Exposé for the bachelor thesis:
Multivariate time series segmentation with ClaSP

Supervised by : Dr. Patrick Schäfer,
Prof. Dr. Ulf Leser,
Arik Ermshaus

Institution: HU Berlin

Alina Hartwich

June 2023

1. Introduction

In many areas of our everyday life, data is recorded over a certain period of time, analyzed and later used to make predictions. For example, smartphones or smartwatches use multiple integrated sensors [1] to record human movement data every day in order to draw conclusions on how athletic a person is. Another example is weather forecasting, which is based on recording and analyzing weather data such as the solar flare [12]. In medicine, EEG recordings are used to predict whether a patient is suffering from a panic attack or not [13]. In order to be able to make these predictions, it is important to be able to distinguish between small temporal sections, so called subsequences, in the data. Following data preprocessing, it becomes easy for humans to identify similar subsequences and distinguish them from others. However, this is nontrivial for a computer. The research area of time series analysis helps to identify those subsequences. [2]

A sequence of data points, that are recorded over a certain period of time, commonly with a constant sampling rate, are called time series. Time series can be univariate or multivariate, depending on how many dimensions are included in the recorded time series. For example the smart watch data mentioned above are multivariate time series, because each of the integrated sensors records an x-axis, y-axis and z-axis multivariate time series for each recorded process.

In the context of human activity recognition (HAR) a (multivariate) time series always captures a specific process, in that each process state is a segment and a subset of dimensions of the multivariate time series contains important information about the recorded process.

Important information in a process are, for example, the health status of a person, which was recorded via the EEG measurement. In order to extract this information from the multidimensional process and to predict the health status of a person correctly, time series segmentation is used. Time series segmentation (TSS) is particularly concerned with finding changes in a time series, which are called segments. TSS is intended to identify these segments and to learn the underlying process based on the segments.

In terms of EEG measurements, the underlying process could be a change in the cardiac output indicating that the person is experiencing a panic attack. TSS for univariate time series considers one dimension. Algorithms for

univariate TSS exist like ClaSP [5], BinSeg [3] and FLOSS[4]. This thesis addresses multivariate TSS by extending ClaSP. This challenge involves finding most important dimensions of the time series and then use these for multivariate TSS. Since it is often too time-consuming to consider all possible subsets of dimensions in a multivariate TSS, the challenge is to extract the important dimensions from all available ones without neglecting any crucial dimension but reducing noisy dimensions. The multivariate TSS can consider more dimensions as opposed to univariate TSS, so the health status of a person can be considered in a more differentiated way. Without multivariate time series segmentation, it may not be possible to correctly identify a person's health status from a single dimension.
Especially in the field of multivariate TSS, there currently do not exist many domain agnostic algorithms, which is why I will research this topic in my bachelor thesis.

In the following, I will focus on the goal of the expose to adapt the ClaSP algorithm for multivariate time series segmentation in the course of my bachelor thesis. To be able to do this, first important definitions and the underlying algorithms are explained. Then the method is described, how the algorithm could be be adapted and finally the evaluation of the adapted algorithm is described.

2. Goal of the thesis

The goal of this work will be to extend the ClaSP algorithm [5], to segment multivariate time series.
Until now, ClaSP can only segment univariate time series, but as described above there are more and more recorded multivariate time series available. For example, the EEG data are multivariate time series from which one can infer a person's health status. As mentioned above, without multivariate time series segmentation, it may not be possible to correctly identify a person's health status. Therefore, it is important to extend ClaSP so that it can segment multivariate time series.
In the related research area of pair motif discovery, there exists an algorithm called mSTAMP [6], that can recognize pair motifs in multivariate time series. In order to identify these pair motifs, mSTAMP does not test all combinations of the dimensions, as this approach would lead to a very high level of complexity. Instead, mSTAMP uses only a subset of all available dimensions, minimizing

