Data-driven Kernel-based Probabilistic SAX for Time Series Dimensionality Reduction

Konstantinos Bountrogiannis[1,2], George Tzagkarakis[1], and Panagiotis Tsakalides[1,2]
[1]*Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Greece*
[2]*Department of Computer Science, University of Crete, Heraklion, Greece*
E-mails: kbountro@ics.forth.gr, gtzag@ics.forth.gr, tsakalid@ics.forth.gr

***Abstract*—The ever-increasing volume and complexity of time series data, emerging in various application domains, necessitate efficient dimensionality reduction for facilitating data mining tasks. Symbolic representations, among them symbolic aggregate approximation (SAX), have proven very effective in compacting the information content of time series while exploiting the wealth of search algorithms used in bioinformatics and text mining communities. However, typical SAX-based techniques rely on a Gaussian assumption for the underlying data statistics, which often deteriorates their performance in practical scenarios. To overcome this limitation, this work introduces a method that negates any assumption on the probability distribution of time series. Specifically, a data-driven kernel density estimator is first applied on the data, followed by Lloyd-Max quantization to determine the optimal horizontal segmentation breakpoints. Experimental evaluation on distinct datasets demonstrates the superiority of our method, in terms of reconstruction accuracy and tightness of lower bound, when compared against the conventional and a modified SAX method.**

***Index Terms*—Data-driven probabilistic SAX, kernel density estimation, symbolic representations, Lloyd-Max quantization**

I. Introduction

Representing and interpreting complex time-varying phenomena is a challenging task in several application domains. Such issues become even more demanding in view of the large volumes of time series data, emerging thanks to the advances of computing technologies. Typical examples include healthcare data, microarray gene expression data in genetics, and large panel and e-commerce data in finance and marketing, just to name a few. Efficiently mining this data deluge necessitates the extraction of descriptive motifs in appropriate lower-dimensional spaces, which provide a meaningful, yet compact, representation of the original inherent information to be further employed for executing high-level tasks, such as event detection and classification [1]–[3], among others.

The family of symbolic aggregate approximation (SAX) methods [4] has a prominent role among the several existing motif discovery techniques. Due to its conceptual simplicity and computational tractability, SAX has been widely used in monitoring, processing and mining data of numerous sources, including physiological data [5], smart grids [6], building systems [7], and stock market [8].

This work is funded by the Interreg V-A Greece-Cyprus 2014-2020 programme, co-financed by the European Union (ERDF) and National Funds of Greece and Cyprus, under the project SmartWater2020.

Although SAX and its variants [9]–[13] can lead to high-precision results in the case of data characterized by Gaussian statistics, however, their performance may degrade dramatically in more general cases. Indeed, in practical scenarios, where the underlying probability distribution of a time series deviates significantly from a Gaussian, or when the distribution changes across time, then, the previous SAX-based techniques are not capable of adapting to the time-evolving statistics. As a result, their low-dimensional representation and motif interpretation power diminishes.

Motivated by the above limitations, this work aims at negating any assumptions regarding the probability distribution of a given time series by designing a map between the time series space and the space of symbols from a predefined alphabet, which adapts directly to the data statistics. To this end, our proposed method first applies a kernel density estimator (KDE) [14] on the time series to approximate accurately the underlying probability density function (pdf). Then, the output of the KDE is coupled with a Lloyd-Max quantizer [15] to estimate the optimal horizontal segmentation breakpoints, which are further used to define the map between the time series samples and the alphabet's symbols.

Overall, the contribution of this paper is an efficient data-driven SAX-like method, hereafter referred to as probabilistic SAX (pSAX), which: (i) achieves a more accurate low-dimensional representation of a time series, and (ii) enables the construction of a distance measure in the symbols space that is the closest to that of the Euclidean distance in the raw data space, when compared with its SAX-based alternatives.

The rest of the paper is organized as follows: Section II introduces briefly the conventional SAX and discusses the differences between our proposed methodology and prior studies. Section III describes in detail our proposed pSAX method, along with a new distance measure. Section IV evaluates the performance of our method on real datasets and compares its accuracy with the conventional SAX method and one well-established SAX-based counterpart. Finally, Section V summarizes the main outcomes of this work and gives directions for further extensions.

