International Journal of Hybrid Information Technology
Vol. 10, No.4 (2017), pp. 27-40
http://dx.doi.org/10.14257/ijhit.2017.10.4.03

Improving U-shapelets Clustering Performance: An Shapelets Quality Optimizing Method

SiQin Yu[1], Qiuyan Yan[*1,2] and Xinming Yan[1]

[1]*School of Computer Science and Technology, China University of Mining Technology, Xuzhou, China, 221116*

[2]*School of Safty Engineering, China University of Mining Technology, Xuzhou, China, 221116*

ysq@cumt.edu.cn, yanqy@cumt.edu.cn, yanxm@cumt.edu.cn

Abstract

Unsupervised shapelets (u-shapelets) are time series subsequences that can best separates between time series coming from different clusters of data set without label. Because of the high computational cost, the u-shapelets are prohibited for many large dataset. Nevertheless, almost all of the current methods try to improving the u-shapelets based clustering method through reducing the computation time of u-shapelets candidate set. In this paper, we proposed a novel method improving efficiency of u-shapelets in terms of improving the u-shapelets quality. There are three contributions in our work: firstly, we show that by using internal evaluation measure instead gap score can improve quality of u-shapelets. Secondly, a novel method was proposed that applying diversified top-k query technology to filter similar u-shapelets, especially selecting the k most representative u-shapelets on the entirely shapelets candidates. Lastly, extensive experimental results show that combining internal evaluation measure and diversified top-k u-shapelets technology, our proposed method outperforms not only u-shapelet based methods, but also typical time series clustering approaches.

Keywords*: time series; clustering; u-shapelets; diversifying query*

1. Introduction

Time series clustering has attracted considerable interests over the past decades, which has become an important topic in numerous domains of research, including finance [1], meteorology [2], medicine [3], biology [4], engineering [5], and others [6]. Recently, an approach for time series clustering, u-shapelets has aroused wide concern. U-shapelets are the unsupervised extension of shapelets [7]. The shapelets are highly discriminative and descriptive which can best separate between time series from different classes of dataset. Because of powerful distinctive ability and no need for data label in advance, U-shapelets had been widely discussed in many fields [8].

The initial u-shapelets approach [8] constructs a distance map between the u-shapelets and time series instance, which can simply pass it into an off-the-shelf clustering algorithm such as k-means. In order to identify those u-shapelets that yields the optimal distance map, all the subsequences of the dataset are examined as potential u-shapelet candidates. If there are n time series in a dataset, each time series is m points long, and the length of candidates is constant, then the number of all candidates in a dataset is O(nm). Additionally, a distance computation from one candidate to all time series takes $O(nm^2)$ time. Hence, it takes $O(n^2m^3)$ to discover a u-shapelet.

The bottleneck of the existing approaches includes two parts: computing all of the distances between time series subsequences in the dataset and choosing the best subsequence as a u-shapelet. The discovery time of u-shapelets has been minimized by

ISSN: 1738-9968 IJHIT

transforming series to the SAX representation and examining only a special 1% of the data [10]. Furthermore, speed-ups have also been attempted by using of the complexity-based lower bound [9]. All these works improve the efficiency by approximation, and may lose the best representative shapelets. In this work, we show how we improve the efficiency and accuracy through exactly u-shapelets computing. Our key observations here are:

- Selection high quality u-shapelets set is an operative way to enhance the finial cluster efficiency. Through selection the representative and discriminatory u-shapelets can remove the redundancy efficiently. It is easy understood that the less u-shapelets left, the smaller dimension distance map has, and the less time of final cluster method consume.
- High quality u-shapelets should have the best distinctive and descriptive ability which can exactly measure the clustering results quality, which are also an operative way to enhance the finial cluster accuracy.

Existing methods use gap score to evaluate the quality of u-shapelets. From our point of view, the gap score only considers the separation between pre-step clusters, the right evaluation measure should take both the compactness of the pre-step clusters and separation of all the current subsets into consideration. To confirm this idea, we investigate different internal cluster quality measures in u-shapelet extraction process. Moreover, in order to select the most distinctive shapelets, we removed the redundancy from the candidates set. Existing methods define a parameter theta which depend on pre-step cluster results more when get rid of the redundant candidates. In this paper, we proposed a novel distinct shapelets selection method which applied diversified top-k query technology to filter similar candidates and selected the k most representative u-shapelets in candidates set.

Therefore, our contributions can be summarized as follows:

1. Different cluster quality measures are investigated for using as quality measures in u-shapelet extraction and the optimal u-shapelets quality measure is determined.
2. Combing the new quality u-shapelets measure, we introduce a diversified top-k query technology to filter similar u-shapelet candidates, and select the best k u-shapelets.
3. Evaluating our proposed algorithms by comparing them with the shapelet-based algorithm and traditional algorithm.

2. Definitions and Background

For completeness, we begin by reviewing all necessary definitions and defining the key items in this paper. Then we briefly review the brute force algorithm and present our motivation.

2.1. Definition

Definition 1: Distance Between time series and subsequence. The distance between a time series T and a subsequence S of length l is the minimum distance between S and all possible subsequences of length l in T, denoted as sdist(S, T).

Definition 2: U-shapelet candidate. A u-shapelet candidate is a tuple <S, d> where S is a subsequence, d is distance threshold which can separate the dataset into two smaller groups, DL and DR. The number of time series in DL and DR are nL and nR, respectively.

