Noname manuscript No.
(will be inserted by the editor)

An Effective and Versatile Distance Measure for Spatiotemporal Trajectories

Somayeh Naderivesal · Lars Kulik · James Bailey

Received: date / Accepted: date

Abstract The analysis of large-scale trajectory data has tremendous benefits for applications ranging from transportation planning to traffic management. A fundamental building block for the analysis of such data is the computation of similarity between trajectories. Existing work for similarity computation focuses mainly on the spatial aspects of trajectories, but more rarely takes into account time in conjunction with space. A key challenge when considering time is how to handle trajectories that are sampled asynchronously or at variable rates, which can lead to uncertainty. To tackle this problem, we quantify trajectory similarity as an interval, rather than a single value, to capture the uncertainty that can result from different sampling rates and asynchronous sampling. Based on this perspective, we develop a new trajectory similarity measure, Trajectory Interval Distance Estimation (TIDE), which models similarity computation as a convex optimisation problem. Using two real datasets, we demonstrate that our proposed measure is extremely effective for assessing similarity in comparison to existing state of the art measures.

Keywords Spatiotemporal Trajectory · Similarity · Distance · Measure · Uncertainty

1 Introduction

Location-aware devices have become an integral part of our daily lives. With their widespread use, a tremendous amount of data in the form of spatiotemporal datasets have become available, including taxi-data [1] and Cabspotting [2] data. Activities that mine and

S. Naderivesal
School of Computing and Information Systems
The University of Melbourne
E-mail: naderis@student.unimelb.edu.au

L. Kulik
School of Computing and Information Systems
The University of Melbourne
E-mail: lkulik@unimelb.edu.au

J. Bailey
School of Computing and Information Systems
The University of Melbourne
E-mail: baileyj@unimelb.edu.au

extract knowledge, such as moving objects clustering [3], finding the most similar trajectory for a given trajectory [5] or predicting object trajectories based on historical data [6] are examples of the range of current applications. All these applications require the ability to compare trajectories using a distance (similarity) measure.

Since trajectories have both temporal and spatial dimensions, different types of similarity measures have been developed to compare them, including relative-motion-pattern (REMO-pattern) [7], shape-based (spatial) similarity [8,30] and spatiotemporal similarity [9]. In this paper, we focus on spatiotemporal similarity, which has only been considered in a few existing works.

Developing a similarity measure for trajectory data is a challenging task when both temporal and spatial dimensions are considered. For example, in a social network application, suppose we are looking for people with similar trajectories, so we may recommend that they become socially connected to each other. Since people use different devices and applications to record their location data, each user,s trajectory is likely to have a different sampling rate [11]. One user's device may record location data every 5 seconds, while another user's records every 10 seconds. If these users travel together, then for half of the timestamps in the first trajectory, there are no recorded points in the second trajectory. Moreover, even if the number of sampled points in both trajectories is the same, there is no guarantee that the sampled points are recorded synchronously. Taking a different example, suppose one is extracting vehicle convoys (a group of vehicles that travel together) from a spatiotemporal dataset. Assume that the sampling frequency is 1 per Δt and one device starts to record its location at time t, while another starts at time $t+\varepsilon$ ($0 < \varepsilon < \Delta t$). Even though these two trajectories may have the same number of sampled points, for every timestamp in one trajectory, there is no location data in the other trajectory. In summary, when the task is measuring "spatiotemporal" similarity of trajectories, the sampling frequency and asynchronous sampling are key issues to consider.

Several measures have been proposed to quantify the similarity or distance of trajectories including Euclidean distance [12], Dynamic Time Warping (DTW) [13], Edit distance with Real Penalty (ERP) [15], Edit Distance on Real sequence (EDR) [16], Fréchet distance [33–35] and recently Edit Distance with Projection (EDwP) [17]. In addition, measures like Longest Common Subsequence (LCSS) [9,14] and DISSIM [5] have been introduced for trajectory similarity search. Existing measures have the following limitations:

L1 *Two trajectories that have been sampled at different rates might refer to the same trip.* For example, in Fig. 1(a), we wish to determine for the trajectories of the circle and square objects, T^1 and T^2, if they describe the same trip. Since the circle object does not have a sampled point at time t_2, there are two possible paths that it has taken, Path1 and Path2. Here, a key constraint is the speed limit. If the speed limit allows the circle object to travel along the Path2 (same as the square object's path), then it might be close to the square object at time t_2. In this situation, these two trajectories could represent the same trip. Another possibility is that it is in the Path1 and they represent two different trips. However, no existing measure is able to distinguish between these situations.

