Paper:

Combining The Global and Partial Information for Distance-Based Time Series Classification and Clustering

Hui Zhang *, Tu Bao Ho *, Mao-Song Lin **, and Wei Huang *

* School of Knowledge Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan
E-mail:{zhang-h,bao,w-huang}@jaist.ac.jp
** School of Computer Science, Southwest University of Science and Technology, Mianyang, Sichuan 621002, China
E-mail: lms@swust.edu.cn
[Received ; accepted]

Many time series representation schemes for classification and clustering have been proposed. Most of the proposed representation focuses on the prominent series by considering the global information of the time series. The partial information of time series that indicates the local change of time series is often ignored. Recently, researches shown that the partial information is also important for time series classification. However, the combination of these two types of information has not been well studied in the literatures. Moreover, most of the proposed time series representation requires predefined parameters. The classification and clustering results are considerably influenced by the parameter settings, and, users often have difficulty in determining the parameters.

We attack above two problems by exploiting the multi-scale property of wavelet decomposition. The main contributions of this work are: (1) extracting features combining the global information and partial information of time series (2) automatically choosing appropriate features, namely, features in an appropriate wavelet decomposition scale according to the concentration of wavelet coefficients within this scale. Experiments performed on several benchmark time series datasets justify the usefulness of the proposed approach.

Keywords: Time Series, Haar Wavelets, Classification, Clustering, Feature Extraction

1. Introduction

Time series data accounts for a huge amount of the data stored in financial, gene expression, medical and science databases. Many algorithms have been proposed for mining time series data [19, 28]. Time series classification and time series clustering are two important aspects of time series mining. Time series classification has been successfully used in various applications such as medical data analysis [12, 33], sign language recognition [15], speech recognition [23], etc. Time series clustering is a popularly used preprocessing technique in stock market analysis [9], gene expression data analysis [13], and so on. The high dimensionality, high correlation between data, and high noise embedded in time series make time series classification and clustering challenging tasks. For efficiency, most of the proposed methods classify or cluster time series on the high level representation of time series that takes the global information of time series rather than classifying or clustering them directly. The representation includes Fourier Transforms [1, 27], Piecewise Linear Representation (PLR) [21], Piecewise Aggregate Approximation [17, 34], Regression Tree Representation [10], Haar Wavelets [3], and Symbolic Representation [24]. Recently, Jin et al. proposed a time series representation scheme based on the partial information of the time series and showed that the partial information is also important for time series classification [14]. However, little work has done to combine these two types of information in time series mining literatures.

Most of the proposed representation schemes in the literatures require predefined parameters and the corresponding classification or clustering algorithms are considerably influenced by these predefined parameters [20]. For example, when setting the number of straight lines as the input parameter for the PLR algorithm, the range of selection is limited from one to the length of raw time series data. If the input parameter is one, the representation is just a linear regression of the whole data set, and the classification accuracy in this case will be lower than using raw data for most data and algorithms. If we choose the length of raw time series data as the input parameter, the represented data is actually the raw data. The selection is not trivial or easy for the users, and as a consequence, they usually have difficulty in determining the parameters. Note that a domain-transform technique such as Fourier transform doesn't need input parameters itself, but the later feature extraction process needs input parameters in most cases.

In this paper we introduce a time series representation combining both the global information and the partial information of the time series data by Haar wavelet decomposition. Time series classification and clustering algorithms are performed with the selected features of the representation. We propose a novel non-parametric feature extraction algorithm that extracts the approximation (global information) and change amplitudes (partial information) of time series. The Euclidean distances between the features of shifted time series and original time series are smaller than those between the raw data. Hence, distance-based classification and clustering algorithms are suitable for the features. The appropriate features, i.e., features within the appropriate wavelet decomposition scale, are chosen with respect to the concentration of features. The appropriate features are robust against noise embedded in the time series. Experiments performed on several benchmark datasets demonstrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. Section 2 briefly discusses the related work. Section 3 introduces our feature extraction algorithm and the corresponding distance-based classification and clustering algorithms. Section 4 contains an comparative experimental evaluation with the proposed approach. Finally, Section 5 concludes the paper with summarizing the main contributions of the work.