Definition 3: Distance map [8]. A distance map contains the sdists between each of the u-shapelets and all the time series in the dataset. If there are m u-shapelets for a dataset of N time series, the size of the distance map is [N×m] where each row is a time series entity and each column is a distance vector of a u-shapelet.

Definition 4: Diversified top-k query [11]. Given a list of search results L = $\{v_1, v_2 \ldots v_n\}$. For each $v_i \in L$, the score of v_i is denoted as score(v_i). For any two results v_i, $v_j \in L$, there is a user defined similarity function sim(v_i, v_j) and a threshold τ. If sim(v_i, v_j) > τ, the result v_i is similar to v_j, denoted as $v_i \approx v_j$.

Given an integer k where $1 \leq k \leq n$. The diversified top-k query is to search a list of k results R which satisfied the following conditions:

1) $R \subseteq L$ and $|R| \leq k$.

2) For any $v_i \in R$ and $v_j \in L\text{-}R$, score(v_i) > score(v_j) where L-R = $\{ v | v \in L, v \notin R\}$.

3) For any two results v_i, $v_j \in L$ and $v_i \neq v_j$, if $v_i \approx v_j$, then $\{ v_i, v_j \} \not\subset R$.

Definition 5: Similar shapelets. There are two u-shapelet candidates $<S_1, d_1>$ and $< S_2, d_2>$. If dist(S_1, S_2) < min(d_1, d_2), the candidate S_1 is consider similar with the candidate S_2,denoted as $S_1 \approx S_2$, where dist(S_1, S_2) is the two subsequences' distance.

Definition 6: Top-k U-shapelets. Given a set of u-shapelet Candidates= $\{s_1, s_2, \ldots, s_n\}$, and an integer k where $1 \leq k \leq n$. For each candidate $s_i \in$ Candidates, the quality value of the s_i is Q(s_i). The diversified top-k u-shapelets, denoted as Ush, is a list of result that satisfied the following conditions:

1) Ush $\subseteq$ Candidates and |Ush| $\leq$ k.

2) For any $s_i \in$ Ush and $s_j \in$ Candidates-Ush, if $s_i \approx s_j$, then Q(s_i)>Q(s_j) where Candidates-Ush=$\{s | s \in \text{Candidates}, s \notin \text{Ush} \}$

2.2. Brute Force U-shapelets Selection Method

The current u-shapelets based clustering methods are all extensions of the original work [8], and they are mainly to improve the speed of the u-shapelets' discovery process. The original work of finding the u-shapelets is defined in Algorithm 1.

Algorithm 1. U-shapeletsSelection(D, sLen)

Input: D: dataset; sLen: u-shapelet length
Output: Set: set of u-shapelets

```
Set = []    //set of u-shapelets, initially empty
ts = D(1,:)  //a time series of the dataset
while true
 Set=[ ]
 for sl = sLen(1) to sLen(end)
   Candidates = GenerateAllCandidates(ts ,sLen);
   for each cand in Candidates
     [gap,dt] = computeGap(cand,D)
...end for
.end for
 index1= max(gap)
 Set = Candidates(index1)
 dis = computeDistance(Candidates(index1),D)
 D_L = find(dis<dt)
 if length(D_L) == 1,break;
 else
   index2 = max(dis), ts = D(index2,:)
   θ = mean(disD_L) + std(disD_L)
   D* = find(dis<θ), D = D – D*
 end if
end while
return Set
```

The brute force algorithm iterative searches the u-shapelets. In each iteration, it includes two parts: first, they generate all possible u-shapelet candidates and calculate each candidate's gap score (lines 5-10); second, they select the candidate with maximum gap score to be a u-shapelet and use the parameter theta to remove the redundant candidates (lines 11-20). The algorithm terminates when the size of D_L is just one.

There are three problems in Algorithm 1:

(1) High computation complexity. The excessive amount of u-shapelet candidates makes the brute force algorithm intractable for large datasets. As has been illustrated, the time complexity of a u-shapelet is $O(n^2m^3)$.

(2) The measure quality of u-shapelet only considers the separation between two clusters. A u-shapelet uses a distance threshold to divide a dataset D into two clusters, D_L and D_R. And then function ComupteGap() uses the gap score between the distance vectors of D_L and D_R to evaluate the quality of a u-shapelet. The function ComupteGap() only measures the separation between two clusters and ignores the compactness of the cluster itself. We consider that a good quality measure should have ability of attaining high intra-cluster similarity and low-inter-cluster similarity.

(3) The parameter θ depends on the pre-step cluster result. From lines 18-19 in Algorithm 1, it can be seen that the value of parameter θ which is used to get rid of the redundant shapelets is depending on the former cluster D_L. Once D_L is not appropriate, the incorrect parameter theta will influence the process of searching the other u-shapelets, and eventually led to a decline in clustering accuracy.

In order to solve the three problems mentioned above, we try to find a new quality measure for u-shapelet and select the distinct u-shapelets independent with the pre-cluster results, from these two points, improve the final cluster accuracy and efficiency.

3. The Proposed Method

In this section, we first discuss the u-shapelet quality measures and find the new measure to replace traditional gap score (Section 3.1). Then we propose a new method applying diversified top-k query technology to filter similar u-shapelet candidates and select the best k distinct shapelets to improve the clustering accuracy (Section 3.2).