II. Background and Related Work

This section describes briefly the conventional SAX method, and discusses how our proposed approach differs from prior SAX-based representations.

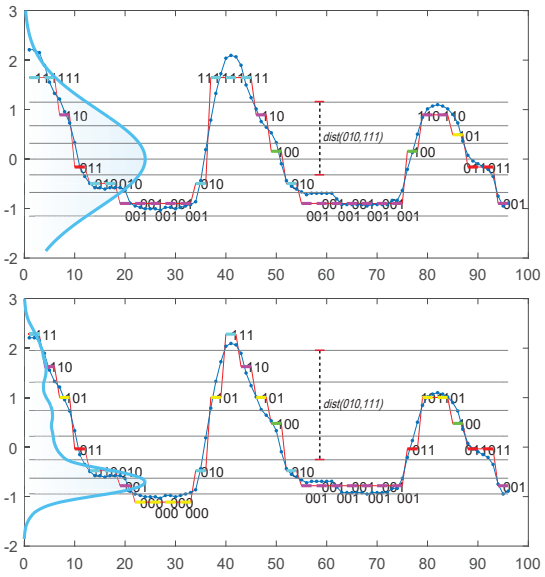

Fig. 1: SAX representation of a time series: each PAA segment is assigned a codeword, depending on which of the $\alpha = 8$ equiprobable intervals it falls in. Top plot: Standard Gaussian distribution; Bottom plot: KDE-estimated distribution.

A. Symbolic approximations and lower bounding distance

In the following, $\mathcal{T}^N$ denotes the set of time series of length N, $\mathcal{Y}^M$ is the set of real vectors of length M, and $\mathcal{C}_A^M$ the set of all vectors of M codewords belonging to an alphabet A of size $|A| = \alpha$. Let $U = (u_1, u_2, \ldots, u_N) \in \mathcal{T}^N$ be a discrete time series of N samples. Without loss of generality, U is first Z-normalized to zero mean and unit variance. Then, the core of a symbolic aggregate approximation (SAX) [4] consists of two steps, coupled with the definition of an appropriate distance measure in the lower-dimensional space, which lower bounds the Euclidean distance in the original space.

The first step of SAX implements a piecewise aggregate approximation (PAA) $\mathcal{T}^N \rightarrow \mathcal{Y}^M$, which transforms a given time series $U \in \mathcal{T}^N$ into a vector $Y = (y_1, \ldots, y_M) \in \mathcal{Y}^M$, with $M < N$. For this, U is divided into M equal size segments and the average value is calculated for each segment. In the second step, a discretization $\mathcal{Y}^M \rightarrow \mathcal{C}_A^M$ is applied to Y, which maps the averages into a predefined set of symbols. More precisely, the Z-normalized time series is assumed to follow a standard Gaussian distribution. Under this assumption, the M averages in Y are quantized within α equiprobable intervals under the standard Gaussian pdf curve. Each quantization interval is bounded by two cutlines and is assigned a codeword from the alphabet A (ref. top plot in Fig. 1 for a visualization of the two-step process). The two-step transformation $\mathcal{T}^N \rightarrow \mathcal{Y}^M \rightarrow \mathcal{C}_A^M$ produces the SAX representation of length M from the alphabet A.

Given two distinct time series U, $S \in \mathcal{T}^N$, their Euclidean distance is defined by

$$d(U,S) = \sqrt{\sum_{i=1}^{N} (u_i - s_i)^2} \ . \tag{1}$$

To guarantee that no false dismissals occur when performing high-level tasks, such as similarity search, it is desirable to define a distance measure in the lower-dimensional space $\mathcal{C}_A^M$ that lower bounds the Euclidean distance in the original space $\mathcal{T}^N$ [16]. Let $C, Q \in \mathcal{C}_A^M$ be the symbolic representations of the time series U and S, respectively. Then, a distance measure in the quantized space of alphabet symbols, which lower bounds the Euclidean distance in the original time series space is defined as follows,

$$mindist(C,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \left(dist(c_i, q_i)\right)^2} \ , \tag{2}$$

where $dist(c_i, q_i)$ is the absolute difference of the two closest cutlines that respectively bound c_i and q_i (ref. Fig. 1 for an example). Furthermore, if $Y \in \mathcal{Y}^M$ is the PAA of U and $Q \in \mathcal{C}_A^M$ is the SAX representation of S, a tighter lower bounding distance measure is defined by

