1d-SAX: A Novel Symbolic Representation for Time Series

Simon Malinowski, Thomas Guyet, René Quiniou, Romain Tavenard

1d-SAX : a Novel Symbolic Representation for Time Series

Simon Malinowski[1], Thomas Guyet[1], René Quiniou[2] and Romain Tavenard[3]

[1] AGROCAMPUS-OUEST/ IRISA-UMR 6074, F-35042 Rennes, France
email : first.last@agrocampus-ouest.fr
[2] Centre de Recherche INRIA Rennes Bretagne Atlantique, France
[3] IDIAP Research Institute, Martigny, Switzerland

Abstract. SAX (Symbolic Aggregate approXimation) is one of the main symbolization technique for time series. A well-known limitation of SAX is that trends are not taken into account in the symbolization. This paper proposes 1d-SAX a method to represent a time series as a sequence of symbols that contain each an information about the average and the trend of the series on a segment. We compare the efficiency of SAX and 1d-SAX in terms of i) goodness-of-fit and ii) retrieval performance for querying a time series database with an asymmetric scheme. The results show that 1d-SAX improves retrieval performance using equal quantity of information, especially when the compression rate increases.

1 Introduction

Time series data mining (TSDM) has recently attracted the attention of researchers in data mining due to the increase availability of data with temporal dependency. TSDM algorithms such as classification/clustering of time series, pattern extraction, similarity search require a distance measure between time series. The computation of these distances is mainly done using the classical Euclidean distance or the Dynamic Time Warping distance. These computations may lead to untractable costs for long series and/or huge databases. Hence, many approximate representations of time series have been developed over the last decade. Symbolic representation is one technique to approximate time series. The most used symbolization technique is called SAX (Symbolic Aggregate approXimation) [7]. It is a very simple technique to symbolize time series without the need for any a priori information. It has been shown to provide good performance for TSDM purposes. Some extensions to the SAX representation have been proposed to take into account the slope information in the time series segments [2, 8, 11]. We propose in this paper a novel symbolic representation for time series, based on the quantization of the linear regression of segments of the time series. We first show that this novel symbolic representation fits the original data more accurately than the SAX representation for the same number of symbols available. Then, this symbolic representation is used to make efficient similarity search in a time series database. We propose an asymmetric querying scheme based on our symbolic representation and compare its performance with the one based on SAX representation.

2 Background and Related Work

In the domain of time series data mining, approximated representations of time series are needed to enable efficient processing. Many methods have been proposed for dimensionality reduction, most of them being numerical *e.g.* discrete Fourier transform (DFT), discrete wavelet transform (DWT), singular value decomposition (SVD), principal component analysis (PCA), adaptive piecewise constant approximation (APCA), piecewise aggregate approximation (PAA), *etc* (see [1] for a survey). Symbolic methods have also been widely used because, beyond simplicity, readability and efficiency for time series representation, algorithms from other domains such as text processing and information retrieval, or bioinformatics can be used. Among symbolic representation methods, SAX proposed by Lin *et al.* [7] earned a large success. SAX is based on PAA and assumes that PAA values follow a Gaussian distribution. SAX discretizes PAA values according to equal-sized areas under the Gaussian curve yielding so-called breakpoints. Using lower bounds that are cheap to compute helps focusing on a small subset of the database sequences for which exact distance can later be computed [10].