3.1. Alternative u-shapelet Quality Measures

As discussed in Section 2, a right evaluation measure should take both the compactness and separation of subsets into consideration. In this section, in order to find the most appropriate quality measure of u-shapelet, we investigate some internal clustering quality measures which consider compactness and separation respectively. We select the most used three internal clustering measures: The Root-mean-square standard deviation, R-squared and I index as the research objects. However, extensions to other internal cluster measures are trivial.

In order to easy understand the conception of the three cluster quality measures, we first introduce the notations used in the definition of each measures: *Dis* is a distance vector which is the distances between u-shapelet and all the time series in dataset, *n* is the number of time series in dataset, *g* is the center of whole distance set *Dis*, *P* is the number of dimensions of *Dis*, *NC* is the number of groups, C_i is the i-th group, c_i is the center of group C_i, and $d(x, y)$ is the distance between points *x* and *y*. In our experiment, we choose the arithmetic mean to compute the values of *g* and c_i.

a) The Root-mean-square standard deviation (RMSSTD) [12]: this measure is the square root of the pooled sample variance of all variables:

$$RMSSTD = \left(\frac{\sum_i \sum_{x \in C_i} \left\| x - c_i \right\|^2}{P \sum_i n_i - 1} \right)^{1/2} \tag{1}$$

b) The R-squared (RS) [13]: this measure is the ratio of sum of squared distances between objects in different groups to the total sum of squares:

$$RS = \frac{\sum_{x \in Dis} \|x - g\|^2 - \sum_i \sum_{x \in C_i} \|x - c_i\|^2}{\sum_{x \in Dis} \|x - g\|^2} \tag{2}$$

c) The I index(I): The index [14] adopts the maximum distance between group centers to measure separation and distance from a data point to its group center for compactness.

$$I = \left(\frac{1}{NC} \frac{\sum_{x \in Dis} d\left(x, g\right)}{\sum_i \sum_{x \in C_i} d\left(x, c_i\right)} \max_{i,j} d\left(c_i, c_j\right) \right)^P \tag{3}$$

These internal clustering measures consider the information intrinsic to the data alone. RMSSTD measures the compactness of groups so the value of RMSSTD should be as small as possible. The RS is intuitive and simple measures the separation between groups. Thus, the RS value should be high. Moreover, I index are the ratio of the separation to the compactness. A large separation and a small compactness determine well-defined groups. Hence, the I index value should be high.

We use the method proposed in [10] to evaluate those new quality measures. The SUSh algorithm requires setting a parameter to decide the length of u-shapelet. For fairness, we choose same value when we perform experiments to compare gap score, RMSSTD, RS and I index. Table 1 show the performance of SUSh using four quality measures on 22 datasets which are described in Table 2. The winning method that achieves the highest accuracy/time on each dataset is distinguished in bold. In accuracy, the I index has the best performance on 11 of 22 data sets. Additionally, the I index is better than the Gap, the RMSSTD and the RS in 19, 19 and 13 datasets, respectively. Furthermore, the RS gets the most accurate on 7 of 22 datasets and the RMSSTD worst. It can be seen taking both compactness and separation into consideration can get the best performance, only considering the separation is in the second place, only using the compactness is the worst. In runtime, we can observe that the clustering times of four quality measures are quite similar and there is no evidence that either measure is better. And the purpose of this paper is improving the performance of u-shapelets in terms of improving the u-shapelets quality. Thus, we prefer to select the more effective measure when the runtimes are similar. We conclude that the I index should be a good choice for u-shapelet quality which can effectively improve accuracy of clustering and does not increase the discovery time.

Table 1. Clustering Accuracy and Discovery Time of Sush with Different Quality Measure on 22 Datasets

Dataset	Clustering Accuracy(Rand Index)				Discovery Time(seconds)				slen
	Gap	**RMSSTD**	**RS**	**I index**	**Gap**	**RMSSTD**	**RS**	**I index**	
50Words	0.3167	0.5654	**0.5747**	0.5681	611.7	636.9	550.9	**523.5**	50
Adiac	0.3822	0.4988	**0.5819**	0.5486	341.3	222.1	**197.7**	198.6	50
Beef	0.5927	0.5044	**0.6355**	0.6092	11.5	**9.0**	12.4	11.3	50
CBF	**0.7441**	0.6003	0.7325	0.7233	364.6	330.8	318.5	**316.0**	35
Coffee	0.8578	0.5795	0.8400	**0.8749**	4.6	4.9	4.3	**3.7**	50
Cricket_X	0.4916	0.4877	0.6244	**0.6373**	1320	**813.9**	955.9	871.9	35
Diatom.	0.6802	0.6048	0.7542	**0.7569**	89.9	**55.4**	64.1	64.2	150
ECG200	0.5782	**0.6010**	0.5998	0.5952	5.8	4.9	4.6	**4.5**	50
ECGFiveDays	0.8466	0.5321	0.8007	**0.8794**	243.5	**181.1**	194.0	190.2	50
Face(Four)	0.9300	0.6940	**0.9511**	0.9463	56.9	**35.9**	49.8	48.9	60
Fish	0.4808	0.5883	**0.7434**	0.6332	269.3	**182.2**	236.2	210.8	50
Gun_Point	0.5641	0.5200	**0.5846**	0.5424	11.8	10.7	**7.0**	10.4	50
Lighting2	0.4923	0.4917	0.4974	**0.4993**	195.2	123.5	**89.7**	105.1	50
Lighting7	0.5906	0.6164	0.6822	**0.6940**	30.9	22.5	**21.7**	23.9	120
MedicalImages	0.3852	0.4925	0.4937	**0.5134**	200.0	201.7	175.2	**172.7**	50
MoteStrain	0.5104	**0.5282**	0.5109	0.5115	266.9	215.6	219.7	**209.7**	30
OliveOil	0.6746	0.7863	**0.7993**	0.7764	9.3	9.9	**6.6**	7.1	100
SwedishLeaf	0.3338	0.4107	0.4363	**0.5005**	309.7	**248.8**	281.5	291.9	50
SynthetieControl	0.8669	0.7102	0.8653	**0.8725**	62.8	**26.6**	53.3	46.9	30
Trace	**1**	0.7639	0.8487	0.8744	66.4	36.2	**35.5**	41.2	35
TwoLeadECG	0.5194	0.6285	0.5817	**0.6482**	**0.2**	0.4	0.5	0.3	30
WordsSynonyms	0.3164	0.3380	0.5569	**0.5966**	593.9	**482.9**	513.2	563.3	50
Total Wins	**2**	**2**	**7**	**11**	**1**	**10**	**6**	**6**	
AVG	**0.5979**	**0.5701**	**0.6680**	**0.6728**	**230.3**	**175.3**	**181.5**	**178.0**	

