Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform

Hui Zhang and Tu Bao Ho
School of Knowledge Science, Japan Advanced Institute of Science and Technology,
Asahidai, Nomi, Ishikawa 923-1292. Japan
E-mail: {zhang-h,bao}@jaist.ac.jp

Yang Zhang
Department of Avionics, Chengdu Aircraft Design and Research Institute,
No. 89 Wuhouci street, Chendu, Sichuan 610041. P.R. China
E-mail: v104@sohu.com

Mao-Song Lin
School of Computer Science, Southwest University of Science and Technology,
Mianyang, Sichuan 621002. P.R. China
E-mail: lms@swust.edu.cn

Keywords: time series, data mining, feature extraction, clustering, wavelet

Received: September 4, 2005

Time series clustering has attracted increasing interest in the last decade, particularly for long time series such as those arising in the bioinformatics and financial domains. The widely known curse of dimensionality *problem indicates that high dimensionality not only slows the clustering process, but also degrades it. Many feature extraction techniques have been proposed to attack this problem and have shown that the performance and speed of the mining algorithm can be improved at several feature dimensions. However, how to choose the appropriate dimension is a challenging task especially for clustering problem in the absence of data labels that has not been well studied in the literature.*

In this paper we propose an unsupervised feature extraction algorithm using orthogonal wavelet transform for automatically choosing the dimensionality of features. The feature extraction algorithm selects the feature dimensionality by leveraging two conflicting requirements, i.e., lower dimensionality and lower sum of squared errors between the features and the original time series. The proposed feature extraction algorithm is efficient with time complexity $O(mn)$ when using Haar wavelet. Encouraging experimental results are obtained on several synthetic and real-world time series datasets.

Povzetek: Članek analizira pomembnost atributov pri grupiranju časovnih vrst.

1 Introduction

Time series data are widely existed in various domains, such as financial, gene expression, medical and science. Recently there has been an increasing interest in mining this sort of data. Clustering is one of the most frequently used data mining techniques, which is an unsupervised learning process for partitioning a dataset into sub-groups so that the instances within a group are similar to each other and are very dissimilar to the instances of other groups. Time series clustering has been successfully applied to various domains such as stock market value analysis and gene function prediction [17, 22]. When handling long time series, the time required to perform the clustering algorithm becomes expensive. Moreover, the *curse of dimensionality*, which affects any problem in high dimensions, causes highly biased estimates [5]. Clustering algorithms depend on a meaningful distance measure to group data that are close to each other and separate them from others that are far away. But in high dimensional spaces the contrast between the nearest and the farthest neighbor gets increasingly smaller, making it difficult to find meaningful groups [6]. Thus high dimensionality normally decreases the performance of clustering algorithms.

Data Dimensionality Reduction aims at mapping high-dimensional patterns onto lower-dimensional patterns. Techniques for dimensionality reduction can be classified into two groups: feature extraction and feature selection [34]. Feature selection is a process that selects a subset of original attributes. Feature extraction techniques extract a set of new features from the original attributes through some functional mapping [43]. The attributes that are important to maintain the concepts in the original data are selected from the entire attribute sets. For time series data, the extracted features can be ordered in importance by using a suitable mapping function. Thus feature extraction is much popular than feature selection in time series mining

community.

Many feature extraction algorithms have been proposed for time series mining, such as Singular Value Decomposition (SVD), Discrete Fourier Transform (DFT), and Discrete Wavelet Transform (DWT). Among the proposed feature extraction techniques, SVD is the most effective algorithm with minimal reconstruction error. The entire time-series dataset is transformed into an orthogonal feature space in that each variable are orthogonal to each other. The time-series dataset can be approximated by a low-rank approximation matrix by discarding the variables with lower energy. Korn et al. have successfully applied SVD for time-series indexing [31]. It is well known that SVD is time-consuming in computation with time complexity $O(mn^2)$, where m is the number of time series in a dataset and n is the length of each time series in the dataset. DWT and DFT are powerful signal processing techniques, and both of them have fast computational algorithms. DFT maps the time series data from the time domain to the frequency domain, and there exists a fast algorithm called Fast Fourier Transform (FFT) that can compute the DFT coefficients in $O(mn\log n)$ time. DFT has been widely used in time series indexing [4, 37, 42]. Unlike DFT, which takes the original time series from the time domain and transforms it into the frequency domain, DWT transforms the time series from time domain into time-frequency domain.

Since the wavelet transform has the property of time-frequency *localization* of the time series, it means most of the energy of the time series can be represented by only a few wavelet coefficients. Moreover, if we use a special type of wavelet called *Haar wavelet*, we can achieve $O(mn)$ time complexity that is much efficient than DFT. Chan and Fu used the Haar wavelet for time-series classification, and showed performance improvement over DFT [9]. Popivanov and Miller proposed an algorithm using the Daubechies wavelet for time series classification [36]. Many other time series dimensionality reduction techniques also have been proposed in recent years, such as Piecewise Linear Representation [28], Piecewise Aggregate Approximation [25, 45], Regression Tree [18], Symbolic Representation [32]. These feature extraction algorithms keep the features with lower reconstruction error, the feature dimensionality is decided by the user given approximation error. All the proposed algorithms work well for time series with some dimensions because the high correlation among time series data makes it possible to remove huge amount of redundant information. Moreover, since time series data are normally embedded by noise, one byproduct of dimensionality reduction is noise shrinkage, which can improve the mining quality.

However, how to choose the appropriate dimension of the features is a challenging problem. When using feature extraction for classification with labeled data, this problem can be circumvented by the wrapper approach. The wrapper approach uses the accuracy of the classification algorithm as the evaluation criterion. It searches for features better suited to the classification algorithm aiming to improve classification accuracy [30]. For clustering algorithms with unlabeled data, determining the feature dimensionality becomes more difficult. To our knowledge, automatically determining the appropriate feature dimensionality has not been well studied in the literature, most of the proposed feature extraction algorithms need the users to decide the dimensionality or give the approximation error. Zhang et al. [46] proposed an algorithm to automatically extract features from wavelet coefficients using entropy. Nevertheless, the length of the extracted features is the same with the length of the original time series that can't take the advantage of dimensionality reduction. Lin et al. [33] proposed an iterative clustering algorithm exploring the multi-scale property of wavelets. The clustering centers at each approximation level are initialized by using the final centers returned from the coarser representation. The algorithm can be stopped at any level but the stopping level should be decided by the user. There are several feature selection techniques for clustering have been proposed [12, 15, 41]. However, these techniques just order the features in the absence of data labels, the appropriate dimensionality of features still need to be given by the user.

In this paper we propose a time-series feature extraction algorithm using orthogonal wavelet for automatically choosing feature dimensionality for clustering. The problem of determining the feature dimensionality is circumvented by choosing the appropriate scale of the wavelet transform. An ideal feature extraction technique has the ability to efficiently reduce the data into a lower-dimensional model, while preserving the properties of the original data. In practice, however, information is lost as the dimensionality is reduced. It is therefore desirable to formulate a method that reduces the dimensionality efficiently, while preserving as much information from the original data as possible. The proposed feature extraction algorithm leverages the lower dimensionality and lower errors by selecting the scale within which the detail coefficients have lower energy than that within the nearest lower scale. The proposed feature extraction algorithm is efficient that can achieve time complexity $O(mn)$ with Haar wavelet.

The rest of this paper is organized as follows. Section 2 gives the basis for supporting our feature extraction algorithm. The feature extraction algorithm and its time complexity analysis are introduced in Section 3. Section 4 contains a comprehensive experimental evaluation of the proposed algorithm. We conclude the paper by summarizing the main contributions in Section 5.

