Geoinformatica
https://doi.org/10.1007/s10707-021-00449-8

Individual and collective stop-based adaptive trajectory segmentation

Agnese Bonavita[1] · **Riccardo Guidotti**[2] · **Mirco Nanni**[3]

Received: 30 September 2020 / Revised: 16 April 2021 / Accepted: 12 August 2021 /
Published online: 08 October 2021

Abstract
Identifying the portions of trajectory data where movement ends and a significant stop starts is a basic, yet fundamental task that can affect the quality of any mobility analytics process. Most of the many existing solutions adopted by researchers and practitioners are simply based on fixed spatial and temporal thresholds stating when the moving object remained still for a significant amount of time, yet such thresholds remain as static parameters for the user to guess. In this work we study the trajectory segmentation from a multi-granularity perspective, looking for a better understanding of the problem and for an automatic, user-adaptive and essentially parameter-free solution that flexibly adjusts the segmentation criteria to the specific user under study and to the geographical areas they traverse. Experiments over real data, and comparison against simple and state-of-the-art competitors show that the flexibility of the proposed methods has a positive impact on results.

Keywords Mobility data mining · Segmentation · User modeling

1 Introduction

Thanks to the wide diffusion of localization technologies and mobile services based on the positioning of users and devices, the availability of mobility traces is increasing fast in several application domains. Location-based services provided through smartphones are nowadays extremely popular, from nearby restaurant suggestions to travel assistants, and in the near future all circulating vehicles will be equipped with localization capabilities for

✉ Agnese Bonavita
agnese.bonavita@sns.it

Riccardo Guidotti
riccardo.guidotti@unipi.it

Mirco Nanni
mirco.nanni@isti.cnr.it

[1] Scuola Normale Superiore, Pisa, Italy

[2] University of Pisa and ISTI-CNR, Pisa, Italy

[3] ISTI-CNR, Pisa, Italy

their continuous monitoring. The vast amounts of data produced open the door to several useful applications that can yield better services, more sustainable cities, improved living conditions, etc. All this starts from appropriate mobility analysis operations able to transform raw data into useful information in the form of a deeper domain knowledge, patterns, models, and forecasts.

In mobility analytics one of the fundamental concepts is *movement*, namely the part of mobility data that describes a transfer from one place where the individual (or the object) was staying, to another one were the user will stop. Identifying movements in the raw stream of positions, for instance the continuous flow of GPS traces of a vehicle, is essential to many tasks, as it enables the development of mobility data models [19, 38] and applications like carpooling [2, 17], trajectory prediction [45] and car crash prediction [16], which are based on stop locations and the transitions between them. Errors in identifying stops and movements greatly affect the results of modeling, and therefore the overall performances. While it is simple to define a *stop* in geometrical terms, it is much less clear how to define *significant stops*, i.e. stops that might have some meaning for the user (for instance, stopping to do some activity before leaving), as opposed to stops that are simply incidental (for instance, due to a small traffic jam).

Practitioners in the mobility analytics domain defined several simple strategies to select stops in the mobility data stream (a brief account of literature on this topic is provided in the next section), each of them apparently capturing well some specific concept or some application-specific idea of stop. For instance, some solutions simply identify the moments where the object did not move, based on some thresholds, while others select the stops that have a duration compatible with some specific task, for instance discarding stops at a supermarket if their duration is physically too short to be able to enter, buy and exit. However, most existing solutions suffer from two important limitations: *(i)* they are based on critical thresholds that the user needs to choose accurately, and in most cases it is difficult to understand what value is the best; *(ii)* such thresholds are global, i.e. the same threshold value applies to all the individuals, irrespective of any distinctive characteristics they might have or of the places they visit. The reason of the latter is that, while an overall evaluation might be performed to select a global threshold, doing that separately for each individual might be impossible if their number is huge.

In this work[1] we try to overcome the limitations highlighted above, providing a general methodology that inspects the mobility of the individual, and identifies segmentation thresholds that apparently match their mobility features. The process allows to get rid of virtually any input parameter (the only needed one is a spatial threshold that depends to location accuracy, and therefore is basically fixed for GPS data, ref. Section 4.2), adapts thresholds to each individual and is completely automatic, thus applicable to large pools of users. Moreover, we extend the aforementioned approach by observing the typical stops of other users for areas in which the single individual behavior is not reliable due to a low number of stops, and use the collective behavior to infer a suggested time threshold for the individual in those areas.

The paper is organized as follows: Section 2 discusses the related works and how our proposal differs from existing solutions; Section 3 provides some preliminary definitions; Section 3 defines the problem we want to tackle; Section 4 introduces our first proposed method to solve the problem, named ATS, while Section 5 describes a space-aware refinement

[1]This work extends the paper "Self-Adapting Trajectory Segmentation" presented at BMDA 2020 International Workshop in Big Mobility Data Analytics [4].

named ACTS, with its variants; Section 6 defines some evaluation measures to quantify the quality of a segmentation; Section 7 provides empirical quantitative and qualitative evaluations of results, also comparing against some baselines and competitors; finally, Section 8 closes the paper with some conclusions and pointers towards future developments.

2 Related work

Segmentation is a technique for decomposing a given sequence into homogeneous and possibly meaningful pieces, or *segments*, such that the data in each segment describe a simple event or structure. Segmentation methods are widely used for extracting structures from sequences, and are applied in a large variety of contexts [43], such as: time series [7, 21], genomic sequences [31, 36, 37], text [28], video data. In the latter case, for instance, [33] proposes a trajectory segmentation approach for image motion, formulating it as an optimization problem aimed at minimizing the error between the observed motions and the segments approximating them.

Various simple approaches are currently adopted in practice, the most common being based on spatial and/or temporal constraints. This is the case, for instance, of the application paper in [46] where human trajectories are identified through a predefined rule based on a pair of spatio-temporal parameters regulating the end of a trajectory and the start of the subsequent one; and [20], where the trajectory is divided into subsequent trips if the time interval of "nonmovement" exceeds a certain threshold. In [49] it is described a change-point-based segmentation approach for GPS trajectories according to the transportation means adopting a universal threshold for determining whether a segment is "walk" or "nonwalk". The work in [6] presents a theoretical framework that computes an optimal segmentation through computational geometry methods based on several criteria (e.g., speed, direction, location disk) that must be satisfied in each partition, thus making the approach local. However, the approach is rather general and not clearly applicable to the human trajectory context, where a trip can be complex and not showing the geometrical/movement uniformity the method looks for. Each criterion mentioned above corresponds to thresholds that the user must set, without clear guidelines on how to choose them. Finally, we remark that the implicit objective of such solutions is to identify the situations where the trajectory physically stops, regardless of its significance for the user. That allows to overcome the lack of a specific signal in the input data (e.g. car switch on/off) and the presence of artifacts introduced by GPS errors (e.g. the coordinates of an object change even if in reality it does not move), yet it does not distinguish between significant stops and irrelevant ones, which is a more semantic classification.

The authors of [47] segment the trajectories in two steps. The first segmentation is performed by means of simple policies with respect to temporal and/or spatial predefined constraints. Then, the trajectories are divided into *stops* and *moves* observing variations of the speed of the object. If the variations of the speed is below a speed threshold and there is a sufficient number of observations, then the portion of trajectory is annotated as a stop. The speed threshold is not general but changes according to the user behavior and also to the surrounding of the stop. In [40] it is defined a measure of the density of the points in the neighbourhood of each trajectory point, the Spatio-Temporal Kernel Window (STKW) statistics. To determine the start and end points of segments, the algorithm looks for maximal changes in STKW values. The focus of the approach is on capturing changes of transportation mode, including stops, which are simply points with low speed.

Besides to these methodologies, several other solutions to the trajectory segmentation problem have been proposed in the literature, yet with objectives different from ours. For example, cost-function based strategies were presented in [24, 25], while clustering-based ones are introduced in [29, 30], and a method based on interpolation kernels is described in [10, 11]. All these approaches are more focused on splitting a movement into homogeneous parts, rather than discovering significant stops, which is the purpose of this paper.

From a more specific perspective, we can frame our proposal as a methodology for *stop-detection*, the segmentation being a consequence of selecting stops as cutting points. Along this direction, [44] presents a kernel-based algorithm to detect stop locations and estimate stop durations. The method does not analyze the points sequentially, and instead builds a kernel density surface from which it extracts local maxima that become activity location candidates from which to derive the stay time. In [1] it is presented an algorithmic framework for criteria-based segmentation of trajectories through a start-stop matrix that stores the relation between a trajectory and a criterion. In the criteria-based setting, segments are chosen such that the movement inside each segment is homogenous w.r.t a given criterion (e.g., on speed). The work in [48] describes a solution that derives the users' activity locations and times from data collected by their phones (GPS, GSM, WiFi, etc.). The main steps of the procedure consist in generating a first set of candidate stops according to predefined spatial/temporal windows, then in checking frequently visited places and in merging them, and finally in removing extra stops. A refinement of this procedure is presented in [39]. In [9] it is described a procedure that starts from fixed atomic segment of a homogeneous state, i.e., not moving or moving very little), and iteratively extends the segment until a new state is found. Similarly, [8] illustrates a method for threshold settings for stop detection focusing on periods of non-movement. In [22] stop points are detected using a density-based spatial clustering algorithm where a temporal criterion and gaps are also taken into account. Similarly, in [14] it is proposed a refined version of the DBSCAN clustering algorithm combined with SVM to identify the activity of stop locations. Finally, also [23] describes a cluster-centric trajectory segmentation approach exploiting movement characteristics such as position, direction, and speed of moving objects. Compared to these solutions, our proposal has a twofold objective, since we aim at simultaneously labeling a point as a stop and to refine the trajectory among two consecutive stops.

In this work we provide a segmentation method that, opposed to most of the existing ones, is not based on fixed space and/or time thresholds to be selected by the user – this is the case, for instance, of [20, 27, 46, 47, 49]. Instead, we aim to make the segmentation parameter-free and also adaptive to the single user's data [15, 26], giving the opportunity to have different kinds of segmentation over different users. Also, our approach is complementary to the STKW-based one [40], as the latter aims to differentiate movements with different speed profiles, including stops as a particular example, while we focus on stop timing and try to understand which stops are actually significant (e.g. not too short) for the user. A work that is closer to our proposal is [10], where the authors introduce a new approach for unsupervised trajectory segmentation, called Octal Window Segmentation. The solution is based on a behaviour deviation detection strategy which makes use only of geolocation data over time. Interpolation methods are adopted to generate an error signal, which is then used as a criterion to split the trajectories into sub-trajectories. However, while the general idea has some similarity to ours, the approach seems to be appropriate for identifying route changes (at least in free movement) and changes of speed, yet much less for identifying significant stops.

3 Problem definition

We start by defining trajectory segmentation based on a spatial and a temporal threshold, in a way similar to standard approaches in literature.