3.2. Our Method

Based on the appropriate quality measures, we can improve the quality of u-shapelet candidates. The next key problem is how to remove the redundant shapelets from candidate set independently from the pre-step clusters, as the same time, find the best descriptive and distinctive u-shapelets. To resolve this problem, we introduce the diversified top-k query technology to find top-k optimal u-shapelets, named as DivUshapCluster. The diversified top-k query technology [11] takes both query relevance and diversity into consideration and has already been applied to many areas, such as, document search [15], web search [16], graph search [17] and others [18]. Our proposed diversified top-k u-shapelets selection method is able to obtain the k most representative u-shapelets and those u-shapelets are no similar with each other. Further, we can obtain high quality distance map and improve the performance

The Algorithm 2 details our proposed method. First, we generate a candidate set of constant length from the entire dataset in line 2. For each subsequence, we transform it to the SAX representation and choose some small fraction of subsequences as u-shapelet candidates. Referencing to [10], this transformation can get two orders of magnitude speed up in the u-shapelet discovery process. In lines 3-5, we compute the quality of all u-shapelet candidates which is defined in Algorithm 3. Once the qualities of candidates are measured, a set of u-shapelets that have best qualities and no one similar with others are obtained by the diversified top-k selection method. In lines 6-16, we iterative search the best u-shapelet. In each iteration, we find the u-shapelet with best quality which is already computed in lines 3-5. Once the similar candidates of the selected/best u-shapelet are found, we remove them from the set of candidates (line 14). We repeat this procedure until the number of u-shapelets achieves k. After generating top-k optimal u-shapelets, we compute the distance vector of each u-shapelet and add the distance vector to the distance map in the for loop of lines 17-21. Finally, the distance map is passed into k-means and the cluster label for each time series is returned.

The Algorithm 3 is a subroutine in lines 3-5 of Algorithm 2 to compute the quality of a u-shapelet candidate. We first compute the distance vector *dis* of a candidate and sort the *dis* in a descending order. Then, in line 4-17, in terms of each possible distance threshold *d*, the *dis* is separate into two clusters: D_R and D_L. The I index calculated according to this separation reflects the quality of current u-shapelet candidate. We select the maximum I index value as the quality of candidate and the according u-shapelet tuple is (*ush*, *d*).

Algorithm 2 DivUshapCluster(D, sLen, k, n)

Input: D: dataset; sLen: ushapelet length; k: the number of u-shapelets;
n: number of clusters

Output: cluster label for each time series in the dataset

```
1: Ush= ∅ , i =1, DIS=[ ]
2: CandidateUsh = GenerateCandidates(Data, sLen)
3: for i = 1 to | CandidateUsh |
4:   assessQuality(CandidateUsh[i], Data)
5: end for
6: while i < k
7:   ush = arg max_{ush∈CandidateUsh} ush.quality
8:  Ush.add(ush)
9:  i=i+1
10:  for j = 1 to | CandidateUsh |
11:    if (CandidateUsh[j] ≈ ush)
12:      deletUsh.add(CandidateUsh[j])
13:  end for
14:  CandidateUsh=CandidateUsh-SubUsh
15:  if | CandidateUsh|=0 , break;
16:end while
17:for cnt = 1 to |Ush|
18:  ush = Ush[cnt]
19:  dis = computeDistance(ush, D)
20:  DIS=[DIS dis]
21:end for
22:[cluster_centers, Result] = k-means(DIS, n)
23:return Result
```

Algorithm 3 assessQuality(ush, Data)

Input: ush: a u-shapelet candidate; D: dataset

Output: quality: the quality of the u-shapelet candidate

```
1:  dis = computeDistance(ush, D);
2:  disSorted = sort(dis);
3:  quality = 0;
4:  for l = 1 to |dis| - 1
5:   d = dis(l);
6:   Dr = find(disSorted < d);
7:   Dl = find(disSorted > d);
8:   r = |Dr|/|Dl|;
9:   if 1/k < r < (1-1/k)
10:    ma = mean(disSorted(Dr)); mb = mean(disSorted(Dl));
11:    m = mean(disSorted);
12:    U = sum(abs(dis-m)) * abs(ma-mb);
```

```
    B = sum(abs(dis(Da)-ma))+sum(abs(dis(Db)-mb));
    I = U/(2*B);
  end if
  if I > quality, quality = I;
end for
return quality
```

To present a more direct insight of our algorithm, we test DivUshapCluster method on Coffee dataset. DivUshapCluster use the diversified top-k u-shapelets selection method to filter similar u-shapelets and get the most representative u-shapelets. In Figure 1, DivUshapCluster has selected two optimal u-shapelets and the distance map of two u-shapelets is plot in two-dimensional space. It can be seen that using the distance map, we could get good clustering.

