Journal of Big Data

RESEARCH Open Access

Reactive buffering window trajectory segmentation: RBW-TS

Bakht Zaman[1*], Dogan Altan[1], Dusica Marijan[1] and Tetyana Kholodna[2]

*Correspondence:
bakht@simula.no

[1] Simula Research Laboratory, Oslo, Norway
[2] Navtor AS, Egersund, Norway

Abstract

Mobility data of a moving object, called trajectory data, are continuously generated by vessel navigation systems, wearable devices, and drones, to name a few. Trajectory data consist of samples that include temporal, spatial, and other descriptive features of object movements. One of the main challenges in trajectory data analysis is to divide trajectory data into meaningful segments based on certain criteria. Most of the available segmentation algorithms are limited to processing data offline, i.e., they cannot segment a stream of trajectory samples. In this work, we propose an approach called Reactive Buffering Window - Trajectory Segmentation (RBW-TS), which partitions trajectory data into segments while receiving a stream of trajectory samples. Another novelty compared to existing work is that the proposed algorithm is based on multidimensional features of trajectories, and it can incorporate as many relevant features of the underlying trajectory as needed. This makes RBW-TS general and applicable to numerous domains by simply selecting trajectory features relevant for segmentation purposes. The proposed online algorithm incurs lower computational and memory requirements. Furthermore, it is robust to noisy samples and outliers. We validate RBW-TS on three use cases: (a) segmenting human-movement trajectories in different modes of transportation, (b) segmenting trajectories generated by vessels in the maritime domain, and (c) segmenting human-movement trajectories in a commercial shopping center. The numerical results detailed in the paper demonstrate that (i) RBW-TS is capable of detecting the true breakpoints of segments in all three usecases while processing a stream of trajectory points; (ii) despite low memory and computational requirements, the performance in terms of the harmonic mean of purity and coverage is comparable to that of state-of-the-art batch and online algorithms; (iii) RBW-TS achieves different levels of accuracy depending on the various internal parameter estimation methods used; and (iv) RBW-TS can tackle real-world trajectory data for segmentation purposes.

Keywords: Online trajectory segmentation, Mobility data, Trajectory features, Multidimensional time series, Streaming data

Introduction

Vessel navigation systems, mobile and wearable devices, and several IoT devices continuously generate high-velocity trajectory data streams. These trajectory data streams contain useful information that can be leveraged to assist in various decision-making tasks. For instance, in the maritime domain, automatic identification system (AIS)-based trajectory data can be exploited to improve the safety and efficiency of vessels by assisting in tasks including route planning, collision avoidance, and search and rescue missions [1–3].

Trajectory data mining deals with processing mobility data collected via sensors that transmit a stream of features stamped with time and location [4, 5]. One of the main preprocessing tasks for trajectory mining is trajectory segmentation, in which trajectory data are divided into multiple non-overlapping segments. Many existing trajectory segmentation approaches mostly focus on location information [6–11]. However, in numerous applications, other more suitable trajectory features exist that can be efficacious for segmentation. For instance, to compute the fuel consumption of a vessel for energy optimization purposes, one can partition the trajectory based on speed, and then by computing the average speed in all trajectory segments, estimate the fuel consumed [12]. Therefore, trajectory segmentation algorithms that can consider multiple trajectory features simultaneously, including the speed and direction, can be useful in numerous other scenarios.

Furthermore, in many real-world applications (e.g., maritime domain and e-health monitoring systems), data are streaming and certain decisions depending on segmentation need to be made in real-time. However, existing batch algorithms are not applicable in streaming scenarios, as the entire data set is not available at once, or because of higher computational complexity, as a batch algorithm would be required to be run again when a new data sample has arrived. Therefore, an online trajectory segmentation algorithm is required, which not only relies on spatiotemporal information, but can also incorporate other relevant trajectory features according to the requirements of the target application. To this end, we propose an online segmentation algorithm called reactive buffering window-trajectory segmentation (RBW-TS), which considers the variations in trajectory features during segmentation. Note that the RBW-TS algorithm does not depend directly on time and location. Therefore, it is imperative to select relevant features according to the target application requirements. For instance, the location and direction may be of interest to segment the trajectory in one application, whereas speed and acceleration could be potential features for making a decision about segmenting the trajectory in another application.

The contributions of the paper are as follows:

- An online segmentation algorithm for multidimensional time series is proposed for a streaming data scenario. The algorithm can include multiple appropriate trajectory features required for a specific application. This attribute makes the algorithm general and enables it to be applied in many domains and applications.
- The proposed online algorithm requires less computational resources compared to batch algorithms since it does not process the whole data at once and does not make multiple iterations over the whole data. Moreover, the proposed algorithm requires low memory resources because it does not store the trajectory points but only the estimated parameters.
- We apply the proposed algorithm on three real-world data sets, and compare its performance with the state-of-the-art batch and online trajectory segmentation approaches. The numerical results show that the proposed online segmentation algorithm can extract segments from streaming data. The validation results further

show that RBW-TS achieves a competitive level of performance in terms of the three metrics (purity, coverage, and harmonic mean of purity and coverage) compared to state-of-the-art algorithms. Moreover, it is observed that the selection of different sets of trajectory features leads to various levels of performance of the algorithm.

The rest of the paper is organized as follows. Section Related work discusses the related trajectory segmentation algorithms available in the literature. Section Definitions and terminology details the definitions that are helpful to describe the proposed segmentation algorithm. In Sect. Time series segmentation, the problem of multidimensional time series segmentation is formulated, and the proposed algorithm is presented. Section Experimental evaluation deals with the performance evaluation of the proposed algorithm and comparison with other methods using real-world data sets. The numerical results are reported and discussed. Finally, the conclusions and future work are detailed in Sect. Conclusions and future work.

Related work

In this section, we introduce several important and relevant state-of-the-art trajectory segmentation approaches. We also present a discussion on the strengths and limitations of these approaches.

Stay Point Detection (SPD), proposed in [6], is one of the first algorithms proposed to segment a trajectory based on the two states of Stay and Move. The SPD uses two parameters, time t and distance d, to identify whether a moving object is stationary in the vicinity or in a moving state. The trajectory of the moving object is segmented according to the boundaries of the transition between its states after detecting the state of the moving object. An improvement in the SPD algorithm is the inclusion of speed and bearing in the determination of the state of a moving object. The speed point feature is used in conjunction with an adjusted density-based spatial clustering of applications with noise (DB-SCAN) algorithm called clustering based stops and moves of trajectories (CB-SMOT) in [7]. In a very similar approach, [8] presented direction-based stops and moves of trajectories (DB-SMOT), which uses the bearing feature to detect the state of the movement. Using an optimization-based approach, [9] proposed an algorithm called greedy randomized adaptive search procedure for unsupervised trajectory segmentation (GRASP-UTS) to benefit from the point features of a trajectory in a greedy algorithm. The GRASP-UTS considers semantic features such as distance to shore, which boosted the performance of the method; however, the iterative approach based on kernel optimization in this algorithm makes it computationally more complex than other trajectory segmentation algorithms. Octal window segmentation (OWS) and sliding window segmentation (SWS) are proposed in [10, 11] to detect partitioning points based on identification changes in the behavior of a moving object. These approaches assume that semantic features and environmental conditions can affect the trajectories of moving objects indirectly, and processing the trajectory itself must be sufficient to partition trajectories precisely. For example, movement in a traffic jam causes a moving object to have a low speed. By processing a sliding window, these algorithms limit the amount of data to be processed, which results in the ability to segment trajectories in a stream fashion and less memory and computation resources. Their limitation is the assumption of having access to high-resolution trajectories with sufficient and frequent trajectory samples on the route; hence, a sparse or noisy trajectory would result in performance degradation. On the other hand, since RBW-TS does not process directly the time and location information, a high-resolution trajectory is not required.

Various approaches have been used to address the problem of multidimensional time-series segmentation. One of these approaches is called greedy Gaussian segmentation (GGS), proposed in [13], in which the data in each segment are considered to follow a multivariate Gaussian distribution. The proposed method computes the breakpoints of the segments and then estimates the parameters of each segment. For the combinatorial optimization problem of searching over the possible breakpoints in [13], a dynamic programming-based approach is proposed. A similar approach is presented in [14], where each segment (cluster) is characterized by a correlation network (specifically, the Markov random field (MRF)). The algorithm learns both the breakpoints of segments and the MRF parameters of each segment. The proposed algorithm is called Toeplitz inverse covariance-based clustering (TICC). The main optimization problem is based on a likelihood function, the sparsity of the inverse covariance matrix, and temporal consistency. The main problem is non-convex; hence, alternating minimization is applied in [14]. Recently, a semi-supervised approach has been proposed in [15] that takes into account time-point clustering based on the temporal proximity of time points and the correlation of their corresponding values. Again, this is a position/distance-based approach, and it cannot be generalized to include other features. Other approaches include [16–18], which either cannot handle streaming data scenarios or cannot be generalized for multidimensional feature-based segmentation. Finally, an online segmentation algorithm, named Thresholds [19], takes into account thresholds based on speed and orientation to define safe areas to decide the inclusion of points in a given trajectory (i.e., trajectory sampling). The Thresholds algorithm calculates velocity vectors to determine a joint safe area considering the intersection of the sample-based (i.e., recent data points in the selected sample so far) and trajectory-based (i.e., recent data points in the trajectory so far) velocity vectors. Subsequently, the current candidate point is checked to determine whether it falls within this joint safe area. As a limitation, the Thresholds algorithm [19] is limited in terms of the multidimensional features it takes into account (i.e., velocity and orientation), which also necessitates careful fine-tuning of thresholds to be used for such features.

