Evaluating Dilated Time Series Segmentation in ClaSP

Study Project Exposé

Anika Ilieva

Department of Computer Science, Humboldt-Universität zu Berlin
Supervisor: Dr. rer. nat. Patrick Schäfer
November 1, 2023

1 Introduction

Low-cost, high-resolution sensors are broadly used in various industry fields. Medical surveillance, earth observation, manufacturing and security control are only a few of the common sensor applications. In these fields, sensing devices are used to detect events or changes in the measurements. The produced data is an unlabelled series of data points indexed in time order, called *time series (TS)*. Formally expressed, a time series T is a sequence of $n \in \mathbb{N}$ real values, $T = (t_1, ..., t_n), t_i \in \mathbb{R}$ [SEL21]. As sensors produce vast amounts of time series, the interest in analysing the temporally ordered data has peaked. For instance, TS analysis includes tasks, such as classification [Ism+19; DPW20], similarity search [SH12], motif discovery [Mue+09; SL23a], anomaly detection [Blá+21] and segmentation [Gha+17; SEL21].

Time series segmentation (TSS) is a particularly crucial part of TS exploration. As described in [ESL23], TSS methods aim to identify state changes in real-world processes by distinguishing changes in the statistical distribution or shape of the measured values in a post hoc manner. During a segmentation, an input TS is divided into a sequence of discrete segments that are (semantically) dissimilar to neighbouring segments to reveal the underlying system properties. In this way, happenings of interest, usually causing a state change in a system, can be detected promptly. The task of detecting such signal shifts is called *change point detection (CPD)* [TOV19], while a segmentation represents an ordered sequence of change points. As monitoring data streams for important or unexpected events is crucial for all sensor application fields, various methods for TSS have been proposed, which are briefly outlined in the following subsection.

1.1 State-of-the-art TS Segmenation

Typically, TSS is formulated as an unsupervised learning task, aiming to divide a TS into segments without using labelled or pre-defined categories. Moreover, studies regarding TSS often categorise the existing algorithms as either *domain-specific* or *domain-agnostic* [ESL23].

Many **domain-specific** TSS approaches have been proposed and successfully applied in fields, such as medical condition monitoring [Bos+03] and climate change detection [DVB03]. In [TOV19], the authors review prominent domain-specific CPD and TSS methods and outline three main subcategories: i) Kernel-based methods, ii) Likelihood-based methods, and iii) Graph-based methods. A common drawback of all subtypes is that they require the user to select domain-dependent hyper-parameters, such as window size, CPD threshold, and seasonality. Although domain-specific TSS can also be advantageous when tailored solutions are necessary for a particular domain problem, such approaches have limited versatility and adaptability, high complexity and risk of bias. The drawbacks mentioned above are considered impactful, especially in research, where frequently applying an algorithm to new data should be of reasonable time and computational complexity.

In contrast to domain-dependent TSS techniques, **domain-agnostic** algorithms follow more generalised data-driven approaches that do not require in-depth domain expertise and frequent customisation of model hyper-parameters. Therefore, domain-agnostic TSS approaches allow easy implementation across domains associated with low time and computational costs. Some prominent examples of domain-independent TSS algorithms are FLOSS [Gha+17], Autoplait [MSF14], and HOG-1D [ZI16]. For a long time, FLOSS has been considered the state-of-the-art for CPD and TSS. The approach proposed in [Gha+17] annotates an input TS with a bespoke *arc curve*, which is a vector that contains for each index i the number of arcs that cross over i. The local minima of this number indicate state change points. FLOSS distinguishes itself from many domain-agnostic methods as it enables online or streaming segmentation, while others (e.g., [TOV19; Lai+13]) are only defined for batch data. Moreover, FLOSS does not assume that all data is segmentable and provides ways of dealing with this hurdle. However, all of the above-mentioned domain-agnostic algorithms have similar drawbacks. Utilising domain-dependent TSS may lead to oversimplification, as it may not fully capture the intricacies of a particular domain, leading to lower accuracy compared to a well-tailored domain-specific method.

