A FAST SHAPELET DISCOVERY ALGORITHM WITH SYMBOLIC FOURIER APPROXIMATION

Cun Ji, Yupeng Hu, Shijun Liu*, Chenglei Yang, Li Pan, Lei Wu*
School of Computer Science & Technology, Shandong University, Jinan, China
{jicun, huyupeng, lsj, chl_yang, panli, i_lily}@sdu.edu.cn

Abstract

Time series classification has been attracting great interest over the past decade. One of the most promising recent approaches is to find shapelets. A shapelet is one fragment of a time series that can be used to represent class characteristic of the time series. Classifier based on shapelets is interpretable, more accurate and faster. However, the time taken to find shapelet is enormous. For this, we propose a fast shapelet discovery algorithm with symbolic Fourier approximation. In our algorithm, every time series is carried out by discrete Fourier transform to reduce the noise firstly. After that, we use symbols to approximate represent for the transformed data with the help of multiple coefficient binning. Finally, we use fast shapelet algorithm to get the best shapelet. Results from experiments show that our algorithm has a higher accuracy and interpretable classification results.

Keywords: Time series data, Data mining, Classification, Shapelet, Symbolic Fourier approximation

1. INTRODUCTION

Cloud computing and Internet of Things (IoT) have become core technologies in industrial field in recent years. Usually, the sensors in IoT collect observation data with equal intervals. So data collected in IoT is time series.

A time series is a series of data points indexed (or listed or graphed) in time order. A time series data represents a collection of values obtained from sequential measurements over time, which characterized by its numerical and continuous nature (Esling, 2012; Fu, 2011).

Time series is always considered as a whole instead of individual numerical field. What's more, the high dimensionality, high feature correlation, and typically high levels of noise found in time series provide an interesting research problem (Keogh, 2003; Ye, 2009; Gabr, 2013). Effective time series classification (TSC) has long been an important research problem for both academic researchers and industry practitioners.

In TSC, an unlabeled time series is assigned to one of two or more predefined classes (Keogh, 2003). Now, TSC arises in many real world fields. Examples of real life problems that may be accomplished by classification of time series are electrocardiogram classification, fault detection and identification of physical systems, automotive preventive diagnosis, gesture recognition, alarm interpretation of telecommunication networks, data sensor analysis, speaker identification and/or authentication, aerospace health monitoring, etc.(see (Prieto, 2015)).

There is a plethora of classification algorithms such as classification trees, nearest neighbors, discriminant analysis, iterative classification, etc. can be applied to time series. However, recent empirical evidence has strongly suggested that classification based on time series shapelet outperforms many classification algorithms (He, 2012). Shapelets are discriminative subsequences which have the property that the minimum distance between a shapelet and the time series is a good predictor for time series classification (Wistuba, 2015). The algorithms based on shapelets are interpretable, more accurate and faster than state-of-the-art classifiers (Ye, 2009; Mueen, 2011; Ye, 2011)

After getting shapelets, we can classify industrial time series accurately and we can easily explain why such a classification result is given. However, the time taken to discovery shapelets is significant, even though shapelets are computed offline (Rakthanmanon, 2013).

For this, we propose a fast shapelet discovery algorithm with symbolic Fourier approximation. In our algorithm, every time series is carried out by discrete Fourier transform (DFT) to reduce the noise firstly. After that, we use symbols to approximate represent for the transformed data with the help of multiple coefficient binning (MCB). Finally, we use the fast shapelet discovery algorithm to get the best shapelet.

Our contributions of this paper can be summarized as follows:

1. We propose a fast shapelet discovery algorithm with symbolic Fourier approximation (SFA). In our algorithm, DFT is used to reduce the noise and SFA is used to speed up the time.
2. Comparison experiments among different shapelet discovery algorithms are conducted. The experiments results show that our algorithm has a higher accuracy and interpretable classification results.

The rest of the paper is organized as follows: some definitions are given and related work is discussed in Section 2. Our algorithm, a fast shapelet discovery algorithm with SFA (FS-SFA), is demonstrated in Section 3. Experimental results are presented in Section 4, and our conclusions are given in Section 5.

