Distribution-wise Symbolic Aggregate approXimation (dwSAX)

Matej Kloska and Viera Rozinajova

Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava,
Ilkovičova 2, 842 16 Bratislava, Slovak Republic
{matej.kloska,viera.rozinajova}@stuba.sk
https://www.fiit.stuba.sk

Abstract. The Symbolic Aggregate approXimation algorithm (SAX) is one of the most popular symbolic mapping techniques for time series. It is extensively utilized in sequence classification, pattern mining, anomaly detection and many other data mining tasks. SAX as a powerful symbolic mapping technique is widely used due to its data adaptability. However this approach heavily relies on assumption that processed time series have Gaussian distribution. When time series distribution is non-Gaussian or skews over time, this method does not provide sufficient symbolic representation. This paper proposes a new method of symbolic time series representation named distribution-wise SAX (dwSAX) which can deal with Gaussian as well non-Gaussian data distribution in contrast with the original SAX, handling only the first case. Our method employs more general approach for symbol breakpoints selection and thus it contributes to more efficient utilization of provided alphabet symbols. The method was evaluated on different data mining tasks with promising improvements over SAX.

Keywords: Time series · Kernel density estimator · SAX.

1 Introduction

Whenever we work with any kind of event observations taken according to the order of time, we usually talk about time-order sequences also known as time series.

With the raise of big data and streaming technologies we see various fields such as healthcare, finance, security and industry where intelligent analysis and data mining tasks take place. Aforementioned tasks in context of time series are demanding and usually need different approaches due to data instances temporal ordering characteristic. At the same time, time series usually capture feature rich, highly dimensional data which make processing tasks even harder. In our work, the high dimensionality is related to high number of data points in time series. Dimensionality reduction and descriptive forms of time series representation are recognized as a possible solution for highly performing data mining tasks [6]. A challenging area in the field of effective time series processing is their compact data presentation without sacrificing any significant information [17]. Symbolic representation of time series appears to be the solution to this problem.

The Symbolic Aggregate approXimation algorithm (SAX) [8] is one of the most popular symbolic mapping techniques for time series. SAX as a powerful symbolic mapping technique is widely used due to its data adaptability. It is extensively utilized in sequence classification [13], pattern mining [3], anomaly detection [7] and many other data mining tasks [9, 14, 15]. However, this approach heavily relies on assumption that processed time series have Gaussian distribution [8]. When time series distribution is non-Gaussian or skews over time, this method does not provide sufficient symbolic representation. This paper proposes a new method named distribution-wise SAX (dwSAX) which can deal also with non-Gaussian data distribution in contrast with the original SAX. Our method employs more general approach for symbol breakpoints selection and thus improves tightness of lower bounding without sacrificing other SAX benefits.

This paper is organized as follows. Section 2 describes original SAX method and possibilities to data distribution estimation. Section 3 introduces dwSAX - our improvement of SAX method. Section 4 contains an experimental evaluation of the proposed method on time series clustering and anomaly detection tasks compared to the original SAX method. Finally, Section 5 offers some conclusions and suggestions for future work.

2 Related work

One of efficient data stream processing problems is their high dimensionality, too high number of data points. Possible solution to this issue is efficient symbolic representation that represents highly dimensional data stream through less dimensional symbolic data stream. In past decades, many different time series representations have been introduced. Lin et al. [8] divided methods into data adaptive (eg. Piecewise Linear Approximation, Singular Value Decomposition, SAX) and non data adaptive (eg. Wavelets, Random Mappings, Discrete Fourier Transformation). Recent research [11, 15, 18, 2, 10] shows activities in both method families. In the following sections we discuss fundamentals of original SAX, and techniques for data distribution estimation.

2.1 Symbolic representation - SAX

SAX is one of the best known algorithms for symbolic time series representation. This method makes it possible to represent any time series of length n using a string of any length w ($w \ll n$) with symbols from predefined alphabet. Looking at the mentioned method, it consists of:

1. *dimensionality reduction*: applying Piecewise Aggregate Approximation (PAA) [6], it significantly reduces dimensionality and preprocesses time series for further step,

2. *discretization*: mapping PAA segments into specific symbols from alphabet based on precomputed mapping symbols table.