Definition 1 (Individual Trajectory) Given a user u, her *Individual Trajectory* T_u is the sequence of n points $T_u = \langle p_1, \ldots, p_n \rangle$ that describes her position in time, where each point $p \in T_u$ is defined as a triple $p = (p.x, p.y, p.t)$, representing its spatial coordinates x and y, and the corresponding timestamp t. Moreover, points are in chronological order, i.e. $\forall 1 < i \leq n . p_{i-1}.t < p_i.t$.

First, we associate to each point a value corresponding to the time needed to move away from it farther than a spatial threshold:

Definition 2 (Pseudo-Stop Duration) Given an individual trajectory $T = \langle p_1, \ldots, p_n \rangle$ and a spatial threshold σ, the Pseudo-stop duration associated to point p_i is defined as $SD(T, i) = \min\{p_j.t - p_i.t | i < j \leq n \wedge d(p_i, p_j) > \sigma\}$, where d represents the geometrical Euclidean or geographical distance.

Notice that the last point p_n will have $SD(T, n) = \min \emptyset = \infty$. Then, we define a partitioning of trajectories into segments, each terminating with a point having an high pseudo-stop duration

Definition 3 (Segmented Trajectory) Given a trajectory $T = \langle p_1, \ldots, p_n \rangle$, a spatial threshold σ and a temporal threshold τ, we define the (σ, τ)-segmentation of T as $T^{\sigma,\tau} = \langle S_1, \ldots, S_m \rangle$, such that:

$$
\begin{aligned}
&(i)\ \ \forall i\ s.t.\ 1 \leq i \leq m,\ S_i \text{ is a subsequence } \langle p_s, p_{s+1}, \ldots, p_e \rangle \text{ of } T \\
&(ii)\ \ \textstyle\bigcup_{i=1}^{m} set(S_i) = set(T) \text{ and } i \neq j \Rightarrow set(S_i) \cap set(S_j) = \emptyset \\
&(iii)\ \ \forall S = \langle p_s, p_{s+1}, \ldots, p_e \rangle \in T^{\sigma,\tau},\ SD(T, e) > \tau \wedge \\
&\qquad\qquad \forall j\ s.t.\ s \leq j < e : SD(T, j) \leq \tau
\end{aligned}
$$

where $set(I) = \{p \in I\}$.

Conditions *(i)* and *(ii)* imply that the segments of the segmented trajectory of T form a partitioning of the elements of T in the strictly mathematical sense. Moreover, condition *(iii)* states that all the points in a segment are movement points, i.e., their pseudo-stop duration is smaller than the given threshold, excepted the last point. Therefore, each point in T that has a high pseudo-stop duration will act as a split point, and corresponds to a distinct partition in $T^{\sigma,\tau}$. Also, an implicit consequence of the definition is that partitions are maximal, i.e., they cannot be extended with additional points and still satisfy the requirements of Definition 3.

Existing trajectory segmentation methods assume that the same rules and the same parameters should apply to all moving objects. Since different objects can show very different movement characteristics, the above assumption leads to make choices that on average fit best the dataset, yet potentially making sub-optimal choices on single individuals.

Our objective is to overcome this limitation, making the segmentation process adaptive to the individual and taking into consideration their overall mobility. Our problem statement extends the traditional formulation of segmentation as a threshold-based operation, thus the core issue is to find good parameter values for each user.

Definition 4 (Individual Cut Threshold Problem) Given an Individual Trajectory T_u, and a global spatial threshold σ, the problem is to identify the temporal threshold τ that yields the optimal segmentation $T^{\sigma,\tau}$.

We notice that the problem definition requires a user-provided parameter σ. However, as it will commented later in more detail, this is a single global threshold that only depends on location accuracy and is therefore expected to be rather easy to select for a given data source type.

In this work we also consider a generalization of the problem, where each user is actually associated to a set of thresholds instead of just one. In particular, we assume that the correct temporal threshold can depend on where the user is moving in each specific moment. We do that by first revising our definition of segmentation:

Definition 5 (Space-Adaptive Segmented Trajectory) Given a trajectory $T = \langle p_1, \ldots, p_n \rangle$, a space partitioning G that maps points to geographical cells, a spatial threshold σ and a function $\tau_G : G \rightarrow \mathcal{R}$ that associates a temporal threshold to each cell in G, we define the (σ, τ_G)-segmentation of T as $T^{\sigma,\tau_G} = \langle S_1, \ldots, S_m \rangle$, such that:

$$
\begin{aligned}
&(i)\ \forall i\ s.t.\ 1 \leq i \leq m,\ S_i \text{ is a subsequence } \langle p_s, p_{s+1}, \ldots, p_e \rangle \text{ of } T \\
&(ii)\ \textstyle\bigcup_{i=1}^{m} set(S_i) = set(T) \text{ and } i \neq j \Rightarrow set(S_i) \cap set(S_j) = \emptyset \\
&(iii)\ \forall S = \langle p_s, p_{s+1}, \ldots, p_e \rangle \in T^{\sigma,\tau},\ SD(T, e) > \tau_G(G(p_e)) \wedge \\
&\qquad \forall j\ s.t.\ s \leq j < e : SD(T, j) \leq \tau_G(G(p_j))
\end{aligned}
$$

where $set(I) = \{p \in I\}$.

The change basically consists in replacing the fixed threshold τ of the user with a set of values, one for each geographical cell visited by the user, formalized as a function from cells to thresholds. The problem now, therefore, becomes how to find the assignment of thresholds τ_G.

Definition 6 (Individual Space-Adaptive Cut Threshold Problem) Given an Individual Trajectory T_u, a space partitioning G and a global spatial threshold σ, the problem is to identify the set of temporal thresholds τ_G that yields the optimal space-adaptive segmentation T^{σ,τ_G}.

Since the number of moving objects can be very large, the process must be completely automatized and require no human intervention. In Section 4 we will introduce a simple and effective approach to solve the first problem, and thus find a suitable value of τ for each user, also providing some basic guidelines to choose a value for the global spatial parameter. Then, Section 5 will revise the approach to tackle the space-adaptive problem definition, considering a more flexible context where the temporal threshold of a user can also change based on the areas visited, thus in principle yielding different values for different spatial locations.

4 Self-adaptive trajectory segmentation

The first solution proposed for the individual cut threshold problem consists in fixing the spatial threshold to a global value (i.e. to be used for all users) and then in studying the segmentations that we would obtain by applying different temporal thresholds. We will start

describing the process for choosing the temporal threshold, which is the central part of the solution, and later discuss how the spatial one can be chosen.

4.1 Methodology

When very small values of τ are used, the segmentation obtained will contain a huge number of very short segments, till the extreme case where each point forms its own segment. As the threshold is increased, more and more segments will merge together, since some of the former splitting points will fall below τ. The process is expected to gradually enlarge the trajectory segments by first including simple slowdowns (i.e. not really stop points), then temporary stops (e.g. at traffic lights), and so on. Our approach consists in (virtually) monitoring such progression, and detecting the moment where an anomalous increase in the number of segments is observed, which represents a sort of *change of state*. This follows the same kind of idea adopted in various unsupervised classification contexts, such as the *knee method* for deciding the number k of clusters for the k-means algorithm [42], or analogous solutions to choose the radius for density-based clustering (e.g. DBScan).

In our solution, rather than relying on visual or similar heuristic criteria, we will base the threshold selection on a statistical test. In particular, we will adopt the Modified Thompson Tau Test [5] which, basically, checks whether a given value fits the distribution of the rest of the data or not [18]. Since we look for anomalous values in a sequence, we apply the test iteratively, comparing each value $n(t)$ (the number of segments obtained with $\tau = t$) against the values $n(t')$ obtained for larger thresholds t'. This process yields a set of thresholds that have an anomalous number of partitions as compared to the successive thresholds. Among them, we simply choose the smallest one, thus deciding to select the segments that emerge at the first *change of state*, also representing shorter and finer granularity movements.

The procedure, named ATS (self-Adaptive Trajectory Segmentation) is summarized in Algorithm 1. Step 1 collects the pseudo-stop durations SD of all the points i of the input trajectory, and step 2 computes the frequency F of each value, basically representing the number of new segments obtained using that value as τ w.r.t. the previous smaller thresholds. In our implementation such frequency distribution is computed through smoothed histograms, grouping values into bins of 1-minute width. Figure 1(left) shows the frequency distribution of a sample trajectory, the vertical line corresponding to a possible cut point. The resulting set of segments obtained is described in Fig. 1(right) in terms of segments duration. Finally, step 3 applies the Modified Thompson Tau Test to all possible cut thresholds,

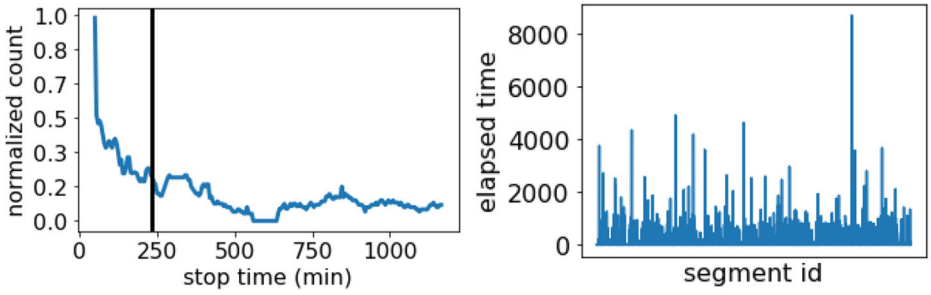

Fig. 1 Frequency distribution of pseudo-stop durations for a user trajetory (left), and the durations of the segments obtained using a specific threshold to cut the trajectory (right). The threshold used corresponds to the vertical line on the left image

corresponding to all the non-zero values of frequency function F, to identify the frequency values that appear to be anomalous with respect to the frequency of larger thresholds. Among all these candidate thresholds, step 4 selects the smallest one as value for τ.

Computational complexity The cost of Algorithm 1 is dominated by step 1, since the computation of each pseudo-stop duration (SD) could in principle require to scan all the remaining points of the individual trajectory, thus yielding a $O(n^2)$ cost, where n is the size of the individual trajectory. However, in practical applications the trajectory portion needed for each SD is relatively small, leading to a quasi-linear cost. The remaining parts of the algorithm can be realized in linear time, including the Modified Thompson Tau Test, which can be computed for each point through incremental updates, and considering that the actual size of F (namely, not considering empty bins where $F(a) = 0$, which can be simply omitted) is at most $n = |T|$.

Algorithm 1 ATS(T, σ).

