An Accelerated U-Shapelet Time Series Clustering Method with LSH

Linghao Luo and Shu Lv
School of Mathematical Sciences, University of Electronic Science and Technology of China, Qingshuihe Campus of UESTC, No.2006, Xiyuan Avenue, West Hi-tech Zone, Chengdu, Sichuan, P.R. China
Email: llh501000@163.com

Abstract. The clustering method of time series is based on the measurement of the whole time series, and the traditional clustering method is used for direct clustering. It was only recently that Keough came up with the concept of Shapelet. He later applied this concept to unsupervised time series clustering. This is a time series clustering method based on the feature of sub-series. This method has good clustering accuracy and stability. However, due to the fact that traversal sequences are all sub-sequences, the original violent search algorithm is very complex and difficult to be applied to a large number of time series data sets. In this paper, on the basis of the original method, you introduce the idea of data retrieval in the data preprocessing stage to quickly screen candidate subsequences. Then we introduce a position-aware hashing algorithm -LSH. The method is used to match the sub-sequences of the pre-processed time series data, and the wide representativeness of the high-quality u-Shapelet is used for selection. Therefore, LSH-us is proposed in this paper to speed up the extraction process of feature sub-sequences. Experimental results show that this algorithm can improve the speed of u-Shapelet time series clustering algorithm and ensure the accuracy of clustering.

1. Introduction and Backgrounds

A time series is a series of observed data arranged in chronological order, and each observation value is collected at a fixed time interval according to a given sampling rate. Time series data mining refers to the process of mining valuable information and useful knowledge from a large amount of data. Time series analysis is an important topic within many fields of research including, aerospace, finance, meteorology, motion capture, etc. However, most research on time series analysis is limited by the need for costly labeled data. This has led to an increase of interest in clustering time series data, which, by definition, does not require access to labeled data. Without any prior knowledge about the data, cluster analysis can solve the classification problem of a large amount of data and tap its internal connections. Recently, time series clustering has become an increasingly important research topic in data mining community. Algorithms for clustering time series are based on similarity, that is, distance-based, feature-based, model-based, and segmentation-based cluster analysis. Among them, in the distance-based method, a common distance metric is Euclidean distance or some improvements based on it as a similarity measure, such as dynamic bending distance (DTW). Traditional clustering analysis is mostly based on vectors, and they cannot solve the time series clustering problem well. Most time
series clustering algorithms focus on processing the entire time series data. In real life, time series data is usually a high-dimensional data that changes with time. If the entire time series data is processed, there will be a lot of noise data that will affect the final clustering effect. Secondly, in actual life, the time series data collected is not necessarily of equal length. In the end, the results we get are only reflected in the data, but not the actual differences between the classes. Most of the research on time series is mainly completed on the premise of assuming the same length of time series, which requires a lot of time and effort to preprocess the data. Some scholars use dynamic bending distance (DTW) for unequal length time series clustering, which is unacceptable for the time consumed by the algorithm. Therefore, it has been proposed to use subsequences as objects for time series clustering analysis, which is a feature-based clustering method. In 2009, Keogh et al. proposed the concept of shapelets [1]. Shapelets are the most discriminative set of sub-sequences of time series. The main idea is to classify according to local features with slight differences between different time series. Keogh's time series classification and analysis method based on shapelets subsequence has also been applied to various fields of time series data mining. After the wide application of classification, Keogh et al. extended shapelets to time series clustering and proposed the concept of unsupervised shapelets [2]. The core problem of this method is how to supervise the extraction of highly representative features in time series.

