2004－ICS－138 (42)
2004／12／7

Multiscale Time Series Clustering: A Consensus Clustering Approach

Hui Zhang† and Tu Bao Ho†

Multiscale analysis has played an important role in signal processing and statistics. Multiscale analysis analyzes the time series data in multiple scales, and each scale catches specific features of the data, thus it gives us the ability of observing time series data in various views. When applying multiscale analysis to time series clustering, it is not trivial to determine the appropriate scale. This problem is circumvented in this paper by getting a consensus of the clustering solutions over each scale. We first perform clustering with the data of each scale, then merge the cluster solutions together. Two contributions of this paper are: we propose a multiscale time series clustering scheme based on the consensus of multiple clustering and a novel consensus clustering algorithm is presented in this paper. The benefits of the proposed approach are experimental evaluated with several real time series data.

1. Introduction

Clustering is a widely used data mining technique for partitioning a data set 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. A number of different clustering approaches have been proposed from different point of view[6] such as hierarchical clustering, distance based clustering, model based clustering. Time series data accounts for a huge amount of data stored in financial, medical, gene expression and science databases and is very popular in real life world and there is increasing interest in clustering time series data in recent years. Distance-based clustering that groups the time series data set by the distance between the subgroups of the data set has been widely used in time series clustering. The K-means algorithm may be the most commonly used distance-based clustering algorithm for its simpleness and efficiency. The consensus clustering approach is an approach of clustering the multiple clustered solutions obtained by reinitializing the start points of clustering algorithms or resampling the data set. It does not matter which clustering algorithm is used for generating each solution and it can even handle multiple clustering algorithms. For simplifying the comparison between different algorithms, we only take the K-means algorithm in our experiments.

In the past decades, wavelets have become one of the most exciting developments in applied mathematics, signal/image processing, and statistics. The most significant contribution of the development of wavelets is to introduce a new framework - multiscale analysis, that is proven to be effective in many applications. Recently there are increasing interests in applying wavelets for time series mining. However, most of the proposed ideas used wavelets for time series matching, few work concern about time series clustering. It is natural of thinking about using wavelets for time series clustering, as comparing the similarities between time series is the key point for both time series clustering and matching.

With multiscale analysis, we can observe the time series data in different aspects, from detail to rough. Nevertheless, it is not easy to choose the appropriate scale for clustering. We don't know which scale is better than others. This problem is circumvented by merging the cluster solutions over each scale in this paper hence we don't need to choose the proper scale. The final cluster result is the consensus of the clusters obtained with all individual scales. It has been demonstrated that classifier consensus systems such as boosting and bagging are successful in many domains. If all clusters obtained in different scales agree on how a dataset should be partitioned, aggregating the clusterings will show improvement over any of the individual clustering.

† School of Knowledge Science
Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa, 923-1292 Japan
email: {zhang-h,bao}@jaist.ac.jp

Most of the proposed consensus clustering algorithms obtain the consensus by first define a similarity matrix between instances by the frequency of two instances locate in the same meta-cluster then performing clustering algorithm with the similarity matrix. This approach needs quadratic memory and computational complexity. We propose a similarity measure between clusters in this paper. The final clustering solution is obtained by corresponding clusters within different scales.

The remainder of this paper is organized as follows. The basic knowledge of Haar wavelet transform and K-means clustering algorithm are introduced in section 2. Some related work also been reviewed in section 2. In section 3, we present our multiscale clustering algorithm. Experimental results are presented in section 4. Finally, section 5 concludes the paper with discussions of some possible future work.

2. Background And Related Work

2.1 Haar Wavelet Transform

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 approximation part, whereas the localized changes are kept in 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.

Haar wavelet is the simplest and most popular wavelet given by Haar. The benefit of Haar wavelet is its decomposition process has low computational complexity. Given a time series with length l, where l is an integral power of 2, the complexity of Haar decomposition is $O(l)$. The concrete mathematical foundation of Haar wavelets can be found in[2].

The length of 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 input time series doesn't satisfy this requirement.

The structure of decomposed hierarchical coefficients are shown in Table 1.