Input : Individual trajectory T, spatial threshold σ
Output: Cut threshold τ
1 $S = \langle\, SD(T, i) \mid 1 \leq i \leq |T| \,\rangle$;
2 $F =$ frequency distribution of S values with 1-minute bins ($F(a) = |\{a \in S\}|$);
3 $C = \{t | t \in range(F) \wedge TT(F(t), \langle F(t') | t' > t \rangle) = true\}$; $TT(a, B)$ = Modified Thompson Tau Test of a vs. set B
4 **return** min C

4.2 Fixing the spatial threshold

In our approach, the threshold σ represents the minimum distance between two (consecutive) points that can be considered as a movement, and the temporal parameter is indeed measured as the time needed to make a movement. A simple way to fix its value is to adopt the minimum value that, according to the accuracy of our dataset, cannot be mistaken for a positioning error, for instance due to GPS uncertainty. In our experiments we adopt road vehicle GPS traces that are expected to have errors not larger than 10 meters, therefore we could fix $\sigma = 20$ (the worst case distance between two points that have the maximal error in opposite directions). We decided to slightly increase it to 50 in order to stay on the safe side, also to take into account that errors are slightly higher than average in urban centers, which is the application context where our experiments are performed. Since we do not have data sources from other kinds of transport (ships, planes, etc.) the selected threshold seems to meet our purposes. However, empirical results confirm that the value of the global parameter σ is not critical, as our approach shows a low sensitivity to it. For this reason, the value we chose in our experiments (50 meters) can be considered a good guess for generic vehicle GPS data. Other data sources with a higher spatial uncertainty might require larger values, to be ascertained through a (one-shot) analysis of the data produced.

5 Individual and collective adaptive trajectory segmentation

The solution described in the previous section strictly follows the problem formulation of (σ, τ)-segmentation given in Definition 3, thus implicitly assuming that a user has a single, optimal threshold that applies well in any area where they move. Clearly, common sense

suggests that this is an artificial assumption, and the threshold that is correct for a user in a given place, might be not optimal for the same user in a different location. To loosen such assumptions, we adopt here the more general notion of space-adaptive segmented trajectory, introduced in Definition 5, and a corresponding strategy to adapt the thresholds also to geographical locations.

The problem, now, consists in finding for each user an assignment of thresholds τ_G that provides a (potentially different) threshold value for each geographical cell in the space partitioning G.

We identify here three possible approaches:

1. *Local individual approach*: following the same idea of ATS, we could restrict the statistical test used to fix the threshold τ only to the points of the user that fall in a given cell $g \in G$. While very appealing, empirical evaluations show that the data samples associated to each cell are too small to apply the test, with very few exceptions. For this reason, alternative solutions were considered.
2. *Local collective approach*: this solution assumes that the time threshold τ is actually a function of the location, and does not directly depend on the user. Therefore, each cell g is associated to a data sample composed of all the points of all users that fall in g. This greatly increases the sample size, yet losing the identity of the single user.
3. *Wisdom-of-the-crowd collective approach*: the idea here is that each user brings an opinion about what is the correct threshold, built from their own mobility data (and therefore their own τ found through ATS), and over each cell g all users vote for the best local threshold value, each vote having a weight proportional to the frequency of visit of the cell. This is a simple application of the classical "wisdom of the crowd" principle [13].

Approaches 2 and 3 provide a candidate threshold value τ^* for a user in a given cell, which can be seen as a suggestion that all users provide as alternative to the individual value τ. Our proposal is to replace τ with τ^* whenever the former has a weak relation with the cell, i.e. for those locations that the user visited only rarely, and that therefore were not significantly involved in the computation of the global τ. Both collective approaches result into a mapping $G_C : G \to \mathcal{R}$ that associates each geographical cell in G to a suggested τ^*. The procedure that implements the management of such suggestions is the same for both approaches, is named ACTS (self-Adaptive and Collective Trajectory Segmentation), and is summarized in Algorithm 2.

Algorithm 2 ACTS($T, \sigma, G_C, min\_stops$).

Input : Individual trajectory T, spatial threshold σ, Cell grids and associated collective threshold multisets G_C, Minimum number of stops min_stops.
Output: Cut thresholds η

$\tau = ATS(T, \sigma)$;
$G_I = \{ (g, freq) \mid g \in G_C \wedge freq = |\{p \in T | p \in g\}| \}$; *// visited cells and frequency*
$\eta = \emptyset$;
for $(g, freq) \in G_I$ **do**
  $\mu = mode(S)$ for $(g, S) \in G_C$;
  if $freq \geq min\_stops$ **then**
    $\eta = \eta \cup \{(g, \tau)\}$; *// individual threshold prevails*
  else
    $\eta = \eta \cup \{(g, \mu)\}$; *// collective threshold prevails*
return η

Besides the individual user trajectories T and the spatial threshold σ, the ACTS procedure takes as input the cell grid G_C containing the peudo-stop times of all the observed users grouped per cell, and the minimum number of stops *min_stops* that an individual user can have in a cell in order to consider the cell "frequently visited". In the first step, ACTS retrieves the user adaptive threshold τ. After that, it identifies the subset $G_I \subseteq G_C$ of cells visited by the user, with their visit frequencies. Then, for each cell g (lines 4–10), if the cell is frequently visited, i.e., the user has in that area at least *min_stops* points, then the individual global threshold τ is used (line 7), otherwise we take the most frequent value among those associated to the cell g (lines 5 and 9).

In order to specify what kind of threshold suggestions we are using in the ACTS procedure, we will refer to it as ACTS_{Local} when G_C is obtained through the local collective approach (number 2 of the list above), and as ACTS_{WOTC} when the Wisdom-of-the-crowd approach is used.

Spatial grid definition In principle, any definition of grid G can be applied to ACTS, provided that it is a partition of space that covers all points in our users' trajectories. In our experiments we opted for a regular grid, which is the most commonly adopted solution in literature, and in particular we implemented it through a standard *geohashing*. *Geohash* [34] is a very efficient mapping of locations into rectangular grids, and allows to change its spatial granularity in a transparent way. Its main limitation is in the fact that grids are predefined worldwide, and the spatial granularity can be changed in a limited set of configurations, the size of each cell doubling when we move from one granularity level to the next one. Other, more sophisticated space partitioning strategies are used in literature, such as regular exagonal grids [41] of quad-tree based adaptive partitioning [12], yet evaluating all of them is out of the scope of this paper. Given an encoding length h, Geohash associates each pair of latitude-longitude coordinates to a string of h letters and digits, which corresponds to define a partitioning into square or rectangular cells, each cell corresponding to the set of points that have the same encoding. In particular, we will consider three levels: $h = 5$, resulting into cells of diameter $\sim$ 4.8 Km; $h = 6$, with diameter $\sim$ 1.22 Km; and $h = 7$, with diameter $\sim$ 0.152 Km.

Algorithm 3 summarizes the overall process, including the generation of grid G and collective suggestions G_C, for both variants of ACTS.

Algorithm 3 $\text{ACTS}_{ALL}(Method, \mathcal{T}, \sigma, h, min\_stops)$.

Input : Method to apply (Local or WOTC), Individual trajectories of all users $\mathcal{T}$, spatial threshold σ, geohash level h, Minimum number of stops *min_stops*.
Output: Segmented trajectories $\mathcal{T}^*$

1 $G = \{geohash(lat, lon) | T \in \mathcal{T} \wedge (lat, lon) \in T\}$;
2 **if** *Method* = *'Local'* **then**
3 $\quad G_C = \{(g, \tau^*) | g \in G \wedge S = \langle SD(T, i) | T \in \mathcal{T} \wedge T[i] \in g \rangle \wedge \tau^* = ATS_{geo}(S, \sigma)\}$;
4 **if** *Method* = *'WOTC'* **then**
5 $\quad G_C = \{(g, \tau^*) | g \in G \wedge S = \langle ATS(T, \sigma) | T \in \mathcal{T} \wedge T \cap g \neq \emptyset \rangle \wedge \tau^* = mode(S)\}$;
6 $\mathcal{T}^* = \emptyset$;
7 **for** $T \in \mathcal{T}$ **do**
8 $\quad \tau_G = ACTS(T, \sigma, G_C, min\_stops)$;
9 $\quad \mathcal{T}^* = \mathcal{T}^* \cup T^{\sigma, \tau_G}$; // (σ, τ_G)-segmentation, see Definition 5
10 **return** $\mathcal{T}^*$

6 Evaluation measures

The reconstruction error generally used for evaluating segmentation problems [3] just measures how well each segment can be approximated with one value, and thus seems not to fit with trajectory segmentation. Therefore, similarly to clustering evaluation, we propose three internal evaluation measures [42]. Let T be the sequence of n points and $T_S = \langle S_1, \ldots, S_m \rangle$ its segmentation. We denote with $A_t = duration(T) = p_n.t - p_1.t$ the total elapsed time from the first point of $p_1 \in T$ to the last point $p_n \in T$, and $A_d = length(T) = \sum_{i=1}^{n-1} d(p_i, p_{i+1})$ the total distance covered by the trajectory, computed by considering every couple of subsequent points in T. Let $M_t = \sum_{S_i \in T_S} duration(S_i)$ be the sum of the segments' duration, i.e., the time spent driving, and $M_d = \sum_{S_i \in T_S} length(S_i)$ be the sum of the segments' length, i.e., the distance traveled. Then, we define the following measures:

- *time precision*: $TP = 1 - M_t / A_t$
- *distance coverage*: $DC = M_d / A_d$
- *mobility f-measure*: $MF_\beta = (1 + \beta^2) \cdot TP \cdot DC / ((\beta^2 \cdot TP) + DC)$

Time precision and distance coverage capture two conflicting effects of segmentation, namely the time covered by stops and the distances covered by the segments (i.e. the movement points). Indeed, a very *aggressive* segmentation will identify a large number of stop points, yielding a high time precision, yet this will make segments shorter, significantly reducing the distance coverage. Similarly, a very *loose* segmentation will yield exactly the opposite results. Any segmentation choice will yield a trade-off between them. Analogously to the f-measure adopted in Information Retrieval, which is a combination of precision and recall measures, our *mobility f-measure* accounts for both aspects simultaneously. In the experiments we adopt $\beta = 0.25$, which weighs *time precision* higher than *distance coverage* by augmenting the relevance of missing precision in stop detection. The reason is that *i)* it is relatively easy to guarantee an high distance coverage, and *ii)* the main focus of the paper is on the temporal aspects of trajectory partitioning.

7 Experiments

We experimented the proposed trajectory segmentation approaches ATS and ACTS over real datasets of GPS vehicle traces. The results commented in the following refer to 2000 users of the area of Rome (Italy), and London (UK). The means and standard deviations of the sampling rate for the users analyzed are 12194.67 ± 22575.66 and 4385.76 ± 9359.14, for Rome and London respectively. The high values and their high variability is due to the presence of several long time gaps, typically due to parking periods.

In the following, we first analyze the personal temporal thresholds returned by ATS, and then provide a quantitative and qualitative evaluation of the results for understanding the benefits of the novel method with respect to existing ones. We compare, in particular, against the common trajectory segmentation method based on fixed parameters ($\text{FTS}_{\text{temp-thr}}$) as proposed in [46]. Moreover, we consider a baseline consisting in a random trajectory segmentation method that splits the sequence of points $T = \langle p_1, \ldots, p_n \rangle$ into m equal-length segments *(i)* with m randomly extracted between 2 and $n/2$ (RTS_1), or *(ii)* with m set to the number of segments returned by ATS (RTS_2).