Keogh proposed the u-shapeletcs BruteForce method. The u-shapelets clustering method can avoid the influence of noisy data to the greatest extent and is suitable for the case where the sequences are not equal in length; use the u-shapelets set to perform the original time series Conversion, this method uses all subsequences in the time series data set as u-shapelets subsequence candidate sets. Each subsequence divides the data set into two subsets. The Gap value is used to measure the separation between the two subsets. The higher the resolution, the better the quality. Perform quality evaluation on all subsequences in the subsequence candidate set to find the best set of u-shapelets. Then use traditional static clustering methods (such as k-means) to cluster the distance matrix after conversion. This method has obtained very good results in the experiment. Compared with other time series clustering methods, u-shapelet is a clustering method based on local feature subsequences. It can be said that after the clustering result comes out, each of the u-shapelet we extract can be used as a core public feature of a class. All the time series of this class have a sub-sequence that has morphological similarity to this u-shapelet. This also provides a basis for why we clustered the corresponding time series in the data set into a class. In the clustering of time series, the morphological characteristics between time series are very important. Selecting a wide range of morphological features of time series has potential significance in other fields of time series data mining, including time series prediction, time series pattern extraction, and time series similarity search. However, the BruteForce method has the disadvantage of high time complexity, which is mainly manifested in that when a subsequence is selected as the u-shapelet, the quality of all subsequences in the subsequence candidate set needs to be evaluated, and the time complexity of evaluating a subsequence is too large. In the process of selecting u-shapelets, the de-redundancy method adopted relies heavily on the result of the division operation of the previous u-shapelet being selected, and each time a u-shapelet is selected, the candidate subsequence set needs to be re-set Evaluation. In the process of searching u-shapelets collection, each u-shapelets sub-sequence in the candidate set needs to be evaluated. The size of the sub-sequence candidate set also greatly affects the time complexity of u-shapelets selection. For larger data sets, the current time series clustering method based on u-shapelets is difficult to deal with.

For the Brute Force method, many scholars have carried out research. Keogh et al. proved that Euclidean distance is more competitive among all distance measurement methods [3]. Rakthanmanon proved that clustering directly on the entire time series will not only cause poor clustering effect and low accuracy, but also cause long running time and low efficiency of the algorithm [4]. On this basis, many scholars speed up the selection process of u-shapelets by using approximate calculations, random extraction, pruning strategies, and machine learning methods. In order to reduce the time complexity, Ulanova, Keogh and others reduced the dimensionality of the time series data in 2015, and transformed the time series data by Symbolic Aggregate Approximate (SAX), and selected 1%
from a large number of subsequences. As a candidate set of subsequences, referred to as Sush method for short [5]. This method is a classic case of saving time and cost. In 2016, Zhang et al. proposed an unsupervised feature learning model to learn u-shapelets from time series [6]. Zakaria et al. used approximate values to estimate the distance between time series, and used a pruning strategy to reduce the evaluation of candidate u-shapelets; both of these methods reduced the u-based to a certain extent while ensuring the accuracy of clustering.

The time complexity of shapelet's time series clustering method [7]. Due to the wide application of shapelets in time series classification algorithms, many researchers have also made a lot of efforts in studying how to quickly extract shapelets and improve shapelets, which also gives more ideas for extracting unsupervised u-shaplet. Li et al. subsequently proposed a discovery method based on principal component analysis (Principle Component Analysis, PCA), which reduced the dimension of the data through the PCA algorithm, thereby speeding up the calculation speed [8]. However, the dimensionality reduction method will lose useful information in the original data and affect the classification accuracy.

### 2. Definitions

#### 2.1. Time Series and Time Series Data

A time series, \( T = t_1, t_2, \ldots, t_n \), refers to a sequence of numerical values of the same statistical indicator arranged in chronological order. A time series data, \( D = \{T_1, T_2, \ldots, T_N\} \) is a data set consisting of \( N \) time series.

#### 2.2. Subsequence

\( S_{i,j} = t_i, t_{i+1}, \ldots, t_{i+j-1} \) is Subsequence of a time series. \( i \) and \( j \) is the starting position and length of the subsequence.

#### 2.3. Length Normalized Euclidean Distance

When measuring the similarity between time series, in order to achieve the scale shift invariance, it is necessary to perform z-normalization operation on the Euclidean distance.

\[
\text{dist}(T_X, T_Y) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (T_{X_i} - T_{Y_i})^2}
\]  

#### 2.4. Distance between Subsequence and Time Series

The distance between a sub-sequence of length \( j \) and a time series of length \( n \) is the minimum value of the Euclidean distance between this sequence and all sub-sequences of length \( j \) of this time series

\[
\text{sdist}(S, T) = \min_{1 \leq i \leq n-j} \text{dist}(S, T_{i,j})
\]

#### 2.5. Orderline

The one-dimensional array generated by the Euclidean distance between the subsequence and all time series is arranged in increasing order. There is an interval between each distance on the Orderline, this interval can divide the array into two datasets: \( D_A \) and \( D_B \).