Table 1 The hierarchical wavelet coefficients

Scale	Coefficients				
0	A_0				
1	A_1				D_1
2	A_2			D_2	D_1
...	...			D_2	D_1
$\log_2(l)$	A_k	D_k	...	D_2	D_1

Scale 0 is the original time series. To calculate the approximation coefficients $A_j = \{a_{0,j}, a_{1,j}, \ldots, a_{l-1,j}\}$ and detail coefficients $D_j = \{d_{0,j}, d_{1,j}, \ldots, d_{l-1,j}\}$ within scale j, the approximation part of scale $j-1$ is divided into an even part including even indexed samples: $even(A_{j-1}) = \{a_{0,j-1}, a_{2,j-1}, \ldots, a_{2l-2,j-1}\}$ and an odd part including odd indexed samples: $odd(A_{j-1}) = \{a_{1,j-1}, a_{3,j-1}, \ldots, a_{2l-1,j-1}\}$. Approximation coefficients A_j are calculated as

$$A_j = \frac{1}{\sqrt{2}}(even(A_{j-1}) + odd(A_{j-1})). \quad (1)$$

The detail coefficients within scale j are calculated as

$$D_j = \frac{1}{\sqrt{2}}(even(A_{j-1}) - odd(A_{j-1})). \quad (2)$$

Therefore, the number of decomposing scales for the input time series is $log_2(l)$ except the scale zero, here l is the length of zero-padded input series.

2.2 K-means Algorithm

For a given data set $D = \{d_1, d_2, \ldots, d_N\}$ containing N instances and a given number of clusters $K, K < N$, where the target of clustering is to divide the N instances into a set of partitions. The partition can be defined as $C \equiv \{C_1, C_2, \ldots, C_K\}$, with $\cup_{k=1}^{K} C_k = D$, and $C_i \cap C_j = \phi, \forall_{i,j} : i \neq j$.

K-means algorithm is a fast and one of the most commonly used clustering algorithms[1]. It groups n samples into K clusters for a given parameter K. K-means is an iterative hill-climbing algorithm to optimize a score function which measures the intracluster similarity or intercluster dissimilarity. The typical score function is the squared-error-criterion defined as $E = \sum_{j=1}^{K} \sum_{d_i \in C_j} |d_i - m_j|^2$, where E is the sum of square error for all samples in the dataset $D = \{d_1, d_2, \ldots, d_N\}$, d_i is a sample, and m_j is the mean of the cluster C_j. m_j is

calculated by

$$m_j = \frac{1}{n_j} \sum_{d_i \in C_j} d_i, j = 1, 2, \ldots, K$$

where n_j is the number of samples in C_j. The k-means algorithm consists of the following steps:

Step 1: Randomly initialize K cluster centers from the N samples $\{d_1, d_2, \ldots, d_N\}$.

Step 2: Assign a sample d_i to a cluster $C_j, j \in \{1, 2, \ldots, K\}$, if and only if $|d_i - m_j|^2 \leq |d_i - m_p|^2, p = 1, 2, \ldots, K$, and $j \neq p$.

Step 3: Recalculate the new mean value for each cluster.

Step 4: If the value of the score function E does not changed in the last iteration, exit. Otherwise go to step 2.

2.3 Related Work

Combining multiple clustering results based on resampling or reinitializing the initial cluster centers has attracted increase attention over the last years[10),9),5),3),11),4)].The key idea of these algorithms is to define a pairwise similarity matrix between instances. Each solution is represented by a $D \times D$ matrix where the (i, j) position is either 1 if observations i and j belong to the same cluster and 0 otherwise. The average of all matrices is used as the input for a hierarchical clustering algorithm. The disadvantage of this approach is that it requires quadratic computational complexity. Our approach is to define a similarity measure between meta-clusters and correspond the clusters by this similarity measure hence different with other algorithms.

All the proposed consensus clustering algorithms yield different clustering solutions from the same data set by resampling or refining the initial cluster centers. The multiple clustering solutions are generated from each scale of wavelet transformed data set in our approach. In this regard, our idea is similar with the iterative multiscale clustering algorithm[8)]. They set the clustering obtained from rougher scale as the initial points of the detailer scale that is different with our clustering consensus approach.