2. Related work

A large number of wavelet-based feature extraction techniques have been proposed. Chan and Fu introduced an algorithm for nearest neighbor querying with Haar wavelet coefficients, and only first few wavelet coefficients were preserved for dimensionality reduction [3, 4]. Popivanov and Miller used first few Daubechies wavelet coefficients instead of Haar wavelet coefficients for time series querying [26]. Shahabi et al. proposed an algorithm called TSA-tree which queries either the approximation part or detail part of Haar wavelet coefficients in a specific scale given by the user [30]. Struzik and Siebes defined a new time series similarity measurement on the correlation of Haar wavelet coefficients [31]. In the signal processing community, Piter and Kamarthi chose clustered wavelet coefficients as features [25]. Tancel et al. only used approximation wavelet coefficients as input to an ART-2 type neural network [32]. Kalayci and Ozdamar suggested a method using eight central detail coefficients of scale m ranging from 1 to 5 as the features [16].

To date, all the proposed methods use global information or partial information, but no proposed algorithm combines these two types of information. Furthermore, no proposed work gives solution for automatically choosing appropriate features for a given dataset. We propose a method of using entropy to choose appropriate scale which is similar in spirit to the wavelet packet algorithm introduced by Coifman et al. [5], in which the entropy is used to select the best basis for a wavelet packet.

3. Time Series Representation and Feature Extraction

Our basic idea is to extract features from time series and perform classification and clustering algorithms with the extracted features. The multi-scale property of wavelets allows us to extract features with the global information and partial information simultaneously. After obtaining the features, distance-based classification and clustering algorithms can be applied in terms of the similarity between the extracted features. Therefore, we decompose our task into three sub-procedures: (1) representing the time series via wavelet coefficients within various scales which contain both the global and partial information of the time series data; (2) retrieving the features by selecting the appropriate scale of the representation; and (3) designing a similarity measuring strategy, in which most proposed similarity models could be applied. The basic idea of Haar wavelet decomposition is introduced in Section 3.1. We give the time series representation and corresponding feature extraction algorithms in Sections 3.2 and 3.3. Section 3.4 presents the method of noise reduction with the features. We suggest the similarity measure strategy and its corresponding classification and clustering algorithms in sections 3.5.

3.1. Haar Wavelet Decomposition

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. One property of wavelets is that the broad trend of the input sequence is preserved in the approximation part, whereas localized changes are kept in the detail parts. No information is gained or lost during the decomposition process. The original signal can be fully reconstructed from the approximation part and the detail parts.

The Haar wavelet is the simplest and most popular wavelet proposed by Haar. The benefit of the Haar wavelet is that its decomposition process has low computational complexity. Given a time series with length n, where n is an integral power of 2, the complexity of Haar decomposition is $O(n)$. The Haar wavelet decomposition process needs a pair of sequences associated with it. The sequences are called wavelet analysis filters, denoted as $\{h_k, g_k\}$. The Haar wavelet analysis filters h_k and g_k are given by

$$h_k = [1/\sqrt{2}, 1/\sqrt{2}] \text{ and } g_k = [1/\sqrt{2}, -1/\sqrt{2}]$$

Haar wavelet decomposition is implemented by a two-step process: down-sampling and a convolution with the down-sampled series and wavelet analysis filters. Concrete mathematical foundations can be found in [2]. The length of the 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 time series if the length of the input time series doesn't satisfy this requirement.

A time series $\overrightarrow{X} = \{x_1, x_2, \ldots, x_n\}$ can be decomposed into an approximation part $\overrightarrow{A_1} = \{(x_1 + x_2)/\sqrt{2}, (x_3 + x_4)/\sqrt{2}, \ldots, (x_{n-1} + x_n)/\sqrt{2}\}$ and a detail part $\overrightarrow{D_1} = \{(x_1 - x_2)/\sqrt{2}, (x_3 - x_4)/\sqrt{2}, \ldots, (x_{n-1} - x_n)/\sqrt{2}\}$. The