Recently, another domain-agnostic approach to TSS has been proposed, called Classification Score Profile (ClaSP) [SEL21]. ClaSP is a novel, highly accurate, parameter-free TSS algorithm that reduces the TSS segmentation problem to a binary TS classification problem of identifying regions by similar shapes. Given the assumption that subsequences extracted from the same TS segment are mutually similar (self-similar) and dissimilar from subsequences belonging to another segment, ClaSP iteratively determines change points by finding where the performance of the binary classifier is highest. As described in [ESL23], the binary classifier is trained to determine whether a subsequence belongs to the left or the right part and its class label. In this way, the ClaSP inventors use established supervised TS analysis methods to solve TSS, an unsupervised TS problem. To this end, the research described in [ESL23] shows that ClaSP is not only fast and scalable but also outperforms FLOSS in terms of accuracy.

1.2 Problem statement

The overview of existing TSS approaches presented above shows a clear trade-off between domain-specific and domain-agnostic TSS algorithms. On the one hand, domain-dependent methods offer results with high accuracy for a specific field of application, while domain-independent ones only achieve limited accuracy. On the other hand, domain-specific approaches require tedious incorporation of domain knowledge and have high complexity and limited applicability. In contrast, domain-agnostic approaches are more adaptable and suitable for a broader range of applications. Although the interest in utilising domain-agnostic TSS approaches has vastly grown, achieving high accuracy in various domains is a hurdle, even for state-of-the-art algorithms. For instance, the developers of ClaSP outline in [SEL21] that making their classifier more powerful is still an open research question.

2 Project Goal

The goal of this project is to address the research gap pointed out in [SEL21], namely to improve the state-of-the-art TSS algorithm ClaSP by making its classification step more accurate without negatively impacting its runtime. In this way, ClaSP might provide accuracy in various domains comparable to the results achieved by domain-specific TSS approaches. Since ClaSP treats the TSS problem as a binary classification, this work will consider established methods for improving the performance of time series classification algorithms, such as applying dilation [DPW20; SL23b]. Furthermore, this project aims to evaluate the attempted improvement in terms of algorithm accuracy and runtime and compare the results to the ones achieved by the proposed initial ClaSP algorithm and some of its competitors, such as FLOSS [Gha+17], Autoplait [MSF14] and BOCD (Bayesian online Changepoint Detection) [AM07].

3 Approach: Dilation Mapping

This project's approach to improving the state-of-the-art domain-agnostic TSS method ClaSP is based on dilation or, more specifically, a dilation mapping. To clarify which part of the algorithm can be enhanced via applying a dilation mapping, this section will first give a brief overview of ClaSP's functionality, followed by an explanation of the inner workings of dilation and dilation mapping.

As presented by the authors of ClaSP in [SEL21], the novel TSS approach is provided with a TS T with length $|T| = n$ as input. First, a classification score profile is computed in the following way: T is partitioned into overlapping windows of fixed length w, which are used to generate hypothetical splits for increasing offsets $i \in [w+1, ..., n-w-1]$, while extracting features for each window. The window length w is the only hyper-parameter of ClaSP. Then, each split is interpreted as a binary classification problem $Y = \{0, 1\}$ by attaching label $0\,(1)$

to all windows to the left (right) of the split point. A binary k-nearest-neighbour classifier (k-NN) is trained on the features and evaluated via a cross-validation score. The result is recorded for each offset i, representing the classification score profile of T. Finally, every local maximum in the profile is interpreted as a potential change point.

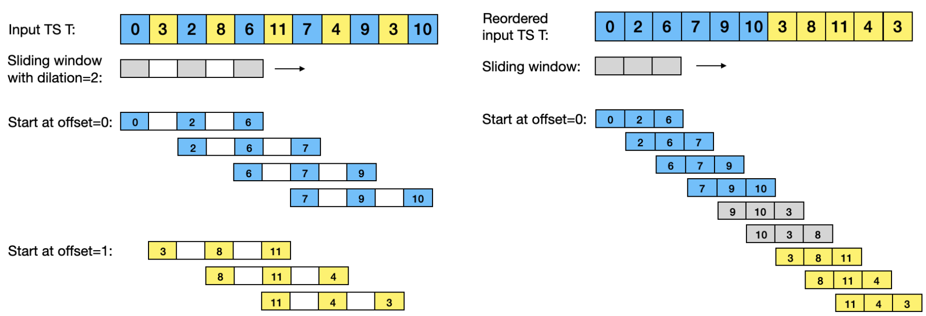

Fig. 1: Example of the difference between the usage of a sliding window with dilation (left) and the primary application of dilation mapping, followed by a standard sliding window operation (right), as shown in [SL23b].