Next, we show the results obtained with the two variants of ACTS, ACTS_{LOC} and Acts_{WOTC}, thus evaluating the impact of considering geography and collective behaviors in the definition

of individual temporal thresholds. Here, we compare our proposed solutions against a state-of-the-art approach for trajectory segmentation exploiting a completely different strategy but relying exactly on the same input data format. We name HEH-D the proposal described in [22] for detecting stop points using the DBSCAN method. In summary, HEH-D first runs DBSCAN on the GPS observations only considering the spatial dimension. Then, it further separates the points in each cluster that have a temporal gap between each other larger than q seconds, and turns into noise the spatio-temporal clusters composed by less than k points. Finally, all the noise points are sorted chronologically and modeled as trajectories while those in the clusters are treated as stop points. According to the suggestions in [22], we adopted the following parameters setting: $min\_pts = 5$, $\epsilon = 50$ meters, $q = 210$ seconds. Also, for the parameter k we evaluated all values between 2 and 6 (the latter being the choice suggested in the paper), and eventually selected $k = 2$ since it yields the best results in terms of mobility f-measure. Additionally, we also experimented with a variant of this method that replaces DBSCAN with OPTICS, named HEH-O in the experiments, for which we adopted the same parameters specified for HEH-D. The idea is that OPTICS typically performs better than DBSCAN when clusters in the data have variable densities, and that might help improving the quality of the segmentation.

Finally, we conlcude the section with an evaluation of run times of our methods when the number of users and the duration of their trajectories vary.

7.1 Self-adaptive temporal threshold (ATS)

We observe in Fig. 2 the distribution of the time thresholds selected by ATS for each user (vertical axis represents value frequencies in log-scale).

Although every user has her own mobility with its own mix of regular and more erratic behaviors [35], we observe two clear peaks in the distributions for both Rome and London. This means that ATS mainly recognizes two different types of users regarding to the minimum duration of the stops. This supports the intuition behind our approach, namely to have a self-adaptive procedure selecting a personalized best temporal threshold for each user. Selecting one single threshold value for all the data might negatively affect the segmentation of some users, partitioning their trajectories either too much or too little. The first peak is at about 600 seconds ($\sim$ 10 minutes), while the second peak is around 1200 seconds ($\sim$ 20 minutes). These values correspond to the temporal thresholds that the ATS procedure uses

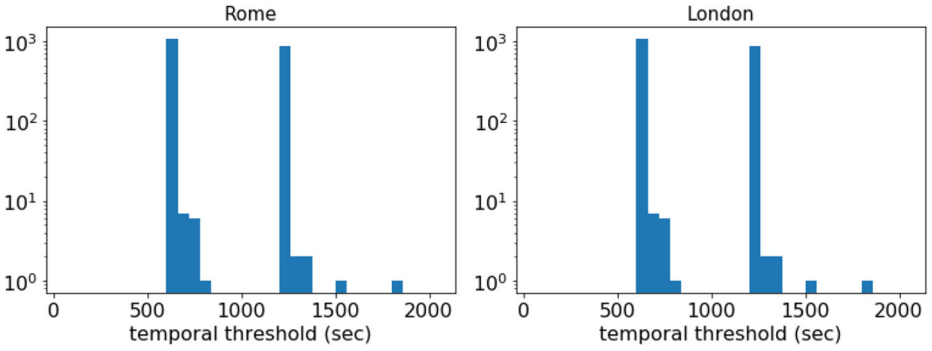

Fig. 2 Time threshold distributions for trajectories obtained with ATS in Rome and London. The peaks show the ideal thresholds to be set to build the trajectories

to cut each trajectory. There is also a minority of users having values relatively far from the peaks.

7.1.1 Comparison of evaluation measures

In this section we compare the ATS approach with the baseline methods taken into account. In Tables 1 and 2 we report the results obtained on our two cities. The first three columns show the evaluation measures described above. The fourth column shows the ratio between the average sampling period of movement points (thus discarding the stop portions of the user's trajectory) and the average sampling period of the full trajectory, while the last one reports the average number of segments obtained and its standard deviation. In general, we can observe that the best results were obtained with the ATS and FTS methods, both for Rome and London. Analyzing the ratio (fourth column) we can see that values are low for both ATS and the FTS ones, meaning that the long stops are ignored (i.e. they are recognized as real stops) and just the short ones are considered. On the contrary, with the random approaches the ratio is bigger because the algorithm function evaluates all stops in the same way. Looking at the number of segments it is possible to note that FTS and ATS methods produce different quantities, especially the FTS_{120} produces less segments in the Rome case and much more in London. About the last two approaches, the RTS_1 method works with a random number of segments, so it is normal that the final result differs from the others, while the RTS_2 takes as number of segments the same of the ATS approach so we aspect to achieve similar results.

For the evaluation measures we can see that ATS reached the goal we expected, i.e. yielding a quality of results which is always comparable or higher than fixed-threshold approaches and more robust. Indeed, for both Rome and London the values obtained by ATS are compatible with the FTS results, even better in the $MF_{.25}$ for Rome and in the distance coverage for London. In particular, in the Rome example, having a high $MF_{.25}$ values means that also the time precision and the distance coverage are well correlated, leading to satisfying result. Looking at the FTS_{120} results, we can note that the time precision is high but the distance coverage is very low, because the algorithm builds short trajectories with few points. An analogous reasoning can be done analyzing the FTS_{1200} method, which produces an excellent distance coverage score but a lower time precision. The ATS solution reaches a good balance, thanks to its adaptive behaviour that allows to control and correct the trajectory fragmentation, and all its evaluation measures are always either the best or the second best of the group.

Table 1 Evaluation on Rome data

method	$MF_{.25}$	TP	DC	$ratio_{\text{sr}}$	*#segms (avg* $\pm$ *std)*
ATS	**.951**	.951	.981	0.049	837.34 $\pm$ 854.52
FTS_{120}	.925	**.996**	.456	**0.015**	592.26 $\pm$ 652.78
FTS_{1200}	.948	.947	**.997**	0.053	746.28 $\pm$ 733.96
RTS_1	.279	.268	.722	0.700	2094.85 $\pm$ 2472.36
RTS_2	.124	.118	.877	0.883	899.59 $\pm$ 926.03

The first three columns show the measures illustrated in Section 6. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments

Table 2 Evaluation on London data

method	MF_{25}	TP	DC	$ratio_{\text{sr}}$	$\#segms\ (avg \pm std)$
ATS	.955	.953	**.999**	0.047	433.92 ± 513.72
FTS_{120}	**.958**	**.961**	.944	**0.040**	1131.83 ± 1431.81
FTS_{1200}	.952	.950	**.999**	0.050	359.55 ± 410.61
RTS_1	.267	.256	.695	1.007	2833.72 ± 4203.05
RTS_2	.035	.033	.958	1.008	445.65 ± 527.97

The first three columns show the measures illustrated in Section 6. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of trajectories

For a better understanding of the quality of ATS, the distribution of $MF_{.25}$ values for the different approaches on the two datasets is shown in Fig. 3 through a boxplot visualization. For the Rome case we can observe that with the ATS approach the median value is the highest (closest to 1) and the inter-quartile range is smaller than the other two, meaning that we have a smaller variabiliy and thus more robust results. The London case appears to be different, and the best $MF_{.25}$ values are obtained with the FTS_{1200}, although the median is very similar to ATS and the box is only slightly narrower. This indicates that in some contexts the flexibility introduced by ATS might be not required, and it only reaches performances similar to those of simpler solutions.

7.1.2 Comparison of segmentation statistics

In the following we analyze other statistical indicators on the trajectory segments extracted by the various methods. Indeed, discovering some hidden correlations between trajectory features and the segmentation approach could lead to a better understanding of the problem and highlight other relevant aspects. In Fig. 4 we report the distributions of the average number of points per segment for Rome and London. For all methods, the majority of segments have less than 20 points, probably meaning that most of the trips take place within the city. However, in the distribution tails some long trajectories with more points emerge. We observe that the distribution peaks of ATS place somehow in between the peaks of the two FTS variants (though closer to FTS_{1200}, especially in London) thus finding a trade-off between them. Moreover, we can see that the distributions are different in the two cities:

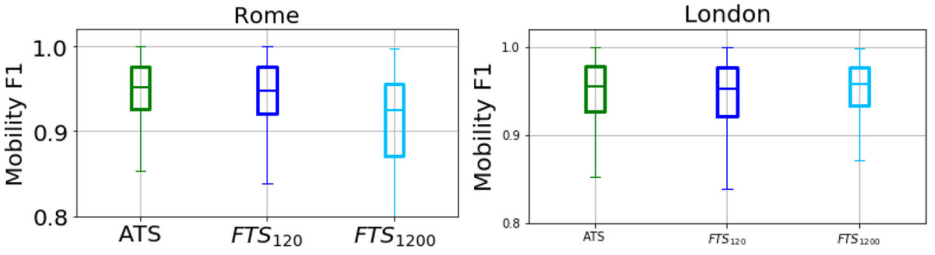

Fig. 3 Boxplots for the $MF_{.25}$ results. On the Rome data ATS yields better results than the FTS solutions, while in London all three produce almost the same results. The variability of ATS results is consistently smaller than the other methods, which is a sign of robustness

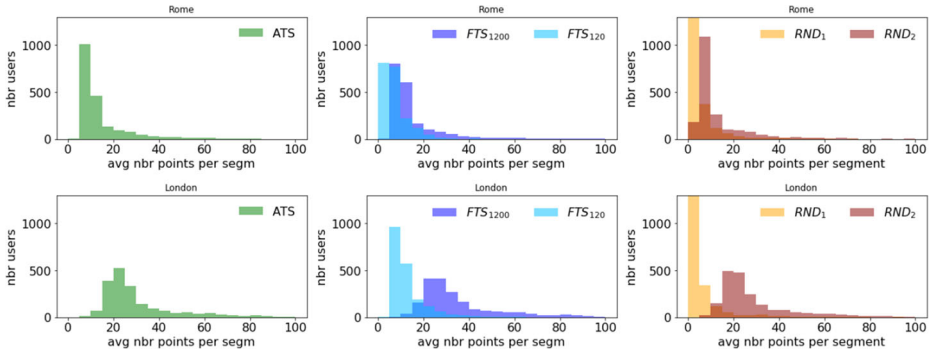

Fig. 4 Distributions of average number of points per segment obtained by ATS

London has a wider distribution than Rome, meaning that the first one has a larger variety of trips.

Figure 5 shows the distributions of the average number of segments per user. In London most of the users have less than 20 trajectory segments. The peak of the distribution is between 5 and 10. Between 30 and 100 segments the distribution remains stable at a small value larger than zero. In Rome we observe a similar result with a peak between 15 and 20. Also in this case, the peak of ATS distribution tends to stay in the middle of the FTS ones.

In Fig. 6 we compare the distribution of average length and average duration of the segments returned by ATS (left) and FTS (right) for the area of Rome. With the ATS method the peak value is around 10km, thus confirming that most of the trips are short, and likely to take place around the city. With the FTS methods the peak position depends on the temporal threshold imposed: with a threshold of 1200 seconds the average distance is similar to ATS, while with 120 seconds it becomes lower and close to 5 km. The results for the RTS methods are omitted, since their plots are very similar to FTS. Also, the plots for London show exactly the same behavior observed on Rome.