The quality of the SAX representation of a time series depends on i) the number of PAA coefficients, *i.e.* the number of segments the time- series is divided in, ii) the number of symbols used for quantization (the alphabet size), iii) the gaussianity assumption. Several works have addressed these problems. In [9], Pham *et al.* alleviate the gaussianity assumption by introducing adaptive breakpoint vectors acting on segment size and alphabet size. However, the simplicity of SAX is lost by introducing a preprocessing phase using a clustering method. Other approaches attempt to enrich the PAA representation and, further, the SAX symbols. Extended SAX (ESAX)[8] associates the symbolic minimum and maximum of the PAA segment to the related SAX symbol as well as the order of their occurrences. This defines an abstract shape that gives finer results in time series retrieval. However, the size of the ESAX representation is 3 times the size of the SAX representation. From an efficiency point of view, authors did not compare their method with a SAX representation of the same size. In [3], authors make use of piecewise linear approximation (PLA), but only in a post-processing step and without quantizing PLA values. Several very recent works attempt to introduce a symbolic representation of the segment trend into the SAX representation. In [2], Esmael *et al.* associate one of the trend values U (up), D (down) and S (straight) to each SAX symbols computed from the segment trend. This yields a symbolic representation alternating SAX and trend values, having twice the size of the SAX representation. Trend approximations are obtained by linear regression and quantizing the line slope. No details are given about slope quantization and about the justification of using only three values. In [11], Zalewski *et al.* represent the slope information by quantizing the differences between the PAA values of two successive segments in the time series. A quantization algorithm separates difference values into k classes and determines the related centroids. The symbolic value affected to some difference value is the symbol associated to the closest centroid. Several quantization meth-

ods are evaluated. The main drawback of this method is the loss of simplicity and readability by switching to a first order derivative representation. In [6], authors introduce TSX, a Trend-based Symbolic approXimation. TSX divides a segment into 3 sub-segments determined by the most peak (MP) point, the most dip (MD) point and the bounds of the PAA segment. Then, TSX associates to each SAX symbol the trend information (symbolic slope) of its related sub-segments. This yields a 4-tuple symbolic series representation of time series. This representation is close to the ESAX representation but it is finer. A static lookup table is given for selecting the slope breakpoints in the TSX representation. However, the authors did not take into account the fact that slope breakpoints are dependent on the selected segment size.

We propose in this paper a symbolic representation that quantizes both the average and slope values of the time series on each segment. It hence produces one symbol per segment, each symbol can be interpreted in terms of an average and a slope value related to the linear regression on the segments.

3 The 1d-SAX Symbolic Representation

Our novel symbolic representation for time series is detailed in this section. We first review the main principles of the SAX method, before explaining how we propose to extend it.

3.1 SAX Representation

SAX transforms a numerical time series into a sequence of symbols taking their values in a finite alphabet. This method is very simple and does not require any *a priori* information about the input time series (apart from the distribution should be Gaussian with zero mean and unit variance). SAX representation is based on three steps:

1. Divide a time series into segments of length L
2. Compute the average of the time series on each segment
3. Quantize the average values into a symbol from an alphabet of size N

SAX makes the assumption that time series values follow a Gaussian distribution. The quantization step makes use of $(N-1)$ breakpoints that divide the area under the Gaussian distribution into N equiprobable areas. These breakpoints can be found in lookup tables. Hence, the average values computed for each segment of the time series (step 2 above) are then quantized according to the breakpoints of the Gaussian distribution. Fig. 1 shows an example of the SAX representation of a time series with $N = 4$.

3.2 1d-SAX Representation for Time Series

We detail in this section a novel symbolic representation of time series. The rationale behind this representation is the following. The SAX representation

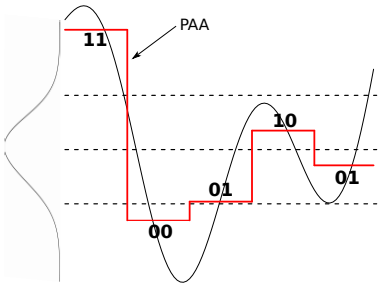

Fig. 1: Example of the SAX representation of a time series with $N = 4$. The dotted lines on the figure represent the three breakpoints of the Gaussian distribution $\mathcal{N}(0, 1)$. SAX symbols are represented by their binary values.

explained above relies only on the average value of the time series on each segment. Hence, two segments having different behaviors but with close averages will be quantized into the same symbol. For instance, a time series with an increasing trend can be mapped into the same bin as a time series with a decreasing trend if their respective means are close.
We propose here to integrate into the SAX representation an additional information about the trend of the time series on each segment. This new representation is denoted 1d-SAX in the following. It is based on three main steps :

1. Divide of the time series into segments of length L
2. Compute the linear regression of the time series on each segment
3. Quantize these regressions into a symbol from an alphabet of size N.