Dilation is a technique that increases the receptive field of a filter, such as a sliding window, by inserting gaps between the entries in the filter. The total number of values is kept constant. An example of the functionality of a sliding window with dilation $d = 2$ (gap of 1 between each value) is shown on the left side of Figure 1. The sliding window is shifted along T. As the dilation is set to 2 for this example, the sliding window operation yields two subsequences starting at uneven (blue) and even (yellow) offsets. Applying dilation has been proven to effectively enable processing data at different scales (e.g., ROCKET TS classifier [DPW20]). As ClaSP also implements a sliding window as one of its first steps towards TSS, adopting dilation is a promising way to improve ClaSP's feature extraction accuracy without significantly increasing its complexity.

However, introducing dilation to an algorithm is associated with much effort in rewriting the code-base. To address this issue, the authors of [SL23b] introduce a modified approach, called dilation mapping, that has a similar effect as dilation and is much more feasible to apply. The proposed transformation of reordering all values in the input TS can be applied as a pre-processing step. An example is shown on the right side of Figure 1 to elaborate this method further. To mimic the functionality of a sliding window with dilation of 2, first, every uneven index of the input TS is taken, and then the resulting subsequence is concatenated to every even index. Afterwards, an ordinary sliding window operation is applied. The example is chosen to be conceptually equivalent to the application of a sliding window with dilation of 2 without reordering the input TS. However, the same process can also be designed for higher dilation factors ($d > 2$).

As proven in [SL23b], the dilation mapping is an operation with linear space and time complexity in the length of the input TS. Without altering any aspect of the ClaSP algorithm itself and by simply re-ordering the time series, the method can become dilated and profit from all the advantages of dilation, such as capturing essential data patterns at different scales. On the downside, as visible in Figure 1, by applying dilation mapping, a few additional windows are generated at the intersection of the re-ordering (coloured in grey). However, the longer the input TS is, the less this issue impacts the quality of the results. Since captured time series representing real-world processes are typically sizable and the application of dilation mapping has been shown to effectively mimic dilation and maintain the advantages it brings in [SL23b], this project will focus on applying dilation mapping in the context of ClaSP.

4 Evaluation

After the dilation mapping implementation step, the modified TSS approach will be evaluated to be compared to its baseline (ClaSP) and some of its competitors. The evaluation will resemble the experiments that the developers of ClaSP describe in [SEL21]. The following subsections give a more detailed overview of the benchmark datasets and the evaluation metrics this project will utilise.

4.1 Benchmark Datasets

For the evaluation of ClaSP, 98 datasets have been used, all of which can be found on the website dedicated to ClaSP [SEL]. 32 of these datasets are segmentation datasets that capture biological, mechanical or synthetic processes and have been used for evaluating FLOSS [Gha+17], which makes them a suitable candidate for comparing the performance of the different approaches. The 66 other datasets are semi-synthetic datasets curated by the authors of ClaSP from the UCR archive [Dau+19]. Starting with 120 UCR archive datasets, ClaSP's developers apply various data cleaning and munging approaches to develop the 66 datasets suitable for TSS methods analysis.

The number of change points that need to be detected varies from dataset to dataset: 49 input TS consist of 2 segments (1 change point), 22 datasets have 3 segments, 10 datasets have 4 segments, 11 datasets have 5 segments, 1 dataset has 6 segments, and 5 datasets have 7 segments. Furthermore, besides the baseline (ClaSP), the following five TSS algorithms will be used for comparison: Autoplait [MSF14], FLOSS [Gha+17], BOCD [AM07], Binary Segmentation (BinSeg) [SK74] and the window-based change point algorithm with L_2 cost function described in [TOV19].

4.2 Metrics

Using the above-mentioned datasaets, the performance of the proposed method will be evaluated in terms of **runtime** and **accuracy**. More precisely, the model accuracy will be calculated using the following evaluation metric, proposed in [Gha+17] and adopted by [SEL21]:

$$error = \frac{1}{n \cdot |cpts_{pred}|} \cdot \sum_{p \in cpts_{pred}} \min_{p' \in cpts_T} |p - p'| \tag{1}$$

Given an input TS T, a set of predicted change points $cpts_{pred}$, and a set of ground truth change points $cpts_T$, with each location in [1 ...n], a normed $error \in$ [0...1] is computed as shown in equation 1. Overall, the relative distances between every predicted change point p and the closest ground truth change point p' are summed up and normalised. Although this metric has the disadvantage of possibly matching multiple predicted change points to the same ground truth change point, this project will also adopt it to allow easier comparison of results to the outcomes of other published algorithms.