2. DEFINITION AND RELATED WORK

DOI: 10.29268/iciot.2016.0015

2.1 Definition

Definition 1: A *time series* (*T)* is an ordered list of real-valued variables; $T = t_1, t_2 ... t_m$. Typically, data points $t_1, t_2 ... t_m$ are arranged by temporal order, spaced at equal time intervals. And *m* is the length of time series *T*. The length of time series is also notations as $|T|$.

Definition 2: A *time series subsequence (S)* is a contiguous sequence of a time series. Subsequence *S* of length *l* of time series *T* starting at position *i* can be written as $S = T_i^l = t_i, t_{i+1} \ ... \ t_{i-l+1}$.

Definition 3: A *dataset* (*D)* is a set of time series; $D = \{T_1, T_2 \ ... \ T_{|D|}\}$. $|D|$ means the number of time series in dataset *D*. Note that the lengths for each time series may be not necessarily equal.

Definition 4: The *distance between time series (dist(T, R))* is a distance function that takes two time series $T = t_1, t_2 ... t_m$ and $R = r_1, r_2 ... r_m$ which are of the same length as inputs and returns a nonnegative value.

In this paper, we use the Euclidean distance which is calculated as (1). It is also applicable to subsequences of same length.

$$dist(T, R) = \sqrt{\sum_{i=1}^{m}(t_i - r_i)^2} \tag{1}$$

Definition 5: The *distance from the time series to the subsequence (subDist(T, S))* is a distance function that takes time series *T* and subsequence *S* as inputs and returns a nonnegative value, which is the minimum possible distance from T to S. It is calculated as (2).

$$subDist(T, S) = \min(dist(S, T_1^l), \ldots dist(S, T_{m-l+1}^l)) \tag{2}$$

In (2), *m* is the length of time series *T* and *l* is the length of subsequence *S*.

Definition 5: *Entropy (e)*: Suppose that dataset *D* contains time series from *c* different classes. And the probability of time series in class *i* is p_i. The entropy of *D* can be calculated as (3).

$$e(D) = -\sum_{i=1}^{c} p_i \log p_i \tag{3}$$

Definition 6: A *split (sp)* is a two-tuples <*s, d*> of a subsequence *s* and distance threshold *d* which can separate the dataset into two smaller datasets, D_L and D_R. The time series in *D* which distance to *s* is bigger than *d* is put into D_R, otherwise it is put into D_L. The number of time series in D_L and D_R are n_L and n_R, respectively.

Definition 7: The *information gain (gain)* of a split *sp* can be calculated as (4).

$$gain(sp) = e(D) - \frac{n_L}{n} e(D_L) - \frac{n_R}{n} e(D_R) \tag{4}$$

Definition 8: A *separation gap (gap)* is the distance between two different sides of the given split *sp*. It can be calculated as (5).

$$gap(sp) = \left| \frac{1}{n_R} \sum_{T \in D_R} subDist(T, s) - \frac{1}{n_L} \sum_{T \in D_L} subDist(T, s) \right| \tag{5}$$

Definition 9: A *shapelet* is a split that separates the dataset into two smaller datasets with the maximum information gain; ties are broken by maximizing the separation gap.

The definition of shapelet is explained briefly here; however, for a more complete definition of shapelet please refer to (Ye, 2009; He, 2012, Mueen, 2011).

2.2 Related Work of Shapelet Discovery Algorithm

Since introduced in 2009 (Ye, 2009), shapelet has aroused the concern of many researchers. Firstly, shapelet classier classifies new instance faster because it is more compact than many of the alternatives. Secondly, shapelets are directly interpretable. Thirdly, shapelets allow for the detection of shape-based similarity of subsequences. This type of similarity is often hard to detect with algorithms based on whole series.

Next, some related work of shapelet discovery algorithm will be discussed.

2.2.1 Brute force shapelet discovery algorithm

The brute force shapelet discovery algorithm is firstly introduced by Ye and Keogh in 2009 (Ye, 2009). The algorithm generates all the possible candidates. Then the algorithm tests all the candidates and returns the best one. The pseudo code of the brute shapelet discovery algorithm is shown in Algorithm 1

There are two parts in Algorithm 1. Form Line 1-7, the algorithm generates all the candidates. The lengths of candidates are between *minLen* and *maxLen*. The parameters *minLen* and *maxLen* are the minimum and maximum length we input. Then the algorithm tests every candidate through comparing the information gain and separation gap from Line 10 to Line 20. At last, the algorithm returns the final *shapelet*.