$$mindist\_PAA(Y,Q) = \sqrt{\frac{N}{M} \cdot \sum_{i=1}^{M} \begin{cases} (\beta_{L_i} - y_i)^2 & \text{if } \beta_{L_i} > y_i \\ (\beta_{U_i} - y_i)^2 & \text{if } \beta_{U_i} < y_i \\ 0 & \text{otherwise} \end{cases}} \ , \tag{3}$$

where β_{L_i} and β_{U_i} are the lower and upper cutlines of codeword q_i. By combining (1) and (3), the tightness of lower bound (TLB) measure is defined as

$$TLB(U,S) = \frac{mindist\_PAA(Y,Q)}{d(U,S)} \ . \tag{4}$$

Note that conventional SAX-based techniques compute the optimal cutlines based on a Gaussian assumption for the data distribution. As such, the larger the deviation of the true distribution from a Gaussian, the greater the information loss, which severely decreases the accuracy of the symbolic representation. To alleviate this drawback, our method adapts directly to the data using a kernel density estimator (KDE). Regarding the computation of the optimal cutlines, the Lloyd-Max quantizer is employed, which minimizes the distortion based on a mean squared error (MSE) criterion.

B. Relation to prior work

The method presented here takes advantage of the model-free data-adaptive nature of KDEs, along with the optimality (in the MSE sense) of Lloyd-Max quantization, for the samples-to-symbols assignment. Previous works also perform a data-driven discretization of time series by employing Lloyd's algorithm [17], [18], self-organizing maps [19], or other clustering methods [20]. A modified version of SAX (a.k.a. aSAX) [9] employs the k-means algorithm for the discretization, which is the only data-adaptive extension of SAX to the authors' knowledge. While the present study is related to prior works in data-driven discretization, our methodology capitalizes on the fact that Lloyd-Max quantization partitions the data according to the estimated underlying probability distribution. In earlier studies, the k-means and the other

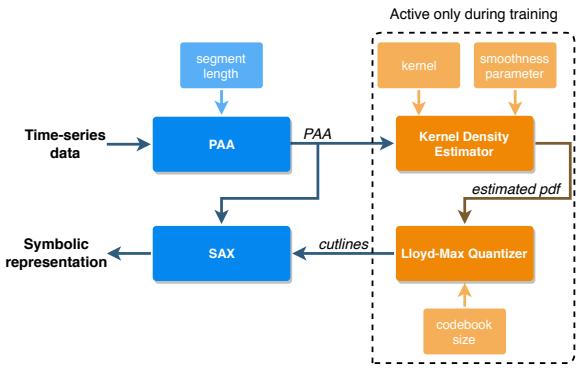

Fig. 2: Overview of our pSAX pipeline. Notice that KDE and Lloyd-Max are active only during training. After training, the cutlines are fixed and used unchanged for future inputs.

clustering methods partition directly the observed samples, overlooking the overall behavior of the data source. As a result, our method increases the symbolic representation's accuracy by better adapting to the probabilistic structure of time series data.

III. Data-driven Probabilistic SAX

In contrast to the conventional SAX-based approaches, our method exploits the efficiency of KDEs [14] to approximate accurately the true distribution by adapting directly to the data, without any prior probabilistic assumption. To further enhance the generalization capability of our method, the KDE-based step is coupled with a Lloyd-Max quantizer [15] to compute the optimal cutlines. Fig. 2 illustrates the overall pipeline of our method to be analyzed below, whilst the bottom plot in Fig. 1 visualizes the outcome of our proposed symbolic representation, as opposed to the conventional SAX (top plot).

A. Kernel density estimator

By definition, a kernel density estimator is the summation of a set of translated and dilated kernel functions centered at each observed sample, as follows,

$$\hat{f}_X(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x_i - x}{h}\right) , \tag{5}$$

where x_i, $i = 1, \ldots, N$, are the observed samples, $K(\cdot)$ is the kernel function which controls the weight given to the neighboring samples of $x \in X$, and h is a smoothness parameter. In our implementation, we employ the Epanechnikov kernel [21]. This choice is motivated by (i) the asymptotic optimality (it minimizes the asymptotic mean integrated squared error) and (ii) the compact support of this kernel, which yields an increased computational efficiency. In advance, the smoothness parameter was chosen according to Silverman's rule of thumb [22, Sec. 3], reduced by a factor.