#### 2.6. U-shapelet

Unsupervised-shapelet is a two-tuple \((S, dt)\). \( S \) is essentially the most representative local subsequence of time series in time series clustering. \( dt \) is the interval where the gap value between \( D_A \) and \( D_B \) is the largest on the orderline.

#### 2.7. U-shapelet Distance Matrix

If the data set has \( n \) time series and u-shapelet candidate set has \( n \), the u-shapelet distance matrix is
\[ DIS = \left( \begin{array}{ccc} sdist(u_1, T_1) & \cdots & \text{dist}(u_n, T_1) \\ \vdots & \ddots & \vdots \\ sdist(u_1, T_n) & \cdots & \text{dist}(u_n, T_n) \end{array} \right) \]  

(3)

2.8. Hash Function

The hash function family \( H = \{ h: S \rightarrow U \} \) is sensitive to \( (r_1, r_2, p_1, p_2) \). If \( \forall v, q \in S \)

\[ \text{If } D(v, q) \leq r_1, \Pr_H[h(v) = h(q)] \geq p_1; \]

\[ \text{If } D(v, q) \geq r_2, \Pr_H[h(v) = h(q)] \leq p_2; \]

In order to make the LSH function family effective, \( r_1 < r_2, p_1 > p_2 \) must be guaranteed.

2.9. Hash Bucket

There may be multiple elements in the same location in the hash bucket to deal with the hash collision problem. In this way, each position in the hash table represents a hash bucket.

3. Our Approach

In view of the shortcomings of the original BruteForce algorithm, which is too high in time and cannot be applied to large-scale data sets, this paper makes improvements on the basis of the original algorithm, introduces a local sensitive hash algorithm, and tries to ensure that the clustering effect of the algorithm is not reduced [9]. The above makes the efficiency of the algorithm significantly improved.

3.1. BruteForce Method

The Brute Force method is the first method proposed by the earliest author to apply shapelet to unsupervised time series clustering research. BruteForce algorithm is an iterative method to extract u-shapelet. First, given the length of the stator sequence, the sliding window technique is used to extract all the sub-sequences of the time series in the data set into the u-shapelet candidate set. In order to reduce the complexity of the algorithm, the author no longer selects all time series when extracting subsequences, but selects the first time series. First extract a subsequence in the set, use formula (2) to calculate the distance between this subsequence and all time series, sort the \( N \) distance values to generate an orderline, and according to the different intervals on the orderline, find a Maximize the interval of classification and evaluate with gap value.

\[ \text{gap} = \mu_B - \mu_A - \sigma_B - \sigma_A \]  

(4)

After finding the interval of the maximum gap value, assign this value to this subsequence, which is the quality value of this subsequence. The process of assigning quality values to all subsequences in the subsequence candidate set. Each subsequence has a corresponding gap value. Sort the gap values from small to large, and select the first subsequence as u-shapelet to enter the u-shapelet set. Next, in the time series data set \( D \), delete the time series similar to the selected u-shapelet. In order to ensure that the selected time series are clustered as much as possible, BruteForce uses \( \rho \) values to divide and establish a standard. When the distance value on the orderline is less than \( \rho \), the corresponding time series is similar to the selected u-shapelet.

\[ \rho = \mu_A + \sigma_A \]  

(5)

After removing the similar time series, the data set \( D \) has less time series than the original. Then enter the second step of the iteration, using the sliding window to extract all subsequences. The subsequent steps are the same as above, and continue to find the next u-shapelet. When the maximum gap value interval is found on the orderline, \( D_A \) contains only one distance value, then the iteration is
terminated. We believe that when $D_A$ has only one time series left, it is too unique to represent a class, so the u-shapelet extraction process is stopped. After obtaining the u-shapelet set, calculate the distance between each u-shapelet and all time series using formula (2) to generate u-shapelet distance matrix. Use this method to convert high-dimensional time series into static data that is easy to calculate. Use traditional clustering methods such as k-means to perform clustering on DIS with (3).

