Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Implementing a gap-invariant distance measure in ClaSP

Bachelor Thesis Exposé

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)

eingereicht von:	Moritz Spühler
geboren am:	02.02.2003
geboren in:	Berlin
Gutachter/innen:	Dr. Patrick Schäfer

1 Introduction

The increasing number and quality of sensors across various domains has led to a massive generation of data over time. This type of sequential data is commonly referred to as *time series* (TS).

One of the central tasks when working with time series is the detection of *change points*, i.e., identifying positions in the series where the state of the underlying data-generating process changes. While such changes are often easily recognized by humans, algorithms frequently struggle due to the complexity of the relationship between raw data and the process generating the data.

Many algorithms for this task—referred to as *change point detection* (CPD) — are domain-specific. In contrast, domain-agnostic CPD methods often still require expert knowledge to tune hyperparameters effectively.

Closely related to CPD is *time series segmentation* (TSS), which aims to divide a time series into segments based on these change points.

Schäfer et al. [5] introduced ClaSP, a domain-agnostic algorithm for TSS. ClaSP transforms the segmentation task into a *classification score profile* from which change points can be identified as local maxima. For detecting a single change point, the time series is split at each possible index, and the resulting segments are compared in terms of their internal similarity and mutual dissimilarity. A k-nearest neighbors (kNN) classifier is trained for each split, and its classification accuracy serves as a measure of separability. So far, ClaSP uses the *z-normalized Euclidean distance* (ED) to compute similarities between subsequences. For more than one change point, the algorithm is run recursively on each of the produced segments. In a later extension, ClaSP was made parameter-free and improved on multiple change points detection [2].

In this work, we propose replacing the distance metric in ClaSP with one that better captures higher-level semantic similarities. ED only looks at raw data differences, which can prevent it from detecting more complex similarities between segments, e.g. of variable length. Specifically, we draw inspiration from the approach by Imani et al. [3], who introduce a distance measure based on splitting a *subsequence* into a prefix, a *don't-care* region, and a suffix. This *prefix–suffix distance* enables the discovery of semantically similar subsequences even if their lengths or internal structures vary - see figure 1 [1]. Due to its partial invariance to gaps, this method has demonstrated strong expressiveness and robustness in detecting complex patterns [3].

Therefore, we believe that the proposed changes will enable ClaSP to better detect higher-level semantically similar subsequences, leading to improved change point detection on datasets derived from more complex underlying processes.

[1] Figure taken from Imani et al. [3].

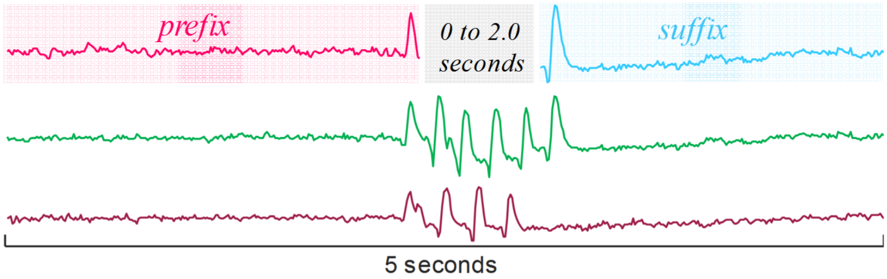

Figure 1: *top)* An idealized version of Asian citrus psyllid phloem-ingestion behavior. It consists of a highly conserved prefix and suffix, but with zero to two seconds of much more variable behavior in-between. *bottom)* Two realizations of this high-level similarity in data.

2 Research State

2.1 Definitions

In order to precisly talk about the proposed change to ClaSP, we introduce necessary definitions following Ermshaus et al. [2] and Imani et al. [3]:

Definition 2.1 (Process)**.** A *process* $P = (S, T)$ consists of one or more discrete states $s_1, \ldots, s_l \in S$ that are pairwise separated by transitions $(s_i, s_k) \in T \subseteq S \times S$ with $s_i \neq s_k$.

Many real-world processes evolve over time by transitioning between internal states. To observe these transitions, we typically measure an output over time, resulting in a time series.