Concept of this method is illustrated in Figure. 1. Throughout this paper we use common notation used also in original SAX paper [8].

Table 1: A summarization of common notation used in this paper and the original SAX paper. [8]

C	A time series $C = c_1, ..., c_n$ where $c_i \in R$
$\bar{C}$	A Piecewise Aggregate Approximation of a time series $\bar{C} = \bar{c}_1, ..., \bar{c}_w$
$\hat{C}$	A symbol representation of a time series $\hat{C} = \hat{c}_1, ..., \hat{c}_w$
w	The number of PAA segments representing time series C
a	Alphabet size (e.g., for the alphabet = {a,b,c}, $a = 3$)

Dimensionality reduction Intuition based on aforementioned description is to reduce time series from n dimensions into w dimensions. This goal is simply achieved by division of the time series into w equal sized pieces. For each piece, mean value is calculated and this value represents underlying vector of w original values. Total vector of all pieces becomes new reduced representation of the original time series.

More formally, a time series C of length n can be reduced into a w-dimensional time series by a vector $\bar{C} = \bar{c}_1, ..., \bar{c}_w$ where i^{th} element of $\bar{C}$ is calculated as follows [8]:

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{1}$$

Discretization Discretization step replaces PAA segments obtained in the length reduction step by alphabet symbols. Assuming data are normalised before reduction (with zero mean) and have highly Gaussian distribution, the replacement is performed as follows. We first precompute a table of breakpoints. Lin et al. [8] defined breakpoints as sorted list of numbers $B = \beta_1, ..., \beta_{a-1}$ such that the area under a $N(0, 1)$ Gaussian curve from β_i to $\beta_{i+1} = 1/a$ (β_0 and β_a are defined as $-\infty$ and ∞, respectively).

The Gaussian curve enables efficient breakpoints table precomputation, thus the discretization step is trivial in comparison to the single vector lookup operation.

Finally, formal definition of SAX as proposed by Lin et al. [8]: A subsequence C of length n can be represented as a word $\hat{C} = \hat{c}_i, ..., \hat{c}_w$ as follows. Let α_i denote

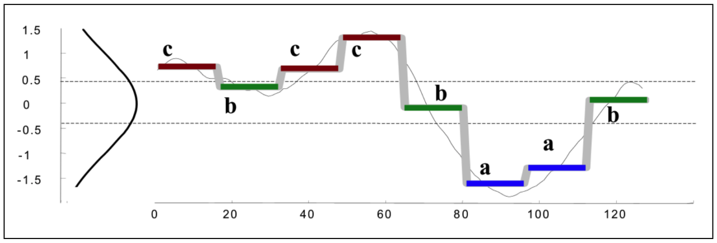


Fig. 1: Concept of Original SAX. Background grey thin line is replaced with bold line segments (PAA). PAA segments are mapped by normal distribution sketched on y axis into symbols, $a = (-\infty; -0.43\rangle, b = (-0.43; 0.43\rangle, c = (0.43, \infty)$. [8]

the i^{th} element of the alphabet, i.e., $\alpha_1 = a$ and $\alpha_2 = b$. Then the mapping from a PAA approximation C to a word $\hat{C}$ is obtained as follows:

$$\hat{c}_i = \alpha_j, \quad iif \quad \beta_{j-1} \leq \bar{c}_i < \beta_j \tag{2}$$

2.2 Techniques for distribution estimation

In the previous section we discussed internals of the original SAX method. SAX uses a Gaussian distribution to derive the regional breakpoints resulting in the generation of an equiprobable set of symbols. As we already mentioned, our method wants to make a new SAX method Gaussian distribution requirement free. In this section we want to mention other methods how to estimate data distribution and, based on them, improve SAX by a different way of setting the regional breakpoints. At the beginning, we want to state a common intuition to graphically represent data distribution - histogram plotting from underlying time series data points.