The obvious weakness of the brute force shapelet discovery algorithm is the relatively slow training time. The time complexity of the brute force algorithm is $O(n^2m^4)$, *n* is the number of time series in dataset and *m* is the length of time series.

Table I. Brute Shapelet Discovery Algorithm

```
Algorithm 1: BruteForceShapelet
Input:  D(the time series dataset), maxLen , minLen
Output: shapelet(the final shapelet)
For every time series T in D
    For l = minLen to maxLen
     For i= 1 to |T|- l +1
         candidates.add([illegible])
     End For
  End For
End For
bsf_gain =0
bsf_gap=0
For every candidate cand in candidates
  For every split sp of cand
     calculate gain as (4) and gap as (5)
     If (gain>bsf_gain)
                 or(gain=bsf_gainandgap>bsf_gap)
        bsf_gain =gain
        bsf_gap = gap
        shapelet = cand
     End If
    End For
End For
return shapelet
```

2.2.2 Some speed up methods

Some methods have been proposed to speed up the shapelet discovery algorithm.

Ye and Keogh developed two speedup methods: Subsequence Distance Early Abandon (SDEA) and Admissible Entropy Pruning (AEP) (Ye, 2011). In SDEA, once the distance is larger than the current smallest distance, the computation is abandoned. SDEA can help reduce the runtime by a factor of two. AEP calculates a cheap-to-compute upper bound of the information gain, and uses this to admissibly prune certain candidates. AEP reduces the runtime by more than two orders of magnitude.

The current state-of-the-art algorithm which guaranteed to find the same shapelet with the brute force algorithm is introduced in (Mueen, 2011). They accelerated the time required to find the same shapelet by reusing computations and pruning the search space. Their algorithm uses a matrix to cache the distance computations for future use, and then applies the triangle inequality to prune some candidates. This results in a time complexity of $O(n^2m^3)$ and a memory footprint of up to $O(nm^2)$. But note that the speedup method can handle time series without normalization only.

Improvements in (Ye, 2011) and (Mueen, 2011) guaranteed to find the same shapelet with the brute force algorithm. Some speed up methods which not guaranteed to find the same shapelet with the brute force algorithm are also put forward.

Chang et al. introduced an implementation on highly parallel Graphics Process Units (GPUs) (Chang, 2012). Through hardware-based optimization, they significantly reduce the running time of the shapelet discovery algorithm. But the cost of hardware optimization is expensive.

He et al. reduced the running time by elaborating the usage of infrequent shapelet candidates (He, 2012). They supposed that discriminative subsequences are usually infrequent compared to other subsequences. This assumption may not be tenable in some specific database.

Until now, the fastest Shapelet Discovery Algorithm is Fast Shapelet (FS) (Rakthanmanon, 2013). Exploiting projections on the SAX representation is also used to find shapelet in FS. The time complexity of ST is O(nm^2). An example of SAX is given in *Figure 1*.

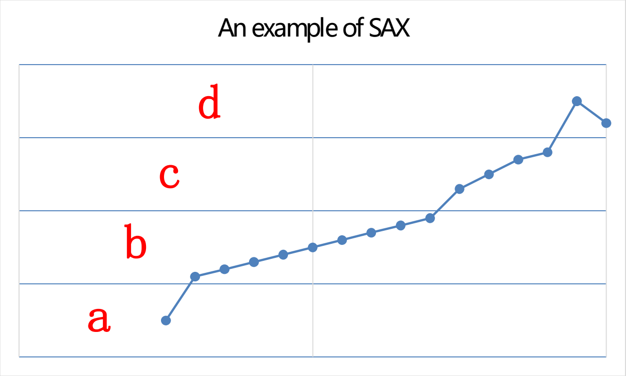

Figure 1. An example of SAX

From *Figure 1*, we can clearly find SAX did not consider the distribution of the data value. More than half of the data value is transformed to *b*.