Definitions and terminology

In this section, we present the definitions of the concepts used to formulate and explain the main ideas of our proposed segmentation algorithm.

- *Trajectory point:* A minimal trajectory point (l_i) is represented as: $l_i = (x_i, y_i, t_i, o_i) \in \mathcal{L}$, where x_i is the longitude of a moving object which varies from 0° to $\pm 180^\circ$, y_i is the latitude which varies from 0° to $\pm 90^\circ$, t_i is the time when x_i and y_i were collected, o_i is the identifier of a moving object, and $\mathcal{L}$ is the set of all trajectory points. A trajectory point may contain additional elements which would represent diverse features of the moving object in the application at hand. The sequence of spatio-temporal points characterizes a trajectory.

- *Raw Trajectory*: A *raw trajectory*, or simply *trajectory*, is a time-ordered sequence of spatio-temporal points. A formal definition of a raw trajectory for a moving object o is given by: $\tau_o =< l_0, l_1, ..., l_n >, l_j = (x_j, y_j, t_j, o_j), l_j \in \mathcal{L}, 0 \leq j \leq n$ where, $\forall_{l_u, l_v \in \tau_o}\ o_u = o_v$, $\forall_{l_u, l_v \in \tau_o}$ if $\mathrm{u} \leq \mathrm{v} \implies \mathrm{t_u} \leq \mathrm{t_v}$. As motivated earlier in the introduction section, we split a trajectory into smaller parts, called *segments* or *subtrajectories*, defined next.
- *Segment or Subtrajectory:* A *segment* or *subtrajectory* is a set of consecutive trajectory points belonging to a raw trajectory that represents a useful pattern or behavior of the moving object.
- *Trajectory Point Feature:* A trajectory point feature is an attribute that describes the state of a moving object. Examples of trajectory point features include the speed, direction, velocity, and acceleration, etc. These features can be present in the observations of the trajectory samples or can be computed from these observations. A combination of these point features of trajectories can be used for segmentation.
- *Buffering State and Buffering Window:* Buffering is the state in which the algorithm is idle and waiting for new trajectory points to fill the buffering window. A buffering window of size w contains the initial w trajectory points of each segment.
- *Segmentation State:* The state of the algorithm when it processes the newly arrived sample to decide whether the new trajectory point belongs to the current segment or not.

Time series segmentation

In this section, first, we present the model and problem formulation for the batch scenario. Then, the online segmentation of multidimensional time series is discussed, and the proposed segmentation algorithm is detailed.

Model

The model considered in this work is based on n-dimensional features of trajectory points. Each *trajectory point* is mapped to an n-dimensional vector containing various features. The (n-dimensional) feature vector corresponding to the trajectory point l_i can be represented as:

$$\boldsymbol{z}_i = [d_i, v_i, a_i, \ldots]^\top \in \mathrm{R}^{\mathrm{n}},$$

where d_i is the direction, v_i is the velocity, and a_i is the acceleration of the moving object. This is just an example of a feature vector corresponding to a trajectory point. Some of these features are available in trajectory samples observed, while others can be computed from the observations. Note that time and location information are not explicitly used for trajectory segmentation. Therefore, we exclude time and location from placing them in the feature vector.

Batch problem formulation of multidimensional time series segmentation

The problem of multidimensional time series segmentation in batch scenario (i.e., when the whole data is available for computation) can be formulated as follows: Given the trajectory feature vectors $\{\boldsymbol{z}_i\}_{i=1}^{T}$, compute the set of breakpoints of segments $\mathcal{S} = \{b_1, \ldots, b_K\} \subset \{1, \ldots, T\}$, where $K \ll T$. In the batch scenario, the goal is to compute the breakpoints of the segments, i.e., $b_1, \ldots, b_K$, where K is the number of breakpoints.

We assume that $\boldsymbol{z}$ follows a multidimensional Gaussian distribution, i.e., $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathrm{R}^n$ is the mean and $\boldsymbol{\Sigma} \in \mathrm{R}^{n \times n}$ is the covariance matrix. The probability density function of the multivariate Gaussian distribution is given by:

$$p(\boldsymbol{z}) = \frac{1}{((2\pi)^n |\boldsymbol{\Sigma}|)^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z} - \boldsymbol{\mu})\right), \quad (1)$$

where $|\cdot|$ denotes the determinant of the input matrix. We assume that the feature vector denoted by $\boldsymbol{z}_i$ corresponding to the trajectory sample at the i-th time instant is an independent sample drawn from $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, and the multidimensional time series is partitioned into K segments, and the k-th segment is identified by the parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of the Gaussian distribution.

We revisit the batch problem of multidimensional time-series segmentation when the feature vector follows a Gaussian distribution. Due to the nature of the problem at hand, now the goal is to estimate the locations of the breakpoints as well as the parameters of the segments. Thus, the number of parameters to be estimated becomes significantly large. The large number of parameters to be estimated makes the problem difficult to solve. However, there are available heuristic approaches such as [13] presenting a solution to the underlying problem.

Online multidimensional time series segmentation

In an online scenario, where the observations (trajectory samples) are streaming, when a new sample arrives, the online segmentation algorithm is required to determine whether the new sample belongs to the current trajectory segment. Each trajectory segment is characterized by a multivariate Gaussian distribution, following the approach in [13]. To make a decision about the trajectory sample, we pose a detection problem: the null hypothesis ($\mathcal{H}_0$) is that the present sample belongs to the current segment, whereas the alternate hypothesis ($\mathcal{H}_1$) is that the present sample does not belong to the current segment, mathematically presented as: $\mathcal{H}_0 : \boldsymbol{z}_i \sim p(\boldsymbol{z}_i|\mathcal{H}_0), \mathcal{H}_1 : \boldsymbol{z}_i \nsim p(\boldsymbol{z}_i|\mathcal{H}_0)$. In order to find a rule for deciding whether a trajectory sample belongs to a segment, we need a test statistic T for a threshold γ: $T(\boldsymbol{z}_i) \lessgtr^{\mathcal{H}_0}_{\mathcal{H}_1} \gamma$. However, since the parameters (mean and covariance) are unknown, it is difficult to derive such a test statistic. Therefore, we resort to other heuristic approaches. Due to multidimensional features, we cannot use common approaches for univariate Gaussian random variables, such as the 3-standard deviation rule for γ. One of the most popular approaches is to use the Mahalanobis distance, given by $d_i = ((\boldsymbol{z}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i - \boldsymbol{\mu}))^{-1/2}$, as a measure for hypothesis testing whether the sample $\boldsymbol{z}_i$ belongs to the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Observe that $d_i^2 = (\boldsymbol{z}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i - \boldsymbol{\mu})$ is a random variable. The probability distribution of d^2 is given by χ_n^2 chi-squared with n degrees of freedom [20]. Given the number of features n and the confidence level, we can find the threshold for d such that a sample belongs to a given distribution. Specifically, in order to cover $(1 - \alpha)$ probability with an ellipsoid of radius d, we need $d = \sqrt{\chi_n^2(\alpha)}$, where $\chi_n^2(\alpha)$ is the upper 100α percentile from the Chi-squared distribution with n degrees of freedom [20 Result 4.7], [21].

Reactive buffering window trajectory segmentation

Based on the strategy mentioned in the previous subsection, the proposed online segmentation algorithm works as follows. At each time instant τ, the algorithm receives a trajectory point, and a feature vector is computed. If the current feature vector is different from the previous feature vectors, the algorithm waits for w non-identical feature vectors to fill the buffer. Once the buffer is filled, the estimates of the mean and covariance are computed. The algorithm can select from a set of different alternative estimators of the mean and covariance for this purpose. Immediately after the time instant when the buffer is filled, the algorithm computes the distance of the feature vector from the distribution of the present segment in order to decide whether the current belongs to it. If the distance is smaller than a prespecified threshold, the parameters of the current segment are recursively updated using the previously estimated parameters and the current feature vector. If the distance is greater than the threshold for a consecutive r number of samples, it is decided that a new segment is started. The step-by-step procedure of the proposed RBW-TS algorithm for online multidimensional trajectory segmentation is presented in **Algorithm 1**.