2 The basis of the wavelet-based feature extraction algorithm

Section 2.1 briefly introduces the basic concepts of wavelet transform. The properties of wavelet transform supporting our feature extraction algorithm are given in Section 2.2. Section 2.3 presents the Haar wavelet transform algorithm

used in our experiments.

2.1 Orthogonal Wavelet Transform Background

Wavelet transform is a domain transform technique for hierarchically decomposing sequences. It allows a sequence to be described in terms of an approximation of the original sequence, plus a set of details that range from coarse to fine. The property of wavelets is that the broad trend of the input sequence is preserved in the approximation part, while the localized changes are kept in the detail parts. No information will be gained or lost during the decomposition process. The original signal can be fully reconstructed from the approximation part and the detail parts. The detailed description of wavelet transform can be found in [13, 10].

The *wavelet* is a smooth and quickly vanishing oscillating function with good localization in both frequency and time. A *wavelet family* $\psi_{j,k}$ is the set of functions generated by dilations and translations of a unique *mother wavelet*

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k),\ j,k \in \mathbb{Z}$$

A function $\psi \in L^2(\mathbb{R})$ is an *orthogonal wavelet* if the family $\psi_{j,k}$ is an orthogonal basis of $L^2(\mathbb{R})$, that is

$$< \psi_{j,k}, \psi_{l,m} >= \delta_{j,l} \cdot \delta_{k,m},\ j,k,l,m \in \mathbb{Z}$$

where $< \psi_{j,k}, \psi_{l,m} >$ is the inner product of $\psi_{j,k}$ and $\psi_{l,m}$, and $\delta_{i,j}$ is the Kronecker delta defined by

$$\delta_{i,j} = \begin{cases} 0, \text{ for } i \neq j \\ 1, \text{ for } i = j \end{cases}$$

Any function $f(t) \in L^2(\mathbb{R})$ can be represented in terms of this orthogonal basis as

$$f(t) = \sum_{j,k} c_{j,k}\psi_{j,k}(t) \tag{1}$$

and the $c_{j,k} =< \psi_{j,k}(t), f(t) >$ are called the *wavelet coefficients* of $f(t)$.

Parsevel's theorem states that the energy is preserved under the orthogonal wavelet transform, that is,

$$\sum_{j,k\in\mathbb{Z}} | < f(t), \psi_{j,k} > |^2 = \|f(t)\|^2,\ f(t) \in L^2(\mathbb{R}) \tag{2}$$

(Chui 1992, p. 226 [10]). If $f(t)$ be the Euclidean distance function, Parsevel's theorem also indicates that $f(t)$ will not change by the orthogonal wavelet transform. The distance preserved property makes sure no false dismissal will occur with distance based learning algorithms [29].

To efficiently calculate the wavelet transform for signal processing, Mallat introduced the Multiresolution Analysis (MRA) and designed a family of fast algorithms based on it [35]. The advantage of MRA is that a signal can be viewed as composed of a smooth background and fluctuations or details on top of it. The distinction between the smooth part and the details is determined by the resolution, that is, by the scale below which the details of a signal cannot be discerned. At a given resolution, a signal is approximated by ignoring all fluctuations below that scale. We can progressively increase the resolution; at each stage of the increase in resolution finer details are added to the coarser description, providing a successively better approximation to the signal.

A MBA of $L^2(\mathbb{R})$ is a chain of subspace $\{V_j : j \in \mathbb{Z}\}$ satisfying the following conditions [35]:

(i) $\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \ldots \subset L^2(\mathbb{R})$
(ii) $\bigcap_{j\in Z} V_j = \{0\}, \bigcup_{j\in Z} V_j = L^2(\mathbb{R})$
(iii) $f(t) \in V_j \Longleftrightarrow f(2t) \in V_{j+1}; \forall j \in \mathbb{Z}$
(iv) $\exists\phi(t)$, called *scaling function*, such that $\{\phi(t-k) : k \in \mathbb{Z}\}$ is an orthogonal basis of V_0.

Thus $\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)$ is the orthogonal basis of V_j. Consider the space W_{j-1}, which is an orthogonal complement of V_{j-1} in V_j: $V_j = V_{j-1} \bigoplus W_{j-1}$. By defining the $\psi_{j,k}$ form the orthogonal basis of W_j, the basis

$$\{\phi_{j,k}, \psi_{j,k}; j \in \mathbb{Z}, k \in \mathbb{Z}\}$$

spans the space V_j:

$$\underbrace{V_0 \bigoplus W_0}_{V_1} \bigoplus W_1 \bigoplus \ldots \bigoplus W_{j-1} = V_j \tag{3}$$

Notice that because W_{j-1} is orthogonal to V_{j-1}, the ψ is orthogonal to ϕ.

For a given signal $f(t) \in L^2(\mathbb{R})$ one can find a scale j such that $f_j \in V_j$ approximates $f(t)$ up to predefined precision. If $d_{j-1} \in W_{j-1}, f_{j-1} \in V_{j-1}$, then f_j is decomposed into $\{f_{j-1}, d_{j-1}\}$, where f_{j-1} is the approximation part of f_j in the scale $j-1$ and d_{j-1} is the detail part of f_j in the scale $j-1$. The wavelet decomposition can be repeated up to scale 0. Thus f_j can be represented as a series $\{f_0, d_0, d_1, \ldots, d_{j-1}\}$ in scale 0.

2.2 The Properties of Orthogonal Wavelets for Supporting the Feature Extraction Algorithm

Assume a time series $\overrightarrow{X}$ ($\overrightarrow{X} \in \mathbb{R}^n$) is located in the scale J. After decomposing $\overrightarrow{X}$ at a specific scale j ($j \in [0, 1, \ldots, J-1]$), the coefficients $H_j(\overrightarrow{X})$ corresponding to the scale j can be represented by a series $\{\overrightarrow{A_j}, \overrightarrow{D_j}, \ldots, \overrightarrow{D_{J-1}}\}$. The $\overrightarrow{A_j}$ are called approximation coefficients which are the projection of $\overrightarrow{X}$ in V_j and the $\overrightarrow{D_j}, \ldots, \overrightarrow{D_{J-1}}$ are the wavelet coefficients in $W_j, \ldots, W_{J-1}$ representing the detail information of $\overrightarrow{X}$. From a single processing point of view, the approximation coefficients within lower scales correspond to the lower frequency part of the signal. As noise often exists in the high

frequency part of the signal, the first few coefficients of $H_J(\overrightarrow{X})$, corresponding to the low frequency part of the signal, can be viewed as a noise-reduced signal. Thus keeping these coefficients will not lose much information from the original time series $\overrightarrow{X}$. Hence normally the first k coefficients of $H_0(\overrightarrow{X})$ are chosen as the features [36, 9]. We keep all the approximation coefficients within a specific scale j as the features which are the projection of $\overrightarrow{X}$ in V_j. Note that the features retain the entire information of $\overrightarrow{X}$ at a particular level of granularity. The task of choosing the first few wavelet coefficients is circumvented by choosing a particular scale. The candidate selection of feature dimensions is reduced from $[1, 2, \ldots, n]$ to $[2^0, 2^1, \ldots, 2^{J-1}]$.

Definiton 2.1. *Given a time series $\overrightarrow{X} \in \mathbb{R}^n$, the features are the Haar wavelet approximation coefficients $\overrightarrow{A_j}$ decomposed from $\overrightarrow{X}$ within a specific scale j, $j \in [0, 1, \ldots, J-1]$.*

The extracted features should be similar to the original data. A measurement for evaluating the similarity/dissimilarity between the features and the data is necessary. We use the widely used sum of squared errors (square of Euclidean distance) as the dissimilarity measure between a time-series and its approximation.