3. Multiscale Time Series Clustering

The motivation behind this algorithm originates from the observation that the general shape of the time series is captured by the approximation Haar wavelet coefficients (AHWC). The AHWC within lower resolution (higher scale) correspond to more rough shape and that within higher resolution (lower scale) catch more detail information of the original series. All the AHWC share the features of the time series from rough to detail. In this case, the AHWC can be viewed as the result of smoothing the time series with kernel-varying functions. As the clustering solution generated by K-means algorithm crucially depends on the initial points, and the AHWC within each scale are different with each other, the clusterings obtained over the AHWC of different scales should also be different. However, the clustering solutions share common information from the original time series. Therefore, the more the attained solutions are agree to each other, the more the clusters are robust to the noise and variability of initialization. In this point, we have more confidence to the consensus of clusterings.

The multiscale time series clustering algorithm is summarized in pseudo-code format in Algorithm 1. In the remainder of this section we illustrate in detail with each of the procedure's steps.

Algorithm 1 Multiscale time series clustering procedure

input: a set of time series $D = \{d_1, d_2, \ldots, d_N\}$, number of clusters K.
for scale = 0 to the maximum scale **do**
  $M \leftarrow \phi$ {the joint clustering matrix, initially empty}
  $A^s \leftarrow$ Wavelet transform(D) {Obtain the Approximation Haar Wavelet Coefficients (AHWC) within the scale}
  $M^s \leftarrow$ Cluster(A^s, K) {cluster A^s into K clusters}
  $M \leftarrow M \cup M^s$
end for
$C \leftarrow$ correspond the meta-clusters within M
$P \leftarrow$ reclustering(C) {Cluster M into K clusters based on the corresponded matrix C}
output: The partition P

3.1 Clustering with The Approximation Haar Coefficients of Each Scale

We assume all the time series in the D have the same dimensionality l (This is a requirement for comparing Euclidean distance in K-means algorithm). If the time series data in D have different dimensionality, we can interpolate them to make the dimensionality to be the same. Given a time series d_n, once performed the Haar wavelet transform, we obtain a set of AHWC series $\{A_n^0, A_n^1, \ldots, A_n^{\log_2^{(l)}}\}$ as introduced in section 2.1. Here we take the original time series as the AHWC within the scale 0. Note that AHWC within the highest scale $s = \log_2^{(l)}$ only have one item, which is proportional to the mean of the time series. As the time series data often need to be normalized to reduce the influence of the scope, this item should be the same for all time series data. So we don't take the coefficients within the highest scale into calculation. Therefore, we have $\log_2^{(l)}$ AHWC series for each time series.

The K-means algorithm introduced in section 2.2 is performed over the AHWC within each scale. The clustering solution M^s corresponding to the scale s is a $1 \times N$ matrix whose ith element indicates the cluster that the ith instance belongs to. The value of the entries is an integer number ranging from one to the user specified number of clusters K. A $\log_2(l) \times N$ joint matrix M corresponding to $\log_2(l) * K$ clusters is generated by merging the M^s of each scale.

3.2 Corresponding Multiple Clustering Solutions

In order to get the final clustering solution, one needs to solve the problem of grouping the joint matrix $\log_2(l) * K$ clusters into K groups in an "optimal" fashion. Unlike the classifier combination problem, the correspondence between different clustering solutions is unknown. For example, consider two clustering solutions A and B obtained from five instances with two groups. The clustering solution A is $\{1, 2, 2, 1, 2\}$ and B is $\{2, 1, 1, 2, 1\}$, where the ith element of A and B is the group to which instance i is assigned. Although these two solutions appear to be different, they are in fact the same. The cluster 1 of A and cluster 2 of B are identical, and cluster 2 of A agrees with the cluster 1 of B. If the correspondence problem is solved, it's easy to merge the solutions together.