Algorithm 1 Online Multidimensional Features based Trajectory Segmentation

```
Input: Buffering window length $w$, distance threshold $\gamma$, robustness factor $r$, mean $\hat{\boldsymbol{\mu}}_t$ and covariance $\hat{\boldsymbol{\Sigma}}_t$ estimation algorithm
Output: Trajectory Segments, Segment Parameters
Initialization: $t = 0$
for $\tau = 1, \ldots,$ do
    Get trajectory point $l_\tau$
    Compute the feature vector $\boldsymbol{z}_\tau$
    if $\|\boldsymbol{z}_\tau - \boldsymbol{z}_{\tau-1}\|_2 > 0$ then
        $t = t + 1$
        if the number of samples in the window $\leq w$ then
            if the number of samples in the window $= w$ then
                if estimation algorithm is provided then
                    Compute $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{\Sigma}}_t$ by the given estimation algorithm
                else
                    Empirical Mean and Covariance Estimation:
                    Sample mean $\hat{\boldsymbol{\mu}}_t = \frac{1}{w}\sum_{i=1}^{w} \boldsymbol{z}_i$
                    Sample covariance $\hat{\boldsymbol{\Sigma}}_t = \frac{1}{w-1}\sum_{i=1}^{w}(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_t)(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_t)^\top$
            else
                Store $\boldsymbol{z}_t$ in the buffer
        else
            Compute $\hat{\boldsymbol{\Sigma}}_t^{-1} = \hat{\boldsymbol{\Sigma}}_{t-1}^{-1} - \frac{\hat{\boldsymbol{\Sigma}}_{t-1}^{-1}\boldsymbol{z}_t\boldsymbol{z}_t^\top\hat{\boldsymbol{\Sigma}}_{t-1}^{-1}}{1+\boldsymbol{z}_t^\top\hat{\boldsymbol{\Sigma}}_{t-1}^{-1}\boldsymbol{z}_t}$ (Sherman-Morrison Formula)
            $d = \left((\boldsymbol{z}_t - \hat{\boldsymbol{\mu}}_{t-1})^\top\hat{\boldsymbol{\Sigma}}_t^{-1}(\boldsymbol{z}_t - \hat{\boldsymbol{\mu}}_{t-1})\right)^{\frac{1}{2}}$
            if $d < \gamma$ then
                Update the sample mean $\hat{\boldsymbol{\mu}}_t = \frac{\boldsymbol{z}_t}{t} + \frac{(t-1)\hat{\boldsymbol{\mu}}_{t-1}}{t}$
                Update the sample covariance $\hat{\boldsymbol{\Sigma}}_t = \frac{\boldsymbol{z}_t\boldsymbol{z}_t^\top}{t} + \frac{(t-1)\hat{\boldsymbol{\Sigma}}_{t-1}}{t}$
            else
                if $r$ consecutive time instants $d > \gamma$ then
                    End the current segment
                    Start a new segment
                    $t = 0$
                else
                    Ignore the outlier/noisy sample
        Output: $\boldsymbol{z}_t, \hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t$, current segment
```

In the following, we discuss the main elements of the RBW-TS algorithm.

- *Constant trajectory features:* When the dynamic point features of a trajectory, such as speed, direction, and acceleration, are not changing, this means that we are not required to initiate a new segment. In other words, when the latitude and longitude change but the dynamic features are constant, the samples corresponding to constant dynamic features belong to the same segment. Hence, the proposed algorithm ignores such samples. This strategy also avoids processing repeated samples, meaning that the same observation is repeated multiple times.
- *Buffering window length:* At the start of each new segment, the algorithm waits for w number of trajectory points in order to fill the buffer. We call this *buffering* of the online algorithm for streaming data. During the buffering stage, the algorithm does not perform segmentation activity. The buffering window length w is an important parameter for the proposed online algorithm, as the performance of the algorithm is affected by the value of w. Therefore, w needs to be carefully fine-tuned for each application. Buffering is essential for the proposed online algorithm since the first step after commencing a new segment in the proposed algorithm is to estimate the parameters of the current segment. Once the parameters are estimated, the algorithm can correctly classify the newly arrived samples based on the estimated parameters of the segment. The higher the value of w, the better the parameter estimates (mean and covariance). On the other hand, the lower the value of w, the algorithm would be able to detect short segments of trajectories. Hence, there is a trade-off between accurately estimating the parameters and detecting short segments.
- *Parameters estimation:* When the buffer fills, i.e., the number of samples in the buffer equals the length of the buffering window w, we compute the estimates of the mean and covariance of the multivariate Gaussian random variable. The most common estimates of the mean and covariance are the sample mean and unbiased sample covariance. It is important to mention that the parameters are estimated only once when the buffer is filled. In the next time instants, the parameters (mean and covariance) are updated recursively using the previous estimates and the new feature vector.
- *Mahalanobis distance:* The inverse of the covariance matrix is required in order to compute the Mahalanobis distance d. Given the inverse covariance matrix estimated at the previous time instant, the inverse covariance matrix is updated when a new sample arrives. To this end, the Sherman-Morrison formula [22] is used, which reduces the computational burden compared to computing the inverse of the current covariance matrix. Note that the inverse of the covariance matrix is computed only once at the start of a new segment when the buffer fills. The threshold for the Mahalanobis distance d is a function of the number of features and the confidence interval [21, 23, 24]. Once the number of features is specified, the threshold can be computed for a given confidence interval.
- *Inverse covariance matrix:* Analysis of the inverse covariance matrix can lead to explaining the relations among various features and also can provide insights into the trajectory segments [25]. For instance, if the (i, j)-th element of the inverse covariance matrix is zero, then the i-th and j-th variables are conditionally independent,

given the other variables. Thus, an undirected graphical model can be considered, and hence, the corresponding relationship graph among features can be drawn [26].

- *Computational complexity and memory requirements:* The proposed algorithm RBW-TS does not require high computational resources as it is an online algorithm where no iterations over the whole data are involved. At each time instant, most of the mathematical operations are either $\mathcal{O}(n)$ or $\mathcal{O}(n^2)$, where n is the number of features. Thus, the overall computational complexity of the RBW-TS becomes $\mathcal{O}(n^2)$. Similarly, there is no major memory requirement as the samples are not stored; only the estimated parameters are stored, and these parameters have a memory requirement of $\mathcal{O}(n^2)$.
- *Robustness:* To add robustness to the proposed algorithm against impulsive noise and outliers, the decision of a new segment is not based on a single sample. A sample may be an outlier or a noisy sample, and it is usually highly unlikely to receive multiple consecutive noisy samples. Therefore, the idea is that if a consecutive r number of samples do not fulfill the distance condition, then a new segment is started. The range of values for r is $2 \leq r \leq w$. A commonly acceptable value for r would be $r = 2$.

Robust estimation of mean and covariance

The length of the buffering window w should be greater than the number of features n. Otherwise, the estimated covariance matrix will no longer be positive definite. In addition, for a small value of w but greater than the number of features n, the classical estimators for the mean and covariance suffer from performance degradation. Therefore, to avoid ill-conditioned and poor-quality covariance matrix estimates, we employ the following estimators in our proposed algorithm.

- *Ledoit-Wolf estimator:* A well-conditioned and accurate estimator of the covariance matrix is proposed in [27]. The proposed estimator is distribution-free, simple to compute, and easy to interpret. In this method, a scaled version of the identity matrix, which can be treated as a regularization term, is added to the sample covariance matrix. The magnitude of shrinkage is computed by the Ledoit-Wolf formula. The regularization ensures that the estimated covariance matrix is always positive definite [28].
- *Shrinkage-based estimator:* A shrinkage-based estimator for covariance estimation is proposed in [29], where the shrinkage parameter need to be specified. The estimated covariance matrix will be positive definite and hence invertible.
- *Oracle approximating shrinkage (OAS) estimator:* OAS estimator, proposed in [29], is an improved version of Ledoit-Wolf estimator. First, a closed-form formula for the oracle estimator is derived under the Gaussian assumption. Then, an approximation method is used to derive the OAS estimator. Its convergence rate and accuracy are significantly improved when the data follows Gaussian distribution.
- *Minimum covariance determinant (MCD) estimator:* MCD, introduced in [30], is a robust estimator of the covariance matrix. The idea of the MCD estimator is that the data may contain outliers, and these outliers affect the estimates. Hence, in

MCD, the estimates are computed by considering only a subset of data, which has the minimum determinant of the covariance.

- *Elliptic envelope estimator:* Elliptic envelope estimator is also a robust estimator of the covariance proposed in [31], where the outliers are decided by drawing an ellipse around the data.
- *Graphical lasso estimator:* Given the sample covariance matrix, an inverse covariance matrix also known as the precision matrix, is estimated by imposing $\| \cdot \|_1$, i.e., sparsity regularization on the values of the inverse covariance matrix [32].