Hence for each segment, the linear regressions are computed and then quantized into a finite alphabet. The linear regression on each segment is computed using the least square estimation : let $V_1, \ldots, V_L$, be the values of a time series V on the time segment $T = [t_1, \ldots, t_L]$. The linear regression of V on T is the linear function $l(x) = sx + b$ that minimizes the distance between l and V on T. It is entirely described by the two values s and b. s represents the slope of l and b the value taken by l for $x = 0$. The least square optimization leads to:

$$s = \frac{\sum_{i=1}^{L}(t_i - \overline{T})V_i}{\sum_{i=1}^{L}(t_i - \overline{T})^2}, \text{ and } b = \overline{V} - s \times \overline{T}, \tag{1}$$

where $\overline{T}$ and $\overline{V}$ represent respectively the average values of V and T.
In the following, we choose to describe a linear regression of V on a time segment T by its slope value (s above), and the average value a of l on the segment. a is defined by $a = 0.5 \times s \times (t_1 + t_L) + b$. After this step (second step above), the original time series is represented by a pair (s, a) on each segment it has been divided in. We then need to quantize these pairs into an alphabet of N symbols. For that purpose, the two values are quantized separately and later combined into a symbol. Statistical properties of the linear regression ensure that both the

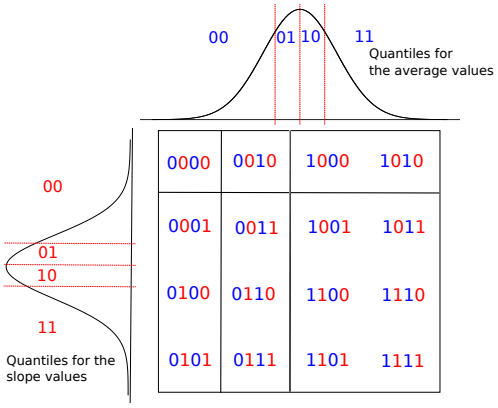

Fig. 2: Obtention of the 1d-SAX binary symbols from the quantization of both the average and the slope values of the linear regression. Here, the number of symbols is equal to 16, 4 levels are given to average values and 4 to the slope.

distribution of the average values and the slope values are Gaussian of mean 0. The variance of the average values is equal to 1, while the one of the slope values σ_L^2 is a decreasing function of L. According to these properties, quantization of the average and slope values can be done as in the SAX representation. The average values are quantized on N_a levels ($N_a < N$) according to the N_a quantiles of the Gaussian distribution $\mathcal{N}(0,1)$, while the slope values are quantized on N_s levels ($N_s < N$) according to the N_s quantiles of the Gaussian distribution $\mathcal{N}(0,\sigma_L^2)$. The choice of this parameter σ_L^2 is important. From the analysis on many time series with Gaussian distribution, $\sigma_L^2 = 0.03/L$ appears to be a good approximation. We assume here for clarity purposes that N_a and N_s are powers of two, *i.e.* $N_a = 2^{n_a}$, and $N_s = 2^{n_s}$. The quantization of the average value leads to a n_a-bit symbol, while the quantization of the slope value leads to a n_s-bit symbol. n_a is then interleaved with n_s to give a (n_a+n_s)-bit symbol, that represents the quantized value of the linear regression on $N = 2^{(n_a+n_s)}$ levels. Fig. 2 shows the obtention of $N = 16$ symbols with $N_a = N_s = 4$. $N_a - 1$ breakpoints are computed to define symbols for the average values, N_s breakpoints are computed to define the symbols for the slope values and these symbols are interleaved to get the final symbols on N levels.

The 1d-SAX representation allows for different configurations (for a same number N of levels) depending on the number of levels given to the average values and to the slope values. For instance, a symbolic representation on 64 levels can be obtained with 32 levels for the average and 2 levels for the slope, or 16 for the average and 4 for the slope, *etc.* The impact of these configurations will be discussed in Section 5.