L2 *Two trajectories that have been sampled asynchronously might represent the same trip.* In Fig. 1(b), for certain maximum speeds of the circle and square objects, they might represent the same trip. For example, if the maximum speed of the square object allows, it might be close to the circle object at time t_2 and if the maximum speed of the circle object allows, it might be close to the square object at time t_3 and so forth (two objects are travelling together). However, most of existing works consider those sampled points

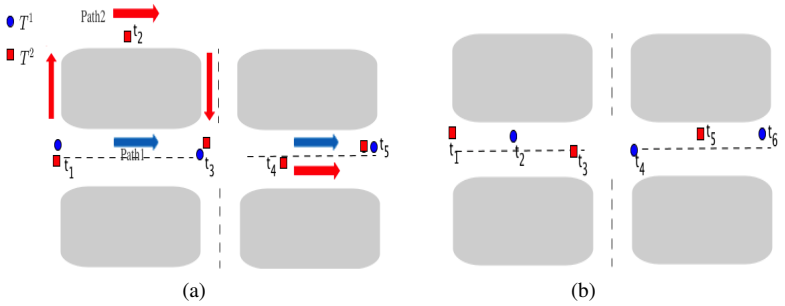


Fig. 1 (a) The circle object might have the same path as the square object (L1) (b) Both objects might move together but sample their trajectory asynchronously (L2)

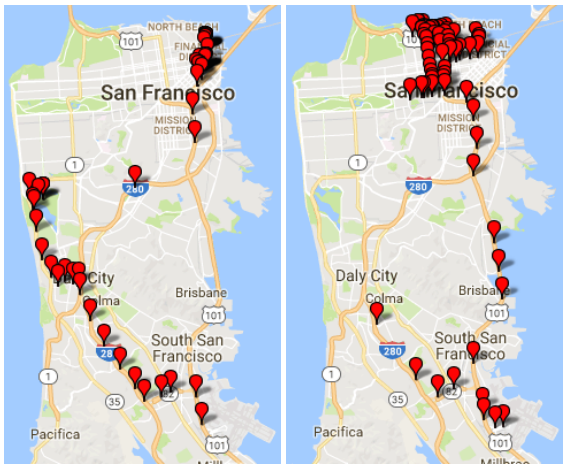
referring to trajectories of two different trips, because they cannot match any of the sampled points in T^1 and T^2.

L3 *The time-complexity of a spatiotemporal similarity measure is an important criterion.* A typical trajectory analysis compares thousands of trajectories, where every trajectory might have hundreds or thousands of sampled points. Identifying the most similar trajectory to a given query trajectory from a dataset may require a large number of comparisons. Most existing measures have quadratic time-complexity for similarity comparison.

In many types of trajectories including road-network trajectories, objects do not travel along a straight path. Fig. 2(a) illustrates a sample trajectory of the Cabspotting dataset. If we use a lower-sampled version of this trajectory (half of the sampled points selected randomly) as a query trajectory, we expect its most similar trajectory to be its original (reference) trajectory. However, using the time-constrained DTW [22], another trajectory is extracted as its most similar trajectory (Fig. 2(b)) which is the 10^{th} similar trajectory to the reference trajectory. This real-world example shows different sampling rates has an adverse effect on the effectiveness of a baseline measure such as DTW.

Zheng et al. proposed two approaches in [28] and [29] for dealing with trajectories with low sampling rates in road networks. To answer range queries, they used road networks and the maximum allowed speed on the road segments to compute the probabilistic location of an object at every timestamp between the time of two consecutive sampled points. However, these two approaches and also the approach in [32] are limited to movements along road networks. Moreover, they approximate consecutive non-sampled points independently, while traveling between those points might not be feasible (see Example 4). In Sect. 6, we will discuss that these methods can be used as a complementary method to assign a probability to each of similar trajectories when they are travelling in road network.

