Anomaly Detection for Discrete Sequences: A Survey

Guillaume Jarry, Mouad Id Sougou

Context and Overview

- Survey article, many algorithms presented in various field related to anomaly detection in discrete sequence, theoretical unification necessary

=> GitHub Library (Section **IV** of the article), started from scratch

https://github.com/JarrryGuillaume/DAD_library.git

Implementation

Techniques implemented in the library

- SAX algorithm
- Kernel Based Methods
- Window Based Methods
- Markov Based Method

Tested and benchmarked on :

- Real Continuous Data : Dodger Dataset
- Synthetic Markov Chains

Problem Definition

- A discrete sequence is an ordered set of symbols where the symbols belong to a finite alphabet. Ex : [“A”, “C”, “D”, “A”, “A”, “B”]

- Our goal is given a dataset of n discrete sequences **S** = {S1, S2, . . ., Sn} containing **only normal ones**, infer an anomaly score of a test sequence Sq

SAX Method :

- Useful to discretize continuous time series into symbolic sequence :
- First **Z-normalize the data**, compute breakpoin $\beta_1 < \beta_2 < \cdots < \beta_{\alpha-1}$

$$\mathbb{P}(Z < \beta_1) = \frac{1}{\alpha} \quad \mathbb{P}(Z < \beta_2) = \frac{2}{\alpha} \quad \ldots \quad \mathbb{P}(Z < \beta_{\alpha-1}) = \frac{\alpha - 1}{\alpha}$$

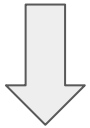
= > Then cut the series into segments of equal length and for each assign :

$$s_a = \begin{cases} \text{symbol}_1 & \text{if } \overline{X}_a < \beta_1 \\ \text{symbol}_2 & \text{if } \beta_1 \leq \overline{X}_a < \beta_2 \\ \vdots & \\ \text{symbol}_M & \text{if } \overline{X}_a \geq \beta_{M-1} \end{cases}$$

Kernel Based Techniques :

- Uses a kernel to compute a score between two sequence. From the article, nLCS (longest common sequence) :

$$\mathbf{nLCS}(S_i, S_j) = \frac{|\mathrm{LCS}(S_i, S_j)|}{\sqrt{|S_i||S_j|}}$$



Medoids	K-Nearest
Compute similarity matrix then medoids in the train set	Asses the test sequence directly on the train set and return kth score

Markov Based Techniques :

- **Fixed Markovian techniques** uses a **fixed-length** history k to estimate the conditional probability of a symbol sq

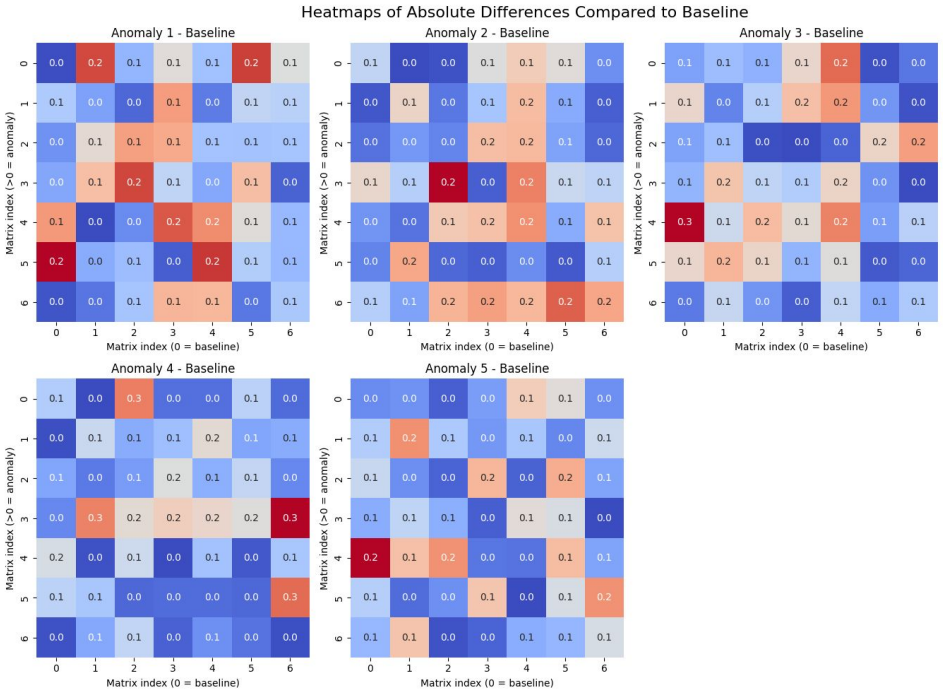
- **Variable Markovian techniques** addresses the limitations of the fixed approach by allowing the history **size to vary** using Probabilistic Suffix Tree (PST)