For this, we apply SFA technology to FS. So the distribution of the data value can be considered.

3. Fast Shapelet Discovery Algorithm With SFA (FS-SFA)

3.1 Overview of FS-SFA

In this section, FS-SFA is given. As shown in *Figure 2*, we firstly used DFT to reduce the noise in our algorithm. After that, the transformed data is approximate represent with the help of MCB. Next, we use fast shapelet algorithm to get the best shapelet, so that we can get a classification decision tree. Finally, we use the decision tree to classification the testing time series.

We will descript some processes in detail in the following subsections

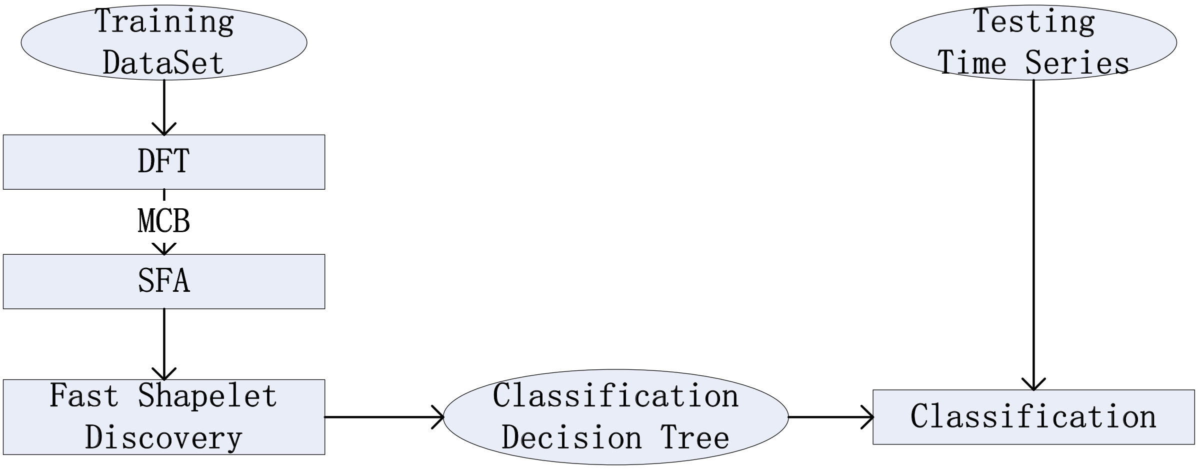

Figure 2. The diagram of FS-SFA

3.2 DFT

The DFT decomposes a time series T of length n into a sum of orthogonal basis functions using sinusoid waves (Sch äfer, 2015). Usually, we use a complex number [illegible] ([illegible]) to represent each waves, for i [illegible] 0, 1, …, [illegible] 1 . After DFT, the format of a time series is as (6).

[illegible]([illegible]) [illegible] [illegible], …, [illegible]
[illegible] ([illegible], …, [illegible]) (6)

We can transform a time series $T = t_1, t_2 \ldots\ t_n$ to DFT format as (7).

[illegible] $\sum$ [illegible] · [illegible] $2\sqrt{-}$ [illegible], [illegible] 0,1, …, [illegible] 1 (7)

3.3 SFA

The SFA (Sch äfer, 2012) is a symbolic representation of time series (Sch äfer, 2015). Before SFA, we need to get MCB at first. The MCB quantization intervals are computed from the samples. In MCB, the distribution of the data value is considered. The number of points in each interval is basically the same. An example of MCB is shown in *Figure 3*.

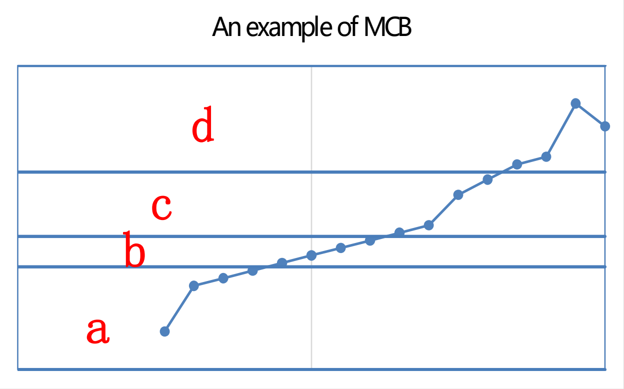

Figure 3. An example of MCB

With the help of MCB, we convert a Fourier transformed time series to SFA word. The alphabet can be determined as (8).

