 31(2):115–124, 1982.

[40] J. Scargle. Timing: New Methods for Astronomical Time Series Analysis. *American Astronomical Society, 197th AAS Meeting,# 22.02; Bulletin of the American Astronomical Society,*, 32:1438, 2000.

[41] B. Scholkopf, A. Smola, and K. Muller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[42] A. Singhal and D. Seborg. Clustering multivariate time-series data. *Journal of Chemometrics*, 19, 2005.

[43] A. Sudjianto and G. Wasserman. A nonlinear extension of principal component analysis for clustering and spatial differentiation. *IIE transactions*, 28(12):1023–1028, 1996.

[44] D. P. Tamraparni Dasu, Deborah F. Swayne. Grouping Multivariate Time Series: A Case Study. *KDD-2006 workshop report: Theory and Practice of Temporal Data Mining,ACM SIGKDD Explorations Newsletter*, 8(2):96–97, 2006.

[45] T. Teräsvirta, C. Lin, and C. Granger. Power of the Neural Network Linearity Test. *Journal of Time Series Analysis*, 14(2):209–220, 1993.

[46] A. Trouve and Y. Yu. Unsupervised clustering trees by nonlinear principal component analysis. *Pattern Recognition and Image Analysis*, 2:108–112, 2001.

[47] A. Veeraraghavan, R. Chellappa, and A. Roy-Chowdhury. The Function Space of an Activity. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pages 959–968, 2006.

[48] R. Vilalta, C. Giraud-Carrier, P. Brazdil, and C. Soares. Using Meta-Learning to Support Data Mining. *International Journal of Computer Science Applications*, 1(1):31–45, 2004.

[49] X. Wang and C. McGreavy. Automatic classification for mining process operational data. *Ind. Eng. Chem. Res*, 37(6):2215–2222, 1998.

[50] X. Wang, K. Smith, and R. Hyndman. Characteristic-Based Clustering for Time Series Data. *Data Mining and Knowledge Discovery*, 13(3):335–364, 2006.

[51] W. Willinger, V. Paxson, and M. Taqqu. Self-similarity and heavy tails: Structural modeling of network trac. *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions, Birkhauser Verlag, Boston*, 1998.

[52] A. Wolf, J. Swift, H. Swinnery, and J. Vastano. Detecting Lyapunov Exponents from a Time Series. *Physica D*, 16:285–317, 1985.

[53] S. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(2):413–428, 2000.