The histogram is former nonparametric density estimator with strong use in explorative data analysis for displaying and summarizing data. Bin width is an important parameter that needs selection prior histogram construction. It is evident that the choice of the bin width has a strong effect on the shape of the resulting histogram.

There were proposed many ways to determine optimal bin width $\hat{h}$ with n observed instances such as [16]:

$$\hat{h} = \frac{range\_of\_data}{1 + log_2 n} \tag{3}$$

or alternatively more general formula based on Mean Integrated Squared Error (MISE):

$$\hat{h} = \hat{C} n^{-1/3}, \tag{4}$$

where $\hat{C}$ is any selected statistic. Most known example of above mentioned formula is normal reference rule [12, 16]:

$$\hat{h} = 3.49 \hat{\sigma} n^{-1/3}, \tag{5}$$

where $\hat{\sigma}$ is an estimate of the standard deviation.

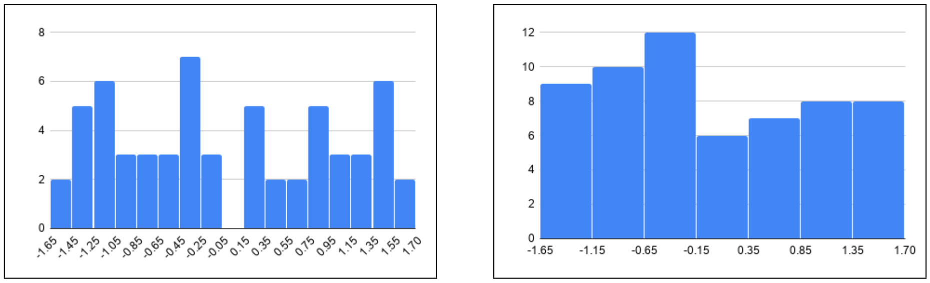


Fig. 2: Comparison of two histograms for the same dataset with bin widths 0.2 and 0.5 respectively. Incorrectly selected bin width causes visually different data distribution.

In past four decades there was a research in the field of distribution estimation using continuous functions - density estimators. Intuition behind kernel estimators is to describe underlying data histogram with smooth continuous line. More formally, given a set of N training data $y_n, n = 1, ..., N$, a kernel density estimator (KDE), with the kernel function K and a bandwidth parameter h, $(h \in R; h > 0)$, gives the estimated density $\hat{f}(y)$ for data y as follows [4]:

$$\hat{f}(y) = \frac{1}{N} \sum_{n=1}^{N} K(\frac{y - y_n}{h}) \tag{6}$$

Kernel function K should satisfy positivity and integrate-to-one constraints [4]:

$$K(y) \geq 0, \qquad \int_{R^+} K(y)dy = 1 \tag{7}$$

The quality of a kernel estimate depends less on the chosen K than on the bandwidth value h. It's crucial to choose the most suitable bandwidth as a value that is too small or too large will result in not useful estimation. Small values of h lead to undersmoothing estimates while larger h values lead to oversmoothing [5]. Figure 3 illustrates possible cases of incorrectly selected bandwidth parameter h.

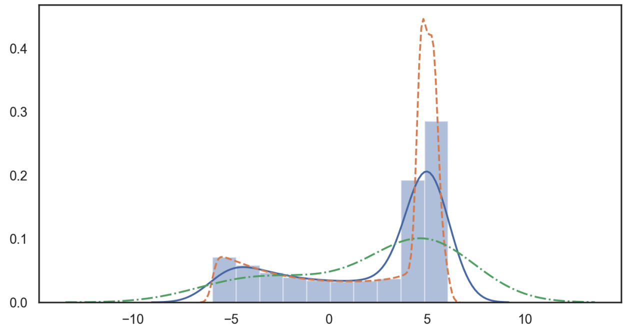


Fig. 3: Implications of different KDE bandwidth parameter selection Gaussian kernel. Optimal (blue line), oversmoothing (green line) and undersmoothing (orange line) bandwidths.