This technique defines, for a given set of parameters (L, N_a, N_s), $N = N_a \times N_s$ binary symbols that represent linear functions on the segment $[1, \dots L]$. A symbol obtained from a segment of a time series is a binary word ω of N bits. From ω, we

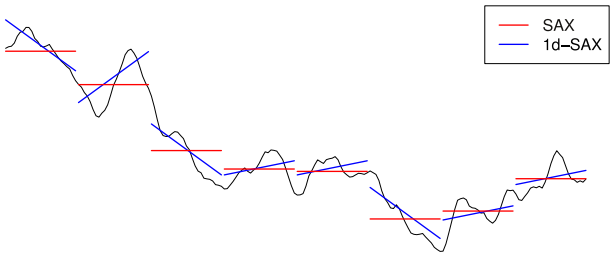

Fig. 3: A time series together with its SAX and 1d-SAX representation. The numbers of levels for the quantization here is 64. 1d-SAX uses here 16 levels for the average values and 4 for the slope values.

can extract ω_a and ω_s, the binary words representing respectively the quantized average and slope values of the linear regression of the time series on the segment. The value of ω_a indicates that the average value of the linear regression on the segment lies between two breakpoints of the $\mathcal{N}(0,1)$ distribution : β_k^a and β_{k+1}^a for instance. Similarly, the slope values of the linear regression on the segment lies between two breakpoints of the $\mathcal{N}(0,\sigma_L^2)$ distribution : β_l^s and β_{l+1}^s. We can get from these intervals an approximation of the average and slope values of the linear regression by taking the median values on each interval. These median values are also given by the quantiles of the Gaussian distribution. Following that procedure, we can obtain a numerical approximation of a time series from its quantized version (with SAX or 1d-SAX).

Fig. 3 shows an example of a time series together with its SAX and 1d-SAX representations. On this example, 64 levels have been used for the quantization, 1d-SAX uses here 16 levels for the average values and 4 for the slope values. We can see on this example that the 1d-SAX representation fits the time series more accurately than the SAX representation. This result will be highlighted in the experimental results section.

4 Asymmetric Querying for Time Series Database Search

We have applied this novel time series representation to the 1-nearest neighbor search (1-NNS) problem. The aim of this application is the following. Let us consider that we have a database D containing $\#D$ time series. Given a query q, we want to find the time series in D that is more similar to q. We assume in the rest of this paper that all the time series in the database have the same length, also equal to the length of the queries. The brute force method consists in calculating the distances between the query and all the series of the database and return the one that is most similar to q. The number of distances to compute is hence equal to $\#D$. Taking advantage of the approximate representation to speed-up the search in big database of time series is interesting in that case. SAX representation has been for instance used to index and mine terabytes of time series [10].

We define in this section an asymmetric querying method for approximate search in time series database. The term asymmetric here means that the queries are not quantized to avoid having a double quantization error (when both queries and series of the database are quantized). Performing asymmetric querying has been shown to improve the accuracy of the distance approximation for vector searches [4]. We propose a method based on this idea to perform approximate search in time series database.
Let us assume that D contains time series, as well as their symbolic representation (1d-SAX) for a given set of parameters (L, N_a, N_s). This set of parameters completely defines the $N_a \times N_s$ symbols $s_1, \ldots, s_N$ that are used to quantize the time series. The numerical approximation of these symbols can be computed as explained at the end of Section 3. The algorithm to search the 1-NN of a query q is :

1. Split q into segments of length L : $q = q_1, \ldots, q_w$
2. Compute the Euclidean distances between every segment of q and the symbols $s_j, 1 \leq j \leq N$. These distances are put in a lookup table $A = (a_{i,j})$ of dimension $w \times N$, where $a_{i,j} = ED(q_i, s_j)^2$. ED represents the Euclidean distance.
3. For every time series d in D, the quantized version of d, $\hat{d} = \hat{d_1}, \ldots, \hat{d_w}$ is available. An approximate distance $Dist_{asym}$ between q and $\hat{d}$ is obtained by

$$Dist_{asym}(q, \hat{d}) = \sum_{i=1}^{w} ED(q_i, \hat{d_j})^2 = \sum_{i=1}^{w} a_{i,s_{\hat{d}_j}}, \tag{2}$$

   which is just obtained by accessing the lookup table and summing over w elements.

