Symbolic representation for time series

Sylvain W. Combettes, Charles Truong, and Laurent Oudre

Université Paris-Saclay, Université Paris Cité, ENS Paris-Saclay, CNRS, SSA, INSERM, Centre Borelli

Introduction

Why use symbolic representations of time series?

- Need for an actionable representation that takes into account the temporal information.
- Used in many data mining tasks: classification, clustering, indexing, anomaly detection, etc.
- 2 main advantages over other representations:
  - Reduced memory usage.
  - Often a better score on data mining tasks thanks to the smoothing effect induced by compression.

2 main steps for symbolic representations

1. Segmentation step: a real-valued signal of length n is split into w segments ($w < n$).
2. Quantization step: each segment is mapped to a discrete value taken from a set of A symbols. Example of set of symbols with $A = 5$: $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$.

Related work

Table 1: Summary of some popular symbolic representations.

Method	Segmentation	Feature extraction	Quantization
SAX [2] (2003)	uniform	mean	Gaussian bins
1d-SAX (2013)	uniform	mean, slope	Gaussian bins
CSAX (2020)	uniform	mean, complexity estimate	Gaussian bins

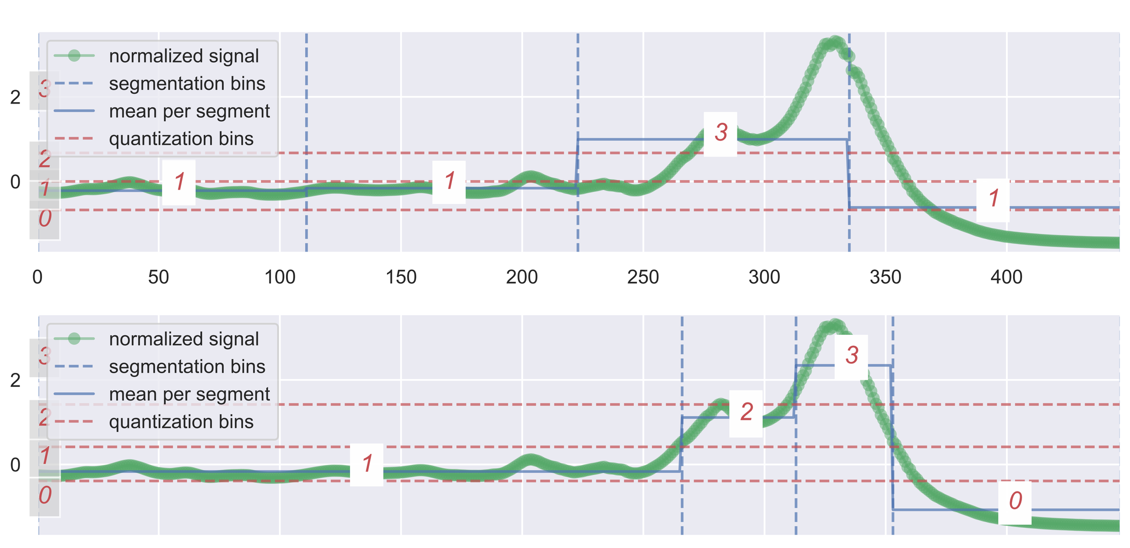

Figure 1: Example of a SAX (top) and our method ASTRIDE (bottom) representations of a signal. The resulting symbolic sequence is `1131` for SAX, and `1230` for ASTRIDE. SAX can not take into account the peaks.

Our method: ASTRIDE

ASTRIDE (Adaptive Symbolization for Time seRIes DatabasEs): adaptive symbolic representation for a data set of N univariate time series of length n, with a compatible distance measure.

Steps of ASTRIDE

1. Segmentation: change-point detection (on the mean) with a fixed number of change-points ($w - 1$), where w is the desired number of segments.
2. Quantization: quantiles, leading to A bins.
3. Distance: general edit distance between the resulting symbolic signals.