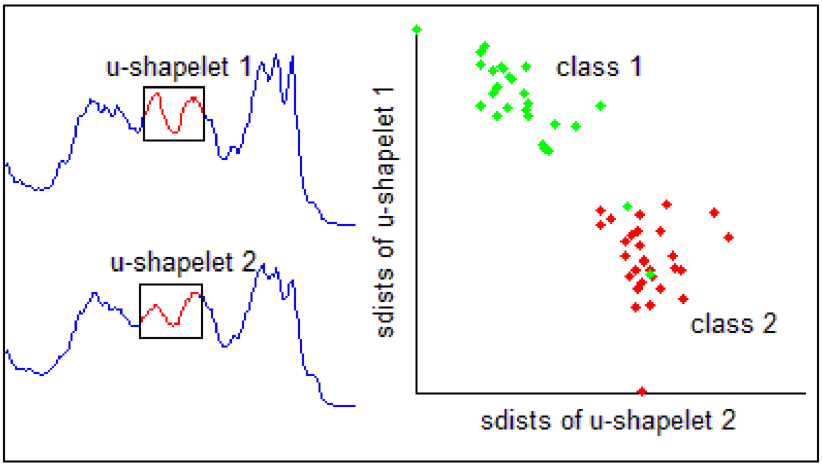


Figure 1. Coffee Dataset: Two U-Shapelets (Marked with Red) Selected from Divushapcluster and on Right the Distance Map of the U-Shapelets

4. Experiment Evaluations

In this section, in order to evaluate the performance of the proposed method (DivUshapCluster), we compare DivUshapCluster against two u-shapelet based methods (Brute Force algorithm [8] and Scalable U-shapelet [10], denoted as SUSh) and three traditional clustering methods (K-means [19], hierarchical [20] and spectral method [21]) in Section 4.2.

4.1. The Experimental Settings

We ran our experiments on a personal computer with Intel(R) Core(TM) i5-3470 CPU 3.20GHz, 4GB RAM, and Matlab R2012b(32-bit). In order to demonstrate the performance of the proposed method, we use 22 datasets from the UCR time series collection which are used commonly. The details of the datasets are shown in Table 2. For each dataset the number of series instances, the number of classes and the length of the time series is presented.

Table 2. Description of Datasets

No	Dataset	Size	Length	Class	No	Dataset	Size	Length	Class
1	50Words	905	270	50	12	Gun_Point	200	150	2
2	Adiac	781	176	37	13	Lighting2	121	637	2
3	Beef	60	470	5	14	Lighting7	143	319	7
4	CBF	930	128	3	15	MedicalImages	1141	99	10
5	Coffee	56	286	2	16	MoteStrain	1272	84	2
6	Cricket_X	780	300	12	17	OliveOil	60	570	4
7	Trace	200	275	4	18	SwedishLeaf	1125	128	15
8	ECG200	200	96	2	19	Synthetie Control	600	60	6
9	ECGFiveDays	884	136	2	20	DiatomSizeReduction	322	345	4
10	Face(Four)	112	350	4	21	TwoLeadECG	1162	82	2
11	Fish	350	463	7	22	WordsSynonyms	805	270	25

4.2. DivUshapCluster

4.2.1. Parameter Verifying

According our conclusion from the previous section, for convenience, we use I index as the u-shapelet quality measure in all following experiments. DivUshapCluster has two parameters, the length of u-shapelets slen and the number of u-shapelets k. The feature and length of time series are variable in different datasets so that the optimal length of u-shapelets is not a constant value. In our experiment, we show those suitable lengths of u-shapelets in different datasets in Table 3. Moreover, to obtain a relatively suitable fixed value of k, we perform our approach in 22 datasets with different value of k. The Figure 2 shows the change of the average accuracy in 22 datasets with varying k. We find that when the accuracy reaches a certain value, it would be stable when k equals to 10. Thus, we choose 10 as the value of k for the following experiments.

