Implementing a gap-invariant distance measure in ClaSP

Masters Expose
WiSe 2023/2024

Authors

Julian Muders
julian.muders@student.hu-berlin.de

Supervisors

Dr. Patrick Schäfer

Arik Ermshaus

Presented on February 20, 2024

1 Introduction

In modern days, data collection is an ubiquitous task. There is an abundance of data sources such as sensors for weather, health data, IT monitoring metrics, or stock prices. This showcases the comprehensive spread of data across a wide range of domains. The collected data is real-valued and temporally ordered. Such structured data is also called *time series*. This vast pool of information provides rich material for analysis, either by humans or by machines, which can yield highly beneficial insights. Contradictory to the ease with which humans are able to analyse and classify data, classifying data and detecting semantic changes in its behaviour can be a hard task for machines.

The challenge lies among others in the associated cost of data analysis [Yeh+16]. Due to the large quantity of data it is challenging to decide which sections of the data should be analysed if the cost of analysing the whole data set exceeds the amount of available resources. Here, the field of time series analysis presents a specific research subarea called *time series segmentation*. This research focuses on dividing a *time series* into sets of segments that hold homogeneous statistical attributes within themselves yet remain heterogeneous when compared with each other, where between two sections a so called *change point* is allocated. Interestingly, though the homogeneous sections exhibit consistent behaviour which requires specific domain knowledge to interpret, the detection of changes that occur between different segments can be more agnostic of their domain [Gha+17] and warrant the creation of domain-independent algorithms like FLOSS [Gha+17] and AutoPlait [MSF14]. Segmentation of time series can be harnessed to provide alerts or updates regarding a change in the state of the system being observed. Additionally, the information of a detected change point could prompt the allocation of further resources for analysing a particular subset. Therefore, change point detection has an expansive array of potential applications.

Within the research area of change point detection, the concept of the *Classification Score Profile* (ClaSP) has been introduced in [SEL21]. This profile describes a transformation on an input time series which allows the reduction of the change point detection problem to finding peaks in the profile. In order to compute the classification score profile, an input time series is split into windows which are then repeatedly assigned to either the left or the right side of a hypothetical split in the time series. In order to reduce these to a profile, a set of *k-Nearest-Neighbour* (k-NN) classifiers will be trained on each of these hypothetical split points and evaluated in a cross-validation setting. The aggregated score of these classifiers then denotes the ClaSP, which indicates the most likely change point in the time series by its global maximum. Usually an Euclidean distance metric is used by the k-NN classifiers to assess how similar any two segments are to each other, which prompts an evaluation of the impact of different distance metrics on the accuracy of ClaSP.

The proposed research aims to address this topic by implementing an algorithm for a different distance metric, the prefix-suffix distance, which was

proposed in [IK19], in order to compare the accuracy of ClaSP when using this new metric as compared to the previously used Euclidean distance metric.

Compared to the Euclidean distance, the prefix-suffix distance metric is gap invariant, it allows for a region between matching regions of the subsequences which is not compared between the two. This enables matching of subsequences of varying lengths and implies that prefix-suffix distance allows for more permissive matching of similar regions which exhibit behaviours of differing lengths.

This expose describes the goal and evaluation criteria for a masters thesis. As such the next section describes the definitions required to convey the proposed research, followed by a section to describe the research goal itself. This is followed by a final section which describes the planned evaluation of the proposed research.

2 Definitions and Notation

This section focuses on the definitions and notations of expressions and concepts which are used throughout this expose, using [SEL21] and [IK19] for reference. First, we will introduce the definition for a Time Series.

Definition 1 (Time Series). A time series (TS) T is a sequence of $n \in \mathbb{N}$ real values, $T = (t_1, \ldots, t_n), t_i \in \mathbb{R}$. The values are also called *data points*.

Using this, we can introduce the definition of a subsequence of a time series.

Definition 2 (Subsequence). Given a TS T of length n, a subsequence $T_{s,e}$ of T with start offset s and end offset e consists of the contiguous values of T from position s to position e, i.e. $T_{s,e} = (t_s, \ldots, t_e)$ with $1 \leq s \leq e \leq n$. The length of $T_{s,e}$ is $|T_{s,e}| = e - s + 1$.

These preliminary definitions allow us to define a segmentation of a time series as well as the general problem of finding a meaningful segmentation. Note that detecting a single change point in a time series will create a meaningful segmenation into two segments for this time series.

Definition 3 (Segmentation). A segmentation of a TS T into $C + 1$ segments is an ordered sequence of change points (or splits) $t_{i_1}, \ldots, t_{i_C}$ with $1 < i_1 < \cdots < i_C < n$.

Definition 4 (Time Series Segmentation). The problem of *time series segmentation* (TSS) is to find a meaningful segmentation of a given TS T under the assumption that T was generated by a process with discrete states. A segmentation is considered meaningful when the change points between two subsequent segments correspond to state changes in the underlying process.