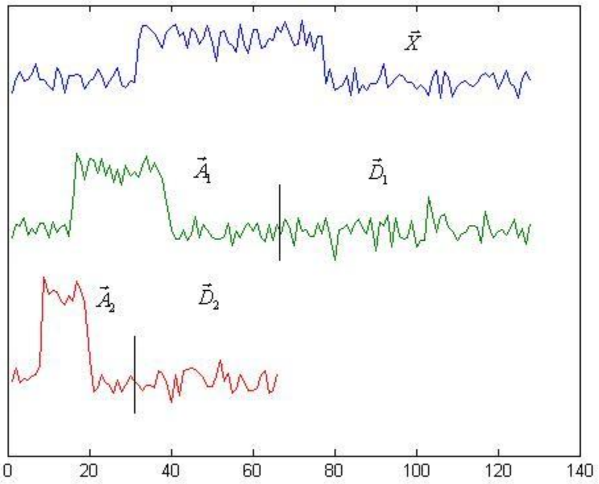

Fig. 1. An example of a time series and its wavelet coefficients to scale 2

$\overrightarrow{A_1}$ are approximation coefficients within scale 1 and $\overrightarrow{D_1}$ are detail coefficients within scale 1. The approximation coefficients and detail coefficients within a particular scale j, $\overrightarrow{A_j}$ and $\overrightarrow{D_j}$, both having length $n/2^j$, 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

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

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

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

The number of decomposing scales for $\overrightarrow{X}$ is $J = log_2(n)$. $\overrightarrow{A_J}$ only has one element denoting the global average of $\overrightarrow{X}$.

3.2. Time Series Representation and Feature Extraction

The concatenation of decomposed wavelet coefficients of a time series $\overrightarrow{X} = \{x_1, x_2, \ldots, x_n\}$ to a particular scale $k \in [1,2,\ldots,J]$ shown in (3) is a representation of $\overrightarrow{X}$. An example of decomposing a time series to scale 2 is illustrated in Fig. 1. The $\overrightarrow{X}$ can be fully reconstructed from $\overrightarrow{W_k(X)}$ without losing any information.

$$\overrightarrow{W_k(X)} = [\overrightarrow{A_k(X)}, \overrightarrow{D_k(X)}, \ldots, \overrightarrow{D_2(X)}, \overrightarrow{D_1(X)}] \quad . \quad (3)$$

Chan et al. proved that the Euclidean distance is preserved through a Haar wavelet transform [3]. Assume we have two time series $\overrightarrow{X} = \{x_1, x_2, \ldots, x_n\}$ and $\overrightarrow{Y} = \{y_1, y_2, \ldots, y_n\}$, the Euclidean distance between $\overrightarrow{X}$ and $\overrightarrow{Y}$ is

$$Disc(\overrightarrow{X}, \overrightarrow{Y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (4)$$

The Euclidean distance between $\overrightarrow{W_k(X)}$ and $\overrightarrow{W_k(Y)}$, $Disc(\overrightarrow{W_k(X)}, \overrightarrow{W_k(Y)})$ is equal to $Disc(\overrightarrow{X}, \overrightarrow{Y})$. In this case, if we just use $\overrightarrow{W_k}$ as the features for a distance based classification or clustering algorithm, the result should be the same with that gotten from the original time series.

The ith element of $\overrightarrow{A_j}$ corresponds to the segment in the series $\overrightarrow{X}$ starting from position $(i-1)*2^j+1$ to position $i*2^j$. The a_j^i is proportional to the average of this segment and thus can be viewed as the approximation of the segment. Thus the approximation coefficients within different scales provide an understanding of the major trends in the data at a particular level of granularity. From Eq. 2, we know that the detail coefficients $\overrightarrow{D_j} = \{d_j^1, d_j^2, \ldots, d_j^{n/2^j}\}$ contain local changes of time series. Thus $|\overrightarrow{D_j}| = \{|d_j^1|, |d_j^2|, \ldots |d_j^{n/2^j}|\}$ denote the amplitude of local changes. We define the concatenation of decomposed wavelet approximation coefficients $\overrightarrow{A_k(X)}$ and the absolute values of decomposed wavelet detail coefficients $|\overrightarrow{D_j(X)}|, j = 1,2,\ldots,k$ to a particular scale k ($k \in [1,2,\ldots,J]$) of a time series $\overrightarrow{X}$ as features.

$$\overrightarrow{F_k(X)} = [\overrightarrow{A_k(X)}, |\overrightarrow{D_k(X)}|, \ldots, |\overrightarrow{D_2(X)}|, |\overrightarrow{D_1(X)}|] \quad (5)$$

This definition helps to overcome the well-known problem posed by the fact that wavelet coefficients are sensitive to shifts of series. The Euclidean distance $Disc(\overrightarrow{F_k(X)}, \overrightarrow{F_k(Y)})$ between the features of two time series $\overrightarrow{X}$ and $\overrightarrow{Y}$ is

$$\sqrt{\sum_i (a_k^i(\overrightarrow{X}) - a_k^i(\overrightarrow{Y}))^2 + \sum_{j=1}^{k}\sum_i (|d_j^i(\overrightarrow{X})| - |d_j^i(\overrightarrow{Y})|)^2} (6)$$