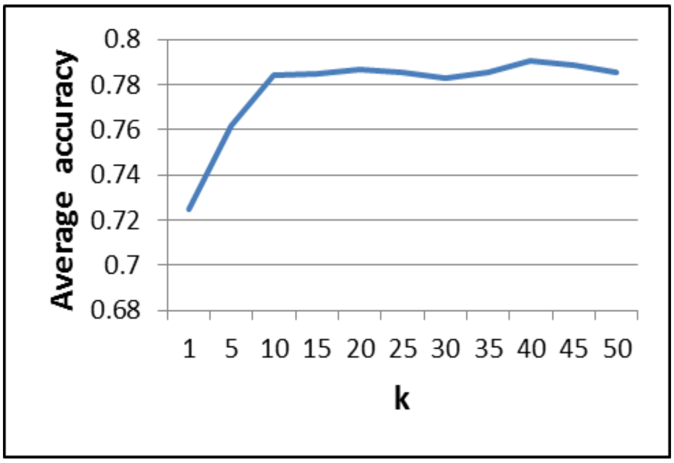


Figure 2. Changes of the Average Accuracy on 22 Datasets with the Increasing of K

4.2.2. Accuracy Testing

Having decided the parameters of our algorithm, we compare DivUshapCluster against two u-shapelet based methods and three traditional clustering methods. As shown in Table 3, DivUshapCluster performs better than others on 22 datasets which wins on 8 out of the 22 datasets and get the highest average accuracy. In experiments, in order to be fair to u-shapelet based approachs, we use the Rand index number they reported on these datasets [22].

In the Table 3, it can also shows that DivUshapCluster is better in 17 datasets and worse in 4 datasets in comparison to the Brute Force algorithm, is better in 17 datasets and worse in 4 datasets in comparison to the SUSh algorithm. This result demonstrates that DivUshapCluster has the ability to improve u-shapelet clustering performance generally. In particular, the accuracy is improved by more than 30% in 6 datasets.

Additionally, comparison to other traditional clustering methods, Table 3 report that DivUshapCluster outperforms hierarchical and spectral clustering on 16 datasets and K-Means on 14 datasets. It is obvious that each method has get best performance in some datasets but DivUshapCluster.

Table 3. Clustering Accurac of Divushapcluster and Rival Methods on 22 Datasets

DataSet	U-shapelet Based Methods			Traditional Clustering Methods			slen
	DivUshap Cluster	Brute Force	SUSh	HC	Spectral	K-Means	
50Words	0.94156	0.64067	0.63998	**0.95172**	0.88476	0.95019	50
Adiac	0.95991	0.30307	0.3109	0.82292	**0.96377**	0.93992	50
Beef	**0.70509**	0.49379	0.49379	0.59266	0.47807	0.65812	50
CBF	0.77943	0.45631	0.46576	0.72401	**0.88744**	0.70276	35
Coffee	**0.96429**	0.52273	0.63701	0.50130	0.80519	0.72918	50
Cricket_X	**0.85973**	0.70162	0.67975	0.81487	0.18437	0.85493	35
Diatom.	0.79119	0.67338	0.69354	0.30590	0.36377	**0.91241**	150
ECG200	0.62819	**0.6495**	**0.6495**	0.49829	0.55558	0.61331	50
ECGFiveDays	**0.91920**	0.50707	0.50707	0.52822	0.50663	0.49996	50
Face(Four)	0.87001	0.93951	**0.94514**	0.77011	0.51255	0.73955	60
Fish	0.82641	0.36838	0.36601	0.77791	**0.83071**	0.78237	50
Gun_Point	0.49774	0.56447	**0.5702**	0.50734	0.49749	0.49749	50
Lighting2	0.50000	0.51309	0.50758	**0.55950**	0.52094	0.51021	50
Lighting7	0.78036	0.6798	0.40609	0.79129	0.45957	**0.79855**	120
MedicalImages	**0.66707**	0.54515	0.51424	0.64695	0.51261	0.66513	50
MoteStrain	0.52574	0.54292	0.50985	0.50200	0.50282	**0.70751**	30
OliveOil	0.80893	0.72994	0.72994	0.78870	**0.85031**	0.83636	100
SwedishLeaf	**0.90546**	0.33618	0.33818	0.65750	0.58168	0.88306	50
SynthetieControl	**0.92922**	0.78701	0.87013	0.86869	0.88230	0.86871	30
Trace	**1**	**1**	**1**	0.75030	0.83569	0.74947	35
TwoLeadECG	0.50198	0.50185	0.50139	0.50545	**0.50859**	0.50207	30
WordsSynonyms	0.88342	0.65328	0.64546	0.89061	0.17036	**0.89508**	50
ToTal Wins	**8**	**2**	**4**	**2**	**5**	**4**	
AVG Accuracy	**0.78386**	**0.595896**	**0.590069**	**0.670738**	**0.604327**	**0.740743**	

4.2.3. Runtime Testing

Up to now, we have already verified that DivUshapCluster outperforms the rival methods in terms of clustering accuracy. In this section, we focused our evaluation on runtime. Both the Brute Force and SUSh are u-shapelet based methods, and it has been demonstrated that the SUSh is two orders of magnitude faster than the Brute Force. Thus we compare SUSh against DivUshapCluster on 22 datasets and Figure 3 illustrate that DivUshapCluster is slightly faster than SUSh in most datasets.

