Change points detection in multivariate signal applied to human activity segmentation

Grzegorz Harańczyk

Kraków, Poland
gharanczyk@gmail.com

Abstract. The detection of change points in multivariate signal without access to annotated data is a challenging task. The fully unsupervised approach requires the development of a robust algorithm that can effectively identify unknown a priori patterns. In this article we will present one of the solutions to “Human Activity Segmentation Challenge” ECML/PKDD’23 ([4]) where the task was to predict the offsets of activity transitions for multivariate time series. The described solution won second place.

Keywords: multivariate signal segmentation · unsupervised learning · change point detection (CPD) · human activity recognition (HAR).

1 Introduction

Detecting change points is a common task when dealing with non-stationary time series and involves the identification of temporal boundaries that separate homogeneous time periods. Its importance was proven in various domains, including finance, environmental monitoring, industrial monitoring, medical condition monitoring, climate change detection, etc.

One of the popular area of application of such methods are human activity recognition (HAR) systems [1]. They are designed to automatically identify and classify human activities based on sensor data. These systems typically involve the use of wearable sensors, such as accelerometers and gyroscopes, to capture the motion and movement patterns of the human body. The data collected from these sensors is then processed and analyzed to recognize and classify different activities which later can be applied as fitness tracking, healthcare monitoring, personal security, gesture recognition etc.

Methods for change point detection can be roughly categorized as *online* [6] or *offline* [7]. Offline algorithms analyze the entire dataset as a whole and retrospectively identify points of change by examining past data. Their objective is typically to identify all the change points in a sequence in a batch processing mode. On the other hand, online (real-time algorithms) operate in parallel with the monitored process. They process each incoming data point as it becomes available, aiming to detect a change point as quickly as possible after it happens, ideally prior to the arrival of subsequent data points. In this article we will focus on the offline scenario.

Traditional change point detection methods often rely on predefined assumptions or manual thresholds, making them less adaptive to complex and dynamic data. To address this challenge, unsupervised machine learning (ML) methods have gained significant attention for their ability to automatically discover change points without prior knowledge or labeled data.

2 Problem statement

Human Activity Segmentation Challenge [4] was organized as one of Discovery Challenges during ECML PKDD 2023 conference. The objective of this challenge was to create completely unsupervised algorithms that address the time series segmentation problem. Many HAR systems currently adopt a strategy of processing fixed-length subsequences extracted from sensor measurements, rather than analyzing complete activity instances. Addressing this challenge requires the automatic partitioning of multi-variate sensor signals into variable-sized segments of activities, the number of which is unknown. Therefore the primary objective of this competition was focused on time series segmentation (TSS), an unsupervised learning problem that aims to identify homogeneous segments of variable lengths within a given time series. TSS is typically employed as a preprocessing step to partition complex time series data for advanced analytical tasks such as classification, anomaly detection, or motif discovery. However, performance in this area remains limited, especially when dealing with real-world time series data where the number of segments is not predetermined.

In order to achieve an accurate solution for the defined task, it was essential to develop robust algorithm capable of segmenting a wide range of different behaviors, while effectively handling multi-dimensional sensor recordings from different devices.

For the evaluation, the ground truth annotations of the activity transitions were used to measure the quality of predicted segmentations. Note that it was not possible to use annotations to build or tune segmentation models. Moreover, embedding human expertise about the given time series into handcrafted models was also explicitly prohibited. Parameters were supposed to be set for the entire data set or learned from the available data. It was also enforced by the validation schema (a part of the score (private score) was hidden until the end of the challenge).

2.1 Notation

In this section, we will introduce some notation that will be used later to facilitate a clearer and more precise description of our solution.