As discussed before, different sampling rates and asynchronous sampling result in uncertainty about the trajectory similarity comparison (L1 and L2). As summarised in Table 1, none of the current measures are completely effective for the similarity computation of spatiotemporal trajectories with different sampling rates and times. Using movement and speed constraints, we tackle this uncertainty by defining a distance (similarity) *interval* (rather than a single value), which denotes the minimum and maximum possible distances of two trajectories as its bounds. However, the computation of distance interval bounds is a computationally expensive optimisation problem. We study two approaches to compute the interval: an exact and an approximate version. For the exact computation, we use a solver to compute



(a) (b)

Fig. 2 (a) A sample trajectory T^1 from Cabspotting dataset (b) The most similar trajectory to a lower sampled version of T^1, under DTW measure

the bounds which results in an exact albeit costly computation (Sect. 5.5). Another approach is to approximate the answer without any significant adverse effect on the effectiveness of the measure. The latter approach runs in quasilinear time and thus addresses L3.

In summary, our main contributions are as follows:

- To address L1 and L2, we introduce the new concept of distance *interval* for spatiotemporal trajectories which has the minimum and maximum distances between two trajectories as its bounds.
- Computing the exact bounds of the distance interval between trajectories is a computationally expensive convex optimisation problem. We introduce an approximation method for computing the minimum and maximum distances in $O(nlogn)$ time.
- We evaluate the efficiency and effectiveness of our measure (called TIDE) for two real datasets. We show that both the exact and the approximate versions of TIDE outperform existing measures. We demonstrate that the accuracy of our approximation method is comparable to the exact method and it is almost as efficient as DISSIM, which runs in linear time.

In the rest of this paper, we first review related work in Sect. 2. In Sect. 3, we formally describe our novel concept of trajectory distance interval. We explain our proposed distance interval computation approaches in Sect. 4. Sect. 5 presents the experimental results and we discuss our results in Sect. 6. We conclude our findings in Sect. 7.

2 Related work

Recently, similarity search for trajectory data has been an important area of research. Since trajectories have both temporal and spatial dimensions, different types of similarity measures have been developed to compare them, including relative-motion-pattern (REMO-pattern) [7,36], shape-based (spatial) similarity [8,30,17] and spatiotemporal similarity [9]. To differentiate between them, consider Fig. 3 that illustrates four spatiotemporal trajectories T^1,

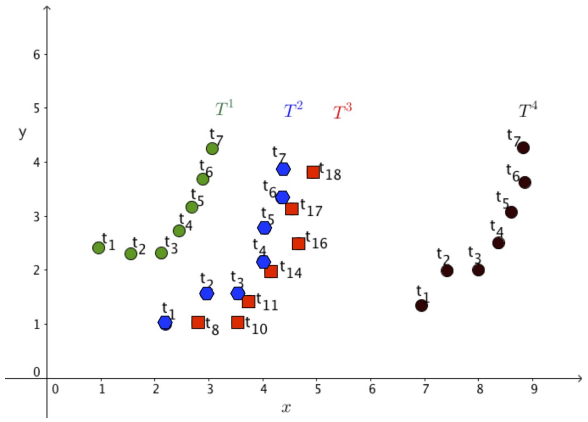


Fig. 3 Illustration of four different spatiotemporal trajectories. T^2 is more similar to T^4 in terms of speed and bearing changes, to T^3 in terms of spatial similarity and to T^1 concerning spatiotemporal similarity

T^2, T^3 and T^4 (for simplicity of illustration, the time dimension is represented as a label for every point). Comparing T^2 with the three other trajectories, it is more similar to T^4 in terms of REMO-patterns similarity (speed and bearing changes) and to T^3 in terms of spatial similarity. If considering both spatial and temporal aspects, T^2 is more similar to T^1.

In some REMO-pattern similarity comparison methods such as [7], trajectories are first transformed into basic motion attributes, i.e., speed and motion azimuth (bearing). Then, they are compared using their motion attributes over time. However, in [36] (MMAD), they transform motions using translations and rotations and then they compare trajectories. In [36], the original problem is an optimisation problem, however, they use an approximation method for a faster comparison between trajectories.