Because $|x| - |y| \leq x - y$, we obtain $Disc(\overrightarrow{F_k(X)}, \overrightarrow{F_k(Y)}) \leq Disc(\overrightarrow{W_k(X)}, \overrightarrow{W_k(Y)})$, and $Disc(\overrightarrow{W_k(X)}, \overrightarrow{W_k(Y)}) = Disc(\overrightarrow{X}, \overrightarrow{Y})$. If $\overrightarrow{X}$ and $\overrightarrow{Y}$ denote the original time series and shifted time series respectively, this inequation is still tenable.

3.3. Appropriate Scale Selection

Normally, for a time series $\overrightarrow{X}$, the first k, $k < n$ coefficients of $\overrightarrow{W_J(X)}$ are taken as features after decomposing $\overrightarrow{X}$ to the scale J [3, 26]. The first few wavelet coefficients are correspond to the approximation wavelet coefficients including the global information of X. However, setting the parameter k is not easy for the users and the later classification or clustering process are affected by poorly chosen parameter settings. A non-parametric feature extraction algorithm without any input parameter is more convenient to the users. For our definition of features, the non-parametric feature extraction algorithm needs to find out which features associated with a specific scale are better than others for classification and clustering automatically.

If the energy of wavelet coefficients within a scale is concentrated in a few coefficients then just those important coefficients can represent the whole, with low error. This scale may give valuable information for classification and clustering. We need a function to describe the energy concentration of the wavelet coefficients. The function should be large when coefficients are largely the same value, and small when all but a few coefficients are negligible. Shannon entropy, which is a measure of impurity within a set of instances satisfies our requirement. It is defined below.

$$H = -\sum_i p_i \log_2 p_i \quad \ldots\ldots\ldots\ldots \quad (7)$$

The appropriate decomposing scale is defined as the scale with the lowest entropy. The appropriate features of a time series are defined as the wavelet coefficients within an appropriate decomposing scale.

$$\text{Appropriate scale} = \underset{k}{\text{argmin}}(-\sum_i p_k^i \log_2 p_k^i) \quad . \quad (8)$$

here $p_k^i = |F_k^i(X)| / \sum_{i=1}^{n} |F_k^i(X)|$ is the proportion between the absolute value of a coefficient in a feature and the sum of the absolute values of a whole feature. p_k^i is proportional to the energy ratio of each coefficient to coefficients within a feature. p_k^i has properties $\sum_i p_k^i = 1$ and $p_k^i \geq 0$.

3.4. Noise Reduction on The Appropriate Features

The idea of wavelet noise reduction is based on the assumption that the amplitude of the spectra of the signal is as different as possible from that of noise. If a signal has its energy concentrated in a small number of wavelet coefficients, these coefficients will be relatively large compared to the noise, which has its energy spread over a large number of coefficients. This allows thresholding of the amplitude of the coefficients to separate signals or remove noise. The thresholding is based on a value τ that is used to compare all the detail coefficients. Appropriate scale, as defined in section 3.3, should be robust to the noise because the energy of true signal gets concentrated in a few coefficients and the noise remains spread out in that scale. Hence the energy of the few coefficients is much larger than that of noises and these large coefficients can dominate the classification or clustering process. To verify the claim, we will compare the classification and clustering results with and without noise-reduction in the experiments.

Donoho and Johnstone [7] gave the threshold as $\tau = \sigma_n \sqrt{2\log(N)}$; here σ_n is the standard variation of noise, and N is the length of the time series. Because we don't know the σ_n of the time series in advance, we estimate it by the robust median estimation of noise method described in [7]. The robust median estimation is the median absolute deviation of the detail wavelet coefficients at scale one, divided by 0.6745.