This completes the discussion about the proposed segmentation algorithm. Next, we present the numerical results for the proposed algorithm.

Experimental evaluation

In this section, we describe the experimental evaluation of the proposed RBW-TS algorithm. We present the research questions investigated, the performance metrics considered in the numerical evaluation, as well as the data sets used in the experiments. Finally, we report and discuss the numerical results.

Research questions

To evaluate the proposed algorithm, we consider the following research questions:

- *RQ1. Detecting True Breakpoints in Streaming Data Scenario:* Can the proposed algorithm be applied in big data applications where the data is streaming? Is the proposed algorithm capable of detecting the true breakpoints in these applications?
- *RQ2. Comparing the Effect of Mean and Covariance Estimators in RBW-TS:* How do different mean and covariance estimators used in the RBW-TS algorithm perform?
- *RQ3. Comparing to Batch Algorithms:* Is the performance of the proposed online algorithm in terms of the harmonic mean of purity and coverage comparable to batch multidimensional segmentation algorithms?
- *RQ4. Comparing to Online Algorithms:* Is the performance of the proposed online algorithm in terms of the harmonic mean of purity and coverage comparable to online segmentation algorithms?
- *RQ5. Impact of Trajectory Features on Segmentation:* How do different combinations of trajectory point features affect the performance of the segmentation algorithm?

Performance metrics

Purity and coverage were formally introduced as evaluation criteria for segmentation algorithms in [33]. We measure the purity and coverage of the estimated segments by comparing them with the ground truth data. Purity shows how much of a trajectory segment is divided correctly as compared to a subject-matter expert segmentation. The coverage quantifies how much the algorithm can cover the segments tagged by a subject-matter expert. Purity is mathematically defined as in [33]:

$$P(S, \Lambda_L) = \frac{1}{k} \sum_{i=1}^{k} \max_{j \in [1,L]} \frac{N_{ij}}{N_i}, \tag{2}$$

where S is the set of segments discovered by the segmentation algorithm, Λ_L is the set of labels (a point feature) provided by a subject-matter expert, k is the number of discovered segments, L is the number of expert labels, and N_{ij} is the number of trajectory points inside a segment s_i with label λ_j. Coverage is defined as [33]:

$$C(S, \Psi_v) = \frac{1}{v} \sum_{i=1}^{v} \max_{j \in [1,k]} \frac{N_{\psi_i \cap s_j}}{N_i}, \tag{3}$$

where S is the set of segments discovered by the segmentation algorithm, Ψ_v is the set of segments by a subject-matter expert, $N_{\psi_i \cap s_j}$ is the number of trajectory points of the segment s_j that belongs to the ψ_i segment, and N_i is the total number of points of the identified segment with segment identifier equal to ψ_i segment. Since the purity and coverage are two orthogonal metrics, we report the harmonic means of purity and coverage, given by $2PC/(P + C)$ where P and C denote purity and coverage respectively, to compare the performance of different algorithms [34].

Evaluation data sets

We apply our online segmentation algorithm on three real-world data sets described next.

Geolife data set

Geolife is a well-known data set for mobility data research collected by Microsoft Research Asia [35]. Trajectory information of object movements is recorded by GPS devices and the transportation mode (labels) such as walking, taxi, bus, car, bike, and subway are included in the data set. In our experiments, we use a subset of the Geolife data set containing 12,955 trajectory points and 181 segments. For this data set, we use the transportation mode as the ground truth for creating the segments. For each trajectory point, several features, such as speed, acceleration, jerk, and bearing, are computed, and a feature vector is formed.

Maritime data set

We use a proprietary maritime data set consisting of AIS trajectory data collected for monitoring maritime traffic, where the vessels transmit their static information (MMSI, IMO number, etc.) as well as their location, speed, direction, and other attributes. The maritime data set is collected and anonymized by Navtor AS, Norway. For this data set, only two features, i.e., speed and direction, are used for the purpose of trajectory segmentation. The data set contains 7 trajectories from Ålesund to Måløy, and each trajectory contains 200 samples on average. These selected trajectories have an adequate number of samples in order to evaluate the proposed algorithm.

ATC pedestrian tracking data set

ATC Pedestrian Tracking data set[1] is introduced in [36], which includes data related to trajectories based on human movements obtained in a shopping center in Osaka, Japan. The data set includes time, person ID, position (x, y, z), velocity, angle of motion, and facing angle features, and these features are calculated by processing the raw data obtained through 3D sensors. In our experiments, we use 11 random trajectories, each including an average of 794 data points. There are no labels available for segments associated with this data set. We have generated labels for this data set by partitioning velocity and angle of motion considering minimum and maximum values into three and four intervals, respectively. Changes on the intervals for consecutive samples are accepted as a segment's starting point.

Results and analysis

In this section, we report and discuss the numerical results addressing the five research questions. We report the results for the three mentioned real-world data sets, where the performance of RBW-TS is evaluated in terms of purity, coverage, and the harmonic mean of purity and coverage.

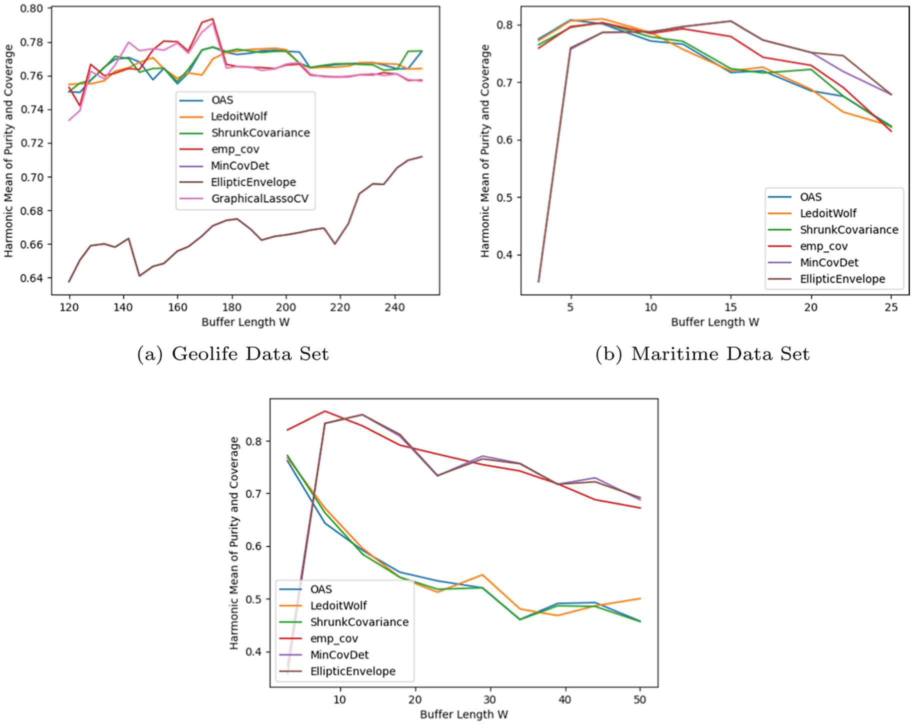
RQ1: detecting true breakpoints in streaming data scenario

RQ1 deals with the performance of the online segmentation algorithm in terms of purity, coverage, and the harmonic mean of purity and coverage, when the data is streaming.

Figure 1 depicts a comparison of the performance of the various estimators for both data sets. In Fig. 1a, we present the harmonic mean of purity and coverage of the proposed algorithm applied to the Geolife data set for different values of the buffer length w for multiple types of mean and covariance estimation algorithms exploited in RBW-TS. A subset of the Geolife data set is divided into 10 parts, and the results are averaged over them. Observe that the best buffer length for this data set is approximately 170 for all the variants of the proposed algorithm. When the individual results of the estimators are analyzed, a general trend of increasing/decreasing in the harmonic mean of purity and coverage is observed. However, each curve is not locally monotonically increasing/decreasing before/after the best buffer length w. This is due to several reasons. First, due to the nature of the harmonic mean of purity and coverage, different rates of increasing/decreasing of purity and coverage would result in different trends of the harmonic mean of purity and coverage. Second, the data contain various segment lengths and trajectories of different sizes. For the maritime data set, Fig. 1b presents the harmonic mean of purity and coverage against buffer size for different covariance estimation algorithms for RBW-TS. We can observe that there is an optimal buffer size w that yields the highest value of the harmonic mean of purity and coverage for the maritime data set too. Note that the graphical lasso estimator is excluded for this data set because, for smaller values of buffer size, the covariance is not accurately estimated, as sometimes the feature vectors of the samples can be very similar.

