VISUALET: Visualizing Shapelets for Time Series Classification

Guozhong Li
Department of Computer Science,
Hong Kong Baptist University
Hong Kong
csgzli@comp.hkbu.edu.hk

Byron Choi
Department of Computer Science,
Hong Kong Baptist University
Hong Kong
choi@hkbu.edu.hk

Sourav S Bhowmick
School of Computer Science and
Engineering, Nanyang Technological
University, Singapore
assourav@ntu.edu.sg

Grace Lai-Hung Wong
Department of Medicine &
Therapeutics, The Chinese University
of Hong Kong
wonglaihung@cuhk.edu.hk

Kwok-Pan Chun
Department of Geography, Hong
Kong Baptist University
Hong Kong
kpchun@hkbu.edu.hk

Shiwen Li
Department of Computer Science,
Hong Kong Baptist University
Hong Kong
19214294@life.hkbu.edu.hk

ABSTRACT

Time series classification (TSC) has attracted considerable attention from both academia and industry. TSC methods that are based on *shapelets* (intuitively, small highly-discriminative subsequences) have been found effective and are particularly known for their *interpretability*, as shapelets themselves are subsequences. A recent work has significantly improved the efficiency of shapelet discovery. For instance, the shapelets of more than 65% of the datasets in the *UCR Archive* (containing data from different application domains) can be computed within an hour, whereas those of 12 datasets can be computed within a minute. Such efficiency has made it possible for demo attendees to interact with shapelet discovery and explore high-quality shapelets. In this demo, we present VISUALET – a tool for visualizing shapelets, and exploring effective and interpretable ones.

CCS CONCEPTS

• **Human-centered computing → Visualization toolkits**;

KEYWORDS

Time-series classification; Shapelet discovery; efficiency; accuracy

ACM Reference Format:
Guozhong Li, Byron Choi, Sourav S Bhowmick, Grace Lai-Hung Wong, Kwok-Pan Chun, and Shiwen Li. 2020. VISUALET: Visualizing Shapelets for Time Series Classification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3417414

1 INTRODUCTION

Time series classification (TSC) has received considerable attention from both academia and industry. Among many others, a classical approach to solve the TSC problem is the whole series-based approach. Such an approach combines classifiers, such as 1-Nearest Neighbor (1NN), and similarity metrics, such as Euclidean Distance or Dynamic Time Warping distance. More details can be found in a comprehensive survey [2].

A recent trend of TSC is to compute small patterns (or features) that can represent and yet distinguish classes of time series. In particular, the seminal work [9] presented *shapelets* for TSC. Intuitively, shapelets can be understood as small highly-discriminative subsequences that maximally represent classes of time series and distinguish time series from one class to another. Shapelet-based methods (*e.g.*, [6, 8]) have repeatedly demonstrated superior accuracies. Some research has combined shapelets with learning approaches (*e.g.*, [3, 4]) to learn a small number of shapelets. Since shapelets themselves are time subsequences, shapelets have been helpful to interpret the classes of time series and interesting cases from different applications have been reported (*e.g.*, Figure 6 & 7).

A fundamental challenge of shapelet-based methods is to efficiently discover high-quality shapelets for building a classification model. Not until recently, *shapelets can be efficiently discovered*. The main technique is to efficiently prune non-discriminative or similar subsequences before potentially costly shapelet discovery. According to [6], the shapelets of more than 65% of the datasets (from different application domains) in the UCR Archive can be computed within an hour, whereas those of 12 datasets of the archive can be returned within a minute. Due to the efficiency, it becomes possible to investigate shapelet discovery using various parameter settings, such as shapelet discovery algorithms, the length of shapelets and the shapelet number to be discovered. The shapelets discovered can then be transformed using [1] and traditional classifiers can be built from transformed shapelets.

In this demo, we present a software called VISUALET that visualizes shapelets and time series. We use the well-known benchmarked datasets, namely the UCR Archive. VISUALET enables demo attendees to study the shapelets discovered by various recent efficient shapelet-based methods, including [3, 4, 6, 8]. For the first time, a demo running on a commodity machine is able to present many shapelets of the UCR Archive.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3417414