In order to solve the problem of finding a meaningful segmentation of a time series into and devise two segments, ClaSP relies on the Distance Profile of a time series as part of its k-NN classification step.

Definition 5 (Distance Profile). A *distance profile* $D \in \mathbb{R}^{n-m+1}$ of a time series T and a given query $T_{i,i+m}$ is a vector which stores $dist(T_{i,i+m}, T_{j,j+m}) \forall j \in [1, 2, \ldots, n-m+1]$ for any distance function $dist$.

The Distance Profile relies on a distance function, of which we will define two. Firstly, the Euclidean distance is currently being used by state-of-the-art ClaSP implementations. Scondly, the prefix-suffix distance which will be implemented as part of this thesis.

Definition 6 (Euclidean distance). The *Euclidean distance* of two subsequences of equal length $T_{a,b}$ and $T_{x,y}$ of a time series T is defined as
$dist_e(T_{a,b}, T_{x,y}) = \sqrt[2]{(t_a - t_x)^2 + (t_{a+1} - t_{x+1})^2 + \cdots + (t_b - t_y)^2}$
with $b - a = y - x = m$

Definition 7 (Prefix-Suffix-Distance). The *prefix-suffix distance* of two subsequences $T_{a,b}$ and $T_{x,y}$ of a time series T, given a *prefix-/suffix length* s and a maximum *don't-care length* r, with $a \neq x$, $b \neq y$, is defined as
$dist_{ps}(T_{a,b}, T_{x,y}) = min\left(dist_e(T_{a,a+s}, T_{x,x+s}) + dist_e(T_{b-s,b}, T_{y-s,y})\right)$ with

$$0 \leq b - a - 2s \leq r$$
$$0 \leq y - x - 2s \leq r$$

where $dist_e(T_{a,a+s}, T_{x,x+s})$ is the distance between the *prefixes* of $T_{a,b}$ and $T_{x,y}$ and $dist_e(T_{b-s,b}, T_{y-s,y})$ is the distance between the *suffixes* of $T_{a,b}$ and $T_{x,y}$ and the regions $T_{a+s,b-s}$ and $T_{x+s,y-s}$ denoting the *don't care regions* of $T_{a,b}$ and $T_{x,y}$.

Note that the subsequences $T_{a,b}$ and $T_{x,y}$ are not required to be of the same length in order to compute their prefix-suffix distance as opposed to the Euclidean distance. Fig. 1 displays two subsequences of a time series with different lengths, which are coloured by prefix, suffix and don't-care regions. This presents a pair of subsequences which are highly similar when compared with the prefix-suffix distance, as they exhibit almost identical prefixes and suffixes with only the length of the don't care regions differing.

Definition 8 (Classification Score Profile). Given a TS T and a window-length w, a *Classification Score Profile* (ClaSP) is a real-valued sequence S of length

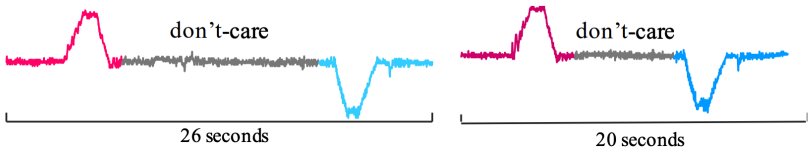

Fig. 1: Two subsequences of a time series measured by [Par+10], which depict the X-axis acceleration of a person using an elevator, Figure taken from [IK19]

n. The i-th value in S is the cross-validation score $s \in [0, 1]$ of a classifier C trained on a binary classification problem with labels $Y = \{0, 1\}$. For index $i \in [w+1, n-w-1]$ training samples are created by assigning label $y = 0$ to all windows to the left $W_L = \bigcup_{j \in [1,\dots,i-w]} T_{j,j+w}$ and $y = 1$ to all windows to the right $W_R = \bigcup_{j \in [i-w+1,\dots,n-w+1]} T_{j,j+w}$. Values $S[1, \dots, w]$ and $S[n-w-1, \dots, n]$ are set to 0, giving a very small blind spot.

Using these definitions we can now lay out the proposed research as well as its evaluation criteria in the following sections.

3 Research Goal

In order to compare the accuracy of the ClaSP algorithm using the two different distance metrics, Euclidean distance and prefix-suffix distance, the focus of this research will be to implement the algorithm for prefix-suffix distance efficiently into the *aeon-Toolkit* [dev]. This toolkit already provides an implementation of the ClaSP algorithm as well as a suitable implementation to calculate the z-normalized Euclidean distance, which will allow a for direct comparison of the two approaches. Since the original proposal for prefix-suffix distance only accounts for use with a 1-NN classifier [IK19], which stands in contrast to the optimal configuration with a 3-NN classifier for ClaSP [SEL21], the proposed research presents two steps of implementation:

1. Implementing the prefix-suffix distance algorithm for use with a 1-NN classifier as described in [IK19]
2. Adapting this implementation for usage with k-NN classifiers, particularly a 3-NN classifier, in an efficient manner if feasible

The first step will facilitate an implementation of the prefix-suffix distance algorithm as it is laid out by the original authors. Once this implementation is in place it needs to be adapted to fit the necessary adapters of the existing ClaSP algorithm. Furthermore the existing ClaSP algorithm needs to be adjusted to accept a 1-NN classifier instead of the current 3-NN classifier. This will allow a first comparison of ClaSP with Euclidean distance to ClaSP with the prefix-suffix distance in terms of accuracy and computational efficiency. The exact evaluation criteria will be outlined in Sec. 4.