[illegible] ≤ [illegible] [illegible]
⇒ [illegible] alphabet[illegible] (8)

For example, the value is *Figure 3* is converted to SFA word *aaaabbbbccccdddd*.

3.4 Fast Shapelet Discovery with SFA

In 2013, Rakthanmanon and Keogh proposed FS (Rakthanmanon, 2013) based on SAX. There are four step in FS: 1) Generating SAX word, 2) Random Masking, 3) Counting Similar Objects and 4) Finding the Best Candidates.

Based on FS, our algorithm is improved by replacing SAX word with SFA word. The algorithm of our algorithm is shown in Table II. We implemented our algorithm reused most code of FS. The difference between FS and FS-SFA is that FS-SFA generating SFA word while FS generating SAX word.

Table II. The Algorithm of FS_SFA

```
Algorithm 2: FS-SFA
Input: D(the time series dataset), m(the length of time series)
Output: shapelet(the final shapelet)
shapelet= new Shapelet();
For sub_len =1 to m
    D'=DFT(D)
    word=SFA(D')
    RandomMasking(word)
    wordCand = FindTopKSAX(wordCand)
End For
shapelet = findingTheBestCandidates(wordCand)
return shapelet
```

4. Experiments

We implemented our method based on the UEA TSC code which is freely accessible from an online repository (Bostrom, 2015). Our code and detailed results can be downloaded from the support page (Liu, 2016).

We performed experiments on datasets from the UCR Time Series Classification Archive (Chen, 2015) and those in the ärff" file format which can be downloaded from the UEA TSC website (Bagnall, 2016).

The UCR Time Series Classification Archive (Chen, 2015) mainly contains five types of time series dataset (Bagnall, 2015). For every kind of problems we used datasets as Table III showed.

Table III. Datasets

Classification Problems Type	Dataset
Sensor Reading	Car, ChlorineConcentration, Coffee, Computers, Earthquakes, ItalyPowerDemand, Trace, Wafer, SonyAIBORobotSurface1, onyAIBORobotSurface2, MoteStrain, OliveOil, Plane
Human Sensor Reading	TwoLeadECG, ECG5000, ECGFiveDays
Motion	CricketX, CricketY, CricketZ, GunPoint, ToeSegmentation1, ToeSegmentation2, Worm, WormsTwoClass
Simulated	Mallat, CBF, SyntheticControl, TwoPatterns
Image Outline	DistalPhalanxOutlineAgeGroup, Adiac, ArrowHead DistalPhalanxOutlineCorrect, FaceFour, Yoga, DistalPhalanxTW, MiddlePhalanxOutlineAgeGroup, MiddlePhalanxOutlineCorrec, MiddlePhalanxTW, OSULeaf, MedicalImages, Symbols, BeetleFly, DiatomSizeReduction

4.1 Accuracy Comparison on Sensor Reading Classification Problem

A major kind of time series data is sensor observation data. First of all, we do a set of comparative experiments between FS and FS-SFA on sensor reading classification problems. The classification results are shown in *Figure 4*.

In sensor reading classification problems, we use 13 datasets. *Figure 4* compares the accuracies between FS and FS-SFA. The points top and to the left of the diagonal line in *Figure 4* indicate instances in which FS-SFA achieved a higher accuracy than SF.

Among these 13 classification problems, FS-SFA works better than FS on 9 datasets. We think the reason of FS-SFA works better is that the time series of sensor reading data has some noise data which reduced by DFT in FS-SFA.