the distances between the motifs and the complexity.
In this work the mSTAMP algorithm, especially the dimensions reduction, will be adapted for the use in the ClaSP algorithm. mSTAMP is a good choice, because the dimension reduction is an important component of multivariate TSS and the algorithm uses the nearest neighbor principle, which is implemented in the ClaSP algorithm too. Furthermore, mSTAMP is accurate, because it finds relevant motifs in a multivariate time series by only considering the most important dimensions that minimize the overall distance. It has a low computational complexity $O(d \log d\, n^2)$ where *d* is the dimensionality of the TS an *n* is the length of the TS. Therefore it makes sense to adapt this algorithm in ClaSP.

After adapting the mSTAMP algorithm for the use in ClaSP, the modified ClaSP algorithm will be evaluated in regard to the accuracy and the run time on multivariate data. In this work, the algorithm will be tested on two different multivariate time series from the domain of human activity recognition and will be compared to other algorithms, such as BinSeg and FLOSS, that can already deal with multivariate TS. The currently implemented ClaSP algorithm is used as a basis. It will also be evaluated whether the adapted multivariate ClaSP algorithm is more accurate than the current implementation, which can only segment univariate time series.

3. Definitions and Background

In this section, important terms for this work such as (multivariate) time series and segmentation are defined. Afterwards, the algorithms ClaSP and mSTAMP are considered in more detail.
Definition 1. A univariate *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. The values are also called data points[5].

For example, the weather data [12] described above are time series.
A time series can be univariate or multivariate. For example, recording only the rainfall is a univariate time series, but if the rainfall and the humidity are recorded over the same time, this data set is called a multivariate time series.

Definition 2. A *d-dimensional time series* $S \in \mathbb{R}^{dxn}$ is a set of co-evolving univariate time series $S^{(i)} \in \mathbb{R}^{n}$, such that $S = \{S^{(1)}, S^{(2)}, ... S^{(d)}\}$ where *d* is the dimensionality of ***S*** and *n* is the length of ***S***. [6]

All co-evolving time series in a d-dimensional time series have the same length *n* and they are synchronized between dimensions. That means, the time intervals between two data points are constant over all dimensions.
These multivariate time series record an underlying process that can change its state over time. These changes of state are called *change points.* [5]
With the help of these change points, the processes can now be divided into specific segments.
The main goal of TSS is to identify all change points in a time series.

Definition 3. Given a process and a corresponding d-dimensional time series *S* , a (*multi-variate) time series segmentation* (TSS) of *S* is the ordered sequence of indices of *S* , i.e., $\{s_{i1}, \ldots, s_{ir}\}$ with $1 < i_1 < \cdots < i_r < n$ at which the underlying multidimensional process changes its state. [5]

Multivariate time series segmentation deals with the problem of dividing a given multivariate time series into meaningful segments. These segments are considered meaningful if the subdivision of the multivariate time series corresponds to the real recorded process states. It is meaningful if the change points, which separates two segments from each other, found by the segmentation correspond to the change points of the real recorded process.

Definition 4. A *k-NN* classifier classifies a given sample by finding the k nearest samples in the pre-labeled training data, using a predefined distance function, such as the Euclidean distance. It then determines the predicted label by taking the majority label of the k-NN training samples [5].
For example, if k = 1, the data point is assigned to the same class as its 1-nearest neighbor.

4. Classification Score Profile (ClaSP)

ClaSP is a self-supervised, hyperparameter free TSS algorithm. So far, it can only solve the TSS problem for univariate time series.