Change-point detection

- All N signals are stacked, producing a single multivariate signal of length n and dimension N.
- ASTRIDE applies multivariate change-points detection with a fixed number of segments (w) on this high-dimensional signal.
- Finding the $w - 1$ instants $t_1^* < t_2^* < \ldots < t_{w-1}^*$ where the mean of signal $y = (y_1, \ldots, y_n)$ change abruptly:

$$(\hat{t}_1, \ldots, \hat{t}_{w-1}) = \underset{(w, t_1, \ldots, t_{w-1})}{\arg\min} \sum_{k=0}^{w+1} \sum_{t=t_k}^{t_{k+1}-1} \| y_t - \bar{y}_{t_k:t_{k+1}} \|^2,$$

  where $\bar{y}_{t_k:t_{k+1}}$ is the empirical mean of $\{y_{t_k}, \ldots, y_{t_{k+1}-1}\}$.
- Reducing the error between the original signal and the best piecewise constant approximation.
- Solved using dynamic programming. Time complexity: $\mathcal{O}(Nwn^2)$.

Levering the general edit distance

1. Preprocessing.
   - Including the segment length information: replicating each symbol proportionally to its segment length. Example: `abd` becomes `aabbbbdd`.
   - Shortening: dividing each length by the minimum length. Example: `aabbbbdd` becomes `abbd`.
2. Applying the general edit distance with custom costs.
   - Edit distance on strings (a.k.a Levenshtein distance): minimal cost of a sequence of operations that transform a string into another.
   - Allowed simple operations and their costs:
     - Substitution: Euclidean distance between the average of all the means corresponding to each symbol.
     - Insertion: max of substitution costs.
     - Deletion: max of substitution costs.
   - Total cost: sum of the costs of the simple operations.

Experimental results (I)

- ASTRIDE is compared to SAX, 1d-SAX, and CSAX on One-Nearest Neighbor (1-NN) classification, with the test accurary, for $A = 9$.
- Evaluated on 86 univariate time series data sets with equal length sourced from the UCR Time Series Classification Archive.

Table 2: Normalized space complexities (nsc) for each symbolization method, with $r = 64$ bits the number of bits to store a real value.

Method	Normalized space complexity
SAX	$\frac{w \lceil \log_2(A) \rceil}{n}$
1d-SAX	$\frac{w \lceil \log_2(A) \rceil}{n}$
CSAX	$\frac{w(\lceil \log_2(A) \rceil + r)}{n}$
ASTRIDE	$\frac{w(N \lceil \log_2(A) \rceil + r)}{Nn}$