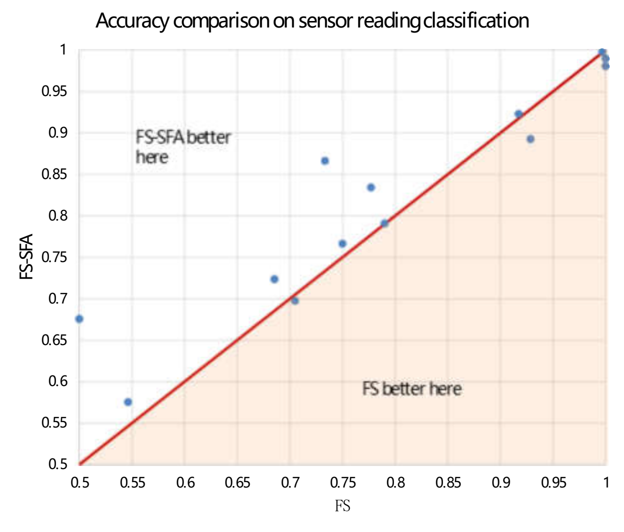

Figure 4. Accuracy comparison on sensor reading classification problems

4.2 Accuracy Comparison on Human Sensor Reading Classification Problem

Human sensor reading classification problems are special sensor reading classification problems. This type of data is usually medical diagnostic data. The classification results between FS and FS-SFA on human sensor reading classification problems are shown in Table IV.

From Table IV, we can get that the accuracies of classification are very similar. The reason may be that noise reduction processing has been carried out for the time series obtained from medical sensors.

Table IV. Accuracy Comparison on Human Sensor Reading Classification Problems

Dataset	FS	FS-SFA
TwoLeadECG	0.924495	0.924495
ECG5000	0.922667	0.921778
ECGFiveDays	0.997677	0.998839

4.3 Accuracy Comparison on Motion Classification Problem

Another kind of time series data is motion time series. The classification results between FS and FS-SFA on motion classification problems are shown in *Figure 5*.

The points top and to the left of the diagonal line in *Figure 5* indicate instances in which FS-SFA achieved a higher accuracy than SF.

From *Figure 5*, we can see that FS and FS-SFA work better on half of the datasets. But on some datasets, FS-SFA works much better than FS. Overall, the classification resulted from FS-SFA is better.

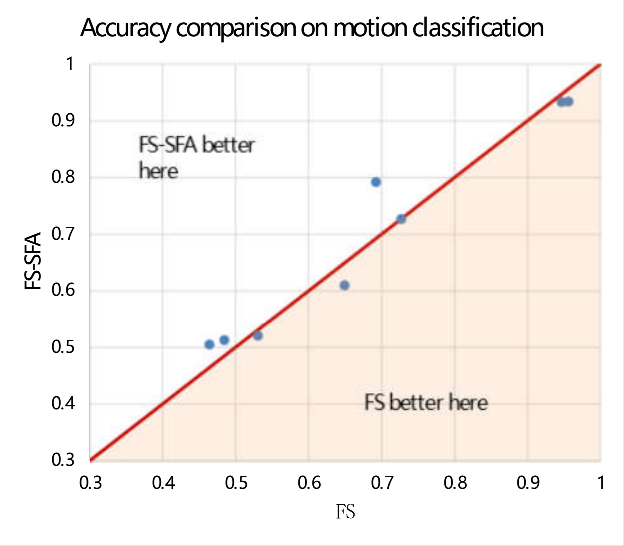

Figure 5. Accuracy comparison on motion classification problems

4.4 Accuracy Comparison on Simulated Classification Problem

Some time series are simulated. Another kind of time series data is motion time series. The classification results between FS and FS-SFA on simulated classification problems are shown in *Figure 6*.

The points top and to the left of the diagonal line in *Figure 6* indicate instances in which FS-SFA achieved a higher accuracy than FS. As shown in *Figure 6*, it's hard to say which method is better.

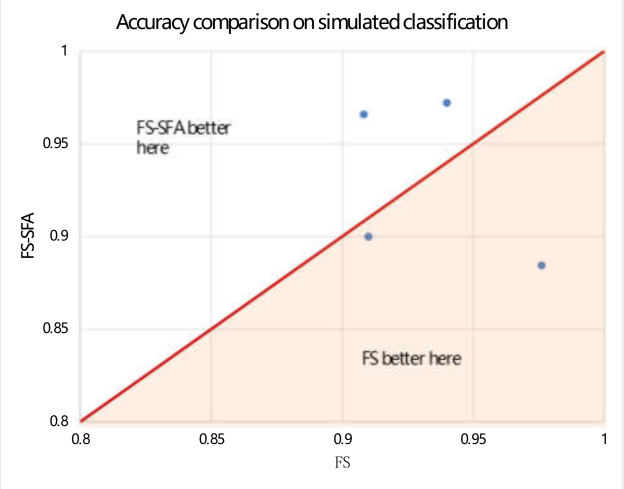

Figure 6. Accuracy comparison on simulated classification problems

4.5 Accuracy Comparison on Image Outline Classification Problem

Last but not the least, some time series is converted from image. The classification results between FS and FS-SFA on motion classification problems are shown in *Figure 7*。