3.2. Locality Sensitive Hashing
Locality sensitive hashing is a popular algorithm suitable for high-dimensional data. Its main idea is that points close to each other will be hashed into the same bucket with a high probability, and points far away will be hashed into the same bucket with a low probability.

Define a series of hash functions $h_1, h_2, \cdots, h_n$. Randomly select k functions to form functions $g(x)$. If you choose $h_1(x)$ to $h_k(x)$,

$$g(x) = (h_1(x), h_2(x), \cdots, h_k(x))$$

(6)

Select L g(x), each function corresponds to a hash bucket. For each point p in the original space, it is mapped into L hash buckets through each function g(x). In this way, each point will appear in a hash bucket of L hash tables. When querying, given a query point q, L is also mapped to q using L g(x) functions, and the point that falls in the same hash bucket as q is used as a candidate result set. Calculate the distance between q and the point in the candidate result set, and select one or K closest points from it. It can be seen that LSH has the potential to perform similar searches in time series data sets.

3.3. LSH-U5

3.3.1. Algorithm Idea. Aiming at the problem of the original BruteForce method with high complexity, this paper introduces the local sensitive hash algorithm (LSH) into the u-shapelet clustering algorithm. The method basis of the LSH-us algorithm in this paper: First, when the two U-shapelets have the same length and their locations are close, their gap values are not much different. Because the two are similar in terms of morphological characteristics. Then, in the process of extracting u-shapelet, we do not need to evaluate the quality of all the sub-sequences of time series. Selecting a sub-sequence within a certain range can represent this sub-sequence well. Second, after u-shapelet is extracted and a distance matrix is generated for clustering, it can be found that the subsequence represented by u-shapelet is the common local feature of its category. All time series in the category label have such a European distance to this u-shapelet within the given range standard. It can be seen that the subsequences that can become u-shapelet must have extensive similarities. If u-shapelet is too "unique", then it cannot be a public feature. Based on the above two ideas of extracting u-shapelet, this paper will start from the direction of filtering sub-sequence to improve the extraction efficiency of u-shapelet, and then improve the clustering efficiency of u-shapelet.

3.3.2. Algorithm Steps. Step 1 Establish LSH Index
1-1 Initialize the index
1-2 Pretreatment: Think of $T = t_1, t_2, \cdots, t_n$ as a n-dimensional space. Map each time series to Hamming space.
1-3 Randomly select K h(x) functions from the hash function family to form g(x). By analogy, repeat L times to generate L g(x) as hash buckets.

Step 2 Subsequence similarity search
2-1 Using sliding window to extract all time series subsequences in D.
2-2 For each subsequence, it is embedded from the Euclidean space to the Hamming space. For L hash buckets, use the hash function to calculate the hash value, and repeat it k times. Search for the hash bucket. All subsequences in the bucket are taken as similar sequences. And record the quantity.
2-3 Sort the number of similar sequences from large to small. Given the distance threshold $\theta$, the sub-sequences below the threshold are selected as similar sequences. Select the subsequences with many similar sequences into the candidate set, the number threshold is 10%.

Step 3 Extract u-shapelet

3-1 For each subsequence in the candidate set, calculate the normalized Euclidean distance from all time series in $D$. Sort all the generated distance values from small to large to generate an orderline, and calculate the gap value of each interval on the orderline. The maximum interval is the gap value of the current subsequence.

3-2 Sort all subsequence gap values and select the largest one to enter the u-shapelet set. And delete the time series in the range of $D$ with $p$ value.

3-3 If the number of distance values in $D_A$ is not less than 2, then return to 3-1, otherwise the loop ends.

Step 4 Clustering

4-1 Generate U-shapelet distance matrix using $D$ corresponding to all elements in U-shapelet set.