B. Lloyd-Max quantizer

The Lloyd-Max quantization algorithm (ref. Alg. 1) reduces iteratively the MSE between the computed codewords and the source by alternating between two necessary optimality

Algorithm 1 Lloyd-Max Quantizer

1: Initialize all codewords c_i, $i = 1, 2, \ldots, M$
2: Initialize distribution domain: $b_0 \leftarrow -\infty$, $b_M \leftarrow +\infty$
3: **while** stopping criteria are not met **do**
4: $b_j \leftarrow \frac{1}{2}(c_j + c_{j+1})$, $j = 1, 2, \ldots, M-1$
{Calculate new boundaries (cutlines)}
5: $c_i \leftarrow \frac{\int_{b_{i-1}}^{b_i} x \cdot \hat{f}_X(x) dx}{\int_{b_{i-1}}^{b_i} \hat{f}_X(x) dx}$, $i = 1, 2, \ldots, M$
{Calculate new centroids (codewords)}
6: **end while**

Method	Time (ms)			
	$\alpha = 16$	$\alpha = 32$	$\alpha = 64$	$\alpha = 128$
pSAX	15.206	20.539	29.546	62.407
aSAX	2.859	4.644	6.402	9.587

TABLE I: Training time of pSAX and aSAX on 10 sequences from the Koski ECG dataset ($M = 40$, $\alpha \in \{16, 32, 64, 128\}$, CPU: Intel i7-6700@3.8GHz).

conditions regarding the update of the boundaries (line 4) and the centroids (line 5). A maximum number of iterations is chosen as the stopping criterion, which is set equal to 100 in our implementation. Furthermore, in contrast to the conventional case, we initialize the quantizer using the k-means++ algorithm [23], yielding an improved convergence to a local minimum, in terms of MSE and convergence speed.

Notice that a training phase is required for our pSAX method. Specifically, the KDE module for pdf estimation is trained first with a sufficient number of PAA segments. Then, the Lloyd-Max quantization intervals are calculated based on the estimated distribution. Nevertheless, this is done only once during initialization. As such, the running time of the training phase does not contribute to the running time of the subsequent dimensionality reduction process. Besides, our algorithm is trained efficiently using a highly reduced set of training sequences (at the order of 10 in this study). Table I shows the training times of pSAX and aSAX. As expected, the training of pSAX takes longer than its aSAX counterpart, which is due to the KDE step and the numerical integrations in the Lloyd-Max step.

C. A novel distance measure

The Lloyd-Max quantizer provides arithmetic values for the codewords, apart from the cutlines. Formally, these values are the centroids of the bounded regions. This feature can be further exploited to define a new distance measure between two symbolic sequences $Q, C \in \mathcal{C}_A^M$, as follows,

$$d_s(Q, C) = \sqrt{\sum_{i=1}^{M} (q_i - c_i)^2} . \tag{6}$$

Although this measure does not lower bound the Euclidean distance, however, it is the closest to the true Euclidean distance in the MSE sense, up to a distortion caused by the KDE and Lloyd-Max steps. Accordingly, a distance measure

between a time series and a symbolic sequence can be derived by utilizing the computed centroids. Specifically, given $U \in \mathcal{T}^N$ and $C \in \mathcal{C}_A^M$, their distance is defined by

$$d_e(U,C) = \sqrt{\frac{1}{N}\sum_{i=1}^{M}\left(\sum_{j=(N/M)(i-1)+1}^{(N/M)i}(u_j - c_i)^2\right)}. \quad (7)$$

In the special case when C is the symbolic representation of U, then d_e is the root mean squared error (RMSE) between U and its reconstruction from C. It is worth noting that the k-means algorithm used by aSAX computes the centroids, too. However, these centroids are computed by using the observed samples, and not by exploiting the underlying probability distribution of the data. KDE is trained with the same set of training samples but, as we will show in the following section, the estimated probability distribution can better approximate the true centroids of the whole dataset.