- **Sparse Markovian techniques** allows using a **sparse history** instead of immediately preceding symbols

Fixed and Variable Method Benchmark
Synthetic data generation :

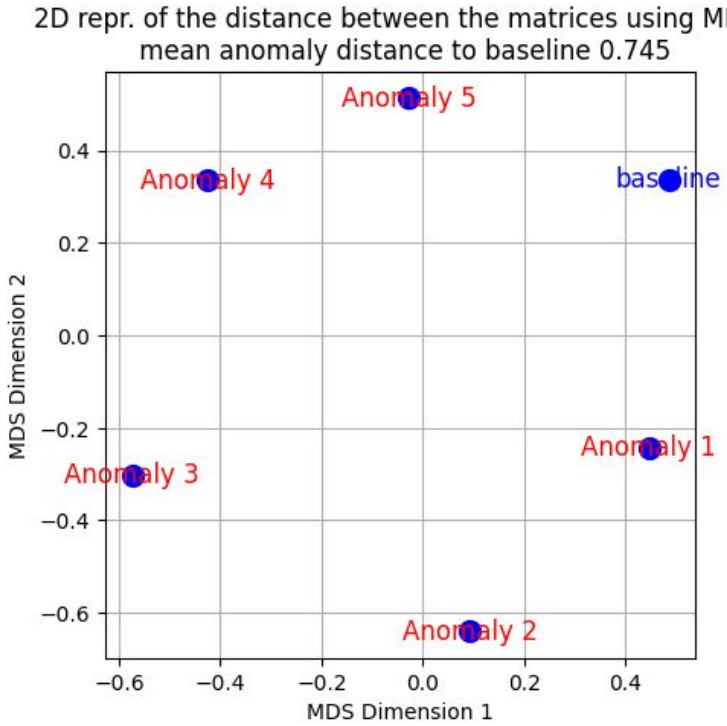
- **1 train set :** 1000 sequence of length 100 generated by a random transition matrix (baseline)
- **1 test set :**
  - 500 sequence generated from baseline
  - 500 sequence generated by 5 other transition matrices, called anomaly matrix (100 each)

Data : Synthetic generation for Markov Benchmark



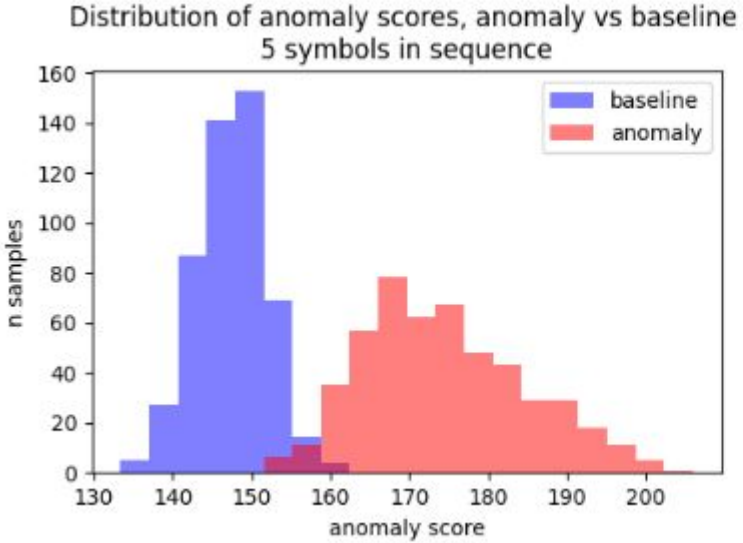
=> Ensure the process are “different” enough : Distance matrix