References

[AM07] Ryan Prescott Adams and David J. C. MacKay. *Bayesian Online Changepoint Detection*. 2007. arXiv: `0710.3742 [stat.ML]`.

[Blá+21] Ane Blázquez-García et al. "A Review on Outlier/Anomaly Detection in Time Series Data". In: *ACM Computing Surveys* 54 (Apr. 2021), pp. 1–33. DOI: `10.1145/3444690`.

[Bos+03] Marcel Bosc et al. "Automatic change detection in multimodal serial MRI: Application to multiple sclerosis lesion evolution". In: *NeuroImage* 20 (Oct. 2003), pp. 643–56. DOI: `10.1016/S1053-8119(03)00406-3`.

[Dau+19] Hoang Anh Dau et al. *The UCR Time Series Archive*. 2019. arXiv: `1810.07758 [cs.LG]`.

[DPW20] Angus Dempster, François Petitjean, and Geoffrey Webb. "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34 (Sept. 2020). DOI: `10.1007/s10618-020-00701-z`.

[DVB03] Jean-François Ducré-Robitaille, Lucie Vincent, and Gilles Boulet. "Comparison of techniques for detection of discontinuities in temperature series". In: *International Journal of Climatology* 23 (July 2003), pp. 1087–1101. DOI: `10.1002/joc.924`.

[ESL23] Arik Ermshaus, Patrick Schäfer, and Ulf Leser. "ClaSP: parameter-free time series segmentation". In: *Data Mining and Knowledge Discovery* 37.3 (Feb. 2023), pp. 1262–1300. DOI: `10.1007/s10618-023-00923-x`. URL: `https://doi.org/10.1007%2Fs10618-023-00923-x`.

[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`.

[Ism+19] Hassan Ismail Fawaz et al. "Deep Learning for Time Series Classification: A Review". In: *Data Min. Knowl. Discov.* 33.4 (July 2019), pp. 917–963. ISSN: 1384-5810. DOI: `10.1007/s10618-019-00619-1`. URL: `https://doi.org/10.1007/s10618-019-00619-1`.

[Lai+13] Claudia Lainscsek et al. "Non-Linear Dynamical Analysis of EEG Time Series Distinguishes Patients with Parkinson's Disease from Healthy Individuals". In: *Frontiers in neurology* 4 (Dec. 2013), p. 200. DOI: `10.3389/fneur.2013.00200`.

[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`.

[Mue+09] Abdullah Mueen et al. "Exact Discovery of Time Series Motifs". In: vol. 2009. Apr. 2009, pp. 473–484. DOI: 10.1137/1.9781611972795.41.

[SEL] Patrick Schäfer, Arik Ermshaus, and Ulf Leser. *ClaSP - Time Series Segmentation*. Accessed: 2023-10-24. URL: https://sites.google.com/view/ts-clasp/clasp.

[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.

[SH12] Patrick Schäfer and Mikael Högqvist. "SFA: A symbolic fourier approximation and index for similarity search in high dimensional datasets". In: Mar. 2012, pp. 516–527. DOI: 10.1145/2247596.2247656.

[SK74] Alastair Scott and Martin Knott. "A Cluster Analysis Method for Grouping Means in the Analysis of Variance". In: *Biometrics* 30 (1974), p. 507. URL: https://api.semanticscholar.org/CorpusID:124273988.

[SL23a] Patrick Schäfer and Ulf Leser. "Motiflets: Simple and Accurate Detection of Motifs in Time Series". In: *Proceedings of the VLDB Endowment* 16 (Feb. 2023), pp. 725–737. DOI: 10.14778/3574245.3574257.

[SL23b] Patrick Schäfer and Ulf Leser. *WEASEL 2.0 – A Random Dilated Dictionary Transform for Fast, Accurate and Memory Constrained Time Series Classification*. Jan. 2023. DOI: 10.48550/arXiv.2301.10194.

[TOV19] Charles Truong, Laurent Oudre, and Nicolas Vayatis. "Selective review of offline change point detection methods". In: *Signal Processing* 167 (Sept. 2019), p. 107299. DOI: 10.1016/j.sigpro.2019.107299.

[ZI16] Jiaping Zhao and Laurent Itti. "Decomposing time series with application to temporal segmentation". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2016), pp. 1–9. URL: https://api.semanticscholar.org/CorpusID:18127054.