IV. Experimental evaluation

This section evaluates the performance of pSAX and compares against aSAX and the conventional SAX, with respect to the achieved TLB (ref. (4)) and RMSE (ref. (7)) values. The methods are compared for a varying symbolic sequence length $M \in \{32, 48, 64, 80\}$ (the lower this number, the higher the dimensionality reduction), alphabet size $\alpha \in \{8, 16, 32, 64, 128\}$ and time series subsequence length $N \in \{480, 1920\}$ (short and long). Four distinct datasets are employed (Koski ECG, Muscle Activation, Rittweger EOG, and Respiration)[1], which are characterized by both structured and complex behaviors. For each dataset, the results are averaged over 8000 Monte Carlo iterations, each one corresponding to a randomly selected segment from the associated time series. The length of the segments N and the parameters M, α are chosen in compliance with the experimental sections in [4], [9]. In order to simulate streaming scenarios, the training samples are taken from the beginning of each dataset. We note that, although our method does not require a Z-normalization of the time series, however, we also Z-normalize the given data for a fair comparison with the other methods.

As a first experiment, we investigate the effect of the symbolic sequence length M on the performance of our method. In particular, Fig. 3 shows the average TLB and RMSE values for the Respiration dataset (characterized by dense spikes). As it can be seen, pSAX achieves a tighter lower bound, yielding a more accurate lower-bounding distance (3) in the lower-dimensional space. Additionally, pSAX achieves a more accurate reconstruction (lower RMSE) against both SAX and aSAX. Furthermore, the superiority of pSAX is more prominent as M increases. Table II shows the average TLB and RMSE values for two additional datasets, namely, the Rittweger EOG (a waveform with varying frequency) and Koski ECG (characterized by a repetitive pattern). As it can be seen, similar results are obtained, demonstrating the superiority of our proposed method.

[1]Datasets available at www.cs.ucr.edu/~eamonn/iSAX/iSAX.html

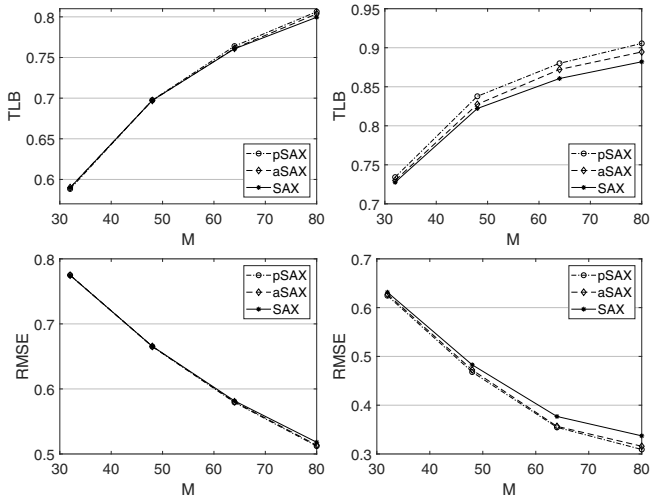

Fig. 3: Tightness of lower bound (top) and reconstruction error (bottom) vs. M for the Respiration dataset ($\alpha = 32$). Left: $N = 1920$ – Right: $N = 480$.

		$M = 32$	$M = 64$	$M = 80$
Dataset	**Method**	**TLB**		
Rittweger EOG	pSAX	**0.8954**	**0.9481**	**0.9570**
	aSAX	0.8930	0.9416	0.9500
	SAX	0.8936	0.9388	0.9463
Koski ECG	pSAX	**0.8205**	**0.9200**	**0.9380**
	aSAX	0.8084	0.8984	0.9219
	SAX	0.7719	0.8292	0.8406
Dataset	**Method**	**RMSE**		
Rittweger EOG	pSAX	**0.3614**	**0.1858**	**0.1606**
	aSAX	0.3638	0.1900	0.1663
	SAX	0.3626	0.2010	0.1782
Koski ECG	pSAX	**0.5025**	**0.2917**	**0.2535**
	aSAX	0.5037	0.2956	0.2574
	SAX	0.5415	0.4443	0.4310

TABLE II: Average TLB and RMSE vs. M, for the Rittweger EOG and Koski ECG datasets ($N = 480$, $\alpha = 64$).