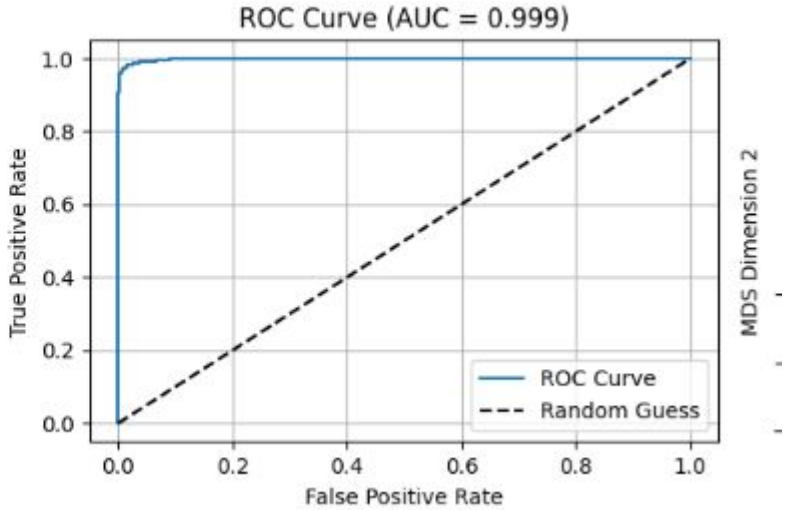
Data : Synthetic generation for Markov Benchmark



Generate distance matrix of the transition matrix and plot them in 2 dimension !

Benchmark Results ! Fixed and Variable Method





AUC Score => Works for every possible threshold of our classifier.

Markov Benchmark, global results !

Method used	Size of transition matrix	AUC SCORE	Baseline-Anomaly Matrix distance
Variable Based	2	0.63	0.417
Variable Based	3	0.84	0.600
Variable Based	5	1.00	0.632
Variable Based	7	1.00	0.782
Fixed Based	2	0.63	0.417
Fixed Based	3	0.84	0.600
Fixed Based	5	0.99	0.632
Fixed Based	7	0.99	0.782

Variable Markov and Fixed Markov did perform exactly the same on this dataset

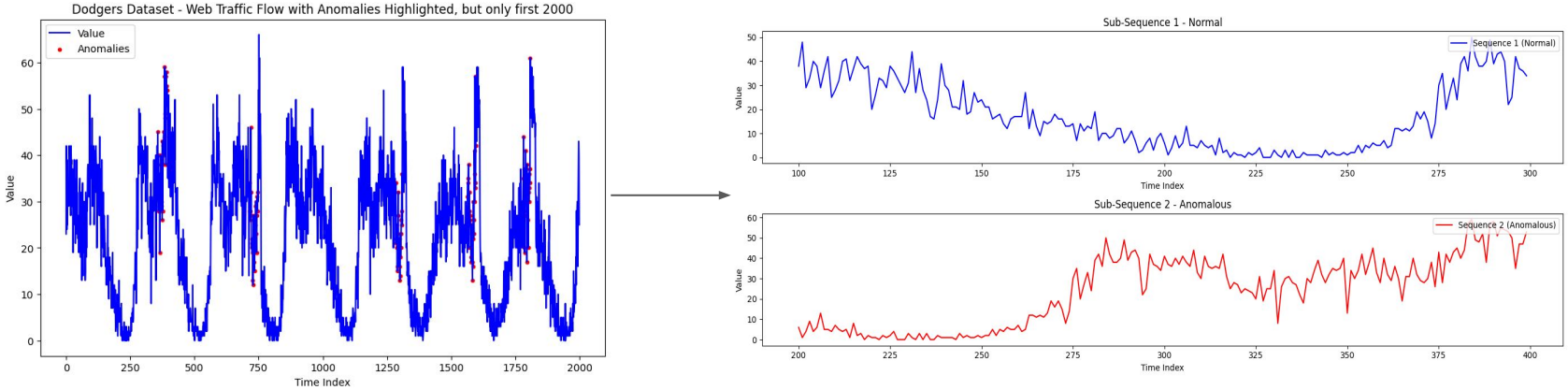
=> Performance improve as number of symbol increase