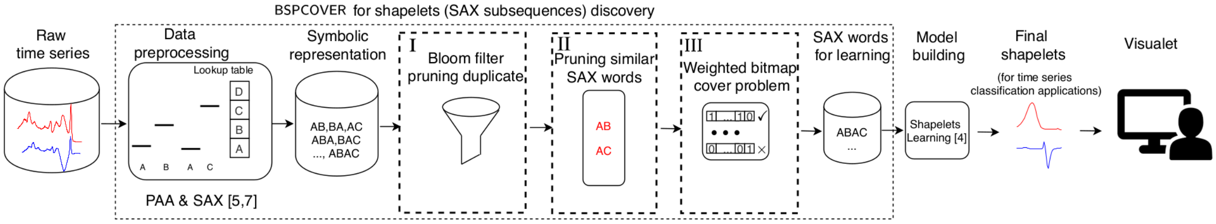


Figure 1: The overview of shapelet discovery in BSPCOVER

2 OVERIVEW OF BSPCOVER

In this section, we briefly introduce the *efficient shapelet discovery method to discover high-quality shapelets* implemented in the demo, namely BSPCOVER [6]. An overview of BSPCOVER is presented in Figure 1.

The first step of BSPCOVER is data preprocessing that simply applies symbolic representation methods, specifically PAA [5] and SAX [7], to reduce the dimensionality of raw time series. In particular, they transform raw data into *symbolic representations* (SAX subsequences, a.k.a SAX words). A sliding window method is adopted to generate numerous SAX words.

The core of BSPCOVER then consists of three main steps. **I.** BSPCOVER derives a bloom filter for each class of time series to efficiently prune the same SAX words that also exist in other classes. Such SAX words have less discriminative power and only deteriorate overall efficiency and accuracy of any classifiers. Since bloom filters do not produce false negatives, the SAX words of a class that do not exist in the bloom filters of other classes are definitely not in those other classes. Such SAX words can be candidates for model building. Figure 2 (①-⑥) shows the main steps of using bloom filters $\mathcal{BF}$ to prune the SAX words set Ω. To quantify the quality of candidate SAX words $\Omega_C^{\mathcal{BF}}$ of a class C, we propose to generate a bitmap structure for each SAX word $\hat{e}$. Figure 3 shows an example of (partial) bitmaps for $\Omega_1^{\mathcal{BF}}$.

II. The second step is a non-metric distance-based pruning of similar SAX words, which eliminates the effect of similar SAX words that exist in all classes. Recall that the distance between the raw time series violates triangle inequality, the symbolic representation of time series inherits this property. In this step, similar SAX words exist in different classes, they are all pruned. In addition, for words in the same class, only one representative word is kept for further processing.

III. For each set of similar SAX words of a class, we consider their term frequency, as a weight. Then, we propose weighted bitmap structures of SAX words to quantify the quality of shapelet candidates, which is utilized to discover a set of SAX words that have the maximal weight and represent all the instances in each class. The shapelet selection problem is then formalized as a weighted bitmap cover problem, which can be reduced from the classical weighted set cover problem. We propose a heuristic algorithm to solve it.

Finally, we compute the discriminative shapelets from the candidates as follows. We transform the SAX words back to their raw representations of time series. Existing learning methods can be applied for modifying the selected shapelets to improve the accuracy further. In the paper, we revise LTS [4] to build one-vs-rest classifiers.

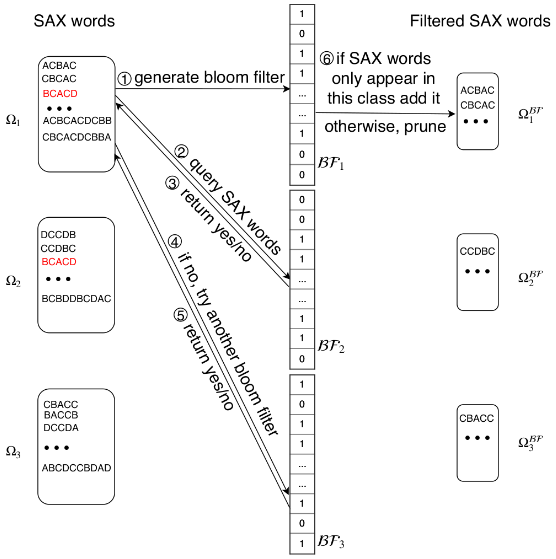


Figure 2: Pruning redundant and non-discriminative SAX words (of 3 classes) using bloom filters

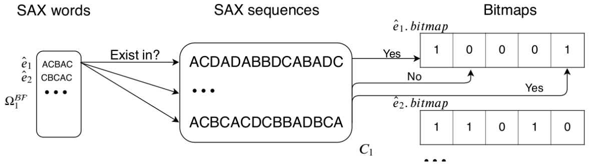


Figure 3: An example of (partial) constructing bitmaps of SAX words in Class 1 (C_1)