After these steps, the approximate distances between q and all the time series in D are available. These distances can be used to select the approximate nearest neighbours of q. The number of elementary arithmetical operations ν_q to compute for a query search using this method is

$$\nu_q = (3L - 1) \times w \times N + (w - 1) \times \#D, \tag{3}$$

where the left part represents the cost of step 2 above and the right part the one of step 3. The number of elementary operations in the case of the brute force method is $(3Lw-1) \times \#D$. The computation cost is lower with this approximate retrieval scheme for large databases where $N \leq \#D$.

5 Experimental Evaluations

We evaluate in this section the performance of our symbolic representation of time series in terms i) of goodness-of-fit to the input data and ii) of retrieval performance when used to query a database of time series using our asymmetric scheme. We have used 7 datasets provided by the UCR Time Series Data Mining

Table 1: Average approximation error (in terms of Euclidean distance) between time series and its symbolic representation (SAX, 1d-SAX with 4 levels for the slopes and 1d-SAX with 8 levels for the slope) for $N = 256$. Results are evaluated for two different values for $L : L_1$ and L_2.

Name Dataset	$w \times L$	L_1	SAX	1d-SAX $Ns = 4$	1d-SAX $Ns = 8$	L_2	SAX	1d-SAX $Ns = 4$	1d-SAX $Ns = 8$
FaceFour	350	10	10.8253	8.1856	**7.5919**	50	17.8261	17.0815	**16.9614**
Gun-Point	150	10	2.2496	1.2082	**0.9691**	25	4.5591	2.5287	**2.1532**
Lightning7	250	10	8.3710	**7.9602**	8.0431	50	11.5377	10.3163	**10.0401**
OSULeaf	10	420	4.4926	2.0337	**1.9354**	35	11.9849	6.8624	**6.2714**
Random walks	500	10	3.9240	2.9393	**2.9345**	50	8.5576	6.3654	**5.8761**
Wafer	150	10	6.5971	5.2787	**4.7991**	25	11.7867	10.9872	**10.7888**
50Words	270	10	4.6606	**3.0908**	3.1276	30	9.9986	6.6672	**5.8782**
Yoga	420	10	2.3708	**1.2983**	1.3187	35	7.4746	3.4236	**2.7674**

archive [5] and one dataset of random walks. Each of these dataset is decomposed into a training set and a test set. The query used for time series retrieval are taken in the test sets while the train sets represent the different databases. For all the results presented in this section, we fixed $\sigma_L^2 = 0.03/L$, which turned out to be a good trade-off for all the datasets.

5.1 Quality of Representation

We first evaluate the proposed symbolic representation of time series in terms of the approximation error induced by the quantization of a time series, that we define here as the Euclidean distance between an original time series and its numerical approximation obtained with the 1d-SAX method. The lowest this distance the better the fit to the original time series. We have plotted in Fig. 4 the average approximation error versus N for the 50Words dataset and two different values of L ($L = 10$ and $L = 30$) and compared the error obtained with SAX and 1d-SAX for the same numbers of symbols used for the symbolization (*i.e.* same quantity of information). We can see on the left part of this figure that when L is small, the gain brought by the slope information begins to be significative for $N \geq 64$. The best configuration from $N = 64$ is the one with $N_s = 4$ (4 levels to quantify the slope values), while the one with $N_s = 8$ gets closer at $N = 256$. When L is higher (right part of Fig. 4), we can see that this phenomenon is amplified: the gain brought by 1d-SAX over SAX is much bigger here, even for small values of N. In addition, we can see that for this value of L, the best configuration tends to be the one with $N_s = 8$. Similar results from all the datasets are given in Table 1. In this table, N is set to 256 and two values of L have been tried for each dataset. The same conclusion can first be drawn: the gain in terms of approximation error amplifies when L increases. This result