4-2 Clustering distance matrix using k-means

4. Experimental Evaluation

The dataset used in the experiment is the UCR database constructed by the University of California Riverside Chen et al. In this paper, 5 general representative data sets in this time series data set are selected as experimental objects. The size of the 5 time series data sets, the number of time series classes included in the data set, and the length of the time series in the data set are all detailed in table 1.

| Type    | Data             | Training set | Test set | Number of clusters | Length |
|---------|------------------|--------------|----------|--------------------|--------|
| Sensor  | Trace            | 100          | 100      | 4                  | 275    |
| Image   | FaceFour         | 24           | 88       | 4                  | 350    |
| Simulated | SyntheticControl | 300         | 300      | 6                  | 60     |
| Sensor  | Lightning7       | 70           | 73       | 7                  | 319    |
| Motion  | GunPoint         | 50           | 150      | 2                  | 150    |

The UCR data set is divided into a training set and a test set. In order to examine the clustering effect of the method in this paper, this paper combines the two into a data set. Each data set already has a label, and there is no dispute about the clustering results in the subsequent comparison process. Therefore, this paper uses Rand Index to measure the final effect of the algorithm on UCR data set clustering. The value range of Rand Index is between 0 and 1. The closer to 1, the better the clustering effect. Let A be the number of object pairs that are placed in the same cluster in cls1 and cls2, B be the number of object pairs indifferent clusters in cls1 and cls2, C be the number of object pairs in the same cluster in cls1 but not in cls2, and D be the number of object pairs indifferent clusters in cls1 but in same cluster in cls2 [2].

$$\text{Rand Index} = \frac{A + B}{A + B + C + D}$$  \hspace{1cm} (7)

In addition to the Rand Index, this article also uses the algorithm running time as a measure. Table 2 is my current experimental results. Dynamic time planning has a wide range of applications in time series data mining. It “time-aligns” two sequences of different lengths representing the same type of things in time, and clusters unequal-length symbol time series obtained after dimension reduction. In the field of time series clustering, the unique nature of DTW enables a more precise expression of morphological features in distance measurement. In the experiment part, the original BruteForce and fast BruteForce (Select a time series to extract sub-sequences) algorithm and the time series clustering method based on DTW are used as the comparison objects, and the u-shapelet time series clustering
method based on local sensitive hashing in this paper is tested. By comparing the running time of Rand Index and the algorithm, the efficiency of LSH-us in the field of time series clustering is investigated. In the experiment, the parameters of LSH-us algorithm are selected as k=10, L=5, θ = 0.6.

| Method       | Data (slen) | Runtime (sec) |
|--------------|-------------|---------------|
| orginalBF    | Trace (40)  | 3898.18       |
|              | FaceFour (50) | 7128.51      |
|              | SyntheticControl (30) | 732.85     |
|              | Lightning7 (100) | 1453.25    |
|              | GunPoint (50)  | 11935.75     |
| fastBF       | Trace (40)  | 0.22          |
|              | FaceFour (50) | 0.09         |
|              | SyntheticControl (30) | 0.22      |
|              | Lightning7 (100) | 0.08       |
|              | GunPoint (50)  | 0.13          |
| DTW          | Trace (40)  | 22.42         |
|              | FaceFour (50) | 66.21        |
|              | SyntheticControl (30) | 22.42     |
|              | Lightning7 (100) | 37.83      |
|              | GunPoint (50)  | 27.59         |
| LSH-us       | Trace (40)  | 8.51          |
|              | FaceFour (50) | 22.16        |
|              | SyntheticControl (30) | 8.51      |
|              | Lightning7 (100) | 14.09     |
|              | GunPoint (50)  | 10.01         |

5. Conclusion and Future Work
This original method based on subsequence clustering has high research value. However, the earliest proposed BruteForce method has high clustering accuracy, but it has extremely high time complexity. As a similar nearest neighbor retrieval method in high-dimensional space, the local sensitive hash algorithm has a large number of applications in image text classification and clustering. Because of the