3 Distribution-wise SAX (dwSAX)

Formerly proposed SAX method is not well suited for time series with non-Gaussian distribution. If we apply Gaussian distribution lookup vector for breakpoints, we get non-optimal, still feasible, breakpoints. Our modified implementation focuses on elimination of this deficiency. Algorithm 1 illustrates overall main function method flow where input is sequence for symbolization, word length, alphabet size and bandwidth for kernel estimator result is sax representation of input sequence. In the next sections we will discuss specific internals of our method:

1. *probability density function estimation*: given normalized time series, we need to estimate probability density function to proceed with more precise breakpoints;
2. *breakpoints vector calculation*: having probability density function, the task is to calculate equiprobable breakpoints covering the domain of pdf.

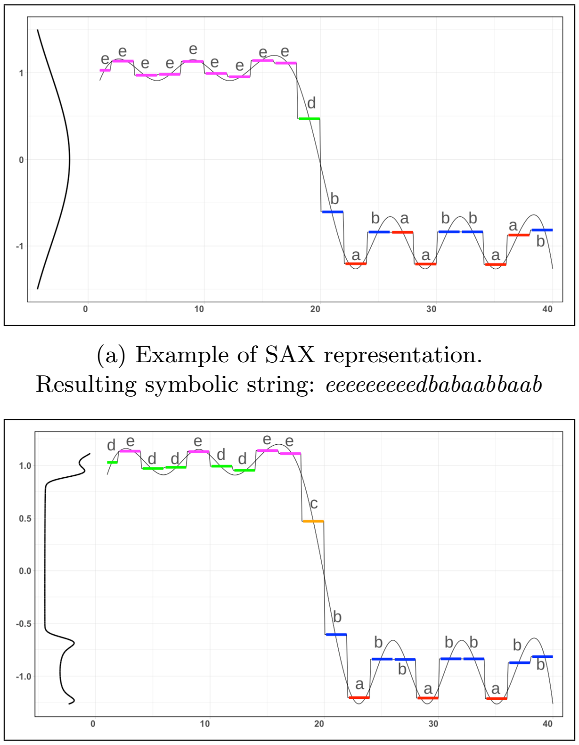
```
Data: Sequence, WordLength, AlphaSize, Bandwidth
Result: SAX Repr
PAA Sequence = ApplyPAA(Sequence, WordLength);
PDF Estimate = EstimatePDF(Sequence);
Breakpoints Vector = CalculateBreakpoints(AlphaSize, PDF Estimate);
SAX Repr = Map(PAA Sequence, Breakpoints Vector);
```

Algorithm 1: dwSAX main algorithm

3.1 Probability density estimation

With transformed PAA time series we can proceed with probability density estimation. Intuition behind probability density estimation is illustrated in figure

3.1. Having precisely estimated probability function will help us in further step breakpoints vector calculation for discretization procedure. Naive solution to this problem seems histograms exploitation as its computational complexity is much lower than the other methods such as KDE. The main drawback of histograms is their discrete representation which is not suitable for breakpoints interpolation. Our method for breakpoints calculation expects continuous probability function suitable for integral calculus with integrate-to-one constraint. KDE appears to be the solution to this problem. This methods needs to specify a kernel function K and a bandwidth parameter h. Selection of appropriate kernel function and bandwidth function depends on data and required precision of overall symbolic representation performance. To our best knowledge, Gaussian kernel gives most relevant results and should be applied as first possible option. The difference between dwSAX and SAX is depicted in Figure 4.



(a) Example of SAX representation.
Resulting symbolic string: *eeeeeeeeedbabaabbaab*

(b) Example of dwSAX representation.
Resulting symbolic string: *deddeddeecbabbabbabb*

Fig. 4: Application of different pdf on breakpoints selection. a) SAX with Gaussian distribution, b) dwSAX with KDE. KDE based breakpoints bring more precise symbolic representation compared with SAX with same alphabet size 5 and PAA=2.