Definition 1. *A multivariate time series T is a sequence of $n \in \mathbb{N}$ real values, $T = (t_1, \ldots, t_n)$, where $t_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$ that contains the observable output of d sensors over time. The values are also called observations or data points.*

Definition 2. *For a given time series T, we define a subsequence $T_{s,e}$ of T with a start offset s and an end offset e which consists of the continuous observations of T from positions s to position e (i.e., $T_{s,e} = (t_s, \ldots, t_e)$ with $1 \leq s \leq e \leq n$).*

Definition 3. *We define segmentation of time series T as set of time series subsequences S_{i_s,i_e} for $i \in I$ such that*

$$\bigcup_{i \in I} S_i = T \tag{1}$$

and

$$S_i \cap S_j = \emptyset \quad \text{for } i, j \in I \tag{2}$$

Each time series segmentation can be expressed as ordered sequence of observations of T such that $t_{i_1}, \ldots, t_{i_S}$ with $1 < i_1 < \ldots < i_S < n$. We call these observations change points.

The set of change points also determines the segmentation of the time series; hence, in this paper, we will use these terms interchangeably.

Definition 4. *We say that coverage $S_{i \in I}$ is finer than $S_{i \in J}$ if each element of $S_{i \in J}$ can be expressed as a union of elements from $S_{i \in I}$. We denote it as $S_{i \in I} \prec S_{i \in J}$*

Definition 5. *For any two time series segmentations we can define their intersection i.e.,*

$$S_{i \in I} \wedge S_{i \in J} = \{s_i \cap s_j \text{ for } (s_i, s_j) \in (S_{i \in I}, S_{j \in J})\} \tag{3}$$

It is easy to observe that $S_{i \in I} \wedge S_{i \in J} \prec S_{i \in I}$ is segmentation of time series T and

$$S_{i \in I} \wedge S_{i \in J} \prec S_{i \in I} \text{ and } S_{i \in I} \wedge S_{i \in J} \prec S_{i \in J} \tag{4}$$

In the context of human activity recognition, our objective is to perform time series segmentation on sensor signals. This segmentation yields consecutive subsequences that correspond to distinct activities, such as walking or running.

Within the specified task of Human Activity Segmentation Challenge, we are presented with time series data that already possess predefined segmentation, representing distinct activities. Our objective is to predict this segmentation accurately. In this particular setup, the initial segmentation is concealed, thus prohibiting the use of supervised machine learning methods. List of original activities is used only for method validation i.e., original segmentation $S_{i \in I}$ will be compared with predicted $\hat{S}_{i \in J}$. Note that we don't know the cardinality of the original segmentation $\#I$ so it is possible that $\#I \neq \#J$;

2.2 Dataset

A dataset of 250 twelve-dimensional multivariate time series was collected for Human Activity Segmentation Challenge. The time series were sampled at a frequency of 50 Hertz (Hz) and contain between seven seconds and fourteen minutes (median 100 seconds) of human motion data (with a cumulative duration of 10.7 hours). Distribution of signal length was presented in Fig. 1.

The recordings were taken by students from Humboldt-Universität zu Berlin and capture few to many potentially recurring activities from a total of one hundred different ones, each lasting for variable time durations. The acquired sensor data encompasses triaxial acceleration, gyroscope, and magnetometer readings, as well as latitude, longitude, and speed, depending on the smartphone utilized. For all time series there were always available measurement of acceleration (*x-acc*, *y-acc*, *z-acc*) and magnetometer measurements (*x-mag*, *y-mag*, *z-mag*) and either set of gyroscope measurements (*x-gyro*, *y-gyro*, *z-gyro*) or measurements of *lat*, *lon* and *speed*. So in our study, each observation in the dataset was represented by a nine-dimensional signal with sampling of 50 values collected per second. Example of such multivariate signal was presented in Fig. 2.

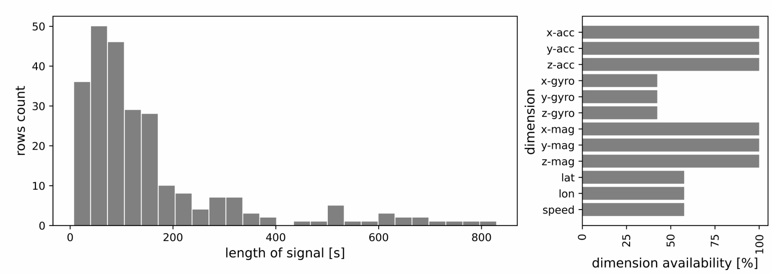

Fig. 1. Distribution of signal length and availability of a given measurement in time series.