The points top and to the left of the diagonal line in *Figure 7* indicate instances in which FS-SFA achieved a higher accuracy than FS.

In image outline classification problems, we use 18 datasets. As *Figure 7* shown, FS-SFA works better on 8 datasets while FS works better on 9 datasets. It's also hard to say which method is better.

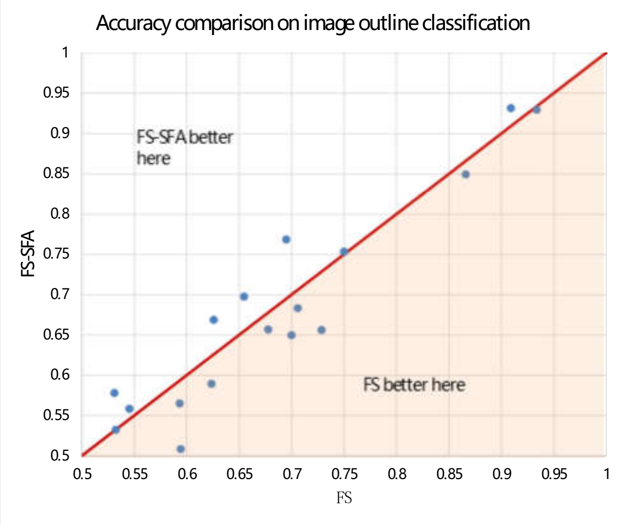

Figure 7. Accuracy comparison on image outline classification problems

4.6 Summary of Experimental Results

There are mainly five types of time series dataset in the UCR Time Series Classification Archive (Chen, 2015). For sensor reading classification problem and motion classification problem, FS-SFA works better than FS. On the other three problems, it's hard to say which method is better. Overall, the classification effect of FS-SFA is better.

5. Conclusion

For the time taken to discovery shapelet is significant, we propose a fast shapelet discovery algorithm with symbolic Fourier approximation in this paper. In our algorithm, we firstly used DFT to reduce the noise. After that, the transformed data is approximate represent with the help of MCB. Next, we use fast shapelet algorithm to get the best shapelet, so that we can get a classification decision tree. Finally, we use the decision tree to classification the testing time series. Comparison experiments among different shapelets discovery algorithms are conducted. The experiments result show that our algorithm has a higher accuracy and interpretable classification results. And our algorithm is more suitable to classify sensor reading problems and motion classification problem.

In our industrial big data platform (Ji, 2015; Ji, 2016), there are much time series of sensor reading data. In the future, we plan to apply the fast shapelet discovery algorithm with symbolic Fourier approximation to our industrial big data platform.

6. ACKNOWLEDGMENT

The authors would like to acknowledge the support provided by the National Natural Science Foundation of China (61402263, 91546203), the National High Technology Research and Development Program of China (2014AA01A302), the special funds of Taishan scholar construction project, the Independent Innovation Projects of Shandong Province (2014ZZCX08201, 2014ZZCX08102, 2014ZZCX03409), and the Science & Technology Development Projects of Shandong Province (2014GGX101028, 2014GGH201007).

7. REFERENCES

Bagnall, A., Lines, J. (2016), The UEA tsc website, retrieved September 12, 2016, from http://timeseriesclassification.com.

Bagnall, A., Lines, J., Hills, J., Bostrom, A. (2015), Time-series classification with COTE: the collective of transformation-based ensembles, *IEEE Transactions on Knowledge and Data Engineering*, *27*(9), pp. 2522-2535.

Bostrom, A., Bagnall, A., Lines, J. (2015), The UEA tsc codebase, retrieved September 12, 2016, from https://bitbucket.org/aaron_bostrom/time-series-classification.