Definiton 2.2. *Given a time-series $\overrightarrow{X} \in \mathbb{R}^n$, let $\widehat{\overrightarrow{X}} \in \mathbb{R}^n$ denote any approximation of $\overrightarrow{X}$, the sum of squared errors (SSE) between $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$ is defined as*

$$SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}}) = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{4}$$

Since the length of the features corresponding to scale j is smaller than the length of $\overrightarrow{X}$, we can't calculate the *SSE* between $\overrightarrow{X}$ and $\overrightarrow{A_j}$ by Eq. (4) directly. One choice is to reconstruct a sequence $\widehat{\overrightarrow{X}} \in \mathbb{R}^n$ from $\overrightarrow{A_j}$ then calculate the *SSE* between $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$. For instance, Kaewpijit et al. [23] used the correlation function of $\overrightarrow{X}$ and $\widehat{\overrightarrow{X}}$ to measure the similarity between $\overrightarrow{X}$ and $\overrightarrow{A_j}$. Actually, $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ is the same as energy difference between $\overrightarrow{A_j}$ and $\overrightarrow{X}$ with orthogonal wavelet transform. This property makes it possible to design an efficient algorithm without reconstructing $\widehat{\overrightarrow{X}}$.

Definiton 2.3. *Given a time-series $\overrightarrow{X} \in \mathbb{R}^n$, the energy of $\overrightarrow{X}$ is:*

$$E(\overrightarrow{X}) = \sum_{i=1}^{n} (x_i)^2 \tag{5}$$

Definiton 2.4. *Given a time-series $\overrightarrow{X} \in \mathbb{R}^n$ and its features $\overrightarrow{A_j} \in \mathbb{R}^m$, the energy difference (ED) between $\overrightarrow{X}$ and $\overrightarrow{A_j}$ is*

$$ED(\overrightarrow{X}, \overrightarrow{A_j}) = E(\overrightarrow{X}) - E(\overrightarrow{A_j}) = \sum_{i=1}^{n} (x_i)^2 - \sum_{i=1}^{m} (a_j^i)^2 \tag{6}$$

The $\widehat{\overrightarrow{X}}$ can be reconstructed by padding zeros to the end of $\overrightarrow{A_j}$ to make sure the length of padded series is the same as that of $\overrightarrow{X}$ and preforming the reconstruction algorithm with the padded series. The reconstruction algorithm is the reverse process of decomposition [35]. An example of $\overrightarrow{A_5}$ and the $\widehat{\overrightarrow{X}}$ reconstructed from $\overrightarrow{A_5}$ using Haar wavelet transform for a time series located in scale 7 is shown in Figure 1. From Eq. 2 we know the wavelet transform is energy preserved, thus the energy of approximation coefficients within the scale j is equal to that of their reconstructed approximation series, i.e., $E(\widehat{\overrightarrow{X}}) = E(\overrightarrow{A_j})$. As mentioned in Section 2.1, $V_J = V_{J-1} \bigoplus W_{J-1}$, we have $E(\overrightarrow{X}) = E(\overrightarrow{A_{J-1}}) + E(\overrightarrow{D_{J-1}})$. When decomposing the $\overrightarrow{X}$ to a scale j, from Eq. 3, we have $E(\overrightarrow{X}) = E(\overrightarrow{A_j}) + \sum_{i=j}^{J-1} E(D_j)$. Therefore, the energy difference between the A_j and $\overrightarrow{X}$ is the sum of the energy of wavelet coefficients located in the scale j and scales higher than j, i.e., $ED(\overrightarrow{X}, \overrightarrow{A_j}) = \sum_{i=j}^{J-1} E(\overrightarrow{D_i})$.

The $H_j(\overrightarrow{X})$ is $\{\overrightarrow{A_j}, \overrightarrow{D_j}, \ldots, \overrightarrow{D_{J-1}}\}$ and $H_j(\widehat{\overrightarrow{X}})$ is $\{\overrightarrow{A_j}, 0, \ldots, 0\}$. Since Euclidean distance also preserved with orthogonal wavelet transform, we have $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}}) = SSE(H_j(\overrightarrow{X}), H_j(\widehat{\overrightarrow{X}})) = \sum_{i=j}^{J-1} E(\overrightarrow{D_i})$. Therefore, the energy difference between $\widehat{\overrightarrow{X}}$ and $\overrightarrow{X}$ is equal to that between $\overrightarrow{A_j}$ and $\overrightarrow{X}$, that is

$$SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}}) = ED(\overrightarrow{X}, \overrightarrow{A_j}) \tag{7}$$

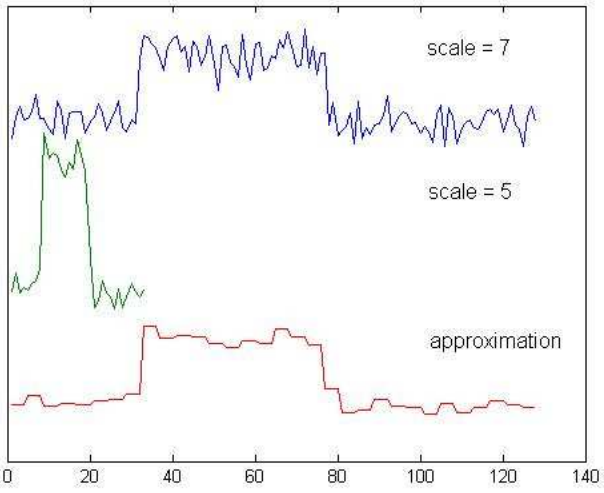

Figure 1: An example of approximation coefficients and their reconstructed approximation series

2.3 Haar Wavelet Transform

We use the Haar wavelet in our experiments which has the fastest transform algorithm and is the most popularly used

orthogonal wavelet proposed by Haar. Note that the properties mentioned in Section 2.2 are hold for all orthogonal wavelets such as the Daubechies wavelet family. The concrete mathematical foundation of the Haar wavelet can be found in [7]. The length of an input time series is restricted to an integer power of 2 in the process of wavelet decomposition. The series will be extended to an integer power of 2 by padding zeros to the end of the time series if the length of input time series doesn't satisfy this requirement.

The Haar wavelet has the mother function

$$\psi_{Haar}(t) = \begin{cases} 1, & \text{if } 0 < t < 0.5 \\ -1, & \text{if } 0.5 < t < 1 \\ 0, & \text{otherwise} \end{cases}$$

and scaling function

$$\phi_{Haar}(t) = \begin{cases} 1, & \text{for } 0 \leq t < 1 \\ 0, & \text{otherwise} \end{cases}$$

A time-series $\overrightarrow{X} = \{x_1, x_2, \ldots, x_n\}$ located in the scale $J = \log_2(n)$ can be decomposed into an approximation part $\overrightarrow{A_{J-1}} = \{(x_1+x_2)/\sqrt{2}, (x_3+x_4)/\sqrt{2}, \ldots, (x_{n-1}+x_n)/\sqrt{2}\}$ and a detail part $\overrightarrow{D_{J-1}} = \{(x_1-x_2)/\sqrt{2}, (x_3-x_4)/\sqrt{2}, \ldots, (x_{n-1}-x_n)/\sqrt{2}\}$. The approximation coefficients and wavelet coefficients within a particular scale j, $\overrightarrow{A_j}$ and $\overrightarrow{D_j}$, both having length $n/2^{J-j}$, can be decomposed from $\overrightarrow{A_{j+1}}$, the approximation coefficients within scale $j+1$ recursively. The ith element of $\overrightarrow{A_j}$ is calculated as:

$$a_j^i = \frac{1}{\sqrt{2}}(a_{j+1}^{2i-1} + a_{j+1}^{2i}), i \in [1, 2, \ldots, n/2^{J-j}] \quad (8)$$

The ith element of $\overrightarrow{D_j}$ is calculated as:

$$d_j^i = \frac{1}{\sqrt{2}}(a_{j+1}^{2i-1} - a_{j+1}^{2i}), i \in [1, 2, \ldots, n/2^{J-j}] \quad (9)$$

$\overrightarrow{A_0}$ has only one element denoting the global average of $\overrightarrow{X}$. The ith element of $\overrightarrow{A_j}$ corresponds to the segment in the series $\overrightarrow{X}$ starting from position $(i-1)*2^{J-j}+1$ to position $i*2^{J-j}$. The a_j^i is proportional to the average of this segment and thus can be viewed as the approximation of the segment. It's clear that the approximation coefficients within different scales provide an understanding of the major trends in the data at a particular level of granularity.