Next, we study the effect of the alphabet size α on the performance of pSAX. To this end, Fig. 4 shows the average TLB and RMSE values for the Muscle Activation dataset (a noisy periodic signal), as a function of α. The experiments show that pSAX achieves a tighter lower bound, along with a better reconstruction quality, when compared against SAX. The same holds against aSAX most of the times, except for a few datasets, when the alphabet size is $\alpha \leq 8$. As expected, the larger the alphabet size the more improved the performance of all the three methods (i.e., higher TLB and lower RMSE). Similar results are shown in Table III for the Respiration and Muscle Activation datasets, demonstrating the efficiency of pSAX in adapting to distinct data generating processes. Note that, for the calculation of RMSE (7), the pSAX codewords are computed using the Lloyd-Max algorithm, whilst for aSAX and SAX they are computed using the k-means and the line 5 in Algorithm 1, respectively.

Overall, we conclude that, under all the experimental parameters settings tested herein, pSAX achieves a tighter lower bound (i.e., closer to 1) and a smaller RMSE when compared with SAX. Moreover, in the vast majority of the settings, pSAX also outperforms aSAX, except for a few cases when the

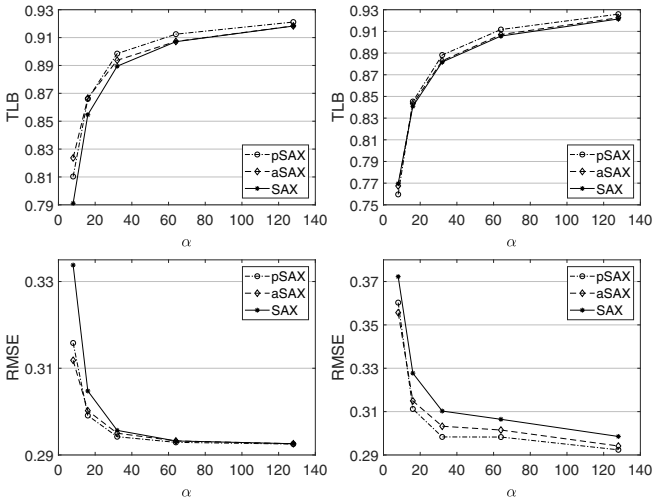

Fig. 4: Tightness of lower bound (top) and reconstruction error (bottom) vs. α for the Muscle Activation dataset ($M = 80$). Left: $N = 1920$ – Right: $N = 480$.

		$\alpha = 8$	$\alpha = 32$	$\alpha = 128$
Dataset	**Method**	**TLB**		
Respiration	pSAX	**0.5006**	**0.5834**	**0.6042**
	aSAX	0.4954	0.5794	0.6032
	SAX	0.4942	0.5664	0.5917
Muscle Activation	pSAX	0.8102	**0.8968**	**0.9223**
	aSAX	**0.8231**	0.8920	0.9196
	SAX	0.7911	0.8878	0.9194
Dataset	**Method**	**RMSE**		
Respiration	pSAX	**0.7894**	**0.7727**	**0.7740**
	aSAX	0.7910	0.7742	0.7741
	SAX	0.8110	0.7952	0.7912
Muscle Activation	pSAX	0.3160	**0.2941**	**0.2924**
	aSAX	**0.3120**	0.2949	0.2925
	SAX	0.3341	0.2955	0.2925

TABLE III: Average TLB and RMSE vs. α, for the Respiration and Muscle Activation datasets ($N = 1920$, $M = 80$).

alphabet size is very small. Finally, note that, when SAX-based methods are used for data indexing purposes, the achieved speedup is generally nonlinear with respect to TLB [10]. Thus, we expect that our pSAX method will require a reduced number of disk accesses, when compared against aSAX and SAX, due to its higher TLB. However, the design of a pSAX-based indexing technique is left as a future thorough study.

V. Conclusions and Future Work

This work proposes a new symbolic representation method, which generalizes previous SAX-based techniques by adapting directly to the underlying probability distribution of a given time series, without any prior model assumption for the data generating process. To this end, our pSAX method exploits the power of KDEs for accurate pdf estimation by a restricted amount of training samples, with the efficiency of Lloyd-Max quantization for optimizing the cutlines and associated codewords. Furthermore, we introduce a novel distance measure in the lower-dimensionality space of symbolic sequences, which can be employed in data mining tasks. Most importantly, our method can be coupled with other variants of SAX in a straightforward way, to improve their performance in the case of non-Gaussian data.