Besides these time series, their sensor names, and the overall sample rate, no other information was provided or permitted for use. Also use of any external data and pre-trained models (as they have been trained on external data) was strictly prohibited.

2.3 Validation procedure

As previously stated, the ground truth segmentation, representing distinct activities was not available during segmentation, only used for the external validation step. To assess the performance of a given solution predicting time series segments, the multi-threshold *F1* score was used. It is defined as follows:

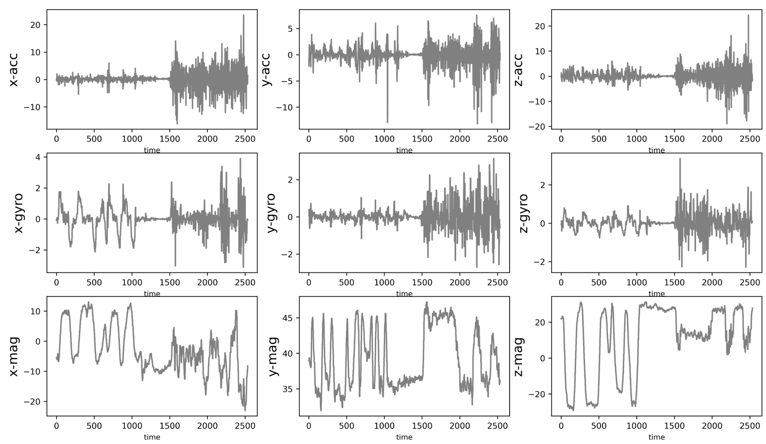

Fig. 2. Example of multivariate signal ($ts\_id = 9$).

For a given time series, we calculate the intersection over union (IoU, also called *Jaccard similarity coefficient* [5]) of overlapping predicted and ground truth segments, to obtain a normed score (higher is better).

$$IoU(S, \hat{S}) = \frac{|S \cap \hat{S}|}{|S \cup \hat{S}|} \tag{5}$$

Then we set a threshold to assess which overlaps are sufficient, from which a confusion matrix is inferred, used to calculate the *F1* score. A true positive (TP) is counted when a single predicted segment matches a ground truth segment with an IoU above the threshold. A false positive (FP) indicates a predicted segment had no associated ground truth segment. A false negative (FN) indicates a ground truth segment had no associated predicted segment.

This computation is repeated for multiple thresholds and the results are averaged to obtain the final normalized score for a given time series:

$$\frac{1}{\#ths} \sum_{t \in ths} \frac{2TP(t)}{2TP(t) + FP(t) + FN(t)} \tag{6}$$

where set of thresholds $ths = \{0.5, 0.55, \ldots, 0.9\}$ and where $\#ths$ denotes the cardinality of the set ths.

To get the final score, this measure is calculated for each of the time series and averaged.

In our analysis, we employed a range of internal metrics, including measures of internal consistency, to evaluate the performance of our models. However, given the specific nature of the task, we also endeavored to utilize the competition scoring system as frequently as possible. After submitting segmentation for all time series in the dataset, the value of $F1$ metric was calculated. It is important to note that utilizing the competition scoring system posed some strategic challenges due to limitations on the number of calls to the scoring API, which were restricted to three calls per day. Additionally, it is worth mentioning that the scoring API provided a single value for the entire solution, encompassing all time series in the public part of validation dataset. The score for another part of validation dataset (private score) was not available until the end of the competition.

3 Approach selection

3.1 Baseline solutions

In the initial stages, we established a set of simple baseline solutions, which were subsequently modified to serve as benchmarks for comparing the performance of our proposed solution. Developing these basic baselines not only facilitated the identification of key aspects that could potentially have a significant impact on performance but also assisted in prioritizing their importance.

Specifically, we generated a series of segmentations using a random selection of change points, as well as a series of segmentations based on equal subsequences with an increasing number of generated segments (see Fig. 3).

We also conducted a series of experiments that involved segmentations using both single and multiple dimensions (see Fig. 4). These experiments not only helped us to reduce the number of components employed in the final model but also enabled an assessment of the performance implications associated with the utilization of complex models.