For the ATC data set, Fig. 1c presents the curve of the harmonic mean of purity and coverage for different values of buffer length. Here, the graphical lasso estimator is again

[1] The data set is publicly available at https://dil.atr.jp/crest2010_HRI/ATC_dataset/.



(a) Geolife Data Set

(b) Maritime Data Set

(c) ATC Data Set

Fig. 1 Harmonic mean of purity and coverage of RBW-TS for different covariance estimation methods vs. Buffer Length for the **a** Geolife, **b** Maritime, and **c** ATC data sets. Hyperparameters values: $1 - \alpha = 0.95, n = 3$

excluded from the comparison because of the unstable behavior for low values of the buffer length. For the empirical covariance estimator, minimum covariance determinant, and elliptical envelope estimator, the optimal values of buffer length lie between 3 and 20. For the remaining estimators, the optimal value of buffer length for the criterion of the harmonic mean of purity and coverage is 3, which suggests that the trajectories have short segments.

To illustrate the results of RBW-TS, a snapshot of the AIS data-based trajectories, as well as the estimated breakpoints of the segments in the region of Ålesund and Måløy plotted in the QGIS software, is depicted in Fig. 2. The true breakpoints are depicted in orange, whereas the estimated breakpoints of the segments are shown in green. The true breakpoints are drawn such that they will be valid for most of the trajectories since they are drawn by considering several trajectories. The figure illustrates that when the two features of speed and direction are used, the proposed segmentation algorithm yields results that are aligned with human intuition. In other words, the estimated breakpoints of the segments are near the place where the direction or the speed changes.

RQ2: comparing the performance of estimation algorithms

We report the performance of all the covariance estimation methods in Figures 3, 4, and 5. First, the box plot for the coverage is presented in Fig. 3a. The median values of the coverage vary across different estimation methods. For the first four estimation methods, i.e., empirical covariance, OAS, Ledoit-Wolf, and shrunk covariance,

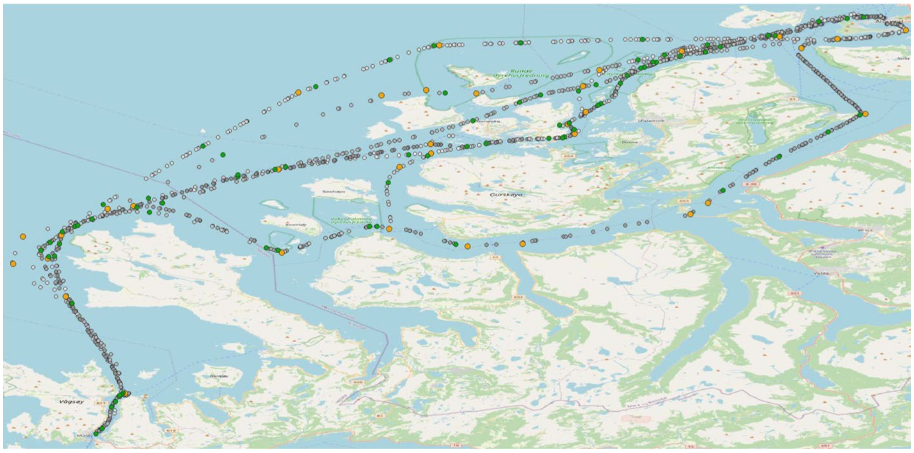
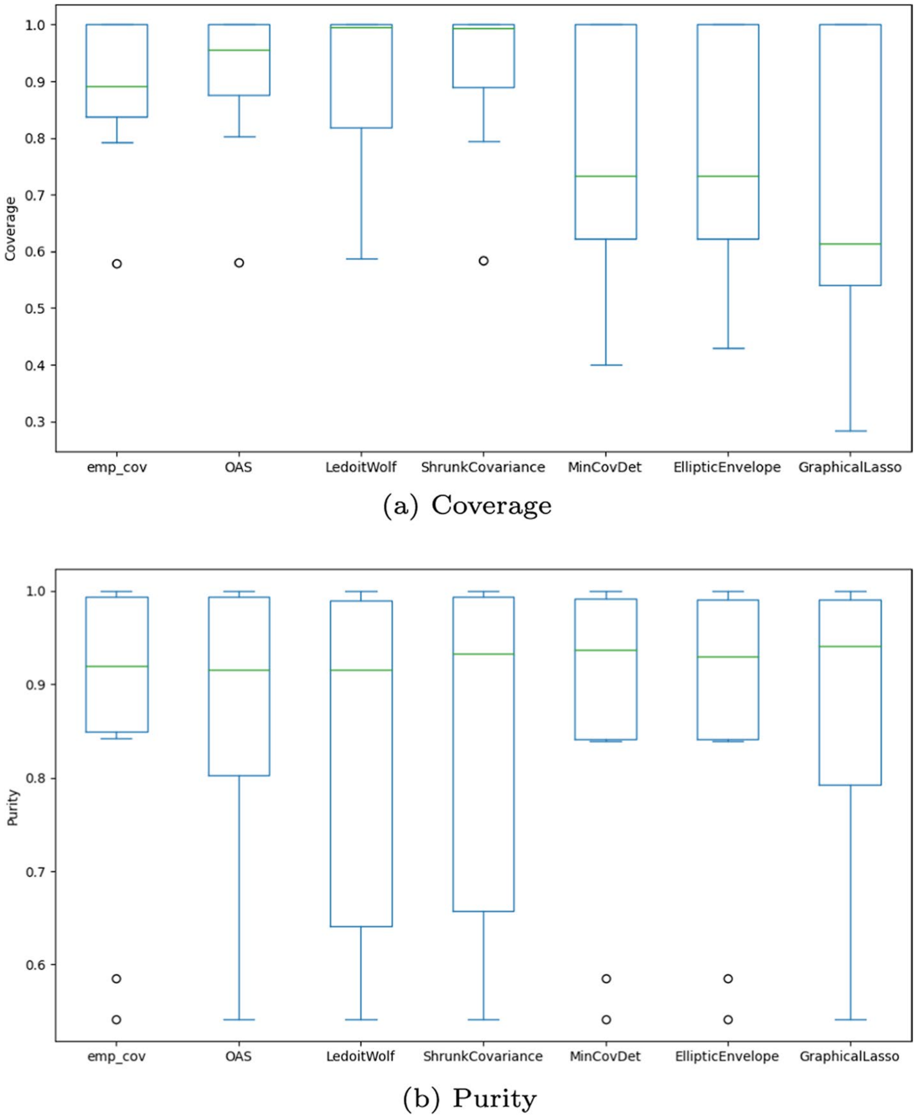


Fig. 2 A snapshot of the result of the segmentation algorithm RBW-TS applied to the maritime data. The orange dots represent the true breakpoints of the segments, and the green dots represent the estimated breakpoints. Here the shrunk covariance estimator is employed to estimate the parameters (mean and covariance) in RBW-TS

the median value of the coverage is greater than or equal to 0.9. The box plot for the purity is presented in Fig. 3b. The median value for the purity is greater than 0.9 for all the estimators. Finally, the harmonic mean of purity and coverage is shown in Fig. 4. There is no significant difference in the results of some of the covariance estimation algorithms. However, these covariance estimation algorithms can produce different results for different types of data. A median value of the harmonic mean of purity and coverage of ≈ 0.87 is observed for four of the estimators. For the remaining three, the harmonic mean of the purity and the coverage is low due to lower values of the coverage. These variations among the estimators used in the the RBW-TS are expected, as all of them follow different strategies to estimate the mean and covariance. By having these different choices of estimators in RBW-TS, one can select the estimator that works the best among the available estimators for a given data set.

For the maritime data set, the harmonic mean of purity and coverage for all the different alternative parameter estimators that can be used in RBW-TS are presented in Fig. 5. All variants of the RBW-TS yield the harmonic mean of purity and coverage values in different ranges. However, their performances are comparable. In other words, all alternative estimators in RBW-TS are able to estimate the mean and covariance with a competitive level of accuracy, hence resulting in acceptable values of the harmonic mean of the purity and coverage.

We have also evaluated the proposed algorithm on the ATC data set in Fig. 6. The empirical covariance, minimum covariance determinant, and elliptic envelope estimators are able to detect the segments more accurately than the remaining estimators. Note that the ATC data set contains samples with high frequency and it may be very sensitive to the selection of the hyperparameters such as buffer length.