In terms of segments duration, ATS yields a distribution with a peak around $1200 - 1500$ seconds ($\sim 20 - 25$ minutes). With the FTS methods the peaks change: for FTS_{120} the peak is around 500 seconds while for FTS_{1200} the peak is centered in 1800 seconds. Also in this case, the results on London are very similar and omitted here.

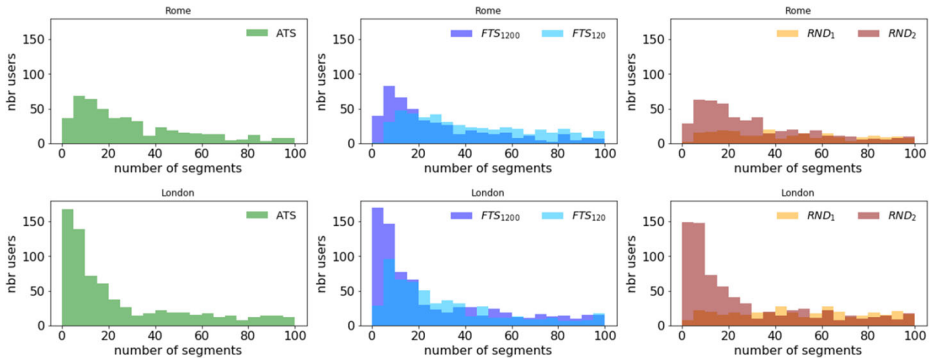

Fig. 5 Distribution of the number of trajectory segments over Rome (top) and London (bottom) with each segmentation method (on the columns, grouped by family)

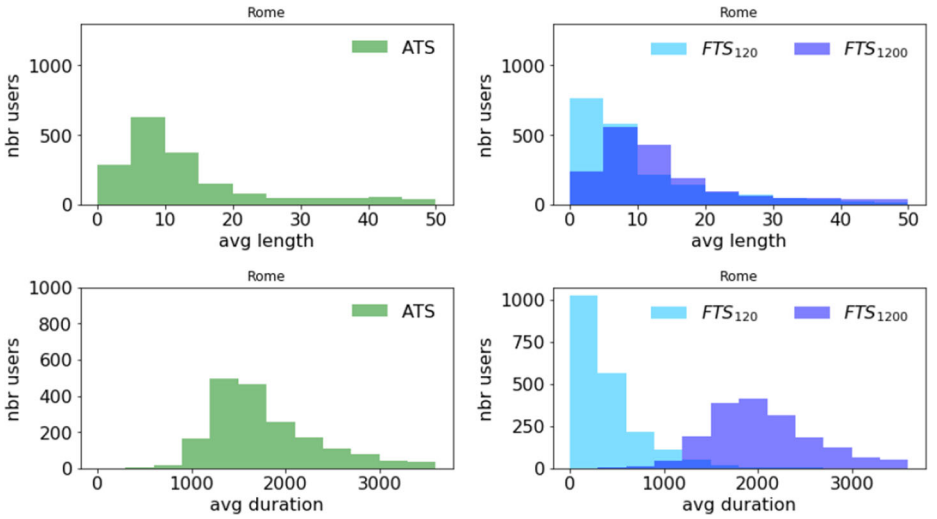

Fig. 6 Distributions of the average length (top) and duration (bottom) for the trajectory segments returned by ATS (left) and FTS (right) for the area of Rome

7.1.3 Case study

In this section we show qualitatively on a case study the effectiveness of ATS with respect to FTS. In Fig. 7 we report the segmentation returned by FTS_{1200} [46] (left) and by ATS (right), the user is traveling from south to north. FTS_{1200} returns two trajectories (green and blue), while ATS returns three trajectories (green, orange and blue). The second line of plots reports the time gap between consecutive GPS points. The colors match the trajectory segments, while stops are highlighted in red. We observe how ATS identifies the short stop of less than 15 minutes at the service area similarly to the subsequent longer stop. On the

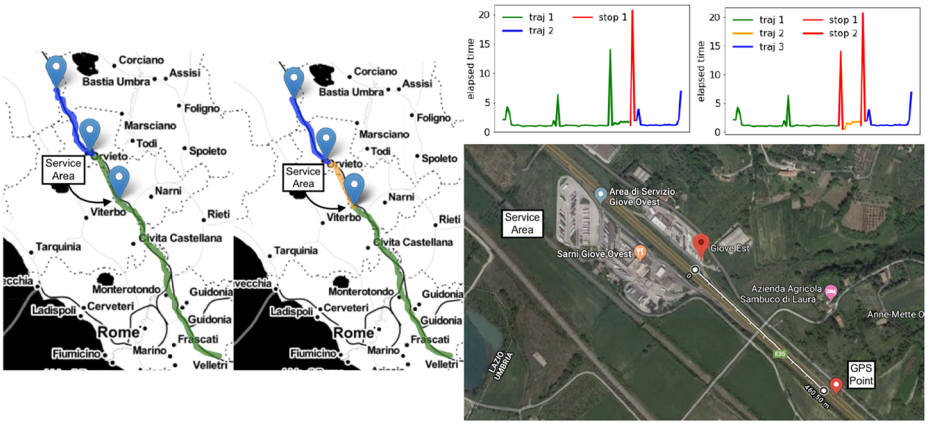

Fig. 7 Trajectory segmentation returned by FTS_{1200} (left) and ATS (right). The user is traveling from South to North. *Top*: spatial representation showing the trajectory segments. *Center*: temporal segmentation showing the inter-leaving time between GPS points. *Bottom*: zoom on the service area highlighted in the top maps where the user probably stops for $\sim$ 15 minutes. Best viewed in color

other hand, FTS_{1200} considers the first stop as part of the green trajectory. The map in the bottom line of Fig. 7 shows the service area which is very close to the GPS points reported on the bottom right corner of the map. This case study highlights how various existing stops under a certain predefined threshold can be missed with a segmentation approach like FTS, while a more data-driven and self-adaptive method like ATS is able to take into account specific user behaviors and return more detailed results.

7.2 Individual and collective adaptive temporal threshold (ACTS)

In this section we show the impact and improvements given by the ACTS methods exploiting the collective behavior over ATS. First of all we choose a reference geohash precision looking for a trade-off between the geographical granularity and the number of pseudostops collected in the cells. We opt for a geohash precision of $h = 6$ corresponding to an area of size $1.22km \times 0.61km$. As shown in the next sections, values $h = 5$ and $h = 7$ yield very similar results, suggesting that any value of h around 6 appears appropriate for this kind of data. The set G of cells obtained this way are used by Algorithm 3 to compute local suggestions for time thresholds by collecting the pseudo stops of all the 2000 users in the dataset under consideration. Then, for the ACTS_{LOC} method a distribution of pseudo-stop durations for every cell is created, which will later pass through the Thompson test. Both ACTS strategies require to define the minimum number of points (visits) of a user in a cell that make it significant for them. In order to avoid any manual setting, after a preliminary experimentation we decided to derive it directly from the distribution of pseudo-stops durations of the dataset, fixing it to its 50-th percentile. In our dataset, in particular, that corresponds to $min\_number = 5$, meaning that when a user passes in a given cell, if they have at least 5 points inside it, for that cell we can use their own individual time threshold computed by ATS; otherwise, we will use the collective threshold assigned to the cell.

We report in Figs. 8 and 9 the distributions of the time thresholds selected respectively by ACTS_{LOC} and Acts_{WOTC} (Rome dataset on the left and London on the right) for each user (vertical axis represents value frequencies in log-scale). Similarly to Fig. 2, we can observe two peaks in the distributions at about 600 seconds ($\sim$ 10 minutes) and 1200 seconds ($\sim$ 20 minutes) for both cities. Compared to Fig. 2, the two ACTS variants show a lower variability and more focused distributions. ACTS_{LOC} and Acts_{WOTC} produce almost identical distributions, however, as will be shown later, their threshold assignments (and therefore the trajectory segmentations they imply) are actually different.

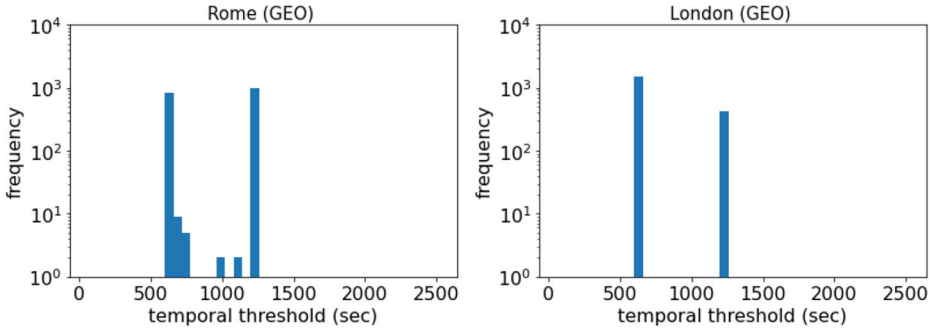

Fig. 8 Time threshold distributions for the users obtained by ACTS_{LOC} in Rome and London. Compared to ATS, the distributions are more concentrated on the two peaks

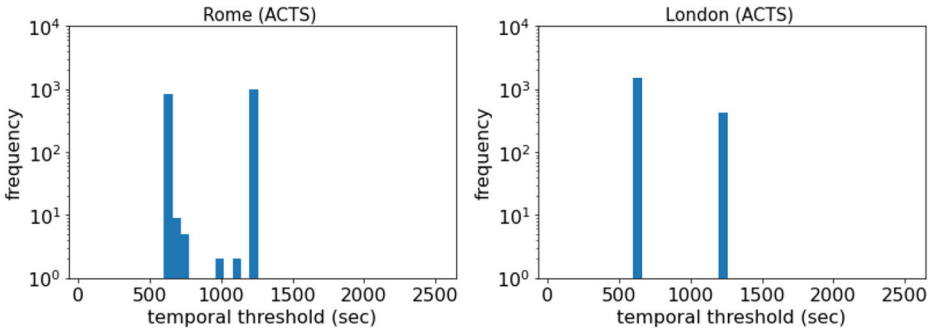

Fig. 9 Time threshold distributions for the users obtained by Acts_{WOTC} in Rome and London. The overall distributions are very similar to ACTS_{LOC}

7.2.1 Impact of ACTS strategies

In this section we provide a first evaluation of the impact of the selection strategies adopted in the two ACTS variants to assign threshold values to grid cells and, as effect, to individuals over those cells. Figure 10 shows the spatial distribution of the number of points (and, therefore, of pseudo-stop duration values associated) that fall in the cells of the two observed areas. In both cases, cells are obtained with a geohash precision $h = 6$. We notice that in the London dataset (right) the higher number of stops are mainly located along locations with high population density and within the urbanized areas. On the other hand, the Rome data (left) shows high values also along the main roads, and covers an area which is larger than the city itself (southern section of the picture), touching a part of Lazio (outside the city) and a part of Tuscany.