Figure 4 summarizes the *whole* process of finding SAX words from raw time series. All the time series instances are transformed into SAX words, as shown in ①. Then, the SAX words are utilized to build the bloom filter $\mathcal{BF}_C$ for each class C, which prunes identical SAX words in other classes (blue dotted rectangle). Next, the bitmap structures for the remaining SAX words are built to quantify their qualities (red dotted rectangle). ② The procedure prunes similar SAX words in all classes. The red SAX words **DCCCC** and **DCCCD** of different classes are similar, whereas the blue SAX words of the

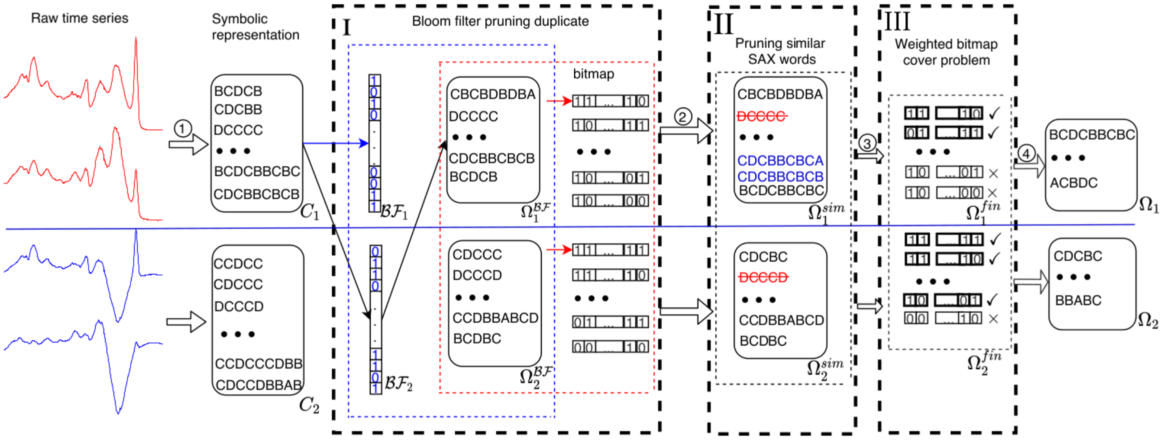


Figure 4: An example of finding SAX words for model building

same class are similar. The red ones are pruned and the blue ones are passed to compute their weights. ③ BSPCOVER utilizes the weight bitmap structures (the last rectangle) to select the minimum subset of bitmaps to cover all the instances.

3 DEMONSTRATION OVERVIEW

The GUI of VISUALET is presented in Figure 5[1]. The software VISUALET was implemented in JAVA[2]. In the left upper part, the `Load Dataset` button is for loading a time series dataset and their class labels. The `BSPCOVER` button is for running our method on the loaded dataset. In the left lower part, the `Load Shapelets` button is for loading shapelets. After loading time series and shapelets, the time series chart panels can be used to visualize them, which can help demo attendees to have a quick overview of a dataset. Some basic information, such as the distance between time series and shapelets, and the class label of time series, are shown in the GUI. On top of the time series charts, the GUI presents some settings of VISUALET, for instance, `Iteration` number, and the `alphabet size` of SAX words, `pCover` number, the imputation methods for handling missing values (`Impute` and the similarity metrics `ED`). On the bottom of Visualet, the `BSPCOVER Console` will output log information.

Scenario 1: Interpretable shapelet exploration. A demo attendee is interested to know what distinguishes the energy consumption behaviors of Italian consumers in summer (Class 1) from those in winter (Class 2). She loads the **ItalyPowerDemand** dataset from the UCR Archive, by clicking `Load Dataset`. By default, VISUALET uses the efficient shapelet discovery method to discover shapelets. VISUALET discovers the shapelets in 2s, whereas the current state-of-the-art [3] took 60 minutes. She loads the shaplets by clicking `Load Shapelets`. She loads some time series of winter energy consumptions to the middle panel. She then loads a specific time series of summer. Finally, she clicks on the shapelet IDs one by one to see why the shapelet similar to time series of one class but not the other. She finds one of the shapelets is particularly helpful in explaining the difference in Figure 6. The power demand in summer is lower than that in winter from 4am to 12pm, which turns out to be the heating in the morning during the winter time.