To do that, it calculates the Classification Score Profile (ClaSP). The input of the algorithm is a univariate time series *T* of length |T| = n. The algorithm starts by "cutting" this time series *T* into overlapping subsequences of equal and fixed length *w*. The resulting *n-w+1* subsequences are called windows. ClaSP calculates the optimal length, here often reffered to as "window size", *w* itself, by using another algorithm called SuSS. This step of dividing the time series into overlapping subsequences, is later used to compare the different subsequences of the time series. Afterwards it calculates the k-Nearest-Neighbour (k-NN) to each window using the STOMP [7] algorithm. The k-NN is later used as a classifier, that classifies similar subsequences contained in the segments. With the resulting distance matrix given by STOMP, the algorithm solves multiple binary self-supervised classification problems. To do that, it iterates through the TS *T* and sets different hypothetical split points. For each possible split point the algorithm labels all windows to the left of the split point with 0 and all windows to the right of the split point with 1. After labeling the windows, the algorithm evaluates the possible split point by using the Leave-One-Out Cross Validation [8], the k-NN classifier and an evaluation function. The classification result of each cross-validation form the Classification Score Profile (ClaSP). The index of the maximum from the Classification Score Profile will be the first change point which separates two different segments from each other. Afterwards the algorithm recursively refines the two segments to find the rest of the change points.

ClaSP finds all change points and solves the TSS problem for univariate time series. The purpose of this bachelor thesis is to extend the ClaSP algorithm using the mSTAMP algorithm so that ClaSP can solve the TSS problem for multivariate time series.
Since mSTAMP uses only the f out of d relevant dimensions from the d-dimensional time series, the challenge now is to adapt this dimension reduction to ClaSP. This should increase the complexity of the algorithm.

5. mSTAMP

mSTAMP is an algorithm, that allows meaningful discovery of multivariate pair motifs. [6]
A pair motif is the most similar subsequence pair [6] of a TS and with the help of mSTAMP the meaningful motif in

a multivariate time series can be found. The goal of mSTAMP is to find the motif by only using the *f* most important dimensions of a *d-dimensional time series,* because it claims, that in most cases all *d* dimensions are not necessary for the discovery of meaningful pair motifs. To find the most important dimensions, mSTAMP sorts the distances of the dimensions for each time-stamp (window) in ascending order and can thus quickly find the *f* out of the *d* best, i.e. the ones with the smallest distance, dimensions. Because of this, mSTAMP does not have to test all combinations of *f* from *d* dimensions. By only using this pre-defined subset, the algorithm is much faster than the naive baseline, that searches in all dimensions with *d* over *f* combinations.

Other than the ClaSP algorithm, mSTAMP does not use a k-NN but the 1-NN. To find the pair motif, mSTAMP computes the z-normalized Euclidean distances between each subsequence and its 1-nearest neighbor and afterwards finds the pair motif by locating the lowest distance. The input of the algorithm is a multidimensional TS *T* and a subsequence length *m*. To store the z-normalized Euclidean distance, mSTAMP calculates a so called *f-dimensional matrix profile*. First, it initializes the f-dimensional matrix profile *P* with infinity. mSTAMP than takes a subsequence of the time series *T* and calculates dimensionwise the distance from a subsequence to each other subsequence and stores them in a so called *distance profile D*. After calculating the distance profile for each subsequence and dimensions, mSTAMP sorts the dimensions of the distance profile *D* in ascending order and calculates the column-wise cumulative sum, i.e. over each time-stamp. Because of this calculation mSTAMP can find the relevant *f* from *d* dimensions without testing all possible combinations using ellbow-plots. Last but not least it updates *P.*

As soon as the algorithm terminates the output *P* is the calculated f-dimensional matrix profile for the given TS *T.* Since mSTAMP can calculate the 1-NN in multivariate time series very precisely and quickly, we can use this implementation, to compute a multivariate k-NN. Therefore this algorithm is particularly well suited for the adaptation in ClaSP. With the help of the mSTAMP algorithm, we will evaluate, if ClaSP can also correctly segment multivariate time series.

As previously mentioned, mSTAMP itself must be adapted before it can be included in ClaSP. Since mSTAMP still uses the 1-NN, this must be rewritten to a k-NN before it can be adapted in ClaSP.