The plots in Fig. 11 compare the temporal thresholds that ATS and Acts_{WOTC} associate to each user for each cell they visit (remind that the value assigned by ATS will be the same for all the cells of a user, while Acts_{WOTC} yields cell-dependent thresholds that will substitute the ATS value when the cell is poorly visited by the user). In both cities we can see that the differences, and thus the impact of Acts_{WOTC} over ATS, is significant and approximately symmetric, i.e. sometimes the initial ATS threshold is increased, some other times it is decreased, with an overall balance between them. The corresponding plots for ACTS_{LOC} vs. ATS are very similar to the previous ones, and is therefore not reported here.

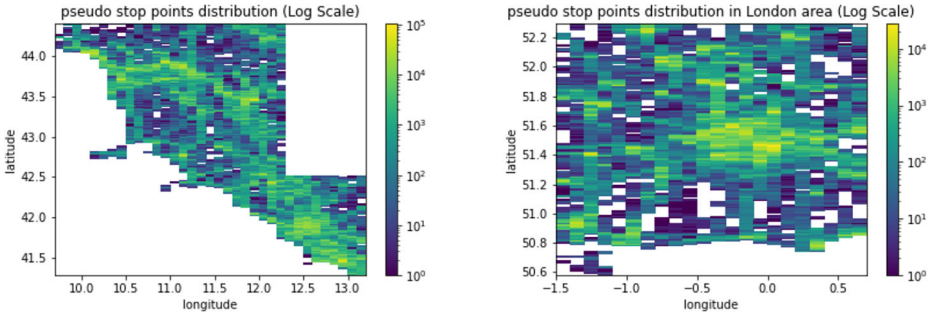

Fig. 10 Points distribution in Rome and London datasets over the geohash grid ($h = 6$)

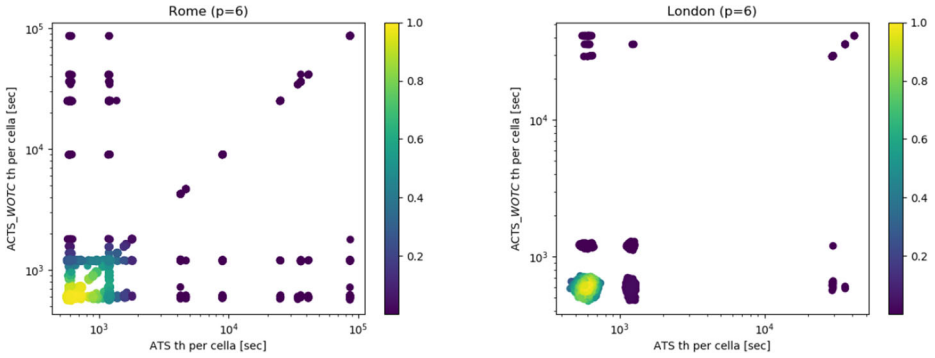

Fig. 11 Comparison of Acts_{WOTC} vs. ATS thresholds for all user-cell pairs, on Rome (left) and London (right). In both cases the difference appear significant and overall symmetricv

As mentioned above, the geohash precision is in principle a parameter that should be chosen by the user. In order to evaluate the sensitivity of the approach over such precision, we show in Figs. 12 and 13 the same scatter plot replicated with precision $h = 5$, corresponding to cells of twice the area w.r.t. the previous case, and $h = 7$, corresponding to cells of half the original area. As we can see, the impact remains virtually the same as $h = 6$, suggesting that this is not a critical parameter – although values much smaller or much larger than these are expected to be not effective, since very small ones yield huge cells potentially covering entire cities, and very large ones create cells that are too small to capture significant amounts of points.

7.2.2 Comparison of evaluation measures

In the following we first compare the ACTS methods against ATS, and later we compare their performances with those of the two variants of the competitor considered, namely HEH-D and HEH-O. All evaluations are based on the metrics defined earlier.

Comparing our approaches, in Table 3 we observe that ACTS_{LOC} improves the performance of ATS in terms of $MF_{.25}$ and TP for Rome dataset. On the other hand, in Table 4 we

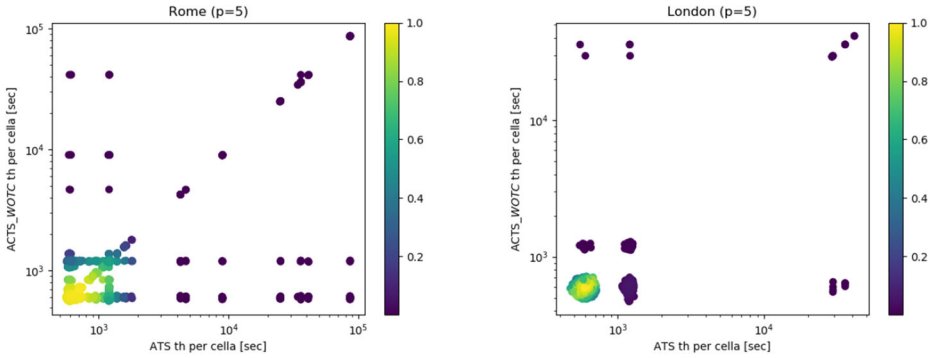

Fig. 12 Comparison of Acts_{WOTC} vs. ATS thresholds for all user-cell pairs, lowering the precision value ($h = 5$), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric

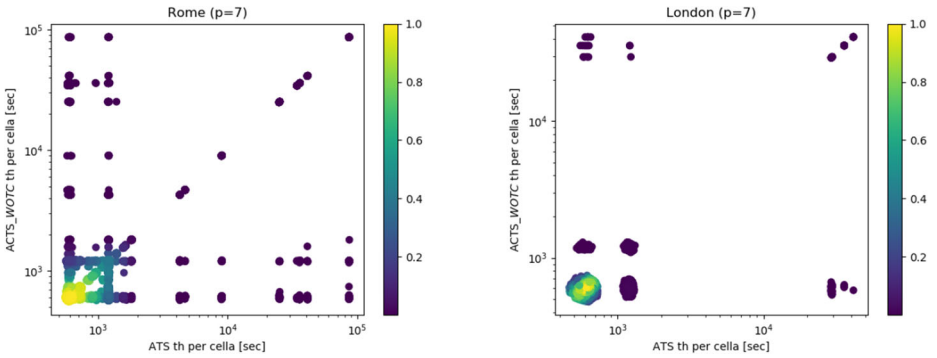

Fig. 13 Comparison of Acts_{WOTC} vs. ATS thresholds for all user-cell pairs, increasing the precision value ($h = 7$), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric

can see that on the London dataset all three approaches are comparable in terms of performance, reaching higher levels of *DC* compared to Rome. In terms of sampling ratio (fourth column) the ACTS methods show an improvement against ATS, since their lower value (more pronounced for ACTS_{LOC}, and a bit marginal for Acts_{WOTC}) means that the former create trajectories with smaller internal time gaps. In addition, by analyzing the number of segments (last column) we can see that values for ACTS_{LOC} are higher than ATS and those for Acts_{WOTC} are comparable, though slightly higher and with a slightly lower standard deviation. These factors, combined with the smaller $ratio_{\text{sr}}$ of ACTS methods with respect to ATS, imply that overall the local and *wisdom of the crowd* mechanisms suggest more changes towards smaller thresholds, therefore leading to more splits.

Figure 14 reports the $MF_{.25}$ for all our approaches and the FTS baselines as boxplots. For the Rome case we can observe that the distribution of values of Acts_{WOTC} is similar to ATS, only slightly more compact, while that of ACTS_{LOC} has slightly higher median and a significantly smaller inerquartile range. The differences in London are much less visible. In summary, the evaluation measures suggest that the ACTS methods achieve a small but interesting improvement over the basic ATS.

The last two lines of Tables 3 and 4 show the measures obtained with the two competitors. We can observe that there is a great discrepancy between them and those obtained with our methods, suggesting that, on our dataset, the clustering-based methods are not able to

Table 3 Evaluation on Rome data

method	MF_{25}	*TP*	*DC*	$ratio_{\text{sr}}$	*#segms (avg* $\pm$ *std)*
ATS	.9513	.9507	.9876	0.0462	851.551 $\pm$ 717.173
ACTS_{LOC}	.9587	.9654	.9174	0.0379	946.743 $\pm$ 785.998
ACTS_{WOTC}	.9514	.951	.9856	0.0459	857.157 $\pm$ 713.349
HEH-O	.1560	.1538	.7874	0.0313	2400757.281 $\pm$ 2760922.811
HEH-D	.1877	.1308	.8586	0.0511	2244814.994 $\pm$ 3521517.705

The first three columns show the measures illustrated in Section 6. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments

Table 4 Evaluation on London data

method	MF_{25}	TP	DC	$ratio_{\text{sr}}$	*#segms (avg ± std)*
ATS	.9547	.9523	.9991	0.0480	433.612 ± 525.916
ACTS$_{LOC}$	.9538	.9517	.9983	0.0472	477.971 ± 588.472
ACTS$_{WOTC}$	.9545	.9523	.9991	0.0478	433.652 ± 525.845
HEH-O	.3660	.3492	0.7513	0.0561	66389.061 ± 150547.085
HEH-D	.8877	.9140	.8401	0.0459	242882.126 ± 963253.651

The first three columns show the measures illustrated in Section 6. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments

segment trajectories in an effective way. In particular, both HEH-D and HEH-O produce highly fragmented segments (see their huge number of segments yielded) leading to a medium-low distance coverage and a very low time precision – the only exception being HEH-D on London, which however further shows how its behaviour is unstable. Figure 15 reports the boxplots showing the distribution of $MF_{.25}$ for the different approaches. We immediately notice that the scores obtained by HEH-O and HEH-D are significantly worse than the others. In light of the results obtained, we will not discuss these two competitors any further in the paper, focusing instead on the behaviour of the other methods.

7.2.3 Comparison of segmentation statistics

Similarly to what done in Section 7.1.2 for ATS, in the following we analyze other statistical indicators on the trajectory segments extracted by the ACTS methods. In Fig. 16 we report the distributions of the average number of segments per user and points per segment for Rome (top) and London (bottom). The average number of segments per user (first column), highlights that in Rome ATS and Acts$_{WOTC}$ yield similar distributions, while ACTS$_{LOC}$ generates more users with an high number of segments. In London the distribution is more skewed towards low numbers of segments, again with ACTS$_{LOC}$ with a peak on higher values. In terms of number of points per segment (right column), we can see that in Rome most segments have between 5 and 15 points, yet ACTS$_{LOC}$ shows a more concentrated peak on 5-10 points, which is coherent with the previous results (more segments are generated, and consequently they are shorter, on average). Something similar happens in London, now the concentration of values being between 15 and 30 points.