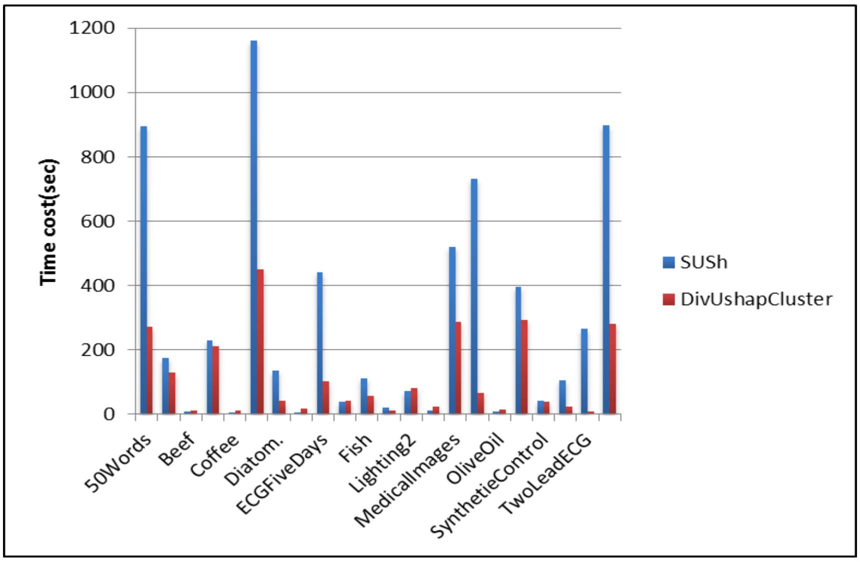
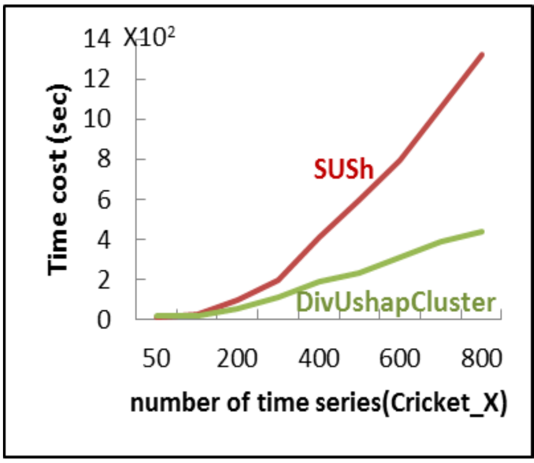
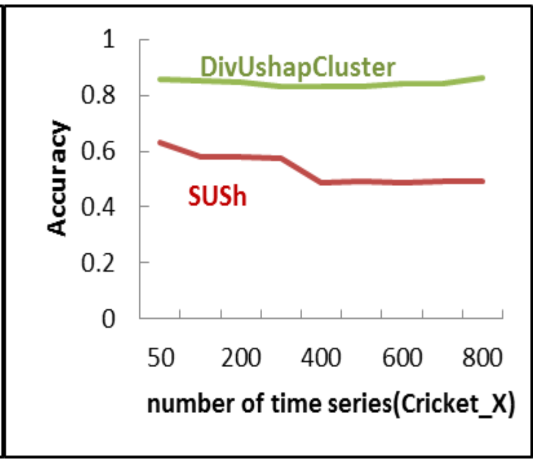


Figure 3. Running Time Comparison between Divushapcluster and Sush on 22 Datasets

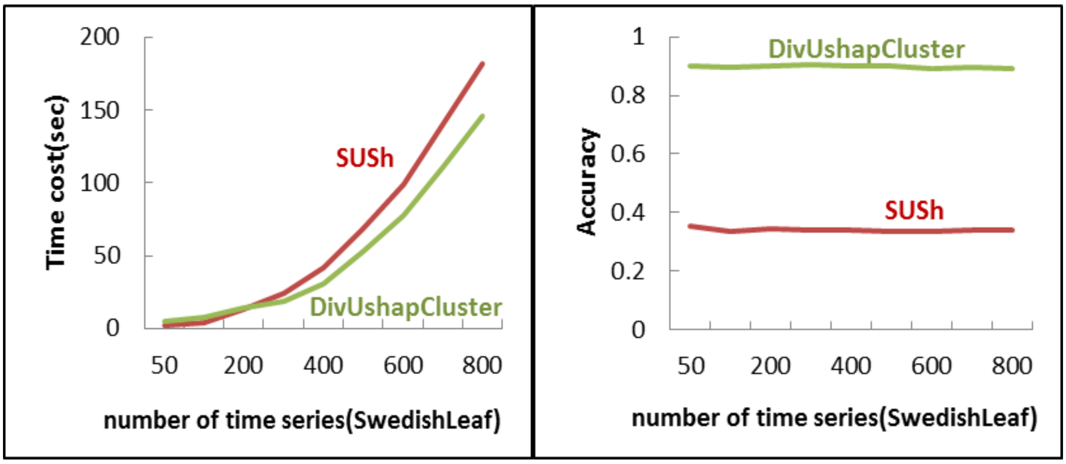
To compare DivUshapCluster and the SUSh in more details, we test on two large datasets in the UCR time series archives, Cricket_X and SwedishLeaf. Figure 4.a) c) shows the runtime when the number of time series is varied from 50 to 800 and Slen is 35. Figure 4.b) d) shows the corresponding accuracy. The running time of SUSh in Figure 4.a) increases from 9 seconds to 22 minutes from n=50 to 800, while our algorithm‘s running time increases from 15 seconds to 7 minutes. Although both algorithms only examining small fraction of all u-shapelet candidates, the u-shapelet extraction process in SUSh need recalculate the distance vectors of candidates when searching a new u-shapelet while DivUshapCluster only need calculate the distance vectors of candidates once. Furthermore, the process of removing the redundant u-shapelets in SUSh follows the Brute Force. As discuss in Section 2, the inappropriate removing operation will influence the process of searching other u-shapelets. The running time and accuracy will be affected eventually. Thus, although there are little difference between the running time of two algorithms in Figure 4.c), it can be seen that in Figure 4.d) the accuracy of DivUshapCluster is much more than SUSh.