To solve the correspondence problem, we define a similarity measure between a group A and another group B as

$$\text{Sim(A,B)} = 1 - \frac{|A \cap B|}{\min(|A|, |B|)}$$

It is easy to prove that the defined similarity measure satisfies the following situations:

(1) $\text{Sim}(A, B) = \text{Sim}(B, A)$
(2) $0 \leq \text{Sim}(A, B) \leq 1$
(3) $\text{Sim}(A, B) = 0, \text{iff } A = B$
(4) $\text{Sim}(A, B) = 1, \text{iff } |A \cap B| = 0$
(5) $\text{Sim}(A, B) = 0, \text{if A } \supset \text{B, or B } \supset \text{A}$

We choose the first row of M as the standard clustering solution (in fact, any row within M can be chosen as the standard solution). The clusters of other solutions within M are corresponded to the the clusters of the standard solution according to the similarities between them. If the standard solution A contains clusters $\{C_1^A, C_2^A, \ldots, C_K^A\}$, and another solution B have clusters $\{C_1^B, C_2^B, \ldots, C_K^B\}$, the label of the kth cluster C_k^B should be assigned to i^*, where

$$i^* = \arg\min_i \text{Sim}(C_k^B, C_i^A)$$

3.3 Obtain The Consensus Solution

After corresponding the clusters, we can find the final clustering solution by Bayesian theorem. From the Bayesian principle, we have

$$p(c_i \mid x) = \frac{p(x \mid c_i)p(c_i)}{p(x)} = \frac{p(x \mid c_i)p(c_i)}{\sum_{i=1}^{K} p(x \mid c_i)p(c_i)}$$

, where x is an instance, c_i is the meta-cluster. Because the prior probability of $p(c_i) = \frac{1}{K}$ is equal for all meta-clusters. We have

$$p(c_i \mid x) = \frac{p(x \mid c_i)}{\sum_{i=1}^{K} p(x \mid c_i)}$$

The instance x is assigned to the i^* meta-cluster, where

$$i^* = \arg\max_i p(c_i \mid x)$$

4. Experimental Evaluation

To show the effectiveness of our approach, we tested our clustering approach on six data sets. We evaluated four algorithms in terms of the clustering quality measured by three objective clustering measurements. Corr is our proposed algorithm described in section 3. As introduced in the section 2.3, other consensus

clustering approaches used the average similarity matrix of instances as the input of a hierarchical clustering algorithm. We implemented the hierarchical clustering algorithm with single, complete, average linkage. Note that the correspondence algorithm doesn't need to do further hierarchical clustering that needs long computational time hence it is much faster than the single, complete and average algorithms.

We first summarize the evaluation measurements used in the experiments. We then briefly describe the datasets used for evaluation. Finally, we report and discuss the results of the evaluation.

4.1 Evaluation Measurements

Evaluating clustering systems is not a trivial task. Since clustering is an unsupervised learning process we lack the information of the actual partitions. The classified data is used in our experiments making comparing the labels between the clustering results and the really labels becoming possible. We used three objective clustering evaluation measurements in our experiments, the Normalized Mutual Information (NMI)[10], Conditional Entropy (CE)[4] and similarity measures (Sim)[7]. All the three measuremensts are in the range of $[0, 1]$. Higher NMI and Sim vlaue mean high agreement between the relly data classes and the clusters. Lower CE value implies high certainty that the relly data classes and the clusters have linear relation. Therefore, we prefer to maximize NMI and Sim and minimize CE.

4.2 Data Description

We used three classified datasets - CC, Trace and CBF from the UCR Time Series Data Mining Archive [⋆1]. Other three data sets are downloaded from internet. All the datasets are normalized to avoid the influence of the data scope.

- 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.
- Cylinder-Bell-Funnel (CBF): The dataset contains three types of time series: cylinder (c), bell (b) and funnel (f). We used 128 examples for each class with length 128 and time step 1.
- The personal income dataset [⋆2] (income) is a collection of time series representing the per capita personal income from 1929-1999 in 25 states of the USA. Group1 consists of the states in which the personal income grows at a high rate. The states for a group in which the personal income grows at a low rate is called group2.
- The ECG dataset (ECG) was obtained from the ECG database at physioNet [⋆3] . We use 3 groups of those ECG time-series in our experiments: Group 1 included 22 time-series representing the 2 sec ECG recordings of people having malignant ventricular arrhythmia; Group 2 included 13 time-series that are 2 sec ECG recordings of healthy people representing the normal sinus rhythm of the heart; Group 3 included 35 time-series representing the 2 sec ECG recordings of people having supraventricular arrhythmia.
- The population dataset (population) [⋆4]was a collection of time-series representing the population estimates from 1900-1999 in 20 states of the US. The 20 states were partitioned into two groups based on their trends: group 1 consisted of states had the exponentially increasing trend while group 2 consisted of states had a stabilizing trend.