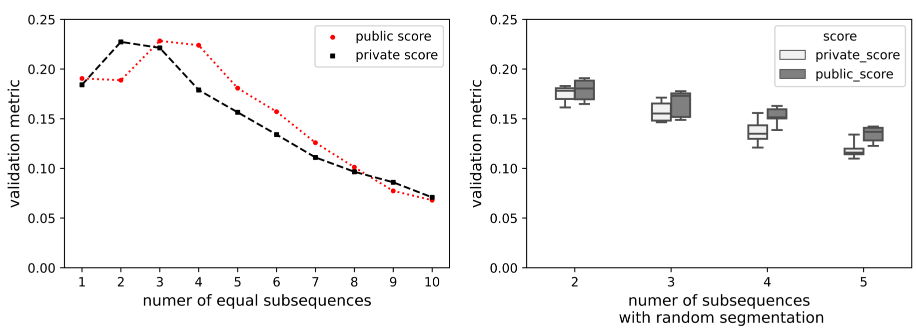

Fig. 3. Performance of simple baseline solutions - equal subsequences with increasing number of generated segments and random segments.

3.2 General idea

In order to maintain control over the final solution while keeping it as simple as possible, we opted to generate segmentations for one-dimensional components of given multivariate signal and subsequently aggregate them to form the final solution. Instead of utilizing all components of the multivariate signal, we selected only three components that exhibited the highest performance in one-dimensional models. Our experimentation involved exploring various methods of aggregating the results from individual components into the final solution, as well as devising a technique to reduce the final solution to prevent overfitting.
Hence, the final solution consists of the following three steps (see Algorithm 1):

- generation of ClaSP change points for selected channels;
- consolidation of change points obtained from various channels;
- elimination of irrelevant change points through pruning.

Algorithm 1 Change Point Detection

```
function ChangePointDetection(data)
    changePoints ← ∅                                          ▷ Initialize empty set
    for channel ∈ {x-acc, y-acc, z-acc} do                    ▷ Loop over channels
        channelData ← extractChannelData(data, channel)       ▷ Extract channel data
        cps ← ClaspProcedure(channelData)                     ▷ Apply clasp procedure
        changePoints ← changePoints ∪ cps                     ▷ Merge change points
    end for
    changePoints ← PruneFunction(changePoints)                ▷ Apply prune function
    return changePoints
end function

procedure ClaspProcedure(channelData)
    ...                                                       ▷ Implementation details - see [2]
end procedure

procedure PruneFunction(changePoints)
    prunedChangePoints ← ∅                                    ▷ Initialize empty set
    for point1, point2 ∈ changePoints do                      ▷ Loop over change points
        if |point1 − point2| ≤ threshold & point1 < point2 then
            prunedChangePoints ← point1                       ▷ Keep only the first change point
        end if
    end for
    return prunedChangePoints
end procedure
```

3.3 ClaSP (Classification Score Profile) algorithm

We conducted experiments using several segmentation methods and ultimately selected the ClaSP (Classification Score Profile) algorithm to generate change points for the chosen channels. In [2], it was demonstrated that ClaSP outperforms existing state-of-the-art methods in terms of accuracy. Additionally, the evaluation of ClaSP's performance involved rigorous experimental analysis using a benchmark dataset consisting of 107 distinct data sets. Remarkably, the results indicated that ClaSP not only achieved improved accuracy but also demonstrated impressive speed and scalability.

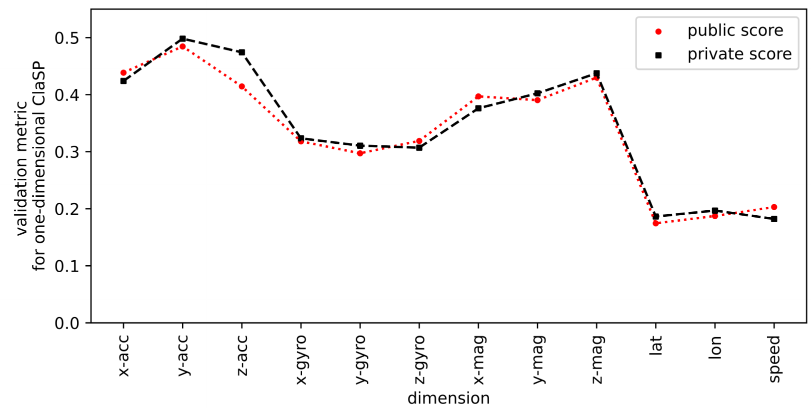

Fig. 4. Performance of ClaSP algorithm applied to single components of multivariate signal.