6. Multivariate ClaSP with mSTAMP

The existing ClaSP implementation [9] will be modified and adapted by this work.
The main goal is to implement the mSTAMP algorithm in the ClaSP algorithm so ClaSP will be able to deal with multivariate TS. Until now mSTAMP only calculates the 1-NN to each window of a time series. Because ClaSP already uses a k-NN classifier, the first step, before mSTAMP can be integrated into ClaSP, will be to adapt mSTAMP to produce k-NNs. For the implementation of the k-NN it is important to think about how many dimensions are relevant for the algorithm. Here, the mSTAMP algorithm, which also considers only a subset of the d dimensions of a TS, is used. The decision of how many dimensions to use plays a crucial role in the evaluation of the algorithm, because the number of used dimensions has a strong influence on the accuracy and speed of the algorithm.
In this work, three possible decisions are examined and evaluated.

1. All dimensions
   One possibility is to use all given dimension of the time series to calculate the k-NN. This can be a good idea because the dimensions of multivariate time series are often related. For example, as described above, sensors often record three axes simultaneously. If one or more dimensions were not considered, one ignores the fact that they are related to each other, so it is possible that the results are not as correct as they should be.
   It is to be tested whether the algorithm becomes more correct when no dimension selection is made compared to when one is made. In this case the mSTAMP algorithm is not needed, but the algorithm covers the case, because mSTAMP can also consider all dimensions.
2. Default value
   The second possibility is to use a default value for the number of dimensions used in a time series. For example it could be a users input how many dimensions the user would like to look at. Since the user will know her data best, a user input can be a good way to consider only the important dimensions and thus save time in the calculation.
3. Threshold value
   The last possibility is a threshold value calculated

by the algorithm itself whether it should stop using more dimensions or not. The elbow plots used in mSTAMP could be used here as a threshold value. Since this ensures accurate results with mSTAMP, it is a good idea to use these elbow plots for the threshold value as well. With the help of the threshold value, the algorithm itself recognizes the best dimensions without getting them in as user input, which may be the best option if the user does not know the data that good.

After checking all three possibilities there is another problem we have to look at. It is the correct calculation of the window size for the ClaSP algorithm with multiple dimensions. Until today ClaSP uses a self learned window size. But because we will not have a univariate time series any more we have to think about how the window size will be calculated.
Here I will look into two different possibilities.

1. One window size for each dimension
   The first possibility will be to let ClaSP calculate one window size for each dimension and work with each dimension and calculated window size separately afterwards.
2. One window size for all dimensions
   The second possibility is to use one window size for all dimensions. First, a window size should be calculated for each dimension of the time series and the later one should be selected from all these window sizes for the entire time series. There are different ways to determine this one window size, for example it would be an option to use the minimum, maximum, median or mode window size of all calculated ones or to calculate an average value of all and use it for the whole time series. The later analysis should check if the results are as accurate as with option 1.

7. Evaluation

The run time of the modified ClaSP algorithm needs to be evaluated on different data sets from the domain of human activity recognition.
Furthermore, the modified multivariate ClaSP algorithm will be compared with other algorithms that can already segment multivariate time series, such as BinSeg, FLOSS, Window and to ClaSP using one dimension. To evaluate the accuracy of ClaSP, the covering score [15] is used.

To evaluate the run time and the accuracy of ClaSP, it is important to have a look at which decision in relation to the number of dimensions for the k-NN and which choice of the window size calculation works best for the algorithm.
To evaluate the different choices in relation to the run time and accuracy of the algorithm, two different data sets are used.