The widely used hard thresholding algorithm is a process of setting the value of detail coefficients whose absolute values are lower than the threshold to zero [6]. The hard thresholding algorithm for features defined in (8) is described in Eq. 9.

$$Thre(|d_j^i|) = \begin{cases} |d_j^i|, & |d_j^i| > \tau \\ 0, & |d_j^i| \leq \tau \end{cases} \quad \ldots\ldots \quad (9)$$

3.5. The Similarity Strategy and Its Corresponding Classification and Clustering Algorithms

For two series with the same length, their corresponding appropriate scales may not be equal. We can't compare the similarity of two sets of appropriate features directly because the meaning of each data entry is different. For example, consider a time series $\overrightarrow{X} = \{x_1, x_2, x_3, x_4\}$ with appropriate scale 1 and a time series $\overrightarrow{Y} = \{y_1, y_2, y_3, y_4\}$ with appropriate scale 2. The features of series $\overrightarrow{X}$ are $\overrightarrow{F_1(X)} = \{a_1^0(\overrightarrow{X}), a_1^1(\overrightarrow{X}), |d_1^1(\overrightarrow{X})|, |d_1^1(\overrightarrow{X})|\}$ and the features of series $\overrightarrow{Y}$ are $\overrightarrow{F_2(Y)} = \{a_2^1(\overrightarrow{X}), |d_2^1(\overrightarrow{X})|, |d_2^2(\overrightarrow{X})|, |d_2^3(\overrightarrow{X})|\}$. Comparing detail coefficients with approximation coefficients will induce errors and be meaningless.

To avoid this problem, we merge the distance of features within different appropriate scales. The distance of two features is replaced by the average of distance computed on two features with different appropriate scales. Suppose a time series $\overrightarrow{X}$ with appropriate scale m and another time series $\overrightarrow{Y}$ with appropriate scale n, the distance between the appropriate features of $\overrightarrow{X}$ and $\overrightarrow{Y}$, $Disc(\overrightarrow{F_m(X)}, \overrightarrow{F_n(Y)})$, is defined as $(Disc(\overrightarrow{F_m(X)}, \overrightarrow{F_m(Y)}) + Disc(\overrightarrow{F_n(X)}, \overrightarrow{F_n(Y)}))/2$. We can simply use distance-based classification and clustering algorithms for the proposed similarity strategy.

3.5.1. 1-*NN* Classification Algorithm Using The Proposed Similarity Strategy

The classification algorithm is implemented by means of a 1-nearest neighbor algorithm (1-*NN*), which we call WCANN (Wavelet Classification Algorithm based on 1-Nearest Neighbor). Table 1 shows an outline of the classification algorithm. The input is $S = \{\overrightarrow{S_1}, \overrightarrow{S_2}, \ldots\}$ consists of a set of labeled time series data (the trained time series datasets) and $\overrightarrow{X}$ (a new emerged testing time series). The output is x_c, the label of $\overrightarrow{X}$. $\overrightarrow{X}$ will be labeled as a member of a class if and only if, the distance between $\overrightarrow{X}$ and one instance of the class is smaller than between other instances.

The classification algorithm for noise reduced coefficients, WCANR (Wavelet Classification Algorithm with Noise Reduction), is similar to the WCANN algorithm. The only difference is that the noise within appropriate features is reduced before classification.

3.5.2. Hierarchical Clustering Using The Proposed Similarity Strategy

Clustering is one of the commonly used data mining tasks for discovering groups and identifying interesting distributions and patterns in the underlying data. Clustering is useful in its own right as a description tool, and also as a subtask in more complex algorithms.

Hierarchical clustering groups data objects into a tree of clusters which is among the best known of clustering methods [8]. The hierarchical clustering algorithms can be divided into *agglomerative* and *divisive* clustering, in terms of whether the hierarchical decomposition is

Table 1. The WCANN algorithm

Input: S, $\overrightarrow{X}$
For each training example $\overrightarrow{S_i}$, calculating its appropriate scale m_i and corresponding appropriate features $\overrightarrow{F_{m_i}(S_i)}$;
Given a testing instance $\overrightarrow{X}$, calculating its appropriate scale n and appropriate features $\overrightarrow{F_n(X)}$;
best-so-far = inf;
for i = 1 to length(S) **do**
  Calculate $Disc(\overrightarrow{F_{m_i}(S_i)}, \overrightarrow{F_{m_i}(X)})$;
  Calculate $Disc(\overrightarrow{F_n(S_i)}, \overrightarrow{F_n(X)})$;
  $Disc(\overrightarrow{F_{m_i}(S_i)}, \overrightarrow{F_n(x)}) = (Disc(\overrightarrow{F_{m_i}(S_i)}, \overrightarrow{F_{m_i}(X)}) + Disc(\overrightarrow{F_n(S_i)}, \overrightarrow{F_n(X)}))/2$
  if $Disc(\overrightarrow{F_{m_i}(S_i)}, \overrightarrow{F_n(x)}) <$ best-so-far **then**
    pointer-to-best-series $k = i$;
    best-so-far = $Disc(\overrightarrow{F_{m_i}(S_i)}, \overrightarrow{F_n(X)})$;
  end if
end for
return x_c = the label of k;

formed in a bottom-up or top-down fashion. As the computation of decomposition is normally simple for the agglomerative procedures, we only use agglomerative clustering algorithm with the proposed similarity strategy.