(a) Coverage

(b) Purity

Fig. 3 Coverage and purity of RBW-TS for different covariance estimation methods applied to the Geolife data set. Hyper-parameters: $w = 250, 1 - \alpha = 0.99, n = 6$

RQ3: comparison with batch algorithm

To compare the performance of the proposed online segmentation algorithm with a batch algorithm, we selected the greedy Gaussian segmentation (GGS) algorithm proposed in [37], which uses multidimensional features for segmentation. Figure 7a presents a comparison of GGS and two variants of the proposed algorithm in terms of the harmonic mean of purity and coverage for the Geolife data set. It can be observed that the batch algorithm GGS obtains a higher median value of the harmonic mean of purity and coverage at the cost of higher computational complexity as the computational complexity of GGS is $\mathcal{O}(n^3)$, where n is the number of trajectory features. Note that the computational complexity of RBW-TS is $\mathcal{O}(n^2)$, as mentioned earlier. Online algorithms have competitive results despite processing the data in a streaming

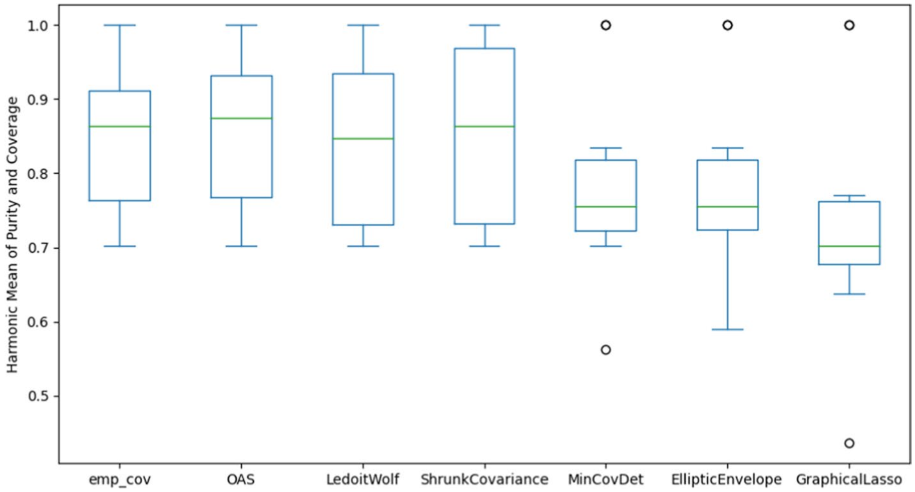


Fig. 4 Harmonic mean of purity and coverage of RBW-TS for different covariance estimation methods applied to the Geolife data set. Hyper-parameters: $w = 250, 1 - \alpha = 0.99, n = 6$

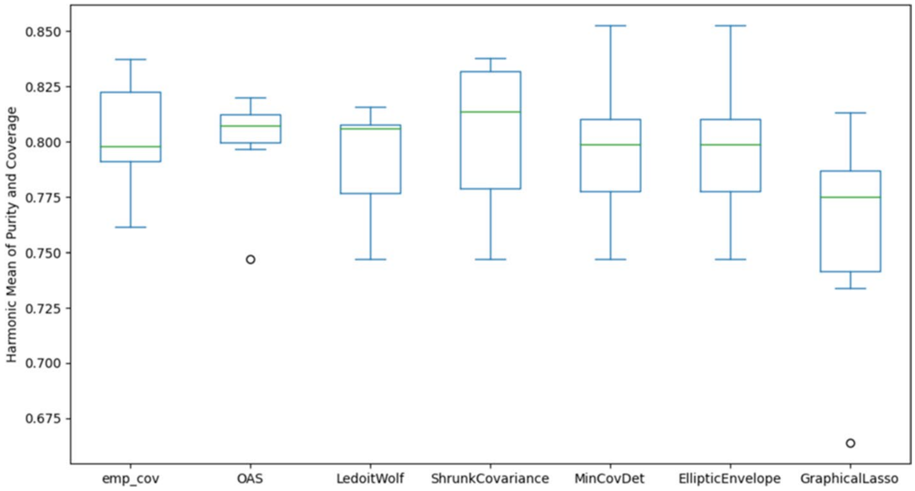


Fig. 5 Harmonic mean of purity and coverage of RBW-TS for different covariance estimation methods applied to the maritime data set. Hyper-parameters values: $w = 7, 1 - \alpha = 0.99, n = 3$

fashion, meaning that the entire data set is not available to the algorithm at once at the beginning of the processing. To demonstrate that the number of breakpoints affects the performance of GGS, Fig. 7b presents a comparison among different values of the number of breakpoints. Observe that there is an optimal value for the number of breakpoints for GGS. Therefore, tuning this parameter is essential. However, due to the nature of this parameter, fixing it in a scenario where we do not know the size of the data is challenging. In contrast, the distance threshold for RBW-TS does not depend on the size of the data and is only related to the dissimilarity of the samples.

For the maritime data set, we repeat the same experiment, and the comparison in terms of the harmonic mean of purity and coverage is shown in Fig. 8. Note that the median values of the harmonic mean of purity and coverage for GGS and our

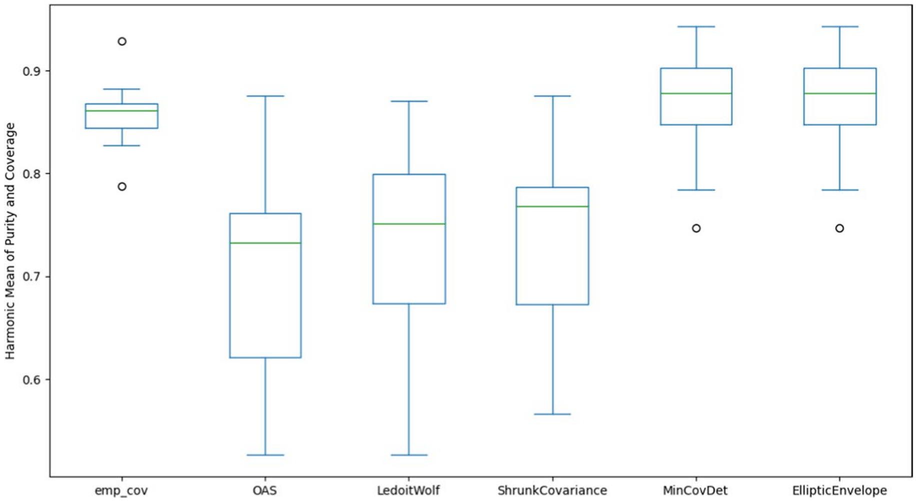
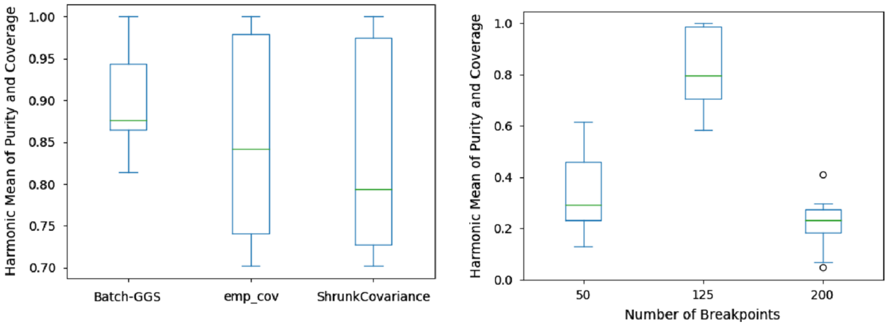


Fig. 6 Harmonic mean of purity and coverage of RBW-TS for different covariance estimation methods applied to the ATC data set. Hyper-parameters: $w = 5, 1 - \alpha = 0.9, n = 2$

proposed algorithm's variants, that is, empirical covariance and shrinkage covariance, are comparable. The slight edge in the performance of GGS is due to the fact that the batch algorithm has the access to full data in advance and runs several iterations over the whole data, thus incurring more computational complexity than online algorithms. It is pertinent to mention that for GGS, the value of the regularization parameter and the number of breakpoints are selected such that it yields the highest value of the harmonic mean of purity and coverage.

The above experiment is also repeated for the ATC data set, and the results are presented in Fig. 9. The figure shows that the performance of RBW-TS in terms of the harmonic mean of purity and coverage is comparable to that of the batch GGS algorithm for two estimators: empirical covariance and minimum covariance determinant. The result of RBW-TS with the shrunk covariance estimator is added to underline that each estimator behaves differently for different data sets.



(a) Harmonic mean of purity and coverage of RBW-TS and GGS batch algorithm applied to the Geolife data set. Hyper-parameters: $w = 200, 1 - \alpha = 0.9, n = 3$ (i.e., speed, bearing, and acceleration), number of breakpoints for GGS = 200, the value of the regularization parameter for GGS = 0.8.

(b) Harmonic mean of purity and coverage of GGS batch algorithm applied to the Geolife data set for multiple values of the number of breakpoints. Hyper-parameters: $n = 3$ (i.e., speed, bearing, and acceleration), the value of the regularization parameter for GGS = 0.01.