The reconstruction algorithm just is the reverse process of decomposition. The $\overrightarrow{A_{j+1}}$ can be reconstructed by formula (10) and (11).

$$a_{j+1}^{2i-1} = \frac{1}{\sqrt{2}}(a_j^i + d_j^i), i \in [1, 2, \ldots, n/2^{J-j}] \quad (10)$$

$$a_{j+1}^{2i} = \frac{1}{\sqrt{2}}(a_j^i - d_j^i), i \in [1, 2, \ldots, n/2^{J-j}] \quad (11)$$

3 Wavelet-based feature extraction algorithm

3.1 Algorithm Description

For a time-series, the features corresponding to higher scale keep more wavelet coefficients and have higher dimensionality than that corresponding to lower scale. Thus the $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ corresponding to the features located in different scales will monotonically increase when decreasing the scale. Ideal features should have lower dimensionality and lower $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ at the same time. But these two objectives are in conflict. Rate distortion theory indicates that a tradeoff between them is necessary [11]. The traditional rate distortion theory determines the level of inevitable expected distortion, D, given the desired information rate R, in terms of the rate distortion function $R(D)$. The $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ can be viewed as the distortion D. However, we hope to automatically select the scale without any user set parameters. Thus we don't have the desired information rate R, in this case the rate distortion theory can't be used to solve our problem.

As mentioned in the Section 2.2, the $SSE(\overrightarrow{X}, \widehat{\overrightarrow{X}})$ is equal to the sum of the energy of all removed wavelet coefficients. For a time series dataset having m time series, when decreasing the scale from the highest scale to scale 0, discarding the wavelet coefficients within a scale with lower energy ratio $(\sum_m E(\overrightarrow{D_j}) / \sum_m \sum_{i=J-1}^{0} E(\overrightarrow{D_i}))$ will not decrease the $\sum_m SSE$ much. If a scale j satisfies $\sum_m E(\overrightarrow{D_j}) < \sum_m E(\overrightarrow{D_{j-1}})$, removing the wavelet coefficients within this scale and higher scales achieves a local tradeoff of lower D and lower dimensionality for the dataset. In addition, from a noise reduction point of view, the noise normally found in wavelet coefficients within higher scales (high frequency part), and the energy of that noise is much smaller than that of the *true signal* with wavelet transform [14]. If the energy of the wavelet coefficients within a scale is small, their will be a lot of noise embedded in the wavelet coefficients; discarding the wavelet coefficients within this scale can remove more noise.

Based on the above reasoning, we leverage the two conflicted objectives by stopping the decomposition process at the scale $j^* - 1$, when $\sum_m E(\overrightarrow{D_{j^*-1}}) > \sum_m E(\overrightarrow{D_{j^*}})$. The scale j^* is defined as the appropriate scale and the features corresponding to the scale j^* are kept as the appropriate features. Note that by this process, at least $\overrightarrow{D_{J-1}}$ will be removed, and the length of $\overrightarrow{D_{J-1}}$ is $n/2$ for Haar wavelet. Hence the dimensionality of the features will smaller than or equal to $n/2$. The proposed feature extraction algorithm is summarized in pseudo-code format in Algorithm 1.

3.2 Time Complexity Analysis

The time complexity of Haar wavelet decomposition for a time-series is $2(n-1)$ bound by $O(n)$ [8]. Thus for a time-

Algorithm 1 The feature extraction algorithm

Input: a set of time-series $\{\overrightarrow{X_1}, \overrightarrow{X_2}, \ldots, \overrightarrow{X_m}\}$
for i=1 to m **do**
  calculate $\overrightarrow{A_{J-1}}$ and $\overrightarrow{D_{J-1}}$ for $\overrightarrow{X_i}$
end for
calculate $\sum_m E(\overrightarrow{D_1})$
exitFlag = true
for j=J-2 to 0 **do**
  for i=1 to m **do**
    calculate $\overrightarrow{A_j}$ and $\overrightarrow{D_j}$ for $\overrightarrow{X_i}$
  end for
  calculate $\sum_m E(\overrightarrow{D_j})$
  if $\sum_m E(\overrightarrow{D_j}) > \sum_m E(\overrightarrow{D_{j+1}})$ **then**
    keep all the $\overrightarrow{A_{j+1}}$ as the appropriate features for each time-series
    exitFlag = false
    break
  end if
end for
if exitFlag **then**
  keep all the $\overrightarrow{A_0}$ as the appropriate features for each time-series
end if

series dataset having m time-series, the time complexity of decomposition is $m * 2(n-1)$. Note that the feature extraction algorithm can break the loop before achieving the lowest scale. We just analyze the extreme case of the algorithm with highest time complexity (the appropriate scale $j = 0$). When $j = 0$, the algorithm consists of the following sub-algorithms:

- Decompose each time-series in the dataset until the lowest scale with time complexity $m * (2n - 1)$;
- Calculate the energy of wavelet coefficients with time complexity $m * (n - 1)$;
- Compare the $\sum_m E(\overrightarrow{D_j})$ of different scales with time complexity $\log_2(n)$.

The time complexity of the algorithm is the sum of the time complexity of the above sub-algorithms bounded by $O(mn)$.

4 Experimental evaluation

We use subjective observation and five objective criteria on nine datasets to evaluate the clustering quality of the K-means and hierarchical clustering algorithm [21]. The effectiveness of the feature extraction algorithm is evaluated by comparing the clustering quality of extracted features to the clustering quality of the original data. We also compared the clustering quality of the extracted appropriate features with that of the features located in the scale prior to the appropriate scale (prior scale) and the scale posterior to the appropriate scale (posterior scale). The efficiency of the proposed feature extraction algorithm is validated by comparing the execution time of the chain process that performs feature extraction firstly then executes clustering with the extracted features to that of clustering with original datasets directly.

4.1 Clustering Quality Evaluation Criteria

Evaluating clustering systems is not a trivial task because clustering is an unsupervised learning process in the absence of the information of the actual partitions. We used classified datasets and compared how good the clustered results fit with the data labels which is the most popular clustering evaluation method [20]. Five objective clustering evaluation criteria were used in our experiments: Jaccard, Rand and FM [20], CSM used for evaluating time series clustering algorithms [44, 24, 33], and NMI used recently for validating clustering results [40, 16].

Consider $G = G_1, G_2, \ldots, G_M$ as the clusters from a supervised dataset, and $A = A_1, A_2, \ldots, A_M$ as that obtained by a clustering algorithm under evaluations. Denote D as a dataset of original time series or features. For all the pairs of series $(\overrightarrow{D_i}, \overrightarrow{D_j})$ in D, we count the following quantities:

- a is the number of pairs, each belongs to one cluster in G and are clustered together in A.
- b is the number of pairs that are belong to one cluster in G, but are not clustered together in A.
- c is the number of pairs that are clustered together in A, but are not belong to one cluster in G.
- d is the number of pairs, each neither clustered together in A, nor belongs to the same cluster in G.