It can be seen from the experimental results in table 2 that the four time series clustering methods have obvious differences in algorithm running time and clustering accuracy. orginalBF, fastBF and LSH-us are all clustering algorithms based on U-shapelet. From the Rand Index, the original BF algorithm has high clustering accuracy. On the Trace dataset, its Rand Index reached 1. In addition to the particularity of the dataset itself, the original BF also showed its powerful clustering accuracy, precisely because the original algorithm traverses all subsequences. Apart from orginalBF, the remaining two clustering methods based on u-shapelet are fast clustering methods. In this paper, the algorithm LSH-us has higher Rand Index than fastBF on five different data sets. DTW is a clustering method using the entire time series. In terms of morphological geometric features, its distance method is more logical. But on the three data sets, the LSH-us Rand Index is also higher than the DTW-based clustering method, and the remaining two data sets are not much different. Although LSH-us' Rand Index is slightly lower than the original Brute Force, it is also within reasonable range. Overall, the data presented by LSH-us on the Rand Index is above average. From the perspective of algorithm running time, the running time of the latter three algorithms is far less than the time spent by the original brute force search algorithm. The time cost of orginalBF differs from them by at least two orders of magnitude. The time taken by fastBF is the least because it only selects the first time series during the extraction of u-shapelet. This also results in the selected u-shapelet from the same time series. As can be seen from the above, the clustering accuracy of this method is not high. DTW takes more time than LSH-us, which is due to the high algorithm complexity of DTW in the distance measurement process. LSH-us generally has a good advantage in operating efficiency. Taken together, the comprehensive results of LSH-us on Rand Index and runtime are better than the first three clustering methods, which also makes LSH-us universally applicable.
universal applicability and efficiency of LSH, this paper introduces LSH into the clustering algorithm based on u-shapelet. During data preprocessing, LSH is used to retrieve the similar sequence of the query subsequence, and then determine whether it has the potential to become a u-shapelet by the number of representatives. Through this, a large number of similar and non-computational subsequences are excluded, thereby reducing the size of the candidate set. From the final experimental results, LSH-us is compared with the original BF, fastBF and DTW-based clustering methods. They represent the original traversal algorithm, fast random screening and dynamic bending distance based on the entire time series. Although their Rand Index or runtime has a slightly better effect than LSH-us. However, the lower time cost is the biggest advantage of LSH-us, while ensuring that the Rand Index is in the appropriate range. Taken together, LSH-us has a more balanced presentation, making it widely applicable and efficient.

After researching this article, it is found that LSH-us has certain problems. First, the accuracy of clustering still needs to be improved. The possible reasons for this problem are: (1) Determination of optimal parameters. Especially for the determination of the length of the subsequence, some researchers have concluded it, but only the range is given. (2) whether the method based on LSH-us can go deeper into the subsequent U-shapelet clustering process. So the future work direction uses other extensions of LSH to apply to U-shapelet clustering methods. This includes KSH, KLSH, etc. On the other hand, combining the above hash retrieval methods, we seek new and improved methods for large-scale data sets and multivariate time series clustering. In the future, we will also seek the optimal clustering method for different kinds of data sets.

Reference
[1] Ye L and Keogh E 2009 Time series Shapelets: A new primitive for data mining Proceeding of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp 947-956.
[2] Zakaria J, Mueen A and Keogh E 2012 Clustering time series using unsupervised-Shapelets IEEE International Conference on Data Mining (IEEE Computer Society) pp 785-794.
[3] Keogh E and Kasetty S 2003 On the need for time series data mining benchmarks: a survey and empirical demonstration Data Mining and Knowledge Discovery 7 (4) 349-371.
[4] Rakthanmanon T, Keogh E, Lonardi S, et al. 2011 Time series epenthesis: Clustering time series streams requires ignoring some data 2011 IEEE 11th International Conference on Data Mining pp 547-556.
[5] Ulanova L, Begum N and Keogh E 2015 Scalable clustering of time series with U-Shapelets Proceedings of the 2015 SIAM International Conference on Data Mining pp 900-908.
[6] Zhang Q, Wu J, Yang H, et al. 2016 Unsupervised feature learning from time series Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence pp 2322-2328.
[7] Zakaria J, Mueen A, Keogh E, et al. 2016 Accelerating the discovery of unsupervised-shapelets Data Mining & Knowledge Discovery 30 (1) 243-281.
[8] Li Z S and He Z F 2014 Time series shapelet extraction method based on principal component analysis Computer System Application 23 (11) 145-149.
[9] Motwani R 1998 Approximate nearest neighbors: Towards removing the CUILBe of dimensionality Proceedings of the 30th Annual ACM Symposium on Theory of Computing (Dallas, USA) pp 604-61.