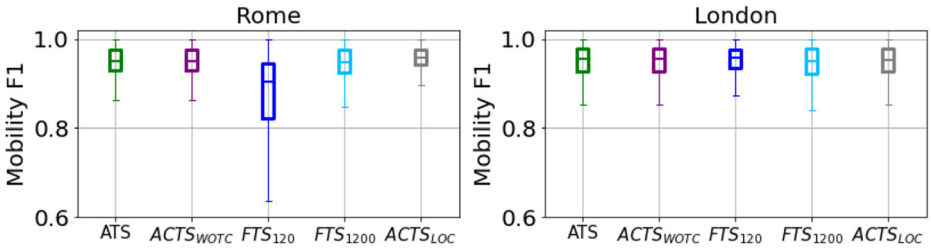

Fig. 14 Boxplots for $MF_{.25}$. On the Rome data ACTS$_{WOTC}$ yields results similar to ATS, while ACTS$_{LOC}$ significantly improves them. On London the differences are less pronounced

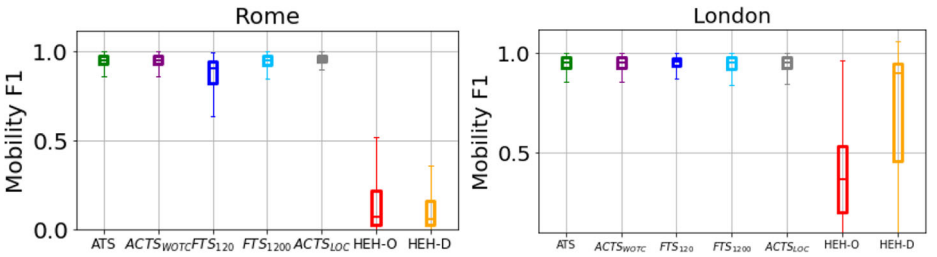

Fig. 15 Boxplots for $MF_{.25}$. In this case it is possible to see the comparison in terms of performance between our approaches and the DBSCAN and OPTICS cluster methods. In both cases the performance of the cluster methods are visibly worse than those achievable with ATS and ACTS

7.2.4 Run times analysis

We present here some performance experiments regarding the scalability of our proposed methods w.r.t. the number of input trajectories (i.e. users) and their duration. In the first experiments, we test how the running time changes by varying the number of users in a range from 200 to 2000 (in steps of 200 and 250) while in the second ones we test it by varying the number of months covered by the data. In particular, in the last case we start from the data of a single month (January) and gradually add the next months one by one, obtaining 12 different datasets. These tests were made on the three ATS/ACTS approaches we proposed, compared with the two methods used as baseline (FTS and RTS). The experimental results for both Rome and London are shown in Fig. 17. As it is possible to notice, the trends of FTS and RTS are linear and very low in both plots, confirming that their simplicity yields very fast executions. As expected, ATS and ACTS have much higher computation times, yet their trends appear to be linear or quasi-linear w.r.t. both the dimensions considered (users and duration), confirming the hypothesis made in Section 4.

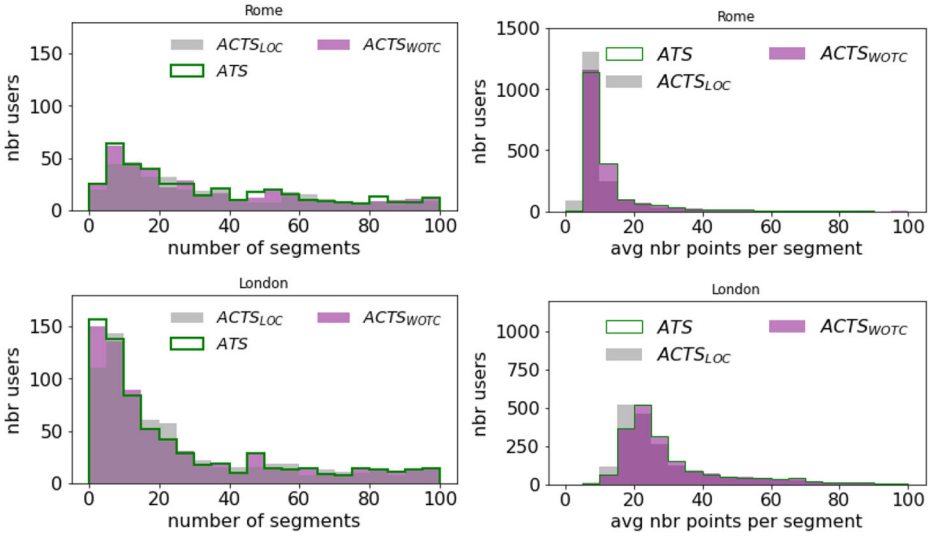

Fig. 16 Distribution of the number of segments, points and trajectories (from left to right) over Rome (top) and London (bottom)

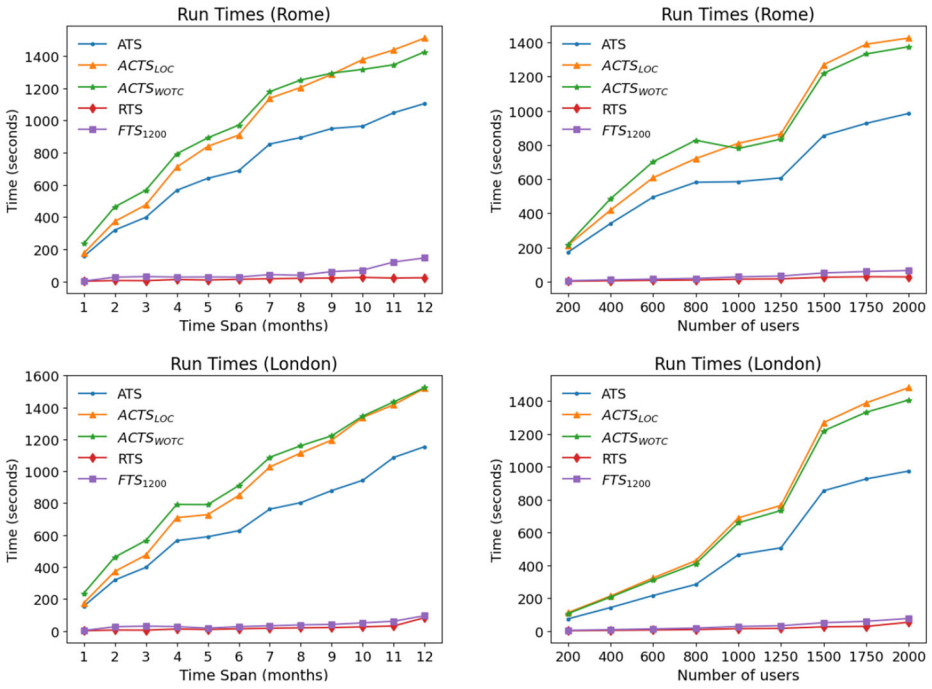

Fig. 17 Run times experiments in function of the number of users and the data collection period. In both cases the time trend grows quite linearly

8 Conclusion

In this paper we have presented a set of user adaptive methods for solving the trajectory segmentation problem, a very common and useful task in mobility data mining, especially in preprocessing phases. The solutions proposed take into consideration the overall trajectory of the user, identifying an individual cut time threshold (each user can potentially have a different threshold) and also combining the information coming from the different users through the spatial regions they share. This process yields thresholds for trajectory segmentation which are not only user-adaptive, but also location-adaptive, thus taking into account that a stop at different places might require time intervals of different duration to be considered a significant stay – and thus a trajectory cut point. The experiments show that the individual and collective adaptive strategies have a significant impact on the thresholds obtained, which lead to a performance improvement in terms of the metrics defined for this purpose.

Having a refined segmentation, as those obtained with ATS and the ACTS family, is very important in applications where the individual behaviour is under study. For this reason, future works on this line include the integration of our methods into existing applications in the domain of crash prediction [16] and simulations of Electric Vehicles mobility [32] which are based on a detailed modeling of users' mobility history.

Also, the results obtained in this paper suggested us to explore the feasibility of some more flexible individual mobility models. In particular, the idea is to depart from the notion of single trips, and instead allow a multi-resolution, hierarchical view where the same

movement is interpreted both as one trip and, possibly, as a sequence of several small ones. The different levels of the hierarchy might be obtained by moving the time threshold τ (the same one that in the present work we tried to fix to either a single value or a few ones for each user) up and down, linking the segments that originate from a split of an existing one. The resulting model would clearly be complex and its computation and management challenging.

Acknowledgements This work is partially supported by the European Community H2020 programme under the funding scheme *Track & Know* (Big Data for Mobility Tracking Knowledge Extraction in Urban Areas), G.A. 780754, https://trackandknowproject.eu/.

Author Contributions Agnese Bonavita: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing, Visualization. Riccardo Guidotti: Conceptualization, Methodology, Investigation, Resources, Writing. Mirco Nanni: Conceptualization, Methodology, Investigation, Resources, Writing, Supervision, Project administration, Funding acquisition.

Funding Open access funding provided by Scuola Normale Superiore within the CRUI-CARE Agreement. This work is partially supported by the European Community H2020 programme under the funding scheme *Track & Know* (Big Data for Mobility Tracking Knowledge Extraction in Urban Areas), G.A. 780754, https://trackandknowproject.eu/.

Availability of Data and Material The datasets adopted in this work are private, and were provided within the scope of the Track & Know project (https://trackandknowproject.eu/).

Code Availability The code is open source, and can be downloaded at:
https://github.com/malciughina/Trajectorypublic/tree/ACTSsegmentation

References

1. Alewijnse SP, Buchin K, Buchin M, Kölzsch A, Kruckenberg H, Westenberg MA (2014) A framework for trajectory segmentation by stable criteria. In: Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems, pp 351–360
2. Berlingerio M, Ghaddar B, Guidotti R, Pascale A, Sassi A (2017) The graal of carpooling: Green and social optimization from crowd-sourced data. Transport Res Part C Emerg Technol 80:20–36
3. Bingham E (2010) Finding segmentations of sequences. In: Inductive databases and constraint-based data mining. Springer, pp 177–197
4. Bonavita A, Guidotti R, Nanni M (2020) Self-adapting trajectory segmentation. In: EDBT/ICDT Workshops
5. Bremer R. (1995) Outliers in statistical data
6. Buchin M, Driemel A, Van Kreveld M, Sacristán V (2010) An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 202–211
7. Bussemaker HJ, Li H, Siggia ED et al (2000) Regulatory element detection using a probabilistic segmentation model. In: Ismb, pp 67–74