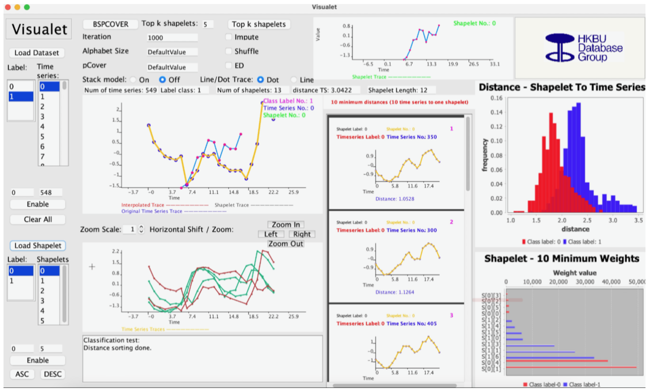
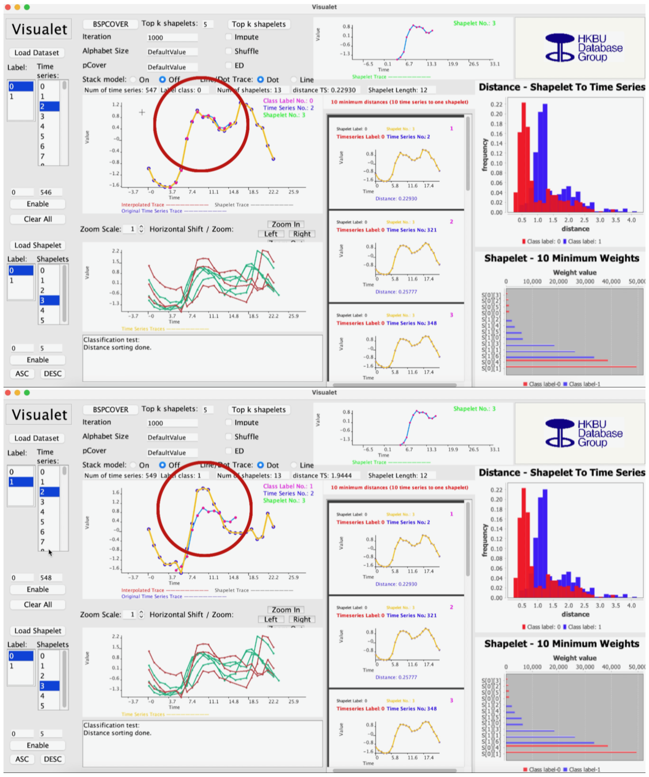

Visualet
BSPCOVER
Top k shapelets: 5
Iteration 1000
Alphabet Size DefaultValue
pCover DefaultValue
☐ Impute
☐ Shuffle
☐ ED
Stack model: ○ On ● Off
Line/Dot Trace: ● Dot ○ Line
Num of time series: 549
Label class: 1
Num of shapelets: 13
distance TS: 3.0422
Shapelet Length: 12
Class Label No.: 1
Time Series No.: 0
Shapelet No.: 0
10 minimum distances (10 time series to one shapelet)
Timeseries Label: 0 Time Series No: 350
Distance: 1.0528
Timeseries Label: 0 Time Series No: 300
Distance: 1.1264
Timeseries Label: 0 Time Series No: 405
Classification test:
Distance sorting done.

Figure 5: The GUI of VISUALET. Top: Settings. Left: Class labels, time series and shapelet IDs. Right (from top to bottom): A specific shapelet, a time series of interest; and some time series of a class.

Scenario 2: Image classification explanation. It is known that TSC can be applied to solve the image classification problem, where images are converted into time series. The **ArrowHead** dataset of the UCR archives is an example. Previously, the discovery of shapelets of **ArrowHead** takes roughly 3.4 hours. VISUALET returns the shapelets within 1 minute. Hence, VISUALET makes it affordable to discover shapelets using different settings. The demo attendees may causally select some shapelets using VISUALET. Such shapelets, when converted back to images, highlight to the fragments of the arrow heads that distinguish their types.

Scenario 3: Data imputation evaluation. Time series of patients' data (*e.g.*, lab test results) from hospitals often contain missing values. The time series can be labeled with a disease. Data imputation methods are often adopted to cleanse the data. The cleansed time series data can be loaded to VISUALET. Their shapelets are discovered. Shapelets that mainly correspond to the imputed subsequences are not acceptable, as a patient cannot be labeled with a disease largely

[1] An online video of VISUALET can be found at https://youtu.be/cAWylZXDXik.
[2] The demo tool is publicly available at https://www.comp.hkbu.edu.hk/~csgzli/tsc/demo.html.