In our pursuit of enhancing performance, we aimed to fine-tune model parameters. In order to accomplish this, we followed the methodology outlined in Section 4.7 of [2] and in the article [3]. However, due to limited validation capabilities, we were unable to guarantee robustness and obtain a superior set of parameters compared to the default configuration.

3.4 Multivariate aggregation

When combining segmentations obtained from one-dimensional time series, two primary strategies emerge as the most intuitive. The first strategy involves deeming a change point as valid for the multidimensional time series if it is valid for any of its individual dimensions. This strategy allows for variations and deviations within individual dimensions while still considering the change point as valid for the overall multidimensional time series. The second strategy entails considering a change point as valid for a multidimensional time series only if it is deemed valid for all of its one-dimensional components. In other words, the change point should exhibit consistency across all dimensions.

Using notation from Section 2.1 we can express it as follows:

Scenario 1: use as a segmentation the intersection of all available one-dimensional segmentations:

$$\bigwedge_d S_{i \in I_d} = S_{i \in I_1} \wedge \ldots \wedge S_{i \in I_d} \tag{7}$$

Note that in our case finally we decided to use only measurements from accelerometers (*x-acc*, *y-acc*, *z-acc*), so in our case multivariate aggregation will be an intersection of these components.

Scenario 2: we define a new segmentation W such that change point $c \in W$ if and only if $c \in S_{i \in I_k}$ for each $k = 1, \ldots, d$. It can be also defined with some precision threshold, i.e., c is close enough to change point in each component segmentation.

For example, if we would get three one-dimensional segmentations: $\{[0, 50], [51, 100]\}$, $\{[0, 50], [51, 80], [81, 100]\}$, and $\{[0, 50], [51, 90], [91, 100]\}$ then Scenario 1 would give us the aggregated segmentation as

$$\{[0, 50], [51, 80], [81, 90], [91, 100]\}$$

and Scenario 2 would give us:

$$\{[0, 50], [51, 100]\}.$$

We opted to proceed with Scenario 1, wherein a change of activity in a single dimension is considered sufficient. We believe that within the context of our use case, this assumption is valid. Specifically, we posit that it is possible for an activity to change solely by altering a single component, without necessitating simultaneous changes across all dimensions.

Upon combining the one-dimensional components, we made an intriguing observation regarding the presence of numerous closely spaced change points. This phenomenon could be attributed to time shifts or delays in detecting activity changes across different components. Given the sampling frequency of 50 Hz, achieving exact alignment of change point values across all dimensions proved to be challenging. Unfortunately, the lack of access to annotated data hindered our ability to empirically validate this hypothesis. Nevertheless, in light of this observation, we made the decision to eliminate redundant change points. Remarkably, this post-processing step yielded a positive impact on the performance of our solution, further reinforcing the significance of addressing the issue of redundant change points in the context of our study.

3.5 Pruning

As previously stated, the sampling rate for all sensors was set at 50 Hz, resulting in the collection of 50 samples per second. Hence, if there is an absolute error of 50 (samples) in predicting a change point, it indicates that a discrepancy change point and the actual change point in the collected data is equal to only one second. From the other side we believe that the transition between distinct activities within the recorded signal may extend for few seconds. To address this problem, without possibility to test it with annotated data, we applied the following procedure:

For a given segmentation of time series $\{cp_i\}_{i \in I}$, in cases where the distance between two change points, denoted as cp_1 and cp_2, falls below a predefined *resolution window* w (i.e., $d(cp_1, cp_2) < w$), we adopt a selection criterion that favors retaining only cp_1. This decision is based on the assumption that cp_1 represents the initial indication of a signal change, while cp_2 and subsequent change points are likely to be observed with a delay. By prioritizing cp_1 in such scenarios, we aim to maintain consistency with the chronological order of change point occurrences, acknowledging the potential presence of temporal delays in the observed signal components.