The used clustering evaluation criteria are defined as below:

1. Jaccard Score (Jaccard):
$$Jaccard = \frac{a}{a+b+c}$$
2. Rand statistic (Rand):
$$Rand = \frac{a+d}{a+b+c+d}$$
3. Folkes and Mallow index (FM):
$$FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$
4. Cluster Similarity Measure (CSM) :
The cluster similarity measure is defined as:
$$CSM(G, A) = \frac{1}{M} \sum_{i=1}^{M} \max_{1 \le j \le M} Sim(G_i, A_j)$$

where

$$Sim(G_i, A_j) = \frac{2|G_i \cap A_j|}{|G_i| + |A_j|}$$

5. Normalized Mutual Information (NMI):

$$NMI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} N_{i,j}\log(\frac{N \cdot N_{i,j}}{|G_i||A_j|})}{\sqrt{(\sum_{i=1}^{M}|G_i|\log\frac{|G_i|}{N})(\sum_{j=1}^{M}|A_j|\log\frac{A_j}{N})}}$$

where N is the number of time series in the dataset, $|G_i|$ is the number of time series in cluster G_i, $|A_j|$ is the number of time series in cluster A_j, and $N_{i,j} = |G_i \cap A_j|$.

All the used clustering evaluation criteria have value ranging from 0 to 1, where 1 corresponds to the case when G and A are identical. A criterion value is the bigger, the more similar between A and G. Thus, we prefer bigger criteria values. Each of the above evaluation criterion has its own benefit and there is no consensus of which criterion is better than other criteria in data mining community. To avoid biassed evaluation, we count how many times the evaluation criteria values produced from features are bigger/equal/smaller than that obtained from original data and draw conclusions based on the counted times.

4.2 Data Description

We used five datasets (CBF, CC, Trance, Gun and Reality) from the UCR Time Series Data Mining Archive [26]. (There are six classified datasets in the archive. The Auslan data is a multivariate dataset with which we can't apply the clustering algorithm directly. We used all the other five datasets for our experiments.) Other four datasets are downloaded from the Internet. The main features of the used datasets are described as below.

– Cylinder-Bell-Funnel (CBF): Contains three types of time series: cylinder (c), bell (b) and funnel (f). It is an artificial dataset original proposed in [38]. The instances are generated using the following functions:

$$c(t) = (6+\eta)\cdot\chi_{[a,b]}(t) + \varepsilon(t)$$
$$b(t) = (6+\eta)\cdot\chi_{[a,b]}(t-a)/(b-a) + \varepsilon(t)$$
$$f(t) = (6+\eta)\cdot\chi_{[a,b]}(b-t)/(b-a) + \varepsilon(t)$$

where

$$\chi_{[a,b]} = \begin{cases} 0, & \text{if } t < a \vee t > b \\ 1, & \text{if } a \le t \le b \end{cases}$$

η and $\varepsilon(t)$ are drawn from a standard normal distribution $N(0,1)$, a is an integer drawn uniformly from the range $[16, 32]$ and $b-a$ is an integer drawn uniformly from the range $[32, 96]$. The UCR Archive provides the source code for generating the samples. We generated 128 samples for each class with length 128.

– Control Chart Time Series (CC): This dataset has 100 instances for each of the six different classes of control charts.

– Trace dataset (Trace): The 4-class dataset contains 200 instances, 50 for each class. The dimensionality of the data is 275.

– Gun Point dataset (Gun): The dataset has two classes, each contains 100 instances. The dimensionality of the data is 150.

– Reality dataset (Reality): The dataset consists 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. Each cluster contains one time series with 1000 datapoints.

– ECG dataset (ECG): The ECG dataset was obtained from the ECG database at PhysioNet [19]. We used 3 groups of those ECG time-series in our experiments: Group 1 includs 22 time series representing the 2 sec ECG recordings of people having malignant ventricular arrhythmia; Group 2 consists 13 time series that are 2 sec ECG recordings of healthy people representing the normal sinus rhythm of the heart; Group 3 includes 35 time series representing the 2 sec ECG recordings of people having supraventricular arrhythmia.

– Personal income dataset (Income): The personal income dataset [1] is a collection of time series representing the per capita personal income from 1929-1999 in 25 states of the USA [1]. The 25 states were partitioned into two groups based on their growing rate: group 1 includes the east coast states, CA and IL in which the personal income grows at a high rate; the mid-west states form a group in which the personal income grows at a low rate is called group 2.

– Temperature dataset (Temp): This dataset is obtained from the National Climatic Data Center [2]. It is a collection of 30 time series of the daily temperature in year 2000 in various places in Florida, Tennessee and Cuba. It has temperature recordings from 10 places in Tennessee, 5 places in Northern Florida, 9 places in Southern Florida and 6 places in Cuba. The dataset is grouped basing on geographically distance and similar temperature trend of the places. Tennessee and Northern Florida form group 1. Cuba and South Florida form group 2.

– Population dataset (Popu): The population dataset is a collection of time series representing the population estimates from 1900-1999 in 20 states of USA [3]. The 20 states are partitioned into two groups based on their trends: group 1 consists of CA, CO, FL, GA, MD, NC, SC, TN, TX, VA, and WA having the exponentially increasing trend while group 2 consists of IL, MA, MI, NJ, NY, OK, PA, ND, and SD having a stabilizing trend.

[1]The 25 states included were: CT, DC, DE, FL, MA, ME, MD, NC, NJ, NY, PA, RI, VA, VT, WV, CA, IL, ID, IA, IN, KS, ND, NE, OK, SD.

As Gavrilov et al. [17] did experiments showing that normalization is suitable for time series clustering, each time series in the datasets downloaded from the Internet (ECG, Income, Temp, and Popu) are normalized by formula $x'_i = (x_i - \mu)/\sigma, i \in [1, 2, \ldots, n]$.

4.3 The Clustering Performance Evaluation

We took the widely used Euclidean distance for K-means and hierarchical clustering algorithm. As the Reality dataset only has one time series in each cluster that is not suitable for K-means algorithm, it was only used for hierarchical clustering. Since the clustering results of K-means depend on the initial clustering centers that should be randomly initialized in each run, we run K-means 100 times with random initialized centers for each experiment. Section 4.3.1 gives the energy ratio of wavelet coefficients within various scales and the calculated appropriate scale for each used dataset. The evaluation of K-means clustering algorithm with the proposed feature extraction algorithm is given in Section 4.3.2. Section 4.3.3 describes the comparative evaluation of hierarchical clustering with the feature extraction algorithm.

4.3.1 The Energy Ratio and Appropriate Scale

Table 1 provides the energy ratio $\sum_m E(\overrightarrow{D_j}) / \sum_m \sum_{j=0}^{J-1} E(\overrightarrow{D_j})$ (in proportion to the energy) of wavelet coefficients within various scales for all the used datasets. The calculated appropriate scales for the nine datasets using Algorithm 1 are shown in Table 2. The algorithm stops after the first iteration (scale = 1) for most of the datasets (Trace, Gun, Reality, ECG, Popu, and Temp), and stops after the second iteration (scale = 2) for CBF and CC datasets. The algorithm stops after the third iteration (scale = 3) only for Income dataset. If the sampling frequency for the time series is f, wavelet coefficients within scale j correspond to the information with frequency $f/2j$. Table 2 shows that most used time series datasets have important frequency components beginning from $f/2$ or $f/4$.