Top k shapelets: 5
Iteration 1000
Alphabet Size DefaultValue
pCover DefaultValue
☐ Impute
☐ Shuffle
☐ ED
Stack model: ○ On ● Off Line/Dot Trace: ● Dot ○ Line
Num of time series: 547 Label class: 0 Num of shapelets: 13 distance TS: 0.22930 Shapelet Length: 12
Zoom Scale: 1 Horizontal Shift / Zoom:
Classification test:
Distance sorting done.

Num of time series: 549 Label class: 1 Num of shapelets: 13 distance TS: 1.9444 Shapelet Length: 12

Figure 6: A shapelet of ItalyPowerDemand highlighting the morning heating demand difference of summer and winter months

due to imputed data. When such shapelets dominate the discovered ones, the data imputation methods should be revised.

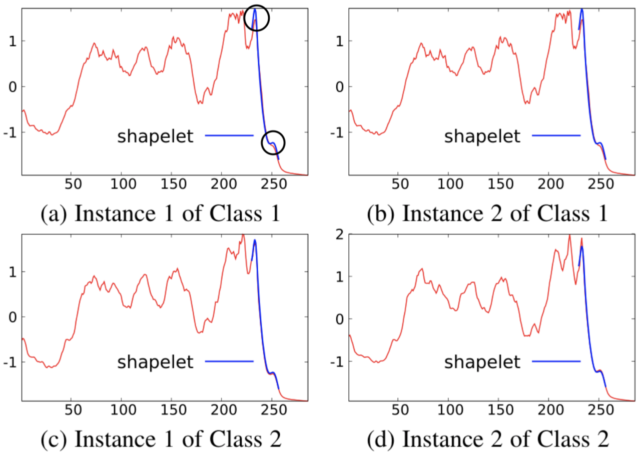
4 CASE STUDY: FOOD CLASSIFICATION

To illustrate the possible interpretability that shapelets offer, we present a use case of food chemistry using the **Coffee** dataset in this demo outline. Food spectrographs have been used in chemometrics to classify food types, a task that has obvious applications in food safety and quality assurance.

Figure 7 shows four time series of **Coffee**. The x-axis of the figures is the wave numbers of the coffee by Fourier transform infrared spectroscopy, whereas the y-axis is the normalized number of different wave numbers. The time series instances (red line) in Figure 7 (a) and (b) belong to Robusta coffee (Class 1), and those (red line) in Figure 7 (c) and (d) belong to Arabica coffee (Class 2). The short blue subsequence shows the shapelet selected by the demonstration software for Class 2 (Arabica coffee).

The shapelet highlights the difference between the two classes. The major difference between two types of coffee beans is precisely the specific wave numbers at 230 and 260 (black circle). Arabica coffee beans have high magnitudes at those frequencies. The classification accuracy of using such shapelets on **Coffee** is 100%.

In the demo, some additional case studies [6], including shapelets of electricity consumption, electrocardiographic applications and image classification, can be demonstrated interactively.



(a) Instance 1 of Class 1

(b) Instance 2 of Class 1

(c) Instance 1 of Class 2

(d) Instance 2 of Class 2

Figure 7: A shapelet of Coffee that highlights the two wave numbers of Robusta (Class 1) and Arabica (Class 2) coffee beans

ACKNOWLEDGMENTS

This work has been supported by the Hong Kong Research Grant Council (HKRGC) 12201119, 12232716, 12201518, 12200817, and 12201018, and National Science Foundation of China (NSFC) 61602395.

REFERENCES

[1] Anthony Bagnall, Luke Davis, Jon Hills, and Jason Lines. 2012. Transformation Based Ensembles for Time Series Classification. (2012), 307–318. https://doi.org/10.1137/1.9781611972825.27

[2] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 31, 3 (2017), 606–660.

[3] Zicheng Fang, Peng Wang, and Wei Wang. 2018. Efficient Learning Interpretable Shapelets for Accurate Time Series Classification. In *ICDE*. 497–508.

[4] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. 2014. Learning time-series shapelets. In *SIGKDD*. 392–401.

[5] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3, 3 (2001), 263–286.

[6] Guozheng Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok Pan Chun, and Grace Wong. 2020. Efficient Shapelet Discovery for Time Series Classification. *TKDE* (2020). https://doi.org/10.1109/TKDE.2020.2995870

[7] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *SIGMOD workshop*. 2–11.

[8] Thanawin Rakthanmanon and Eamonn Keogh. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SIAM*. 668–676.

[9] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *SIGKDD*. 947–956.