4.3 The Performance of The Algorithms

The performance of all algorithms are compared in terms of the mean CE, NMI and Sim obtained from 100 different runs. Table 2 presents the average CE, NMI and Sim obtained after a fixed number of 100 iterations of the four algorithms for the six data set.The corr algorithm performs better than other three algorithms in Trace and income data and takes the same result in population data.The corr algorithm never takes the worst result for all data.

5. Conclusions And Future Work

We have presented a new scheme for multi-scale time series clustering. The consensus of the clustering solutions obtained by clustering

[⋆1] UCR time series mining archive: http://www.cs.usr.edu/ eammon/TSDMA/index.html

[⋆2] income: http://www.bea.gov/bea/regional/spi

[⋆3] ECG: http://www.physionet.org/phsiobank/database

[⋆4] population: http://www.census.gov/population/www/estimates/st-stts.html

Table 2 The average CE, NMI and Sim obtained by four algorithms from 100 distinct runs for the six datasets

CC	single	complete	average	corr
CE	0.6613	0.6349	0.6237	0.6218
NMI	0.7371	0.7048	0.6956	0.7115
Sim	0.6674	0.6682	0.6759	0.6756
CBF	single	complete	average	corr
CE	0.8305	0.7411	0.7456	0.7463
NMI	0.2973	0.3454	0.3423	0.3415
Sim	0.5705	0.5789	0.5792	0.5769
Trace	single	complete	average	corr
CE	0.6904	0.6897	0.6904	0.6890
NMI	0.5288	0.5201	0.5079	0.5367
Sim	0.5656	0.5566	0.5442	0.5701
income	single	complete	average	corr
CE	0.4823	0.4643	0.4660	0.4561
NMI	0.2245	0.2490	0.2468	0.2707
Sim	0.7435	0.7532	0.7503	0.7613
ECG	single	complete	average	corr
CE	0.8300	0.8302	0.8309	0.8307
NMI	0.2415	0.2411	0.2397	0.2410
Sim	0.5857	0.5850	0.5825	0.5835
population	single	complete	average	corr
CE	0.0391	0.0391	0.0391	0.0391
NMI	0.9458	0.9458	0.9458	0.9458
Sim	0.9854	0.9854	0.9854	0.9854

over each scale is taken as the final clustering solution. We have introduced a new clustering correspondence algorithm based on a similarity between meta-clusters. The final solution is calculated by the Bayesian theorem. The proposed consensus clustering algorithm achieve lower computational complexity than the approach of clustering with the similarity matrix between instances. We have empirically demonstrated that the proposed algorithm can offer comparable results than performing hierarchical clustering with the similarity matrix.

We intend to extend the work in the following directions:

(1) Applying the consensus clustering algorithm for clustering multi-array time series. The multi-array time series data have multiple time series associated with each instance. Traditional clustering algorithm cannot be applied to it directly. A consensus solution over the clusters associated with each time series can serve this target.

(2) Applying the algorithm to some real-life data, such as hepatitis data and financial data.

References

1) P.S. Bradley, U.M. Fayyad, and C.Reina. Scaling clustering algorithms to large databases. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 9–15, August 1998.

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) S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

4) X. Fern and C. Brodley. Randomly projection for high dimensional data: A cluster-ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 186–193, August 2003.

5) A. Fred and A. Jain. Data clustering using evidence accumulation. In *Proceedings of The 16th International Conference on Pattern Recognition (ICPR)*, pages 276–280, August 2002.

6) A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

7) 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 Minining*, pages 273–280, November 2001.

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

9) S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene-expression microaray data. *Machine Learning*, 52(1-2):91–118, 2003.

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

11) Y. Zeng, J. Tang, J. Garcia-Frias, and G. Gao. An adaptive meta-clustering approach: Combing the information from different clustering results. In *Proceedings of the IEEE Computer society Bioinformatics Conference*, pages 276–281, August 2002.