Chang, K. W., Deka, B., Hwu, W. M. W., Roth, D. (2012), Efficient pattern-based time series classification on gpu, *Proceedings of 2012 IEEE 12th International Conference on Data Mining*, IEEE, Brussels, Belgium, pp. 131-140.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G. (2015), The ucr time series classification archive, retrieved September 12, 2016, from http://www.cs.ucr.edu/~eamonn/time_series_data/.

Esling, P., Agon, C. (2012), Time-series data mining, *Acm Computing Surveys, 45*(1), pp. 7-7.

Fu, T. C. (2011), A review on time series data mining, *Engineering Applications of Artificial Intelligence, 24*(1), pp. 164-181.

Gabr, M. M., Fatehy, L. M. (2013), Time series classification, *Signal Processing, 2*(2), pp. 123-133.

He, Q., Dong, Z., Zhuang, F., Shang, T., Shi, Z. (2012), Fast Time Series Classification Based on Infrequent Shapelets, *Proceedings of 11 International Conference on Machine Learning and Applications*, Elsevier, Boca Raton, US, pp. 215-219.

Ji, C., Liu, S., Yang, C., Wu, L., Pan, L. (2015), IBDP: An Industrial Big Data Ingestion and Analysis Platform and Case Studies, *Proceedings of 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things*, IEEE, Beijing, China, pp. 223-228.

Ji, C., Shao, Q., Sun, J., Liu, S., Pan, L., Wu, L., Yang, C. (2016), Device Data Ingestion for Industrial Big Data Platforms with a Case Study, *Sensors*,*16*(3), pp. 279-279.

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

Liu, S., Ji, C. (2016), The support page, retrieved September 12, 2016, from http://211.87.227.112:8080/PaperSupport/FS_SFA.html.

Mueen, A., Keogh, E., Young, N. (2011), Logical-shapelets: an expressive primitive for time series classification, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, San Diego, US, pp. 1154-1162.

Prieto, O. J., Alonso-Gonz ález, C. J., Rodr íguez, J. J. (2015), Stacking for multivariate time series classification, *Pattern Analysis and Applications, 18*(2), pp. 297-312.

Rakthanmanon, T., Keogh, E. (2013), Fast shapelets: A scalable algorithm for discovering time series shapelets, *Proceedings of thirteenth SIAM conference on data mining*. SIAM, Austin, US, pp. 668-676.

Sch äfer, P. (2015), The boss is concerned with time series classification in the presence of noise, *Data Mining and Knowledge Discovery, 29*(6), pp. 1505-1530.

Sch äfer, P., H ögqvist, M. (2012), SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets, *International Conference on Extending Database Technology*, ACM, Berlin, Germany, pp.516-527.

Wistuba, M., Grabocka, J., Schmidtthieme, L. (2015). Ultra-fast shapelets for time series classification, *Computer Science*.

Ye, L., Keogh, E. (2009), Time series shapelets: a new primitive for data mining, *Proceedings of 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Paris, France, pp. 947-956.

Ye, L., Keogh, E. (2011), Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, *Data Mining and Knowledge Discovery, 22*(1), pp.149-182.

Authors

Cun Ji obtained his BS degree from Shandong University, China. He is a PhD student of Shandong University. His current research interests include services computing and cloud computing.

Yupeng Hu obtained his BS degree and MS degree from Shandong University, China. He is a PhD student of Shandong University. His current research interests include data mining and machine learning.

Shijun Liu obtained BS degree from Ocean University of China, MS and PhD degrees from Shandong University. He is a Professor of Shandong University. His research interests include services computing and cloud computing.

Chenglei Yang obtained his BS, MS and PhD degrees from Shandong University, China. He is a Professor of Shandong University. His current research interests include human computer interaction and virtual reality.

Pan Li obtained her BS, MS and PhD degrees from Shandong University, China. She is a Lecturer of Shandong University. Her current research interests include cloud computing, and market-oriented resource allocation.

Lei Wu obtained BS degree from Qingdao University of Science & Technology, and PhD degree from Shandong University. She is an adjunct professor of Shandong University. Her current research interests include service computing and cloud computing.