Fig. 7 Comparison of RBW-TS with batch algorithm GGS

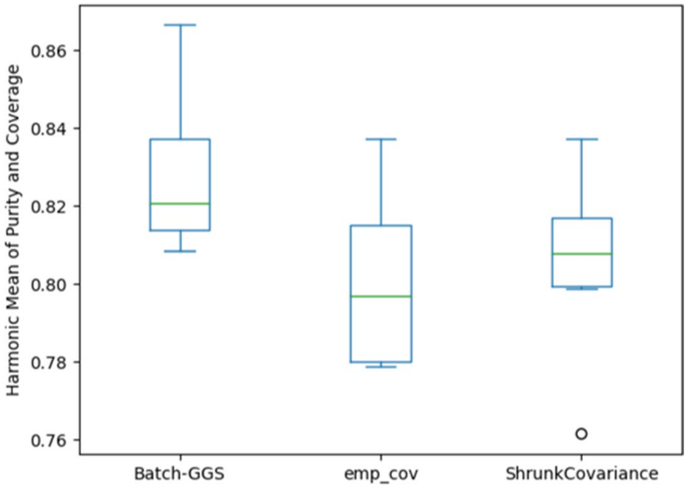


Fig. 8 Harmonic mean of purity and coverage of RBW-TS and GGS batch algorithm applied to the maritime data set. Hyper-parameters: $w = 7, 1 - \alpha = 0.9, n = 3$, number of breakpoints for GGS = 10, the value of the regularization parameter for GGS = 0.001

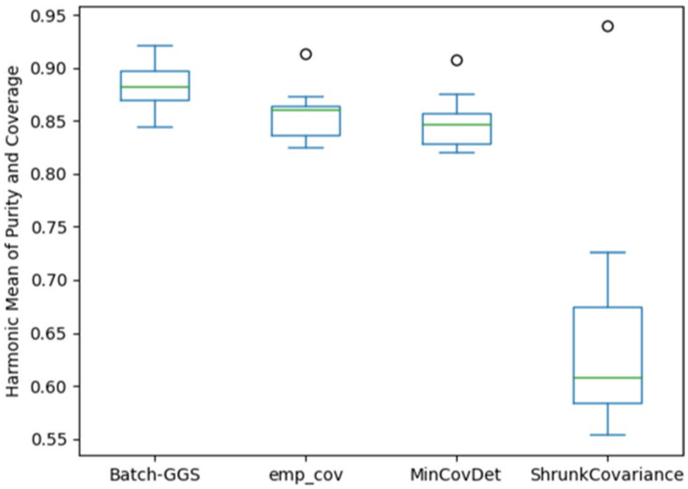


Fig. 9 Harmonic mean of purity and coverage of RBW-TS and GGS batch algorithm applied to the ATC data set. Hyper-parameters: $w = 10, 1 - \alpha = 0.9, n = 3$ (i.e., speed and horizontal and vertical components of the direction of the movement), number of breakpoints for GGS = 50, the value of the regularization parameter for GGS = 0.01

The performance (in terms of the harmonic mean of purity and coverage) of RBW-TS is lower yet comparable to SWS (according to what is reported in [34]) for the same data set. This is expected since the computational complexity of the proposed online algorithm is lower than the aforementioned batch algorithms.

RQ4: comparison with online algorithm

In this research question, we compare the performance of RBW-TS with Thresholds [19] that tracks speed and orientation changes with thresholds to capture segments. We choose Thresholds for comparison as it considers speed and orientation thresholds (dv_s and $d\varphi_s$, respectively) to decide whether a data point in the trajectory represents a

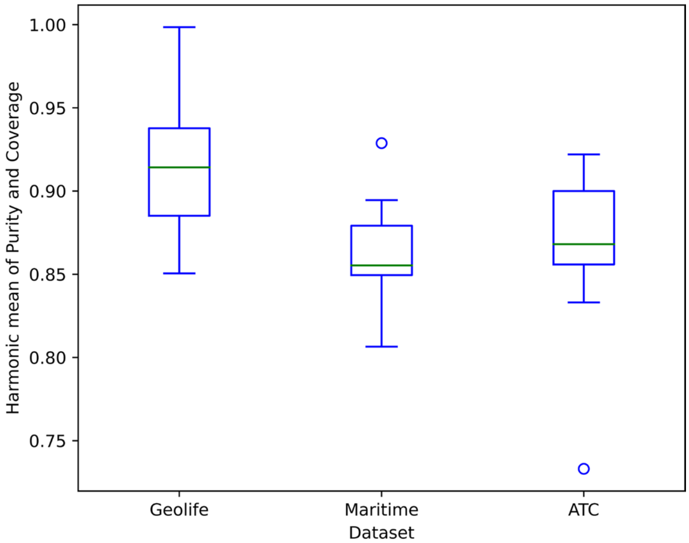


Fig. 10 Harmonic mean of purity and coverage of the Thresholds algorithm [19] applied to the Geolife, maritime, and ATC data sets. Hyper-parameters for Geolife: $dv_s = 20$, Maritime: $dv_s = 5$, $d\varphi_s = 0.5$, ATC: $dv_s = 3$, $d\varphi_s = 1.58$

significant change. The Thresholds algorithm constructs velocity vectors based on previously observed data points (both recently sampled points and trajectory points) and defines a joint safe area for the current candidate data point. In the case of the candidate data point being in the joint safe area, this candidate point is discarded. Otherwise, it is assumed to be a significant deviation considering the thresholds and is added to the sample. Our experiments consider such significant points as the beginning of segments. In our experiments, we use the Haversine distance for the maritime and Geolife data sets for the aforementioned calculations, while the distance formula is used for the ATC data set.

Figure 10 depicts the harmonic mean of purity and coverage for the Thresholds algorithm for three data sets used in this study. Note that we report empirically set the speed (m/s) and orientation (radian) thresholds that yield the best results for each data set. As can be seen from the figure, the Thresholds algorithm yields a value of the median of the harmonic mean of purity and coverage of around 0.86 for the maritime and ATC data sets, while it is around 0.91 for the Geolife data set. Considering the used thresholds for the Geolife data set, using only the speed threshold provides the best results for this data set as the segments for the transportation mode depend mainly on the speed feature. Considering the scores obtained by RBW-TS in Figs. 4-6 for different data sets, it can be deduced that both Thresholds and RBW-TS have the ability to detect the segments in different data sets. However, RBW-TS is more generic and flexible as different estimators can be employed inside it for different scenarios. Moreover, RBW-TS can include multidimensional features for the purpose of segmentation, while the Thresholds algorithm can only include speed and velocity. We use the publicly available code repository[2] for the Thresholds algorithm's implementation.

[2] https://github.com/uestc-db/traj-compression.

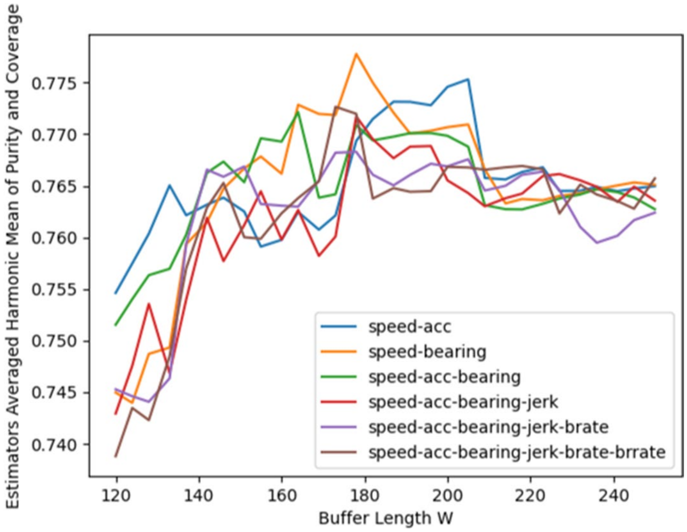


Fig. 11 Harmonic mean of purity and coverage of RBW-TS Averaged over covariance estimation methods applied to the Geolife data set. Hyperparameters values: $1-\alpha=0.9$

RQ5: trajectory point features impact on segmentation

As previously discussed, the selection of trajectory features is important for segmentation. Hence, RQ5 addresses the impact of feature selection on the performance of the segmentation algorithm. Figure 11 compares different sets of features (i.e., speed, acceleration, jerk, bearing, rate of change of bearing, and second derivative of bearing) considered for segmentation in the case of the Geolife data set. For each combination of features, the harmonic mean of purity and coverage is averaged across the estimators. It can be noticed that the speed and bearing features yield the highest values of the harmonic mean among all sets of features for the given data set.

The corresponding results of the impact of feature selection for the maritime data set are not presented, as the true labels of the segments of trajectories are generated by only taking into account the speed and the direction of the vessels. Therefore, in all the previously presented results based on the maritime data set, we only use speed and direction. Similarly, for the ATC data set, the results related to the feature selection are not presented as there are only two relevant features available.