We experimented with different lengths of resolution window, and based on performance on public validation dataset we selected the optimal value i.e., $w = 400$. The selected value turned out to be also optimal for private part of validation dataset, see Fig. 5.

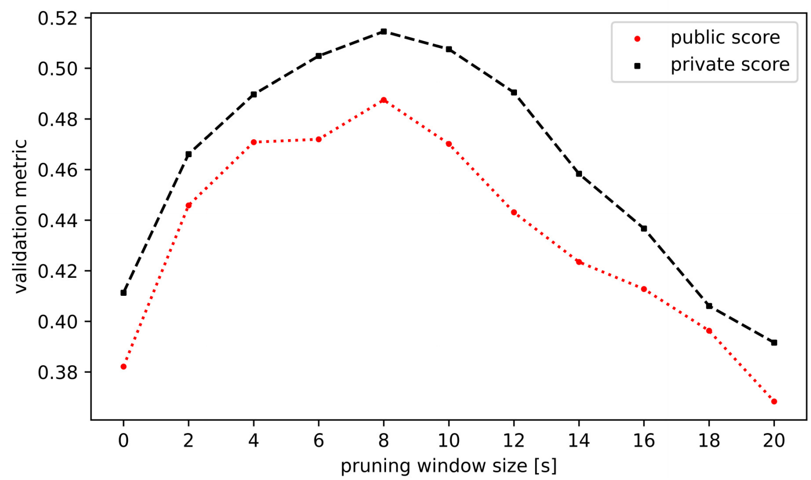

Fig. 5. Impact of pruning with a given resolution window on overall model performance.

4 Summary

We have developed a robust and effective baseline approach for segmenting multivariate signals in the field of human activity recognition. Our proposed solution demonstrates substantial superiority over defined baseline models (see Section 3.1). Notably, our solution achieved very good performance in the 'Human Activity Segmentation Challenge' at ECML/PKDD'23, securing the 2nd place. Furthermore, our approach demonstrated comparable performance on both the public and private parts of the validation dataset, providing evidence of its ability to avoid overfitting.

Given its performance, our baseline approach holds significant value as a simple yet robust reference point for future investigations related to the identification of human activities using multivariate signal-based methods.

The data, Python data loaders and baseline solutions can be downloaded from: github.com/patrickzib/human_activity_segmentation_challenge and the code of the described solution is available at github.com/gharanczyk/ecml_pkdd2023

References

1. Aminikhanghahi, S. and Cook, D. J.: Using change point detection to automate daily activity segmentation. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 2017, pp. 262-267. https://doi.org/10.1109/PERCOMW.2017.7917569
2. Ermshaus, A., Schäfer, P., Leser, U.: ClaSP: parameter-free time series segmentation. Data Mining and Knowledge Discovery **37**, 1262–1300 (2023). https://doi.org/10.1007/s10618-023-00923-x
3. Ermshaus, A., Schäfer, P., Leser, U.: Window Size Selection in Unsupervised Time Series Analytics: A Review and Benchmark. Advanced Analytics and Learning on Temporal Data 83–101 (2023) https://doi.org/10.1007/978-3-031-24378-3_6
4. Ermshaus, A., Schäfer, P., Leser, U., Bagnall, A., Tavenard, R., Leverger, C., Lemaire, V., Malinowski, S., Guyet, T., Ifrim, G.: Human Activity Segmentation Challenge. ECML/PKDD 2023 Discovery Challenge (2023).
5. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et du Jura. Bulletin de la Société Vaudoise des Sciences Naturelles **37**, 547-579 (1901) https://doi.org/10.5169/SEALS-266450
6. Rauhameri, A., Salminen, K., Rantala, J., Salpavaara, T., Verho, J., Surakka, V., Lekkala, J., Vehkaoja, A., Müller, P.: A comparison of online methods for change point detection in ion-mobility spectrometry data. Array **14** (2022) https://doi.org/10.1016/j.array.2022.100151
7. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Processing **167** (2020) https://doi.org/10.1016/j.sigpro.2019.107299