3.2 Breakpoints vector calculation

Having estimated probability function using KDE, we can advance and estimate breakpoints based on probability distribution of time series. The main goal is to efficiently compute those breakpoints as we do not apply only specific Gaussian distribution and its precomputed values table. However, the idea for breakpoints selection is the same - select points from KDE probability density function (pdf) such that they produce equal-sized areas under KDE function curve.

Definition 1. *Let a denote alphabet size, pdf probability density function and β_n, β_{n+1} any two consecutive breakpoints from breakpoints vector B. Then breakpoints vector B is defined as a vector of ordered breakpoints β such that β_n, β_{n+1} follows:*

$$\int_{\beta_n}^{\beta_{n+1}} pdf(y)dy = \frac{1}{a} \tag{8}$$

Discretization process follows the same algorithm as proposed in the original SAX method. For reference see algorithm 2.

```
Data: PAA Sequence, Breakpoints Vector B
Result: SAX Representation
foreach Segment in PAA Sequence do
    for $i \leftarrow 2$ to Length(B) do
        if $B[i-1] \leq Segment$ and $Segment < B[i]$ then
            Append(SAX Representation, Alphabet[i])
        end
    end
end
```

Algorithm 2: dwSAX mapping procedure

4 Evaluation

We evaluated our proposed method modification using data mining tasks such as time series clustering and novelty/anomaly detection with the former method. As far as we know, there is no similar symbolic representation method that we can compare with, except of the SAX. In the next sections we discuss achieved results with their implications in real life method exploitation.

4.1 Clustering

Clustering is by nature one of the most commonly used data mining tasks. Formally, clustering is the division of data into groups of similar objects [1]. Traditionally, clustering techniques are divided into hierarchical and partitioning method families. For purposes of our evaluation, we decided to employ hierarchical clustering and correctly graphically illustrate results. In this data mining task evaluation we used well known Control Chart dataset[1] with selected Normal, Cyclic, Increasing trend and Decreasing trend charts represetatives.

Hierarchical clustering gives us a brief overview how are performing compared similarity measures. Similarity measure is in case of dwSAX and SAX virtually the same but symbolized sequences of time series are expected to be more closely clustered together within specific subtree. Figure 5 shows resulting dendrograms after applying aglomerative hierarchical clustering with Euclidean distance as similarity measure. Both dendrograms seem to be correct at class level series subtrees (second level subtrees from the bottom of dendrogram). Small differences can be observed at intra-class level subtrees clustering where dwSAX clusters more similar series in a common subtree depicting significantly smaller distance between them.

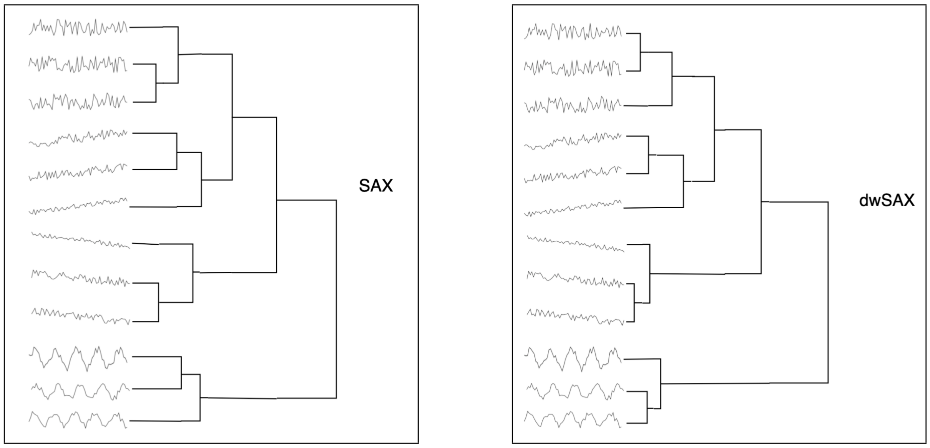


Fig. 5: A comparison of hierarchical clustering of selected Control Chart series. Both methods performs well, dwSAX gives more detailed clustering inside specific series class.