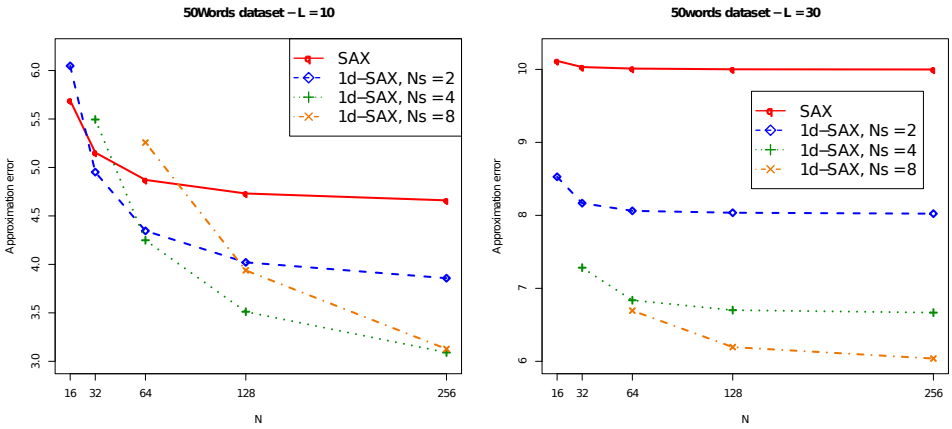

Fig. 4: Average approximation error for the 50Words dataset for two different values of L.

makes sense: representing a time series on a small segment by its average value is less restrictive than on a long segment. We can also draw another conclusion: the number of levels N_a given to the average values should be higher than the one given to the slope values N_s. This means that a balance between N_a and N_s (assuming that their product is fixed and equal to N) has to be found to optimize the performance of the 1d-SAX representation. For all the datasets that we use, the best configuration obtained for small L (less than 25), was quite always for $N_s = 2$ or $N_s = 4$, while it was quite always $N_s = 4$ or $N_s = 8$ when L was long (more than 25).

5.2 Retrieval Performance

We have exploited the property of having a symbolic representation closer to the original time series in a time series retrieval scheme. We present in this section some experimental results of the retrieval scheme presented in Section 4. The results are given in terms of the recall@R measure. This measure reflects the probability of finding the true nearest neighbor in the R sequences of the dataset that are closer to the query q in terms of the approximate distance. It is hence equal to the probability of retrieving the correct 1-NN if computing exact distance only for the best R candidates. Fig. 5 shows the recall@R performance of our scheme using 1d-SAX and SAX for the 50Words dataset and two different values of L. We can see that for small values of L, the gain brought by our symbolic representation is not significant, while this gain increases with the length L of segments. As in Section 5.1, configurations with $N_s = 2$ or $N_s = 4$ are better for small L and configurations with $N_s = 4$ or $N_s = 8$ are better for long L. Recall@1 and Recall@5 values for all the datasets considered in this paper are given in Table 2. These values are obtained for $N = 256$, and only the best configuration for 1d-SAX is given in the table for sake of conciseness.

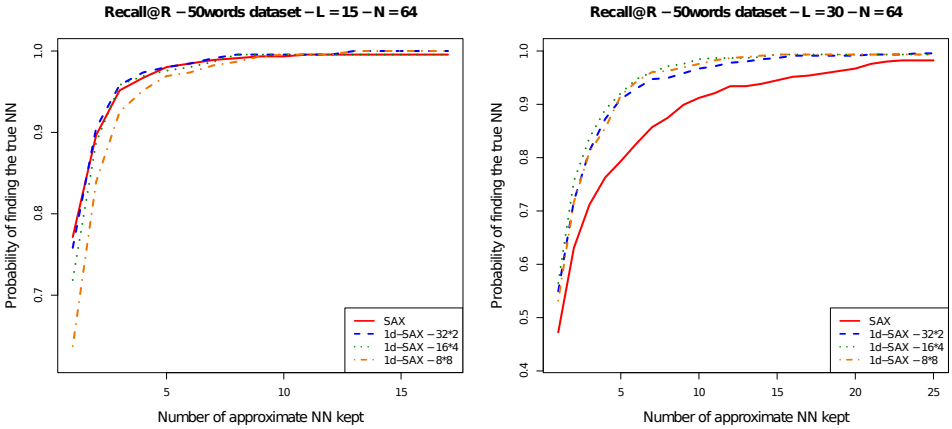

Fig. 5: Recall@R measure for the 50words dataset for $N = 64$ and two different values of L.