Definition 2.2 (Time Series)**.** A *time series (TS)* T is a sequence of $n \in \mathbb{N}$ real values, $T = (t_1, \ldots, t_n),\ t_i \in \mathbb{R}$ that measures an observable output of a process P. The values $t_i \in T$ are also called data points.

To analyze structural changes or repeated behaviors within a time series, we often examine shorter, contiguous sections of it.

Definition 2.3 (Subsequence of a TS)**.** Given a TS T, 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 $\|T_{s,e}\| = e - s + 1$.

If the underlying process changes state, this may be reflected in the time series as a structural change. We formalize such events as change points.

Definition 2.4 (Change Point)**.** Given a process P and a corresponding TS T, a *change point (CP) or split* is an offset $i \in [1, \ldots, n]$ that corresponds to a state transition in P. The problem of change point detection (CPD) is to identify all CPs / splits in T.

To describe a process holistically, it is useful to list all the points in time at which changes occur. This leads to the notion of segmentation.

Definition 2.5 (Segmentation)**.** Given a process P and a corresponding TS T, a *segmentation* of T is the ordered sequence of indices of T, i.e., $t_{i_1}, \ldots, t_{i_S}$ with $1 < i_1 < \cdots < i_S < n$ at which the underlying process P changed its state.

2.2 Change Point Detection

Among existing methods for change point detection, *FLOSS* (Fast Low-cost Online Semantic Segmentation) stands out as a widely adopted and conceptually distinct baseline. It identifies potential change points by drawing *arc curves* from each subsequence to its nearest neighbor and selecting those offsets with the fewest arc crossings, under the assumption that change points disrupt self-similarity in the time series [2].

In contrast, *ClaSP* approaches segmentation through a classification-based paradigm. It evaluates each candidate split point by assigning binary labels to subsequences (0 for those before and 1 for those after the split) and training a classifier to separate them. If the hypothetical split aligns with a true change point, the classifier performs better than at misaligned points. Scanning across all possible splits yields a *classification score profile*, where local maxima indicate intra-segment homogeneity and inter-segment heterogeneity. This in return reflects a potential change point in the TS.

To calculate the similarity between regions in the TS, ClaSP uses z-normalized ED. It gives state-of-the-art results as it captures data differences fairly well.

Imani et al. [3] argue that this distance measure can perfom badly when semantic motifs with higher complexity are present in the data. For a visual reference, see figure 2 [2]. For example, a freethrow in basketball might be preceded by none up to multiple bounces followed by the throw itself. Although displaying the same underlying action, ED would give different distance values for a throw with one bounce in comparison to a throw with three bounces. They propose finding those subsequences by splitting a subsequence into a prefix, don't-care and suffix. Thus, only the prefix and suffix of a subsequence are actually compared, while the part in between does not influence the distance value. The same comparison of a freethrow with a bounce happening in the prefix, none to multiple bounces following in the don't-care part and the throw happening in the suffix would result in similar distances for differently executed freethrows.

Taking that idea of finding similar subsequences in the TS, we can evaluate the intra- and inter-distance of the segments given a splitpoint in a similar manner. We can split them into a prefix, don't-care and suffix and thus, more semantically complex processes might be correctly identified by ClaSP.

2.3 Automatic Parameter Selection

The *semantic-motif-finder algorithm* by Imani et al. [3] takes two inputs: The prefix/suffix length and the maximum don't-care length. In a variation of the algorithm, prefix and suffix can also be set independently. They envision the maximum don't-care length to be a small multiple of the set prefix/suffix length, yet optimal values are still domain

[2] Figure taken from Imani et al. [3].