Once the implementation of the first step is successfully completed, the algorithm for calculating the prefix-suffix distances needs to be extended to output the required information to serve as input to 3-NN classifiers. The main focus of this task lies on keeping the computational complexity of the implementation low. Upon completion of the implementation the evaluation will span a comparison between ClaSP with a 3-NN classifier using the Euclidean distance metrics and ClaSP with a 3-NN classifier using the prefix-suffix distance metric. This evaluation will again consider the accuracy of the model as well as the computational efficiency. Furthermore a comparison in terms of accuracy and computational efficiency to ClaSP with a 1-NN classifier using the prefix-suffix distance will be possible.

4 Evaluation

In order to evaluate the accuracy of ClaSP with the prefix-suffix distance, we will apply the same set of time series as used in [SEL21] using the prefix-suffix distance and a 1-NN classifier for the first described goal. To achieve comparable results, we will apply the ClaSP algorithm with the z-normalized Euclidean distance metric and a 1-NN classifier to the same 98 data sets.

To evaluate the effectiveness of the k-NN adapted prefix-suffix distance metric, which will be implemented as defined in the second goal, we will reuse the results from the [SEL21] paper using the Euclidean distance metric and a 3-NN classifier. This will allow us to compare the results directly with an analysis using the adapted prefix-suffix distance and a 3-NN classifier as well as facilitate a comparison between the prefix-suffix distance metric with two different classifiers, 1-NN and 3-NN. Based on the evaluation results, ClaSP using the prefix-suffix distance metric can be compared to FLOSS, Window-L_2, BOCD, BinSeg-L_2, and Autoplait as seen in [SEL21] for a more comprehensive analysis against some state of the art competitors. All algorithms will use the same evaluation metric as it is used in [SEL21] and defined in [Gha+17], which sums the distances between annotated change points and predicted change points and then normalizes the value to a range of $[0, 1]$, with 0 representing the best possible score.

When assessing the effectiveness of the prefix-suffix distance, the evaluation should take into account that this metric might surpass Euclidean distance only within certain group of data sets. Regardless of the comparative performance of ClaSP when using either of these two metrics, the proposed research is set to include an analysis of specific data sets wherein one metric demonstrates superior performance over the other.

Moreover, it is important to evaluate different values for the new input parameter *don't-care-length* in the evaluation of prefix-suffix distance. To procure the most optimal results for ClaSP with the prefix-suffix distance, this parameter should ideally be chosen as a small multiple of the prefix-/suffix length [IK19]. To determine this parameter's optimal value, a strategy similar to the window size selection process as outlined in [SEL21] should be used.

References

[dev] The aeon developers. https://www.aeon-toolkit.org/. Accessed: 2024-02-10.

[Gha+17] Shaghayegh Gharghabi et al. "Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels". In: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017, pp. 117–126. DOI: 10.1109/ICDM.2017.21.

[IK19] Shima Imani and Eamonn Keogh. "Matrix Profile XIX: Time Series Semantic Motifs: A New Primitive for Finding Higher-Level Structure in Time Series". In: *2019 IEEE International Conference on Data Mining (ICDM)*. 2019, pp. 329–338. DOI: 10.1109/ICDM.2019.00043.

[MSF14] Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. "AutoPlait: automatic mining of co-evolving time sequences". In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. SIGMOD '14. Snowbird, Utah, USA: Association for Computing Machinery, 2014, pp. 193–204. ISBN: 9781450323765. DOI: 10.1145/2588555.2588556. URL: https://doi.org/10.1145/2588555.2588556.

[Par+10] Avinash Parnandi et al. "Coarse In-Building Localization with Smartphones". In: Jan. 2010, pp. 343–354. ISBN: 978-3-642-12606-2. DOI: 10.1007/978-3-642-12607-9_25.

[SEL21] Patrick Schäfer, Arik Ermshaus, and Ulf Leser. "ClaSP - Time Series Segmentation". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM '21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 1578–1587. ISBN: 9781450384469. DOI: 10.1145/3459637.3482240. URL: https://doi.org/10.1145/3459637.3482240.

[Yeh+16] Chin-Chia Michael Yeh et al. "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 2016, pp. 1317–1322. DOI: 10.1109/ICDM.2016.0179.