4.3.2 K-means Clustering with and without Feature Extraction

The average execution time of the chain process that first executes feature extraction algorithm then performs K-means with the extracted features (termed by FE + K-means) with 100 runs and that of performing K-means directly on the original data (termed by K-means) with 100 runs are illustrated in Figure 2. The chain process executes faster than K-means with original data for the used eight datasets.

Table 3 describes the mean of the evaluation criteria values of 100 runs for K-means with original data. Table 4 gives the mean of the evaluation criteria values of 100 runs for K-means with extracted features.

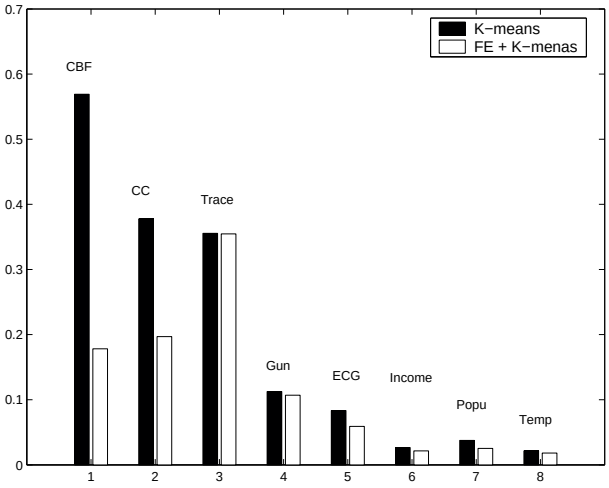

Figure 2: The average execution time (s) of the K-means + FE and K-means algorithms for eight datasets

To compare the difference between the mean obtained from 100 runs of K-means with extracted features and that obtained from 100 runs of K-means with corresponding original data, two-sample Z-test or two-sample t-test can be used. We prefer two-sample t-test because it is robust with respect to violation of the assumption of equal population variances, provided that the number of samples are equal [39]. We use two-sample t-test with the following hypothesis:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

where μ_1 is the mean of the evaluation criteria values corresponding to original datasets and μ_2 is that corresponding to extracted features. The significance level is set as 0.05. When the null hypothesis (H_0) is rejected, we conclude that the data provide strong evidence that μ_1 is different with μ_2, and which item is bigger can be easily gotten by comparing the corresponding mean values as shown in Table 3 and Table 4. We list the results of t-tests in Table 5 (If the mean of the values of a criterion corresponding to extracted features is significantly bigger than that corresponding to the original data, we set the character as '$>$'; if the mean of the values of a criterion corresponding to extract features is significantly smaller than that corresponding to the original data, the character is set as '$<$'; otherwise we set the character as '='). Table 5 shows that the evaluation criteria values corresponding to extracted features are bigger than that corresponding to the original data eleven times, smaller than that corresponding to the original data five times, and equal to that corresponding to the original data twenty four times for eight datasets. Based on the above analysis, we can conclude that the quality of K-means algorithm with extracted features is better than that with original data averagely for the used datasets.

Table 6 gives the mean of the evaluation criteria values of 100 runs of K-means with features in the prior scale.

Table 1: The energy ratio (%) of the wavelet coefficients within various scales for all the used datasets

scale	CBF	CC	Trace	Gun	Reality	ECG	Income	Popu	Temp
1	8.66	6.34	0.54	0.18	0.03	18.22	5.07	0.12	4.51
2	6.00	5.65	1.31	1.11	0.1	26.70	2.03	0.46	7.72
3	7.67	40.42	2.60	2.20	0.29	19.66	1.49	7.31	5.60
4	11.48	19.73	4.45	7.85	3.13	12.15	26.08	8.10	4.57
5	18.97	9.98	6.75	15.58	3.85	8.97	28.49	13.68	9.92
6	32.25	17.87	14.66	54.81	8.94	7.11	26.39	21.94	4.29
7	15.62		39.66	14.43	21.39	3.55	10.46	48.39	16.60
8			29.56	4.02	20.01	1.80			42.62
9			0.46		19.41	1.83			4.16
10					22.84				

Table 2: The appropriate scales of all nine datasets

	CBF	CC	Trace	Gun	Reality	ECG	Income	Popu	Temp
scale	2	2	1	1	1	1	3	1	1

The difference between the mean of criteria values produced by K-means algorithm with extracted features and that of criteria values generated by the features in the priori scale validated by t-test is described in Table 7. The mean of the criteria values corresponding to the extracted features are twelve times bigger, nineteen times equal, and nine times small than that corresponding to the features located in priori scale. The mean of the evaluation criteria values of 100 runs of K-means with features in the posterior scale are shown in Table 8. Table 9 provides the t-test result of the difference between the clustering criteria values of extracted features and the clustering criteria produced by the features within posterior scale. The mean of the criteria values corresponding to the extracted features are ten times bigger than that corresponding to the features located in the posterior scale, twenty nine times equal to that corresponding to the features located in the posterior scale, and only smaller than the features in the posterior scale one time. Based on the result of hypothesis testing, we can conclude that the quality of K-means algorithm with extracted appropriate features is better than that with features in the prior scale and posterior scale averagely for the used datasets.

4.3.3 Hierarchical Clustering with and without Feature Extraction

We used *single* linkage for the hierarchical clustering algorithm in our experiments. Figure 3 provides the comparison of the execution time of performing hierarchical clustering algorithm with original data (termed by HC) and the chain process of feature extraction plus hierarchical clustering algorithm (termed by HC + FE). For clearly observing the difference between the execution time of HC and HC + FE, the execution time is also given in Table 10. The chain process executes faster than hierarchical clustering with original data for all nine datasets.

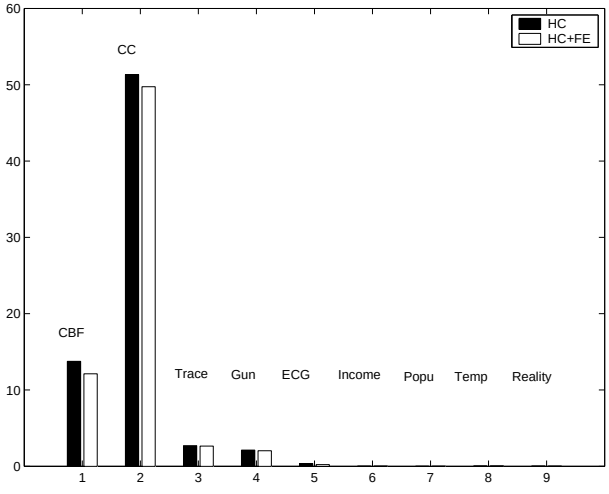

Figure 3: The execution time (s) of the HC + FE and HC

Evaluating the quality of the hierarchial clustering algorithm can be divided into subject way and objective way. Dendrograms are good for subjectively evaluating hierarchical clustering algorithm with time series data [27]. As only Reality dataset has one time series in each cluster that is suitable for visual observation, we used it for subjective evaluation and other datasets are evaluated by objective criteria. Hierarchical clustering with Reality dataset and its extracted features had the same clustering solution. Note that this result is fit with the introduction of the dataset as *Euclidean distance produces the intuitively correct clustering* [26]. The dendrogram of the clustering solution is shown in Figure 4.