Markov Benchmark results

Method used	Size of transition matrix	AUC SCORE	Baseline-Anomaly Matrix distance
Fixed Based	3	0.848	0.696
Fixed Based	5	0.943	0.682
Variable Based	3	0.834	0.696
Variable Based	5	1.00	0.682

Different results in different dataset, similar performance

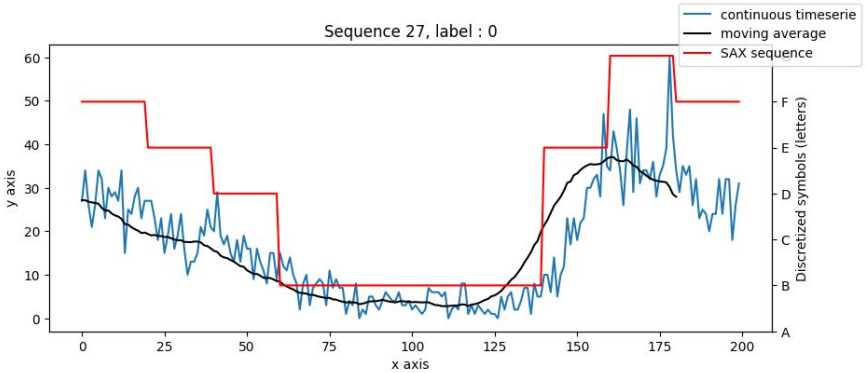
Real Data : Dodgers Dataset Pre Processing



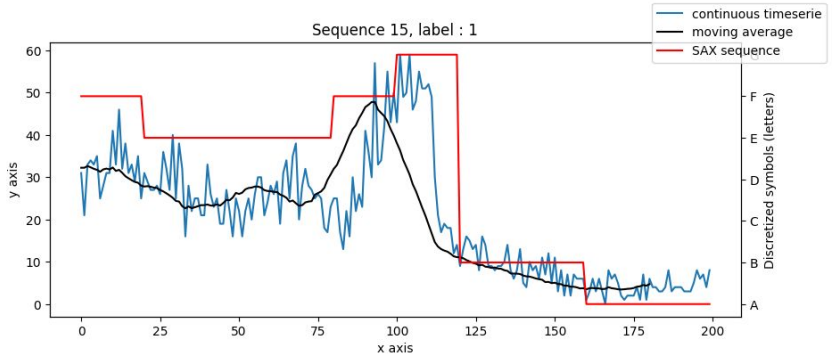
Represents the number of visitors of a website with unusual spikes

Before applying SAX, go from a unique annotated signal to multiple labeled sequences

SAX discretization on the dataset



Normal sequence



Anomalous sequence

Uses a local normalization that adapts to every sequence
Word size of 10 and an alphabet size of 7

SAX impact on the initial dataset : Still separable ?