8. Cich G, Knapen L, Bellemans T, Janssens D, Wets G (2016) Threshold settings for trip/stop detection in gps traces. J Ambient Intell Human Comput 7(3):395–413
9. Das RD, Winter S (2016) Automated urban travel interpretation: a bottom-up approach for trajectory segmentation. Sensors 16(11):1962
10. Etemad M, Júnior AS, Hoseyni A, Rose J, Matwin S (2019) A trajectory segmentation algorithm based on interpolation-based change detection strategies. In: EDBT/ICDT Workshops
11. Etemad M, Soares A, Etemad E, Rose J, Torgo L, Matwin S (2020) Sws: an unsupervised trajectory segmentation algorithm based on change detection with interpolation kernels. GeoInformatica 1–21
12. Fu C, Huang H, Weibel R (2021) Adaptive simplification of gps trajectories with geographic context – a quadtree-based approach. Int J Geogr Inf Sci 35(4):661–688. https://doi.org/10.1080/13658816.2020.1778003
13. Galton F (1907) Vox populi
14. Gong L, Yamamoto T, Morikawa T (2018) Identification of activity stop locations in gps trajectories by dbscan-te method combined with support vector machines. Transport Res Procedia 32:146–154
15. Guidotti R, Monreale A, Nanni M, Giannotti F, Pedreschi D (2017) Clustering individual transactional data for masses of users. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 195–204
16. Guidotti R, Nanni M (2020) Crash prediction and risk assessment with individual mobility networks. In: 2020 21St IEEE international conference on mobile data management (MDM). IEEE, pp 89–98
17. Guidotti R, Nanni M, Rinzivillo S, Pedreschi D, Giannotti F (2017) Never drive alone: Boosting carpooling with network analysis. Inf Syst 64:237–257
18. Guidotti R, Trasarti R, Nanni M (2015) Tosca: two-steps clustering algorithm for personal locations detection. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. ACM, p 38
19. Guidotti R, Trasarti R, Nanni M, Giannotti F, Pedreschi D (2017) There's a path for everyone: a data-driven personal model reproducing mobility agendas. In: 2017 IEEE International conference on data science and advanced analytics (DSAA). IEEE, pp 303–312
20. Guo S, Li X, Ching WK, Dan R, Li WK, Zhang Z (2018) Gps trajectory data segmentation based on probabilistic logic. Int J Approx Reason 103:227–247
21. Himberg J, Korpiaho K, Mannila H, Tikanmaki J, Toivonen HT (2001) Time series segmentation for context recognition in mobile devices. In: Proceedings 2001 IEEE international conference on data mining. IEEE, pp 203–210
22. Hwang S, Evans C, Hanke T (2017) Detecting stop episodes from gps trajectories with gaps. In: Seeing cities through big data. Springer, pp 427–439
23. Izakian Z, Mesgari MS, Weibel R (2020) A feature extraction based trajectory segmentation approach based on multiple movement parameters. Eng Appl Artif Intell 88:103394
24. Júnior AS et al (2015) Grasp-uts: an algorithm for unsupervised trajectory segmentation. Int J Geogr Inf Sci 29(1):46–68
25. Júnior A. S. et al (2018) A semi-supervised approach for the semantic segmentation of trajectories. In: 19Th IEEE international conference on mobile data management (MDM), pp 145–154
26. Keogh E, Lonardi S, Ratanamahatana CA (2004) Towards parameter-free data mining. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 206–215
27. Koutroumanis N, Santipantakis GM, Glenis A, Doulkeridis C et al (2020) Scalable enrichment of mobility data with weather information. GeoInformatica 1–19
28. Lavrenko V, Schmill M, Lawrie D, Ogilvie P, Jensen D, Allan J (2000) Mining of concurrent text and time series. In: KDD-2000 Workshop on text mining, vol 2000, pp 37–44
29. Lee JG et al (2007) Trajectory clustering: a partition-and-group framework. In: ACM SIGMOD. ACM, p 593–604
30. Leiva L, Vidal E (2013) Warped k-means: an algorithm to cluster sequentially-distributed data. Inf Sci 237:196–210
31. Li W (2001) Dna segmentation as a model selection process. In: Proceedings of the fifth annual international conference on Computational biology. ACM, pp 204–210
32. Longhi L, Nanni M (2020) Car telematics big data analytics for insurance and innovative mobility services. J Ambient Intell Human Comput 11:3989–3999
33. Mann R, Jepson AD, El-Maraghi T (2002) Trajectory segmentation using dynamic programming. In: Object recognition supported by user interaction for service robots, vol 1. IEEE, pp 331–334
34. Morton G (1966) A computer oriented geodetic data base and a new technique in file sequencing. In: IBM Research report
35. Pappalardo L, Simini F, Rinzivillo S, Pedreschi D, Giannotti F, Barabási AL (2015) Returners and explorers dichotomy in human mobility. Nature Commun 6:8166

36. Pavlıček A, Pačes J, Clay O, Bernardi G (2002) A compact view of isochores in the draft human genome sequence. FEBS Lett 511(1-3):165–169
37. Ramensky VE, Makeev VJ, Roytberg MA, Tumanyan VG (2000) Dna segmentation through the bayesian approach. J Comput Biol 7(1-2):215–231
38. Rinzivillo S, Gabrielli L, Nanni M, Pappalardo L, Pedreschi D, Giannotti F (2014) The purpose of motion: Learning activities from individual mobility networks. In: 2014 International conference on data science and advanced analytics (DSAA). IEEE, pp 312–318
39. Safi H, Assemi B, Mesbah M, Ferreira L (2016) Trip detection with smartphone-assisted collection of travel data. Transp Res Rec 2594(1):18–26
40. Siła-Nowicka K, Vandrol J, Oshan T, Long JA, Demšar U, Fotheringham AS (2016) Analysis of human mobility patterns from gps trajectories and contextual information. Int J Geogr Inf Sci 30(5):881–906
41. Stough T, Cressie N, Kang E et al (2020) Spatial analysis and visualization of global data on multi-resolution hexagonal grids. Japanese J Stat Data Sci 3:107–128
42. Tan PN, Steinbach M, Kumar V (2018) Introduction to data mining Pearson Education India
43. Terzi E, Tsaparas P (2006) Efficient algorithms for sequence segmentation. In: Proceedings of the 2006 SIAM international conference on data mining. SIAM, pp 316–327
44. Thierry B, Chaix B, Kestens Y (2013) Detecting activity locations from raw gps data: a novel kernel-based algorithm. Int J Health Geographics 12(1):1–10
45. Trasarti R, Guidotti R, Monreale A, Giannotti F (2017) Myway: Location prediction via mobility profiling. Inf Syst 64:350–367
46. Trasarti R, Pinelli F, Nanni M, Giannotti F (2011) Mining mobility user profiles for car pooling. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 1190–1198
47. Yan Z, Chakraborty D, Parent C, Spaccapietra S, Aberer K (2013) Semantic trajectories: Mobility data computation and annotation. ACM Trans Intell Syst Technol (TIST) 4(3):49
48. Zhao F, Ghorpade A, Pereira FC, Zegras C, Ben-Akiva M (2015) Stop detection in smartphone-based travel surveys. Transport Res Procedia 11:218–226
49. Zheng Y, Zhang L, Ma Z et al (2011) Recommending friends and locations based on individual location history. ACM Trans Web (TWEB) 5(1):5

Agnese Bonavita was born in 1993 in Genoa, Italy. In 2015 she graduated in Physics (bachelor degree) at University of Genoa and in 2018 she graduated in High Energy Physics (master degree) at Pisa University with a thesis on a rare Higgs decay at CMS experiment (CERN). She had research experiences in the two most important particle physics laboratories: Fermilab (Chicago, USA) as summer students and Cern (Geneve,CH) as undergraduate student. She's currently a Ph.D. fellow in Data Science in the joint program offered by Scuola Normale Superiore, Scuola Superiore Sant'Anna, University of Pisa, IMT Lucca and National Research Council.

Her research project, included in the smart cities area, is about human mobility models and predictions. In particular her research interest is on electric vehicles simulations in order to understand the benefits of this new technology and on developing a deep learning method able to predict car accidents and improve risk assessment analysis. For a year she took part of the Track&Know project included in Horizon 2020 project which aims to research, develop and exploit a new software framework that aims at increasing the efficiency of Big Data in the mobility field.

Riccardo Guidotti was born in 1988 in Pitigliano (GR) Italy. In 2013 and 2010 he graduated cum laude in Computer Science (MS and BS) at University of Pisa. He received the PhD in Computer Science with a thesis on Personal Data Analytics in the same institution. He is currently an Assistant Professor (RTD-A) at the Department of Computer Science University of Pisa, Italy and a member of the Knowledge Discovery and Data Mining Laboratory (KDDLab), a joint research group with the Information Science and Technology Institute of the National Research Council in Pisa. He won the IBM fellowship program and has been an intern in IBM Research Dublin, Ireland in 2015. His research interests are in personal data mining, clustering, explainable models, analysis of transactional data.

Topics of interest: Explainable Models, Personal Anaytics, Clustering Analysis of Transactional Data.

Mirco Nanni holds a degree and a PhD in Computer Science (University of Pisa, in 1997 and 2002). He is currently a permanent researcher at ISTI - CNR in Pisa, member of the KDD Laboratory. He was visiting researcher at the Computer Science Department of College Park, University of Maryland (1999), SENSEable Lab. at MIT Boston (2008), Transportation Research Institute at Hasselt University of Belgium(2010) and the Applied Movement Behaviour Research Group, University of Brunswick - Canada (2012).

He has been working in the computer science research since late 90's, especially in the databases and data mining areas. He collaborates to the main international conferences in the area, in the roles of author, program committee member or reviewer. He authored 60+ international publications, mainly on several aspects of human mobility, such as data mining on spatio-temporal data (theory, algorithms and applications for clustering, classification and sequential patterns for trajectories of moving objects), applications on transportation and smart cities (traffic models, carpooling systems), mobile phone data analysis (GSM-based estimation of population and flows) and Big Data for Official Statistics (leveraging mining models for extracting reliable statistics).

He participated to several national and international research projects, as researcher or coordinator, the most recent and relevant ones being: PETRA (Personal Transport Advisor: an integrated platform of mobility patterns for Smart Cities to enable demand-adaptive transportation system); ICON (Inductive Constraint Programming); DataSIM (DATA science for SIMulating the era of electric vehicles); LIFT (Using Local Inference in Massively Distributed Systems); MOVE (Knowledge Discovery from Moving Objects).

He served as program chair for workshops "SAWM: Statistical Approaches for Web Mining", 2004, "STDM: IEEE Workshop on Spatio-Temporal Data Mining", 2007, "DAMASCA: DAta Mining And Smart Cities Applications Workshop", 2015 and program vice-chair for ICDM 2008, IEEE Int. Conf. on Data Mining. Moreover, he serves as regular program committee member and reviewer for several of the most prominent conferences and workshops in the fields of databases and data mining, including: ACM CIKM, ECML/PKDD, IEEE ICDM, ACM KDD, SADM, SIAM DM, SSTD, IEEE SSTDM. Finally, he regularly contributes to the revision of papers for major journals in the field, including: VLDB J., TKDE, DKE, GeoInformatica, Information Systems, IJGIS, KaIS, SADM J., Machine Learning J. Since 2007, he regularly holds courses for graduate and undergraduate students, at Università di Pisa, on databases and data mining. *Topics of interest*: Mobility Data Analysis, Big Data for Official Statistics, Clustering methods Temporal and Sequential patterns.