As each run of hierarchical clustering for the same

Table 3: The mean of the evaluation criteria values obtained from 100 runs of K-means algorithm with eight datasets

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3490	0.4444	0.3592	0.3289	0.2048	0.6127	0.7142	0.7801
Rand	0.6438	0.8529	0.7501	0.4975	0.5553	0.7350	0.7611	0.8358
FM	0.5201	0.6213	0.5306	0.4949	0.3398	0.7611	0.8145	0.8543
CSM	0.5830	0.6737	0.5536	0.5000	0.4240	0.8288	0.8211	0.8800
NMI	0.3450	0.7041	0.5189	0.0000	0.0325	0.4258	0.5946	0.6933

Table 4: The mean of the evaluation criteria values obtained from 100 runs of K-means algorithm with the extracted features

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3439	0.4428	0.3672	0.3289	0.2644	0.6344	0.7719	0.8320
Rand	0.6447	0.8514	0.7498	0.4975	0.4919	0.7644	0.8079	0.8758
FM	0.5138	0.6203	0.5400	0.4949	0.4314	0.7770	0.8522	0.8912
CSM	0.5751	0.6681	0.5537	0.5000	0.4526	0.8579	0.8562	0.9117
NMI	0.3459	0.6952	0.5187	0.0000	0.0547	0.4966	0.6441	0.7832

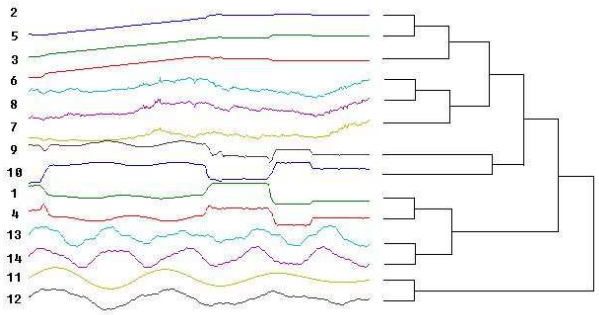

Figure 4: The dendrogram of hierarchical clustering with extracted features and that with original data for Reality dataset

dataset always gets the same result, we don't need multiple runs, the criteria values obtained from extracted features are compared to that obtained from original data directly without hypothesis testing. Table 11 describes the evaluation criteria values produced by hierarchical clustering with eight original datasets. Table 12 gives the evaluation criteria values obtained from hierarchical clustering with extracted features. The difference between the items in Table 11 and Table 12 is provided in Table 13. The meaning of the characters in Table 13 is described as below: '$>$' means a criterion value produced by extracted features is bigger than that produced by original data; '$<$' denotes a criterion value obtained from extracted features is smaller than that obtained from original data; Otherwise, we set the character as '='. Hierarchical clustering with extracted features produces same result as clustering with the original data on CBF, Trace, Gun, Popu and Temp datasets. For other three datasets, the evaluation criteria values produced by hierarchial clustering with extracted features are ten times bigger than, four times smaller than, and one time equal to that obtained from hierarchical clustering with original data. From the experimental results, we can conclude that the quality of hierarchical clustering with extracted features is better than that with original data averagely for the used datasets.

The criteria values produced by hierarchical clustering algorithm with features in the prior scale are given in Table 14. The criteria values corresponding to the extracted features shown in Table 12 are nine times bigger than, five times small than, and twenty six times equal to the criteria values corresponding to the features in the prior scale. Table 15 shows the criteria values obtained by hierarchical clustering algorithm with features in the posterior scale. The criteria values produced by the extracted features given in Table 12 are eleven times bigger than, four times smaller than, and twenty five times equal to the criteria values corresponding to the features in the posterior scale. From the experimental results, we can conclude that the quality of hierarchical clustering with extracted features is better than that of hierarchical clustering with features located in the priori and posterior scale averagely for the used datasets.

5 Conclusions

In this paper, unsupervised feature extraction is carried out in order to improve the time series clustering quality and speed the clustering process. We propose an unsupervised feature extraction algorithm for time series clustering using orthogonal wavelets. The features are defined as the approximation coefficients within a specific scale. We show that the sum of squared errors between the approximation series reconstructed from the features and the time-series is

Table 5: The difference between the mean of criteria values produced by K-means algorithm with extracted features and with original datasets validated by t-test

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	$<$	=	=	=	$>$	$>$	=	=
Rand	$>$	=	=	=	$<$	$>$	=	=
FM	$<$	=	=	=	$>$	$>$	=	=
CSM	$<$	=	=	=	$>$	$>$	=	=
NMI	=	$<$	=	=	$>$	$>$	=	$>$

Table 6: The mean of the evaluation criteria values obtained from 100 runs of K-means algorithm with features in the prior scale

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3489	0.4531	0.3592	0.3289	0.2048	0.6138	0.7142	0.7801
Rand	0.6438	0.8557	0.7501	0.4975	0.5553	0.7376	0.7611	0.8358
FM	0.5200	0.6299	0.5306	0.4949	0.3398	0.7615	0.8145	0.8543
CSM	0.5829	0.6790	0.5536	0.5000	0.4240	0.8310	0.8211	0.8800
NMI	0.3439	0.7066	0.5189	0.0000	0.0325	0.4433	0.5946	0.6933

Table 7: The difference between the mean of criteria values produced by K-means algorithm with extracted features and with features in the priori scale validated by t-test

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	$<$	$<$	=	=	$>$	$>$	=	=
Rand	$>$	$<$	=	=	$<$	$>$	=	=
FM	$<$	$<$	=	=	$>$	$>$	=	=
CSM	$<$	$<$	=	=	$>$	$>$	=	=
NMI	$>$	$<$	=	=	$>$	$>$	=	$>$

Table 8: The mean of the evaluation criteria values obtained from 100 runs of K-means algorithm with features in the posterior scale

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3457	0.4337	0.3632	0.3289	0.2688	0.4112	0.7770	0.8507
Rand	0.6455	0.8482	0.7501	0.4975	0.4890	0.5298	0.8141	0.8906
FM	0.5158	0.6114	0.5352	0.4949	0.4388	0.5826	0.8560	0.9031
CSM	0.5771	0.6609	0.5545	0.5000	0.4663	0.6205	0.8611	0.9216
NMI	0.3474	0.6868	0.5190	0.0000	0.0611	0.0703	0.6790	0.8037

Table 9: The difference between the mean of criteria values produced by K-means algorithm with extracted features and with features in the posterior scale validated by t-test

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	=	$>$	=	=	=	$>$	=	=
Rand	=	$>$	=	=	=	$>$	=	=
FM	=	$>$	=	=	=	$>$	=	=
CSM	=	$>$	=	=	$<$	$>$	=	=
NMI	=	$>$	=	=	=	$>$	=	=

Table 10: The execution time (s) of HC + FE and HC

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp	Reality
HC	13.7479	51.3319	2.7102	2.1256	0.3673	0.0169	0.0136	0.0511	0.0334
HC + FE	12.1269	49.7365	2.6423	2.0322	0.2246	0.0156	0.0133	0.0435	0.0172

Table 11: The evaluation criteria values produced by hierarchical clustering algorithm with the original eight datasets

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3299	0.5594	0.4801	0.3289	0.3259	0.5548	0.4583	0.4497
Rand	0.3369	0.8781	0.7488	0.4975	0.3619	0.5800	0.5211	0.4877
FM	0.5714	0.7378	0.6827	0.4949	0.5535	0.7379	0.6504	0.6472
CSM	0.4990	0.7540	0.6597	0.5000	0.4906	0.6334	0.6386	0.6510
NMI	0.0366	0.8306	0.6538	0.0000	0.0517	0.1460	0.1833	0.1148

Table 12: The evaluation criteria values obtained by hierarchical clustering algorithm with appropriate features extracted from eight datasests