Given a set of N time series to be clustered, we generate a $N \times N$ distance matrix consisted of the pair-distance between each of two time series using the defined similarity strategy. The basic process of agglomerative hierarchical clustering is described as below:

1 Assign each time series to its own cluster. For the given N time series, we now have N clusters, each containing just one time series. Let the distances between the clusters equal the distances between the time series they contain.

2 Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now we have one less cluster.

3 Compute distances between the new cluster and each of the old clusters.

4 Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

4. Experimental Evaluation

To show the effectiveness of our approach, we performed experiments on five benchmark time series datasets. We compared the classification accuracy of our feature extraction algorithm with other feature extraction algorithms. All the feature extraction algorithms were compared with the one-nearest-neighbor algorithm (1-*NN*), evaluated by *leave-one-out* cross validation. We also compared the clustering result of using features to that of using original time series. The single linkage is used for the agglomerative hierarchical clustering.

4.1. Data Description

We used five datasets (CBF, CC, Trance, Gun and Realitycheck) from the UCR Time Series Data Mining Archive [18]. As we need the label information, we only took the classified datasets for experiments. There are six classified datasets in the archive. The Auslan data is a multivariate dataset with which we can't apply our approach directly. All the other five datasets are useded in our experiments. As Realitycheck data only has one instance within each cluster that is too simple for classification, we only used it for clustering. The main features of the used data sets 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 data original proposed by [29]. 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 $[16,32]$ and $b-a$ is an integer drawn uniformly from the range $[32,96]$. UCR Archive provides the source code for generating the samples. We generated 128 examples for each class with length 128.

- Control Chart Time Series (CC): This data set has 100 instances for each of the six different classes of control charts. The dataset can also be downloaded from UCI KDD Archive [11].

- 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 containing 100 instances. The dimensionality of the data is 150.

- Reality Check dataset (Realitycheck): This data set is designed as a simple reality check for new distance measures. The data set consists of data from Space Shuttle telemetry, Exchange Rates and artificial sequences. The data is a normalized data so that the minimum value is zero and the maximum is one.

Table 2. The error rates (%) produced by various feature extraction methods with 1-*NN* classification algorithm for all four data sets

Methods	CBF	CC	Gun	Trace
WCANN	0.26	0.67	4.5	7.5
WCANR	0.26	0.67	4.5	7.5
WC-NN	0.54	1.73	7.25	12.39
DFT-NN	0.41	2.52	6.48	10.84
SVD-NN	0.61	2.29	5.99	12.37
NN	0.26	1.33	5.5	11

4.2. Classification results

Keogh and Kasetty compared 12 distance measurements for time series using 1-*NN* classification algorithm and shown that Euclidean is the best among the compared distance measurements [19]. Therefore we used only Euclidean distance for 1-*NN* algorithm in our experiments. An unclassified instance is assigned to the same class as its closest match in the training set. The accuracy of the classified results is measured by error rates. We compared six algorithms using different features extracted from the CBF, CC, Trace and Gun data sets described above. WCANN and WCANR are our proposed algorithms described in section 3.5. NN is the 1-nearest neighbor algorithm that uses the raw data [19]. WC-NN is the algorithm that uses the 1-nearest neighbor algorithm with decomposed wavelet coefficients having various dimensions [4]. DFT-NN is the 1-*NN* algorithm using first few Fourier coefficients as features [27]. SVD-NN uses Singular Value Decomposition (SVD) to extract features with 1-*NN* algorithm [22]. As WC-NN, DFT-NN and SVD-NN don't give the solution for choosing feature dimensionality, we take the average error rates with all possible feature dimensions produced by these algorithms. Table 2 gives the error rates produced by various feature extraction algorithms with 1-*NN* classification algorithm.