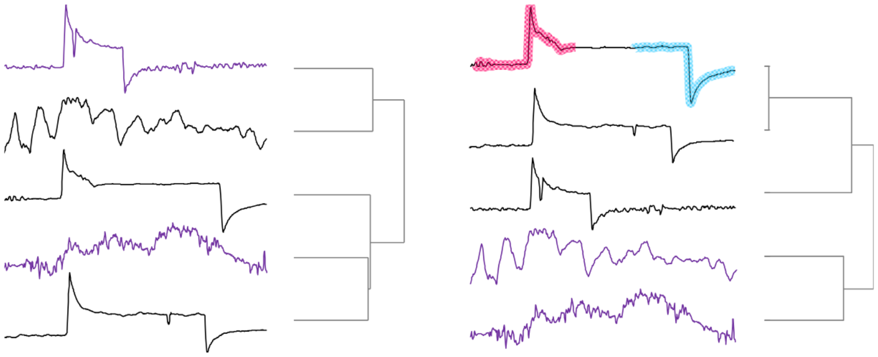

Figure 2: Two clusterings of insect data that include samples of a behavior called "non-ingestion-C". *left*) The clustering produced by Euclidean distance does not correctly group the behaviors, but the semantic subsequence distance *right*) does.

specific. Thus, a domain expert would be needed to approximate those values for an optimal use of the algorithm. ClaSP was made parameter-free in Ermshaus et al. [2] by approximating the algorithm parameters automatically based on the given data with the *SuSS* and automatic change point selection through use of the *Wilcoxon rank-sum test*. We envision achieving something similar by drawing the algorithm parameters directly from the data through an algorithmic analysis.

3 Methods

3.1 Code

Code for ClaSP is already available in the AEON Python toolkit [4] and in the reference implementation by Ermshaus et al. [1]. We will proceed as follows:

1. Integrate the prefix–suffix distance from Imani et al. [3] into AEON's ClaSP module.
2. Generalize the prefix–suffix implementation to support k-NN for arbitrary k (e.g. $k = 1$ and $k = 3$).
3. Implement a data-driven parameter learning function inspired by SuSS [2], to automatically select prefix length, suffix length, and maximum gap length.

3.2 Evaluation

We will evaluate on the 107 hand-selected benchmark datasets from Schäfer et al. [5] and Ermshaus et al. [2]. In addition, we will try to annotate and highlight those datasets that feature higher-level semantic events (e.g. repeating or gap-varying structures) to assess the intended strengths of the prefix–suffix distance.

Our experiments will compare:

- ClaSP with z-normalized Euclidean distance (baseline)
- ClaSP with prefix–suffix distance

Each distance metric will be tested in two classifier settings:

- 1-NN
- 3-NN

We will follow the evaluation protocol of Ermshaus et al. [2]:

1. Segmentation with unknown number of change points (automatic selection).
2. Segmentation with the true number of change points provided as a hyperparameter.
3. Comparison of segmentation accuracy (e.g. F1 score, segmentation error) and runtime.

4 Work Plan

Phase	**Duration**	**Tasks**
Integration of prefix–suffix distance	Weeks 1–2	Implement the Imani et al. [3] prefix–suffix metric in the AEON toolkit's ClaSP module; extend to support arbitrary k in k-NN.
Baseline evaluation with hand-selected parameters	Weeks 3–5	Run ClaSP with Euclidean and prefix–suffix distances using hand-tuned parameters; compare segmentation accuracy and runtime on the 107 benchmark datasets.
Automatic parameter selection (SuSS-like)	Weeks 3–5	Develop and integrate a data-driven parameter-guesser inspired by SuSS [2]; test stability of chosen prefix, suffix, and gap lengths.
Evaluation of automatic parameters	Weeks 6-7	Evaluate ClaSP (both distance metrics) with automatically learned parameters; compare against hand-selected results.
Writing and buffer	Weeks 8–12	Draft thesis chapters in parallel with experiments; reserve final two weeks as a buffer for revisions, proofreading, and potential delays.

References

[1] A. Ermshaus. Claspy repository. `https://github.com/ermshaua/claspy/`.

[2] A. Ermshaus, P. Schäfer, and U. Leser. Clasp: parameter-free time series segmentation. *Data Mining and Knowledge Discovery*, 2023.

[3] S. Imani and E. 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)*, pages 329–338, 11 2019.

[4] M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. Guijo-Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer, and A. Bagnall. aeon: a python toolkit for learning from time series. *Journal of Machine Learning Research*, 25(289):1–10, 2025.

[5] P. Schäfer, A. Ermshaus, and U. Leser. Clasp - time series segmentation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 1578–1587, New York, NY, USA, 2021. Association for Computing Machinery.