Currently, time series and their statistics are considered in a static framework by our method. As a further generalization of pSAX, an online extension over sliding windows is under investigation for real-time applications. To this end, we are interested in tracking the evolution of data statistics across time, while also applying a dynamic quantization scheme, under execution time constraints. Finally, the efficacy of our proposed distance measure will also be evaluated in data indexing scenarios.

References

[1] T. Fu, "A review on time series data mining," *Engin. Appl. of Artif. Intell.*, vol. 24, no. 1, pp. 164–181, 2011.

[2] E. Berlin and K. V. Laerhoven, "Detecting leisure activities with dense motif discovery," in *Proc. ACM Conf. Ubiq. Comp.*, 2012.

[3] A. Balasubramanian and B. B. Prabhakaran, "Flexible exploration and visualization of motifs in biomedical sensor data," in *Proc. Worksh. on Data Min. for Healthcare, in conj. with ACM KDD*, 2013.

[4] J. Lin et al., "Experiencing SAX: A novel symbolic representation of time series," *Data Min. Knowl. Disc.*, vol. 15, no. 2, pp. 107–144, 2007.

[5] P. Ordonez et al., "Using modified multivariate bag-of-words models to classify physiological data," in *IEEE Intl' Conf. Data Min. Work.*, 2011.

[6] Y. Wang, Q. Chen, C. Kang, and Q. Xia, "Clustering of electricity consumption behavior dynamics toward big data applications," *IEEE Trans. on Smart Grid*, vol. 7, no. 5, pp. 2437–2447, Sep. 2016.

[7] C. Miller, Z. Nagy, and A. Schlueter, "Automated daily pattern filtering of measured building performance data," *Autom. in Constr.*, vol. 49, pp. 1 – 17, 2015.

[8] S. Aghabozorgi and Y. W. Teh, "Stock market co-movement assessment using a three-phase clustering method," *Exp. Sys. with Appl.*, vol. 41, no. 4, Part 1, pp. 1301 – 1314, 2014.

[9] N. D. Pham, Q. L. Le, and T. K. Dang, "Two novel adaptive symbolic representations for similarity search in time series databases," in *12th Intl' Asia-Pacific Web Conf.*, April 2010, pp. 181–187.

[10] J. Shieh and E. Keogh, "iSAX: Indexing and mining terabyte sized time series," in *Proc. 14th ACM SIGKDD Intl' Conf. on Knowl. Disc. and Data Min.* Las Vegas, USA: ACM, 2008, pp. 623–631.

[11] S. Malinowski et al., "1d-SAX: A novel symbolic representation for time series," in *Advanc. in Intell. Data Anal. XII*. Springer Berlin Heidelberg, 2013, pp. 273–284.

[12] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189 – 198, 2014.

[13] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation," *DEWS2006 4A-i8*, vol. 7, 2006.

[14] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 09 1962.

[15] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. 6, no. 1, pp. 7–12, March 1960.

[16] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *SIGMOD Rec.*, vol. 23, no. 2, pp. 419–429, May 1994.

[17] G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," in *Proc. Fourth Intl' Conf. on Knowl. Discov. and Data Min.*, ser. KDD'98. AAAI Press, 1998, pp. 16–22.

[18] V. Megalooikonomou, Q. Wang, G. Li, and C. Faloutsos, "A multiresolution symbolic representation of time series," in *21st Intl' Conf. on Data Eng.*, April 2005, pp. 668–679.

[19] T. Kohonen and P. Somervuo, "Self-organizing maps of symbol strings," *Neurocomputing*, vol. 21, no. 1, pp. 19 – 30, 1998.

[20] Y. Huang and P. S. Yu, "Adaptive query processing for time-series data," in *Proc. Fifth ACM SIGKDD Intl' Conf. on Knowl. Discov. and Data Min.*, ser. KDD '99. New York, NY, USA: ACM, 1999, pp. 282–286.

[21] V. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory Probab. Appl.*, vol. 14, no. 1, pp. 153–158, 1969.

[22] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[23] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Ann. ACM-SIAM Symp. on Disc. Alg.*, ser. SODA '07, Philadelphia, PA, USA, 2007.