4.2 Novelty/Anomaly Detection

Novelty or anomaly detection is common data mining task, usually applied in data cleaning / preprocessing step or as targeted task. This process consists of learning phase and detection phase itself. In learning phase, we try to infer normal behavior model from observed data and apply it in detection phase. SAX and dwSAX are candidates for improving such kind of detection. Designed detection model consists of symbolization produced by specific SAX method and Markov chain model for encoding normal behavior motifs from produced symbolic representation. During detection phase, we replay window of last N symbols and sign

[1] Available at: https://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series

current symbol as anomaly when detection statistic is below specific threshold. We compared performance of mentioned methods on two different time series of the same length: a) periodical slightly noisy sine time series, b) periodical time series with strong seasonality. Figure 6 shows the results. In evaluation dataset there were 4 real anomalies and 6 temporal phases with not clear anomalous state. From the detection report we see that both methods are able to help in detection of clear anomalies (4 of 4) in strongly periodical time series. dwSAX slightly outperform SAX in detection of temporal unclear phases (3 of 6).

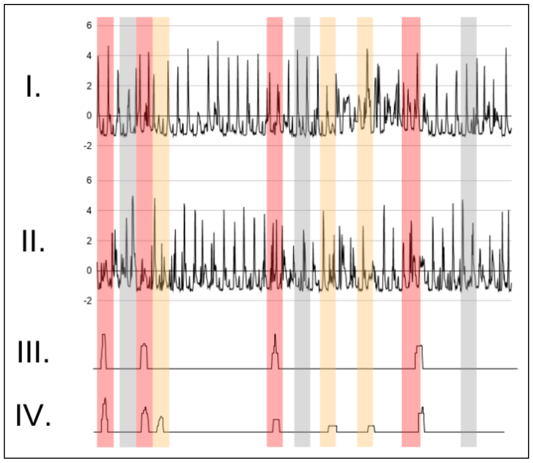


Fig. 6: A comparison of anomaly detection of real life household electricity consumption dataset. I. 1000 points training dataset II. 1000 points test dataset. III. and IV. detection using SAX and dwSAX respectively. Red, orange and grey areas depict commonly detected anomalies, dwSAX only detected anomalies and not detected anomalies respectively.

5 Conclusion and Future work

As stated in the Introduction, our main goal was to improve symbolic representation of time series with non-Gaussian data distribution. Lin et al. [8] proposed a superior method for symbolic representation of time series - SAX. Although this approach is interesting, it suffers from support for time series with non-Gaussian data distribution. We believe that we have designed an innovative solution to this problem. Our approach extends original SAX by means of dynamically captured data distribution of underlying time series and defining alternative vector of breakpoints for characters mapping. Data distribution estimation at its simplest form could be estimated through well-known and widely applicable histograms. However, this approach suffers from the ease of dynamic computation of breakpoints. An alternative solution, though with high overheads is estimation using continuous, function based, estimator. Density estimators appear to be a solution to this problem. The most common variants of these estimators are kernel density estimators.

This method represents a viable alternative to original SAX method. We compared our method with original SAX in two data mining tasks: clustering of time series and anomaly detection. As stated in the Evaluation, our method was able to improve clustering performance by means of significant lowering objective function over original SAX within the same number of iterations. In anomaly detection, both methods detected major anomalies in provided time series. However, our method detected in addition to major anomalies also less evident ones. This was possible achieved by improved breakpoints vector with higher resolution for highly probable values in time series.