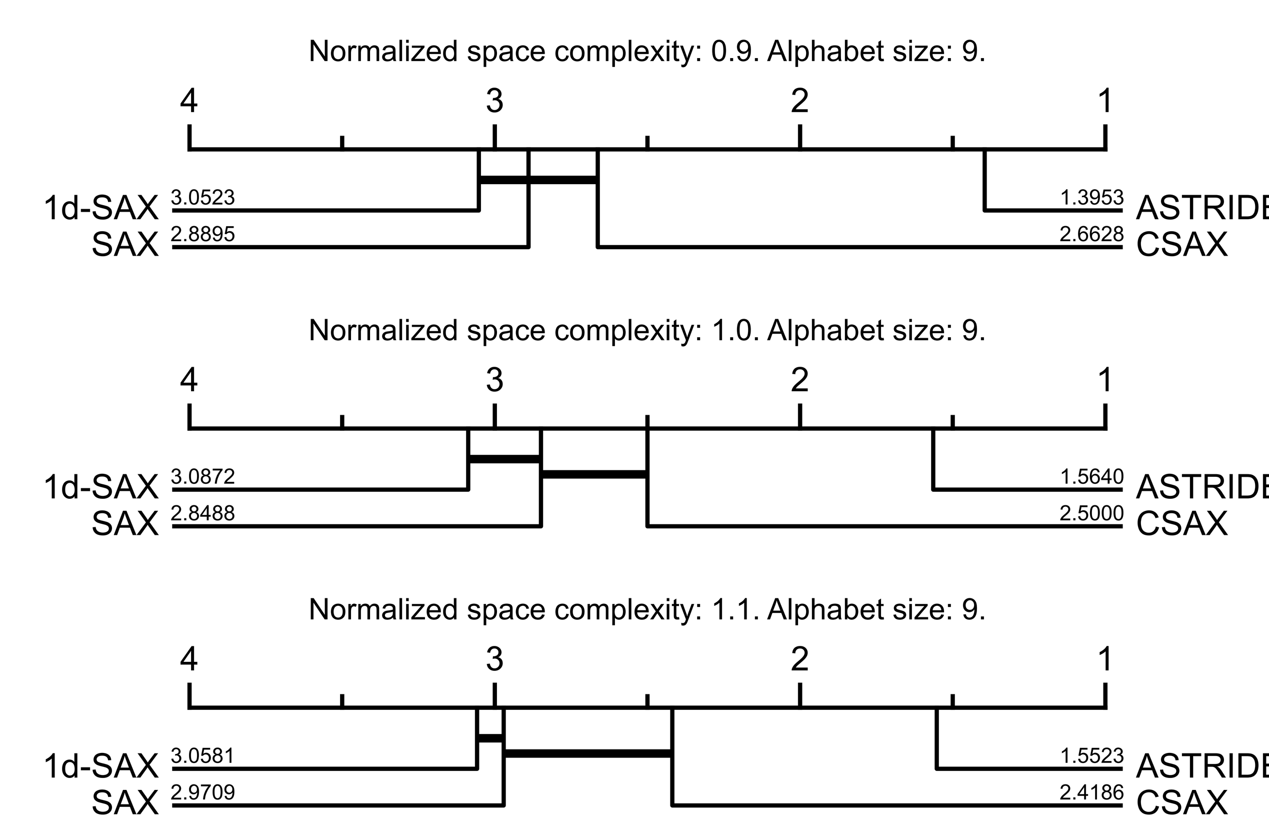

Figure 2: Critical difference diagrams showing the pairwise statistical difference comparison. ASTRIDE is the best symbolization on average over the considered datasets.

Experimental results (II)

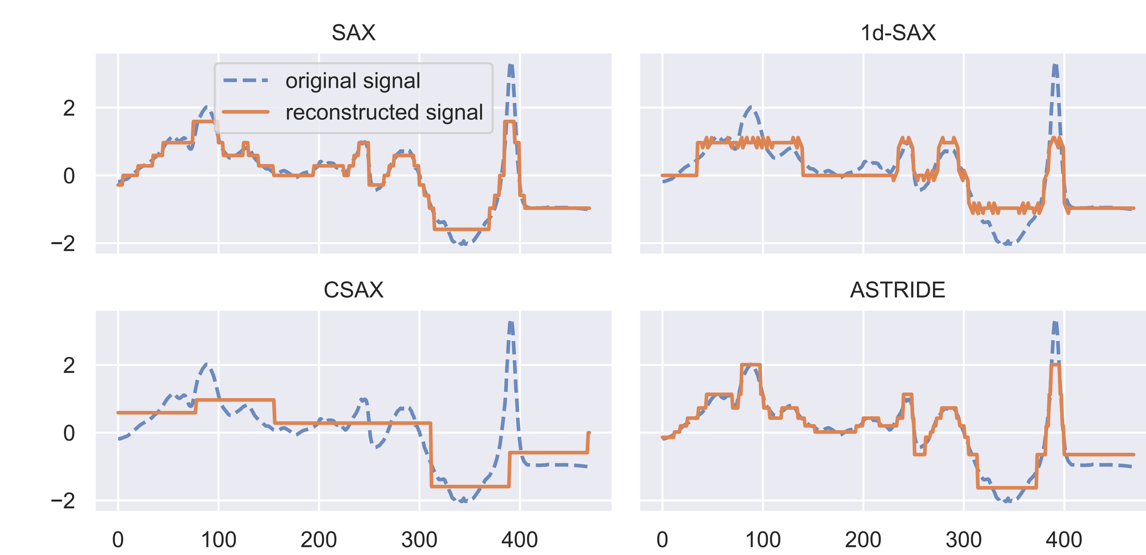

Figure 3: Example of symbolization of a single signal from the Beef data set (UCR archive) of length $n = 470$ for several methods, with $A = 9$ and $nsc = 0.8$.

Table 3: Processing times on the symbolization and 1-NN classification on the ECG200 data set composed of 100 training signals and 100 test signals of length $n = 96$, with $w = 10$ and $A = 9$.

Method	Symbolization (s)	1-NN classification (s)
SAX	0.02	0.11
1d-SAX	0.41	0.21
CSAX	0.58	0.25
ASTRIDE	0.29	0.17

Conclusion

Follow-up paper on adaptive symbolic for a dataset of multivariate time series: d_{symb} method and the d_{symb} playground [1] (Streamlit app).

References

[1] S. W. Combettes, P. Boniol, C. Truong, and L. Oudre. dsymb playground: An interactive tool to explore large multivariate time series datasets. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 2024.

[2] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. 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*, 2003.

sylvain.combettes8@gmail.com

https://sylvaincom.github.io