1. Human Activity Segmentation Challenge Dataset
   The first data set is a data set, that contains 250 twelve-dimensional multivariate TS, sampled at 50 Hertz (Hz). In total it contains 100 different human motion data such as sport activities, household activities and shopping activities. The activities were performed by 16 bachelor student of the Humboldt Universität zu Berlin in 2022.[14]
2. MOSAD
   The second data set is the Mobile Sensing Human Activity Data Set (MOSAD). It contains 14 9-dimensional sensor recordings of three different routines performed by six different participants. The three routines contains household elements like vacuuming and some indoor and outdoor sport activities. [11]

After evaluating the algorithm on all two data sets this work will produce best decisions regarding to the choice of used dimensions and calculated window size.
The accuracies and run times of all possible choices will be computed and compared to each other. In the end the most accurate solution, based on the covering score, will be implemented in the final algorithm. To compare the different run times and accuracies with each other, the results will be shown in boxplots, diagrams and tables.

8. Literature

[1] E. Jovanov, "Preliminary analysis of the use of smartwatches for longitudinal health monitoring," *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, 2015, pp. 865-868, doi: 10.1109/EMBC.2015.7318499 url: https://ieeexplore.ieee.org/document/7318499

[2] Miodrag Lovrić, Marina Milanović und Milan

Stamenković. “Algoritmic methods for segmentation of time series: An overview”. In: Journal of Contemporary Economic and Business Issues 1.1 (2014), S. 31–53

[3] A. J. Scott und M. Knott. “A Cluster Analysis Method for Grouping Means in the Analysis of Variance”. In: Biometrics 30.3 (1974), S. 507–512. issn: 0006341X, 15410420. url: http://www.jstor.org/stable/2529204 (last visited 22. 4. 2023).

[4] Shaghayegh Gharghabi u. a. “Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels”. In: 2017 IEEE International Conference on Data Mining (ICDM). 2017, S. 117–126. doi: 10.1109/ICDM.2017.21.

[5] Patrick Schäfer, Arik Ermshaus und Ulf Leser. “ClaSP - Time Series Segmentation”. In: Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management. CIKM ’21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, S. 1578–1587. isbn: 9781450384469. doi: 10.1145/3459637.3482240.

[6] Yeh, C. C. M., Kavantzas, N., & Keogh, E. (2017, November). Matrix profile VI: Meaningful multidimensional motif discovery. In 2017 IEEE international conference on data mining (ICDM) (pp. 565-574). IEEE. Url:http://www.cs.ucr.edu/~eamonn/Motif_Discovery_ICDM.pdf

[7] Y. Zhu u. a. “Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins”. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). Los Alamitos, CA, USA: IEEE Computer Society, Dez. 2016, S. 739–748. doi: 10.1109/ICDM.2016.0085.

[8] Jason Brownlee *LOOCV for Evaluating Machine Learning Algorithms* https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/. July 2020.

[9] ClaSP Code and Raw Results. 2021. u r l: https://sites.google.com/view/ts-clasp/ (last visited 19. 4. 2023)

[10] SagarDhandare *Feature Scaling In Machine Learning!* https://medium.datadriveninvestor.com/feature-scaling-in-data-science-5b1e82492727. July 2021

[11] Arik Ermshaus, Sunita Singh, and Ulf Leser “Time Series Segmentation Applied to a New Data Set for Mobile Sensing of Human Activities” Published in the Workshop Proceedings of the EDBT/ICDT 2023 JointConference (March 28-March 31, 2023, Ioannina, Greece)

[12] Angryk, R.A., Martens, P.C., Aydin, B. *et al.* Multivariate time series dataset for space weather data analytics. *Sci Data* **7**, 227 (2020). https://www.nature.com/articles/s41597-020-0548-x

[13] OE Dick u. a. “Analysis of EEG patterns in subjects with panic attacks”. In: Human Physiology 46.2 (2020), S. 163–174.

[14] Ermshaus A, Schäfer P, et al. “Human Activity Segmentation Challenge” In: ECML/PKDD 2023 Discovery Challenge (2023), https://ecml-aaltd.github.io/aaltd2023/challenge.html

[15] van den Burg GJ, Williams CK (2020) An evaluation of change point detection algorithms. https://arxiv.org/abs/2003.06222