Conclusions and future work

In this work, we propose an online trajectory segmentation algorithm RBW-TS that can incorporate multiple trajectory features. Given the trajectory features relevant to the target application, the proposed algorithm can be applied to detect segments where the data is streaming. Due to low computational requirements, the proposed online algorithm can be applied in applications where the multidimensional trajectory data is generated at a high frequency. The performance of RBW-TS has been evaluated on three real-world data sets and compared with relevant online and batch segmentation approaches. The numerical results presented in the paper elucidate the competitiveness of the proposed online segmentation algorithm.

The proposed online segmentation algorithm for multidimensional time series data has both advantages and limitations, as expected. However, the advantages have a significant impact in certain scenarios, for instance, when the data is streaming. One of the challenges is setting the buffer size as a hyper-parameter. Unlike other algorithms, tuning buffer size is easy and does not depend on the size of data or the length of trajectories. However, as detecting different length segments depends on the buffer size selection, it is essential to set it effectively. For instance, in the case of large buffer size, segments smaller than the buffer size cannot be detected. Consequently, the buffer size needs to be carefully selected, taking into account the data set (i.e., underlying characteristics of the domain). Second, the RBW-TS can have difficulty detecting segments with a smaller size than the number of the point features (i.e., the degree of multidimensionality) of the trajectories. For instance, when the number of processed features is high, small segments could remain undetected by the algorithm. Future potential extensions of the proposed algorithm include automatic selection of the buffer size, where the buffer size will be jointly learned with segments and adjusted according to the target application. Another potential direction for future work is online activity recognition, e.g., transportation mode, etc., based on the parameters of a segment.

Acknowledgements
We are thankful to Mohammad Etemad for a useful discussion on the topic of the paper.

Author contributions
BZ and DM wrote the main manuscript. DA helped in the writing and experiments section. TK helped in providing and preparing the maritime data set and reviewed the paper. All the authors reviewed the paper.

Funding
This work is supported by ECSEL JU under grant agreement No 101007260 and the Research Council of Norway under Grant Agreement No 329090.

Availability of data and materials
The maritime data set is owned by Navtor AS, Norway. The anonymized data can be made available on special request.

Declarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Competing interests
The authors declare that they have no competing interests.

Received: 28 May 2022 Accepted: 4 July 2023
Published online: 24 July 2023

References

1. Tu E, Zhang G, Rachmawati L, Rajabally E, Huang G-B. Exploiting ais data for intelligent maritime navigation: a comprehensive survey from data to methodology. IEEE Trans Intell Trans Syst. 2018;19(5):1559–82.
2. Lee E, Mokashi AJ, Moon SY, Kim G. The maturity of automatic identification systems (AIS) and its implications for innovation. J Marine Sci Eng. 2019;7(9):287.
3. Dominguez AG. "Smart Ships": Mobile applications, cloud and big data on marine traffic for increased safety and optimized costs operations. In: 2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation; 2014. pp. 303–308.
4. Alessandrini A, Alvarez M, Greidanus H, Gammieri V, Arguedas VF, Mazzarella F, Santamaria C, Stasolla M, Tarchi D, Vespe M. Mining vessel tracking data for maritime domain applications. In: 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW); 2016. pp. 361–367.

5. Rong H, Teixeira A, Soares CG. Data mining approach to shipping route characterization and anomaly detection based on ais data. Ocean Eng. 2020;198:106936.
6. Zheng Y, Zhang L, Ma Z, Xie X, Ma W-Y. Recommending friends and locations based on individual location history. ACM Trans Web (TWEB). 2011;5(1):5.
7. Palma AT, Bogorny V, Kuijpers B, Alvares LO. A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the 2008 ACM Symposium on Applied Computing; 2008. pp. 863–868.
8. Rocha JAM, Times VC, Oliveira G, Alvares LO, Bogorny V. DB-SMoT: A direction-based spatio-temporal clustering method. In: 2010 5th IEEE International Conference on Intelligent Systems (IS); 2010. pp. 114–119.
9. Soares Júnior A, Moreno BN, Times VC, Matwin S, Cabral LDAF. GRASP-UTS: an algorithm for unsupervised trajectory segmentation. Int J Geograph Inform Sci. 2015;29(1):46–68.
10. Etemad M, Júnior AS, Hoseyni A, Rose J, Matwin S. A trajectory segmentation algorithm based on interpolation-based change detection strategies. In: Proceedings of the Workshops of the EDBT/ICDT Joint Conference, Lisbon, Portugal; 2019.
11. Etemad M, Soares A, Etemad E, Rose J, Matwin S. SWS: an unsupervised trajectory segmentation algorithm based on change detection with interpolation kernels. GeoInformatica. 2020. https://doi.org/10.1007/s10707-020-00408-9.
12. Yuan Z, Liu J, Zhang Q, Liu Y, Yuan Y, Li Z. Prediction and optimisation of fuel consumption for inland ships considering real-time status and environmental factors. Ocean Eng. 2021;221:108530.
13. Hallac D, Nystrup P, Boyd S. Greedy gaussian segmentation of multivariate time series. Adv Data Anal Classif. 2019;13(3):727–51.
14. Hallac D, Vare S, Boyd S, Leskovec J. Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery And Data Mining, pp. 215–223. ACM, Halifax NS Canada; 2017.
15. Ertl B, Meyer J, Schneider M, Streit A. Semi-supervised time point clustering for multivariate time series. Proceedings of the Canadian Conference on Artificial Intelligence; 2021.
16. Matsubara Y, Sakurai Y, Faloutsos C. AutoPlait: automatic mining of co-evolving time sequences. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird Utah USA; 2014. pp. 193–204.
17. Singhal A, Seborg DE. Clustering of multivariate time-series data. In: Proceedings of the 2002 American Control Conference, Anchorage, AK, USA; 2002. pp. 3931–39365.
18. Zhang Y-Q, Shi G-Y, Li S, Zhang S-K. Vessel trajectory online multi-dimensional simplification algorithm. J Navig. 2020;73(2):342–63.
19. Potamias M, Patroumpas K, Sellis T. Sampling trajectory streams with spatiotemporal criteria. In: 18th International Conference on Scientific and Statistical Database Management (SSDBM'06); 2006. pp. 275–284. IEEE.
20. Johnson RA, Wichern DW, et al. Applied Multivariate Statistical Analysis, vol. 6. UK: Pearson London; 2014.
21. Bajorski P. Statistics for Imaging, Optics, and Photonics, vol. 808. Hoboken, New Jersey: John Wiley and Sons; 2011.
22. Sherman J, Morrison WJ. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. Ann Math Stat. 1950;21:124–7.
23. Chew V. Confidence, prediction, and tolerance regions for the multivariate normal distribution. J Am Stat Assoc. 1966;61(315):605–17.
24. Wang B, Shi W, Miao Z. Confidence analysis of standard deviational ellipse and its extension into higher dimensional euclidean space. PloS ONE. 2015;10(3): e0118537.
25. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2007;9(3):432–41.
26. Hallac D, Park Y, Boyd S, Leskovec J. Network Inference via the Time-Varying Graphical Lasso. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery And Data Mining, Halifax NS Canada; 2017. pp. 205–213.
27. Ledoit O, Wolf M. A well-conditioned estimator for large-dimensional covariance matrices. J Multivar Anal. 2004;88(2):365–411.
28. Lam C. High-dimensional covariance matrix estimation. Wiley Interdiscip Rev Comput Stat. 2020;12(2): e1485.
29. Chen Y, Wiesel A, Eldar YC, Hero AO. Shrinkage algorithms for mmse covariance estimation. IEEE Trans Signal Proc. 2010;58(10):5016–29.
30. Rousseeuw PJ. Least median of squares regression. J Am Stat Assoc. 1984;79(388):871–80.
31. Rousseeuw PJ, Driessen KV. A fast algorithm for the minimum covariance determinant estimator. Technometrics. 1999;41(3):212–23.
32. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2008;9(3):432–41.
33. Soares Júnior A, Moreno BN, Times VC, Matwin S, Cabral LDAF. GRASP-UTS: an algorithm for unsupervised trajectory segmentation. Int J Geogr Inform Sci. 2015;29(1):46–68.
34. Etemad M. Novel algorithms for trajectory segmentation based on interpolation-based change detection strategies. Halifax, Nova Scotia: Dalhousie University; 2020.
35. Zheng Y, Xie X, Ma W-Y. Understanding mobility based on gps data. In: Proceedings of the 10th ACM Conference on Ubiquitous Computing (Ubicomp); 2008.
36. Bršćić D, Kanda T, Ikeda T, Miyashita T. Person tracking in large public spaces using 3-d range sensors. IEEE Trans Human-Machine Syst. 2013;43(6):522–34.
37. Hallac D, Nystrup P, Boyd S. Greedy gaussian segmentation of multivariate time series. Adv Data Anal Classific. 2018;3(3):727.

Publisher's Note