The WCANN algorithm achieves the same classification accuracy as the WCANR algorithm on all the four data sets. Noise reduction upon the extracted appropriate features doesn't affect the classification result, i.e., the appropriate features are robust against noise in terms of classification. WCANN and WCANR take the same classification accuracy with NN algorithm on CBF data and higher accuracy on other three datasets. WCANN and WCANR algorithm outperform WC-NN, DFT-NN and SVD-NN on classification accuracy for all the datasets used.

4.3. Clustering

Hierarchical clustering is a good way to compare similarity measures with representation, since a dendrogram of size N summarizes $O(N^2)$ distance calculations. We can easily judge which representation measure is better by visually observing which representation appears to create the most natural groupings of the data. Fig. 2 shows the dendrogram produced by agglomerative hierarchical clustering algorithm with the original time series. The extracted features and noise-reduced features produce the same clustering results and the results are shown in Fig. 3. The extracted features are not affected by the noise during the clustering process for the used datasets. The extracted features are robust against noise in terms of clustering. The subgroup contains instances $\{11, 12\}$ and the subgroup contains instances $\{13, 14\}$ are similar. They are successfully clustered into one subgroup with our defined features shown in Fig. 3 and are not in the same subgroup with the original time series shown in Fig. 2.

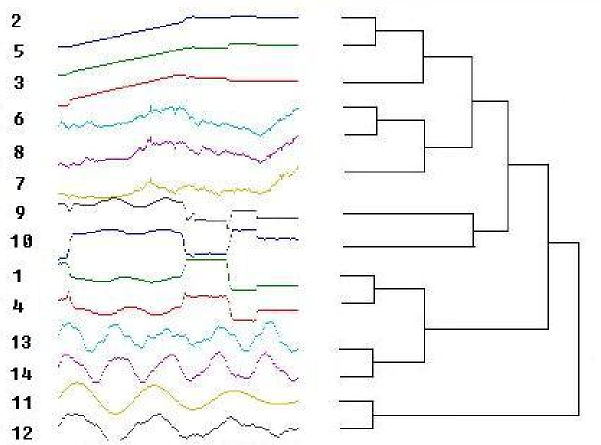

Fig. 2. The dendrogram generated by agglomerative clustering algorithm for original Realitycheck data set, single linkage is used

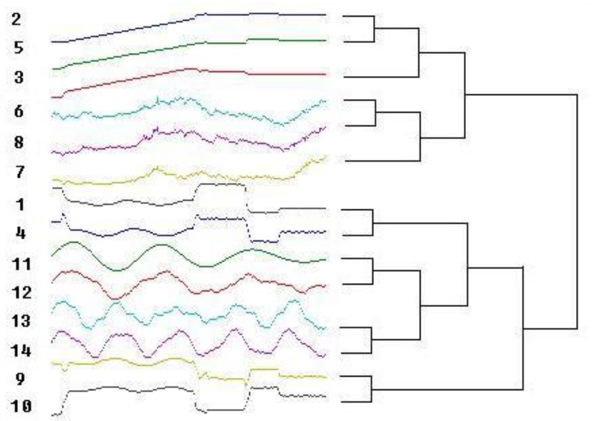

Fig. 3. The dendrogram generated by agglomerative clustering algorithm with the features of Realitycheck data set, single linkage is used

5. Conclusions

We exploit the multi-scale property of Haar wavelet transform to extract features combining the global information and partial information together. We propose a method for automatically choosing the appropriate features by the concentration of the feature coefficients. We propose a distance strategy for the appropriate features with which distance-based classification and clustering algorithms can be easily applied. We conduct experiments on several widely used time series datasets and compare the classification results of our algorithms and those of other four algorithms. Our algorithms outperform other four algorithms on classification accuracy. We also compare the clustering results produced by the appropriate features and the original time series. The appropriate features can create more natural groupings of the data. The experimental results show that the appropriate features always produce the same classification and clustering results with noise-reduced appropriate features. It indicates that several important coefficients within the appropriate features dominate the classification and clustering processes. The extracted appropriate features are robust against noise in terms of classification and clustering.

Acknowledgements

The authors thank Prof. Keogh for providing the experimental data. The authors thank two anonymous reviews for suggestions that markedly clarified and improved the paper.

References:

[1] 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, pp. 69–84, October 1993.

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

[3] F. K. 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.

[4] K. P. Chan and A. W. Fu, “Efficient Time Series Matching by Wavelets”, In Proceedings of the 15th Internation Conference on Data Engineering, pp. 126–133, March 1999.