a) Time Cost on Cricket_X Dataset



b) Accuracy on Cricket_X Dataset



c) Time Cost on SwedishLeaf Dataset d) Accuracy on SwedishLeaf Dataset

Figure 4. Time and Accuracy Comparison between Divushapcluster and Sush on 2 Datasets for Increasing Large Datasets Size

5. Conclusion

U-shapelets are discriminative subsequences of a time series dataset that can best separates time series coming from different clusters of dataset without label. In this paper, we proposed a novel method improving efficiency of u-shapelets method in terms of improving the u-shapelets quality. We demonstrated that the I index should be a good choice for u-shapelets quality which can effectively improve quality of u-shapelets. We proposed a novel method that filter similar u-shapelets and extract the k most representative u-shapelets using a diversified top-k query technology. Extensive experimental evaluations on various datasets have shown that DivUshapCluster outperforms not only u-shapelet based methods, but also typical time series clustering approaches. Furthermore, we demonstrated the running time of our approach is as fast as the current state-of-the-art and even faster on some datasets.

Acknowledgments

This work is supported by the Natural Science Foundation of Jiangsu Province of China (BK20140192).

References

[1] E. Ruiz, V. Hristidis, C. Castillo, A. Gionis, A. Jaimes, "Correlating financial time series with micro-blogging activity", Proceeding of the fifth ACM International Conference on Web Search and Web Data Mining, Seattle, Wa, Usa, **(2012)** February 08-12.

[2] R. Honda, S. Wang, T. Kikuchi, O. Konishi, "Mining of Moving Objects from Time-Series Images and its Application to Satellite Weather Imagery", Journal of Intelligent Information Systems, vol. 19, no. 1, **(2002)**, pp. 79-93.

[3] S. Hirano, S. Tsumoto, "Cluster Analysis of Time-Series Medical Data Based on the Trajectory Representation and Multiscale Comparison Techniques", Proceedings of the Sixth International Conference on Data Mining, Washington, DC, USA, **(2006)** December 18-22.

[4] D. Jiang, J. Pei, M. Ramanathan, C. Lin, C. Tang, A. Zhang, "Mining gene–sample–time microarray data: a coherent gene cluster discovery approach", Knowledge and Information Systems, vol. 13, no. 3, **(2007)**, pp. 305-335.

[5] M. Zhang, A. A. Sawchuk, "Motion primitive-based human activity recognition using a bag-of-features approach", Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, Miami, Florida, USA, **(2012)** January 28-30.

[6] T. W. Liao, "Clustering of time series data-a survey", Pattern Recognition, vol. 38, no. 11, **(2005)**, pp. 1857-1874.

[7] L. Ye, E. Keogh, "Time series shapelets: a new primitive for data mining", Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, **(2009)** June 28 – July 01.
[8] J. Zakaria, A. Mueen, E. Keogh, "Clustering Time Series Using Unsupervised-Shapelets", Proceedings of the IEEE 12th International Conference on Data Mining, **(2012)** December 10-13.
[9] J. Zakaria, A. Mueen, E. Keogh, N. E. Young, "Accelerating the discovery of unsupervised-shapelets", Data Mining and Knowledge Discovery, vol. 30, no. 1, **(2016)**, pp. 243-281.
[10] L. Ulanova, N. Begum, E. Keogh, "Scalable Clustering of Time Series with U-Shapelets", Proceedings of the 2015 SIAM International Conference on Data Mining, **(2015)**.
[11] L. Qin, J. X. Yu, L. Chang, "Diversifying top-k results", Proceedings of The Vldb Endowment, vol. 5, no. 11, **(2012)**, pp. 1124-1135.
[12] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "On Clustering Validation Techniques", Journal of Intelligent Information Systems, vol. 17, no. 2, **(2001)**, pp. 107-145.
[13] M. Hassani, T. Seidl, "Internal Clustering Evaluation of Data Streams". Trends and Applications in Knowledge Discovery and Data Mining, **(2015)**.
[14] U. Maulik, S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no .12, **(2002)**, pp. 1650-1654.
[15] Y. Zhang, J. Callan, T. Minka, "Novelty and Redundancy Detection in Adaptive Filtering", Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland, **(2002)** August 11-15.
[16] R. Agrawal, S. Gollapudi, A. Halverson, S. leong, "Diversifying search results", Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain, **(2009)** February 09-12.
[17] L. Yuan, L. Qin, X. Lin, L. Chang, W. Zhang, "Diversified top-k clique search", Vldb Journal — the International Journal on Very Large Data Bases, vol. 25, no. 2, **(2016)**, pp. 171-196.
[18] E. Demidova, P. Fankhauser, X. Zhou, W. Nejdl, "DivQ: diversification for keyword search over structured databases", Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, **(2010)** July 19-23.
[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of the fiffth Berkeley symposium on mathematical statistics and probability, Berkeley, California, **(1967)**.
[20] L. Kaufman, P. J. Rousseeuw, "Finding groups in data: an introduction to cluster analysis", Technometrics, **(1990)**.
[21] A. Y. Ng, M. I. Jordan, Y. Weiss, "On Spectral Clustering: Analysis and an algorithm", Advances in Neural Information Processing Systems, **(2002)**, pp. 849-856.
[22] Scalable Clustering of Time Series with U-Shapelets, https://sites.google.com/site/ushapelet/