6 Conclusion and Discussion

We propose in this paper a novel symbolic representation for time series. This symbolic representation is based on the quantization of the linear regression of the time series on subsegments. Symbols take into account information about the average values and the slopes values of the time series. One of the main advantage of the proposed method over other representations is that the quantity of information needed to represent a time series is the same as the one needed by SAX, for a same number of symbols N. We have shown that our 1d-SAX method allows for a better fitting of the original time series : the approximation error induced by the symbolization is reduced compared to SAX. Furthermore, we have used this representation to the application of time series retrieval in databases and shown that better performance in terms of recall measure are obtained in comparison with SAX. Future improvements of this method are being thought. We are considering the possibility to extend it to multi-dimensional symbols. Furthermore, we have seen that the different configurations (N_a versus N_s) impact the performance both for the approximation measure. We are also considering the possibility to learn the most adapted configuration while doing the symbolization.

References

1. Ph. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):1–34, 2012.
2. B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser. Multivariate time series classification by combining trend-based and value-based approximations. In *Proc. of the 12th Int. Conf. on Computational Science and Its Applications (ICCSA)*, volume 7336 of *Lecture Notes in Computer Science*, pages 392–403, 2012.

Table 2: Recall@R performance of the 8 datasets considered in this paper. Two values of L are considered for each dataset. N is set to 256 here. Only the performance of the best configuration (N_a, N_s) is put here for sake of conciseness.

Name Dataset	L_1	R@1 SAX	R@1 1d-SAX	R@5 SAX	R@5 1d-SAX	L_2	R@1 SAX	R@1 1d-SAX	R@5 SAX	R@5 1d-SAX
FaceFour	10	0.784	0.795	0.989	1	50	0.454	0.454	0.795	0.818
Gun-Point	10	0.920	0.907	1	1	25	0.713	0.733	0.987	0.993
Lightning7	10	0.397	0.466	0.849	0.877	50	0.205	0.288	0.589	0.726
OSULeaf	10	0.950	0.942	1	1	35	0.479	0.735	0.884	0.996
Random walks	10	0.903	0.902	1	1	50	0.479	0.657	0.943	0.992
Wafer	10	0.653	0.627	0.920	0.919	25	0.472	0.440	0.830	0.824
50Words	10	0.863	0.859	0.998	0.998	30	0.488	0.618	0.800	0.954
Yoga	10	0.963	0.964	1	1	35	0.825	0.834	0.995	0.997

3. N.Q.V. Hung and D.T. Anh. Combining sax and piecewise linear approximation to improve similarity search on financial time series. In *Proc. of the Int. Symp. on Information Technology Convergence (ISITC)*, pages 58–62, 2007.
4. H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.
5. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C.A. Ratanamahatana. The UCR times series classification/clustering homepage. 2011.
6. G. Li, L. Zhang, and L. Yang. Tsx: A novel symbolic representation for financial time series. In *Proc. of the 12th Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI)*, volume 7458 of *LNCS*, pages 262–273, 2012.
7. J. Lin, E.J. Keogh, S. Lonardi, and B.Y. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD workshop on Research issues in Data Mining and Knowledge Discovery*, pages 2–11, 2003.
8. B. Lkhagva, Y. Suzuki, and K. Kawagoe. New time series data representation esax for financial applications. In *Proc. of the 22nd Int. Conf. on Data Engineering Workshops*, pages 17–22, 2006.
9. N. D. Pham, Q. L. Le, and T. K. Dang. Two novel adaptive symbolic representations for similarity search in time series databases. In *Proc. of the 12th Asia-Pacific Web Conference (APWeb)*, pages 181–187, 2010.
10. J. Shieh and E. Keogh. iSAX: Indexing and mining terabyte sized time series. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2008.
11. W. Zalewski, F. Silva, H. D. Lee, A. G. Maletzke, and F. C. Wu. Time series discretization based on the approximation of the local slope information. In *Proc. of the 13th Ibero-American Conference on AI (IBERAMIA)*, volume 7637 of *LNCS*, pages 91–100, 2012.