[5] R. R. Coifman and M. V. Wickerhauser, “Entropy-Based Algorithms for Best Basis Selection”, IEEE Trans. on Information Theory, 38(2):713–718, 1992.

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

[7] D. L. Donoho and I. M. Johnson, “Ideal spatial adaptation via wavelet shrinkage”, Biometrika, 81:425–455, 1994.

[8] R. Duda, P. Hart, and D. Stork, ”Pattern Classification”, John Wiley & Sons, New York, second edition, 2001.

[9] M. Gavrilov, D. Anguelov, and P. Indyk, “Mining the Stock Market: Which Measure is Best?”, In Proceedings of The 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 487–496, August 2000.

[10] P. Geurts, “Pattern Extraction for Time Series Classification”, In Proceedings of the Principles of Data Mining and Knowledge Discovery,5th European Conference, pp. 115–127, September 2001.

[11] S. Hettich and S. D. Bay. “The UCI KDD Archive”. http://kdd.ics.uci.edu, 1999.

[12] T. B. Ho, T. D. Nguyen, S. Kawasaki, S. Q. Le, D. D. Nguyen, H. Yokoi, and K. Takabayashi, “Mining Hepatitis Data with Temporal Abstraction”, In Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining, pp. 369–377, August 2003.

[13] 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.

[14] X. Jin, Y. Lu, and C. Shi, “Similarity Measure Based on Partial Information of Time Series”, In Proceedings of The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 544–549, August 2002.

[15] M. W. Kadous, “Learning Comprehensible Descriptions of Multivariate Time Series”, In Proceedings of the 6th International Conference on Machine Learning, pp. 454–463, September 1999.

[16] T. Kalayci and O. Ozdamar, “Wavelet Preprocessing for Automated Neural Network Detection of EEG Spikes”, IEEE Eng. in Medicine and Biology, 14:160–166, 1995.

[17] 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.

[18] E. Keogh and T. Folias. “The UCR Time Series Data Mining Archive”. http://www.cs.ucr.edu/∼eamonn/TSDMA/index.html, 2002.

[19] 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.

[20] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, “Towards Parameter-Free Data Mining”, In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 206–215, August 2004.

[21] 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, pp. 239–241, August 1998.

[22] 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, pp. 289–300, May 1997.

[23] S. Lawrence, A. Back, A. Tsoi, and C. L. Giles, “The Gamma MLP – Using Multiple Temporal Resolutions for Improved Classification”, In Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing VII, pp. 256–265, Piscataway, NJ, 1997, IEEE Press.

[24] 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, pp. 2–11, June 2003.

[25] S. Pitter and S. V. Kamarthi, “Feature Extraction From Wavelet Coefficients for Pattern Recognition Tasks”, IEEE Trans. on Pattern Recognition and Machine Intelligence, 21(1):83–85, January 1999.

[26] I. Popivanov and R. J. Miller, “Similarity Search over Time-Series Data Using Wavelets”, In Proceedings of The 18th International Conference on Data Engineering, pp. 212–221, Feburary 2002.

[27] 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, pp. 249–257, 1998.

[28] J. F. Roddich and M. Spiliopoulou, “A Survey of Temporal Knowledge Discovery Paradigms and Methods”, IEEE Trans. on Knowledge and Data Engineering, 14(4):750–767, 2002.

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

[30] C. Shahabi, X. Tian, and W. Zhao, “TSA-Tree: A Wavelet-Based Approach to Improve the Efficiency of Multi-Level Surprise and Trend Queries on Time-Series Data”, In Proceedings of the12th International Conference on Scientific and Statistical Database Management, pp. 55–68, July 2000.

[31] Z. R. Struzik and A. Siebes, “The Harr Wavelet Transform in the Time Series Similarity Paradigm”, In Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery, pp. 12–22, September 1999.

[32] I. N. Tansel, C. Mekdeci, and C. McLaughlin, “Detection of Tool Failure in End Milling with Wavelet Transformations and Neural Networks (WT-NN)”, International Journal of Machine Tolls and Manufacture, 35(8):1137–1147, 1995.

[33] Y. Yamada, E. Suzuki, H. Yokoi, and K. Takabayashi, “Decision-Tree Induction from Time-Series Data Based on Standard-Example Split Test”, In Proceedings of the 20th International Conference on Machine Learning (ICML03), pp. 840–847, August 2003.

[34] 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, pp. 385–394, September 2000.