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3299	0.5933	0.4801	0.3289	0.3355	0.5068	0.4583	0.4497
Rand	0.3369	0.8882	0.7488	0.4975	0.3619	0.5200	0.5211	0.4877
FM	0.5714	0.7682	0.6827	0.4949	0.5696	0.6956	0.6504	0.6472
CSM	0.4990	0.7758	0.6597	0.5000	0.4918	0.6402	0.6386	0.6510
NMI	0.0366	0.8525	0.6538	0.0000	0.0847	0.0487	0.1833	0.1148

Table 13: The difference between the criteria values obtained by hierarchical clustering algorithm with eight datasets and with features extracted from the datasets

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	=	$>$	=	=	$>$	$<$	=	=
Rand	=	$>$	=	=	=	$<$	=	=
FM	=	$>$	=	=	$>$	$<$	=	=
CSM	=	$>$	=	=	$>$	$>$	=	=
NMI	=	$>$	=	=	$>$	$<$	=	=

Table 14: The evaluation criteria values obtained by hierarchical clustering algorithm with features in the prior scale

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3299	0.5594	0.4801	0.3289	0.3259	0.5548	0.4583	0.4497
Rand	0.3369	0.8781	0.7488	0.4975	0.3619	0.5800	0.5211	0.4877
FM	0.5714	0.7378	0.6827	0.4949	0.5535	0.7379	0.6504	0.6472
CSM	0.4990	0.7540	0.6597	0.5000	0.4906	0.6334	0.6386	0.6510
NMI	0.0366	0.8306	0.6538	0.0000	0.0517	0.1460	0.1833	0.1148

Table 15: The evaluation criteria values obtained by hierarchical clustering algorithm with features in the posterior scale

	CBF	CC	Trace	Gun	ECG	Income	Popu	Temp
Jaccard	0.3299	0.4919	0.4801	0.3289	0.3355	0.5090	0.4583	0.4343
Rand	0.3369	0.8332	0.7488	0.4975	0.3619	0.5467	0.5211	0.5123
FM	0.5714	0.6973	0.6827	0.4949	0.5696	0.6908	0.6504	0.6207
CSM	0.4990	0.6640	0.6597	0.5000	0.4918	0.6258	0.6386	0.6340
NMI	0.0366	0.7676	0.6538	0.0000	0.0847	0.0145	0.1833	0.1921

equal to the energy of the wavelet coefficients within this scale and lower scales. Based on this property, we leverage the conflict of taking lower dimensionality and lower sum of squared errors simultaneously by finding the scale within which the energy of wavelet coefficients is lower than that within the nearest lower scale. An efficient feature extraction algorithm is designed without reconstructing the approximation series. The time complexity of the feature extraction algorithm can achieve $O(mn)$ with Haar wavelet transform. The main benefit of the proposed feature extraction algorithm is that dimensionality of the features is chosen automatically.

We conducted experiments on nine time series datasets using K-means and hierarchical clustering algorithm. The clustering results were evaluated by subjective observation and five objective criteria. The chain process of performing feature extraction firstly then executing clustering algorithm with extract features executes faster than clustering directly with original data for all the used datasets. The quality of clustering with extracted features is better than that with original data averagely for the used datasets. The quality of clustering with extracted appropriate features is also better than that with features corresponding to the scale prior and posterior to the appropriate scale.

References

[1] http://www.bea.gov/bea/regional/spi.

[2] http://www.ncdc.noaa.gov/rcsg/ datasets.html.

[3] http://www.census.gov/population/www/ estimates/st_stts.html.

[4] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th Conference on Foundations of Data Organization and Algorithms*, pages 69–84, 1993.

[5] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[6] K. Beyen, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In *Proceedings of the 7th International Conference on Database Theory*, pages 217–235, 1999.

[7] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice Hall, Englewood Cliffs, NJ, 1997.

[8] K. P. Chan and A. W. Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering*, pages 126–133, 1999.

[9] K. P. Chan, A. W. Fu, and T. Y. Clement. Harr wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Trans. on Knowledge and Data Engineering*, 15(3):686–705, 2003.

[10] C. K. Chui. *An Introduction to Wavelets*. Academic Press, San Diego, 1992.

[11] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series in Communication, New York, 1991.

[12] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Proceedings of the 9th IEEE International Conference on Tools with AI*, pages 532–539, 1997.

[13] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.

[14] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. on Information Theory*, 41(3):613–627, 1995.

[15] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 247–254, 2000.

[16] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Procedings of the 21th International Conference on Machine Learning*, 2004. Article No. 36.

[17] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market: Which measure is best? In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 487–496, 2000.

[18] P. Geurts. Pattern extraction for time series classification. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127, 2001.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. http://www.physionet.org/physiobank/ database/.

[20] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validataion techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[21] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Fransisco, 2000.

[22] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Trans. On Knowledge and data engineering*, 16(11):1370–1386, 2004.

[23] S. Kaewpijit, J. L. Moigne, and T. E. Ghazawi. Automatic reduction of hyperspectral imagery using wavelet spectral analysis. *IEEE Trans. on Geoscience and Remote Sensing*, 41(4):863–871, 2003.

[24] K. Kalpakis, D. Gada, and V. Puttagunta. Distance measures for effective clustering of arima time-series. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 273–280, 2001.

[25] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction of fast similarity search in large time series databases. *Journal of Knowledge and Information System*, 3:263–286, 2000.

[26] E. Keogh and T. Folias. The ucr time series data mining archive. http://www.cs.ucr.edu/∼eamonn/TSDMA/ index.html, 2002.

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

[28] E. Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, pages 239–241, 1998.

[29] S. W. Kim, S. Park, and W. W. Chu. Efficient processing of similarity search under time warping in sequence databases: An index-based approach. *Information Systems*, 29(5):405–420, 2004.

[30] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[31] F. Korn, H. Jagadish, and C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *Proceedings of The ACM SIGMOD International Conference on Management of Data*, pages 289–300, 1997.

[32] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. 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*, pages 2–11, 2003.

[33] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos. Iterative incremental clustering of time series. In *Proceedings of 9th International Conference on Extending Database Technology*, pages 106–122, 2004.

[34] H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic, Boston, 1998.

[35] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, second edition, 1999.

[36] I. Popivanov and R. J. Miller. Similarity search over time-series data using wavelets. In *Proceedings of the 18th International Conference on Data Engineering*, pages 212–221, 2002.

[37] D. Rafiei and A. Mendelzon. Efficient retrieval of similar time sequences using dft. In *Proceedings of the 5th International Conference on Foundations of Data Organizations*, pages 249–257, 1998.

[38] N. Saito. *Local Feature Extraction and Its Application Using a Library of Bases*. PhD thesis, Department of Mathematics, Yale University, 1994.

[39] G. W. Snedecor and W. G. Cochran. *Statistical Methods*. Iowa State University Press, Ames, eight edition, 1989.

[40] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(3):583–617, 2002.

[41] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of the 16th International Conference on Machine Learning*, pages 389–397, 1999.

[42] Y. Wu, D. Agrawal, and A. E. Abbadi. A comparison of dft and dwt based similarity search in time-series databases. In *Proceedings of the 9th ACM CIKM International Conference on Information and Knowledge Management*, pages 488–495, 2000.

[43] N. Wyse, R. Dubes, and A. K. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann Publishers, Inc, San Mateo, CA, 1980.

[44] Y. Xiong and D. Y. Yeung. Time series clustering with arma mixtures. *Pattern Recognition*, 37(8):1675–1689, 2004.

[45] B. K. Yi and C. Faloustos. Fast time sequence indexing for arbitrary l_p norms. In *Proceedings of the 26th International Conference on Very Large Databases*, pages 385–394, 2000.

[46] H. Zhang, T. B. Ho, and M. S. Lin. A non-parametric wavelet feature extractor for time series classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 595–603, 2004.