The most important limitation lies in unnecessary KDE application in case of highly Gaussain distributed data. Applying both methods in this case will result in very similar symbolic representation. KDE estimates breakpoints similar to precomputed breakpoints from SAX table, but with undoubtly higher computational complexity. On the other had, applying dwSAX without any prior knowledge of data distribution will safely produce efficient symbolic representation. A number of potential shortcomings need to be considered. First, computational complexity of KDE and breakpoints vector recalculations needs to be considered in case of online exploitation. Second, the concept drift at its basis is not covered in proposed method, thought KDE with periodical recalculation is able to overcome a skew in data distribution to some extend. Third, knowledge of breakpoints vector used during discretization is crucial for further operations such as time series indexing. Despite this we believe that our work could be a springboard for research in the field of data distribution-wise symbolic time series representation.

This study has gone some way towards enhancing our understanding of efficient symbolic time series representation. To further our research we plan to design online version of our method to tackle computational complexity with hard online processing constraints. Our results are promising and should be validated by a larger sample size time series from real-life environments.

Acknowledgement

This work was partially supported by the project "Knowledge-based Approach to Intelligent Big Data Analysis" - Slovak Research and Development Agency under the contract No. APVV-16-0213 and "International Centre of Excellence for Research of Intelligent and Secure Information-Communication Technologies and Systems - phase II" - ITMS2014+ 313021W404.

References

1. Berkhin, P.: A survey of clustering data mining techniques. In: Grouping multidimensional data, pp. 25–71. Springer (2006)

2. Eghan, R.E., Amoako-Yirenkyi, P., Omari-Sasu, A.Y., Frimpong, N.K.: Time-frequency coherence and forecast analysis of selected stock returns in ghana using haar wavelet. Journal of Advances in Mathematics and Computer Science pp. 1–12 (2019)
3. Fournier-Viger, P., Lin, J.C.W., Kiran, R.U., Koh, Y.S., Thomas, R.: A survey of sequential pattern mining. Data Science and Pattern Recognition **1**(1), 54–77 (2017)
4. Hwang, J.N., Lay, S.R., Lippman, A.: Nonparametric multivariate density estimation: a comparative study. IEEE Transactions on Signal Processing **42**(10), 2795–2810 (1994)
5. Jones, M.C., Marron, J.S., Sheather, S.J.: A brief survey of bandwidth selection for density estimation. Journal of the American statistical association **91**(433), 401–407 (1996)
6. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**(3), 263–286 (2001)
7. Keogh, E., Lin, J., Fu, A.: Hot sax: Efficiently finding the most unusual time series subsequence. In: Fifth IEEE International Conference on Data Mining (ICDM'05). pp. 8–pp. Ieee (2005)
8. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. pp. 2–11 (2003)
9. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery **15**(2), 107–144 (2007)
10. Mahmoudi, M.R., Heydari, M.H., Roohi, R.: A new method to compare the spectral densities of two independent periodically correlated time series. Mathematics and Computers in Simulation **160**, 103–110 (2019)
11. Sato, T., Takano, Y., Miyashiro, R.: Piecewise-linear approximation for feature subset selection in a sequential logit model. Journal of the Operations Research Society of Japan **60**(1), 1–14 (2017)
12. Scott, D.W.: On optimal and data-based histograms. Biometrika **66**(3), 605–610 (1979)
13. Senin, P., Malinchik, S.: Sax-vsm: Interpretable time series classification using sax and vector space model. In: 2013 IEEE 13th international conference on data mining. pp. 1175–1180. IEEE (2013)
14. Shieh, J., Keogh, E.: i sax: indexing and mining terabyte sized time series. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 623–631 (2008)
15. Tamura, K., Ichimura, T.: Clustering of time series using hybrid symbolic aggregate approximation. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8. IEEE (2017)
16. Wand, M.: Data-based choice of histogram bin width. The American Statistician **51**(1), 59–64 (1997)
17. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery **26**(2), 275–309 (2013)
18. Yang, S., Liu, J.: Time-series forecasting based on high-order fuzzy cognitive maps and wavelet transform. IEEE Transactions on Fuzzy Systems **26**(6), 3391–3402 (2018)