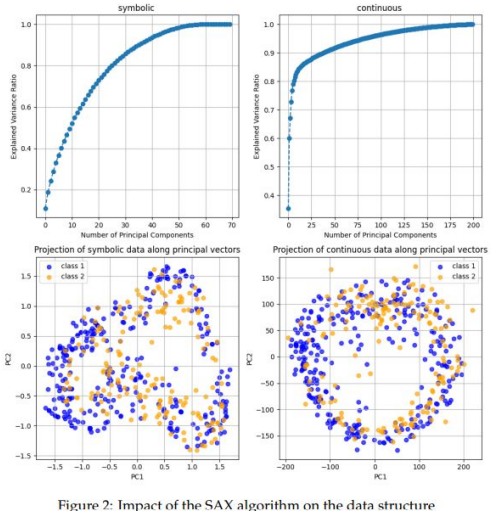
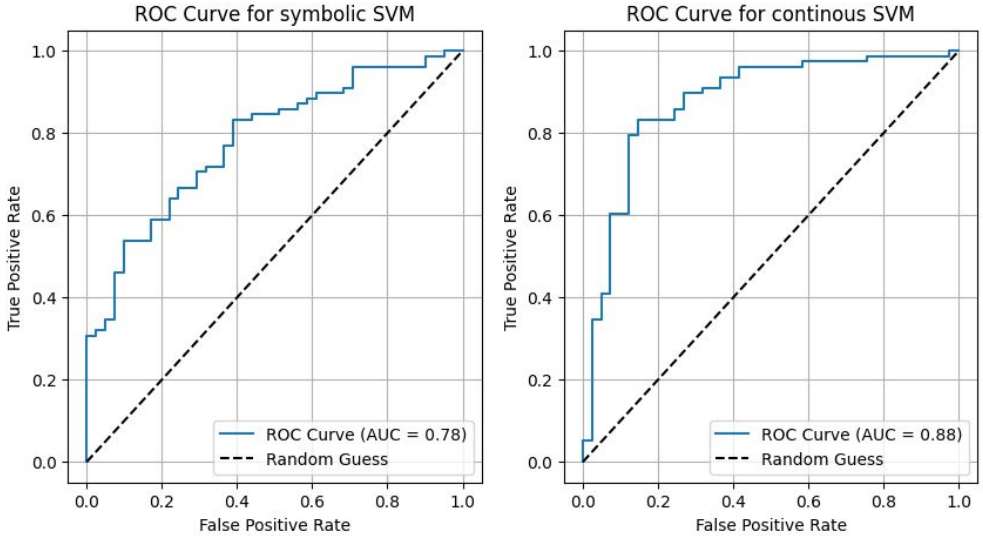


Figure 2: Impact of the SAX algorithm on the data structure



The discretization makes it harder to separate between the two classes.

Important Loss of information contained in the signal after discretization and decorrelation of data

Kernel and Window Based Results

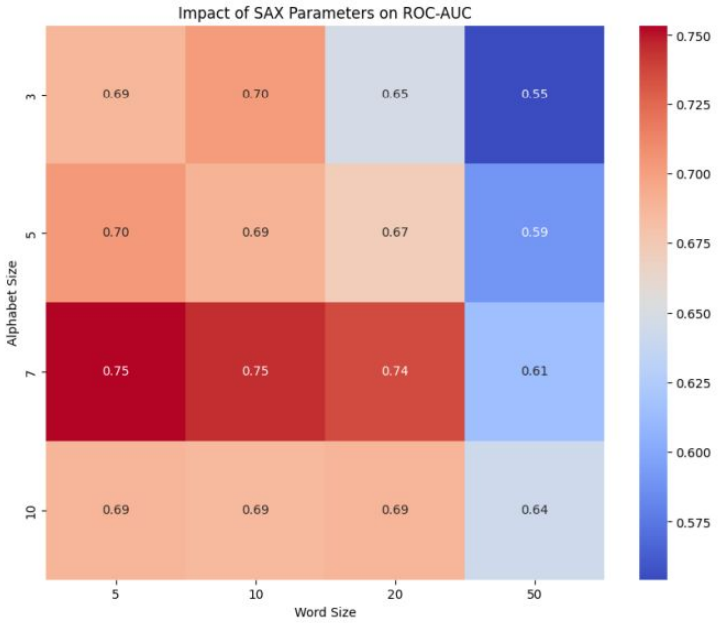
Method	KNN Kernel	Medoids Kernel	Lookahead	Normal Dict	Unsupervised SVM
AUC	0.74	0.67	0.40	0.45	0.42

Table 2: AUC Scores for Kernel-Based and Window-Based Techniques

Kernel-based techniques outperform window-based methods

SAX + Kernel is probably equivalent to looking at significant variation to a trend in the continuous domain.

Impact of SAX parameters on results



Higher alphabet sizes means higher chances of detecting changes in the signal meaning anomalies (Y dimension)

Needs to be balanced with word size as they represent a sliding window and smooths the signal (X dimension)

Let’s contribute !

On aimerait vraiment faire une librairie que d’autre gens puissent utiliser et qui soit d’un bon standard de niveau de code, donc n’hésitez pas à nous dire si cela vous intéresse.

Lien GitHub :
https://github.com/JarrryGuillaume/DAD_library