For shape-based similarity computation, only the spatial dimensions of trajectories (independent of time) are considered. Euclidean distance was applied in [18] and [12] to compare time-series with the assumption that time-series have the same number of data points. This assumption limits the application of Euclidean distance measure (and generally Lp-norm measures) [19]. However, the time-complexity of Euclidean distance is $O(n)$ which is an advantage for this measure [21]. Considering "acceleration and decelerations on the rate of sequences" Dynamic Time Warping (DTW) [13,20] has been proposed to enable time-series to be expanded or compressed to match to each other. In other words, DTW is looking for the maximum similarity between values and matches values by time warping. Chen et al. have proposed EDR [16] and ERP [15] which are based on edit distance. Edit distance or Levenshtein distance has been introduced for string matching and it counts the number of edits (including insertion, deletion and substitution) which is required for transforming one string to another. EDR uses ε as the threshold parameter, but rather than counting matched points, it counts mismatches as the distance between two trajectories. However, ERP has a different approach for measuring the distance. It uses a constant reference point g and computes the distance of every point without a match from g as the penalty. The recent approach Edit Distance with Projections (EDwP) [17] uses projection independent of time which makes it applicable only for spatial similarity measurement.

For the spatiotemporal similarity comparison, both spatial and temporal dimensions are important. As discussed before, DTW is a spatial similarity measure, however, in [22], a warping window has been added to DTW for restricting the amount of time shifting in order

Table 1 Summary of similarity measures

Measure	Time sensitive	Handles difference sampling rates	Handles asynchronous sampling	Original time complexity[1]
Lp-norm	No	N.A.	N.A.	$O(n)$
DTW	No	No	No	$O(n^2)$
Discrete Fréchet	No	No	No	$O(n^2)$
Continious Fréchet	No	Yes	No	$O(n^2)$
LCSS	Yes	No	No	$O(n^2)$
EDR	No	No	No	$O(n^2)$
ERP	No	No	No	$O(n^2)$
DISSIM	Yes	Yes	No	$O(n)$
EDwP	No	Yes	No	$O(n^2)$
MMAD	No	Yes	No	$O(n^2)$
STLIP	No	Yes	No	$O(nlogn)$

to increase the accuracy and decrease the time complexity. Another measure that also limits the extent of time shifting is LCSS [9,14]. LCSS has been introduced for trajectory data and utilises the Longest Common Subsequence which is basically a method for sequence matching. LCSS uses delta δ as a threshold for time shifting and epsilon ε as the maximum difference between spatial dimensions so that they can be considered as a match. Indeed, two points will be counted as a match if their time difference is less than δ and also their spatial distance in every dimension is not greater than ε. DISSIM [5] has been introduced for spatiotemporal trajectories to overcome the previous measures' drawbacks including ignoring time dimension and nonsupporting difference in the sampling frequency. By assuming a direct line between known points, it approximates the non-sampled points by interpolating and defines the distance of two trajectories as the integral of Euclidean distance between them (which is equal to the area between them) for a given time interval. DISSIM utilises time-dependent interpolation for the approximation of the non-sampled points, However, using both interpolation and projection is a drawback for those measures, since they define the non-sampled points by the assumption that there is a direct line between consecutive points, which not necessarily be true. Pelekis et al. proposed a spatiotemporal distance measure, STLIP, in [37], that computes the spatial and temporal distance separately. Using this measure, the spatiotemporal distance between two trajectories is the sum of a) the spatial distance and b) the temporal distance multiplied with spatial distance and k (k represents a user assigned importance to the time factor). However, if two trajectories have a spatial distance of 0, their temporal distance will be ignored.

As summarised in Table 1, none of the current measures are fully satisfactory for the similarity comparison of spatiotemporal trajectories with different sampling rates and times. In this paper, we focus on this last type of similarity, spatiotemporal similarity, which has only been considered in a few existing works.

3 Preliminaries and problem statement

Let $DB = (T^1, T^2, ..., T^m)$ be a set of m spatiotemporal trajectories, while $T^i = (p^i_1, p^i_2, ..., p^i_{n_i})$ is a set of n_i time-stamped coordinates in the form of $p^i_j = (x_{p^i_j}, y_{p^i_j}, t_{p^i_j})$ which are ordered by time.

[1] Time complexity of the original measure

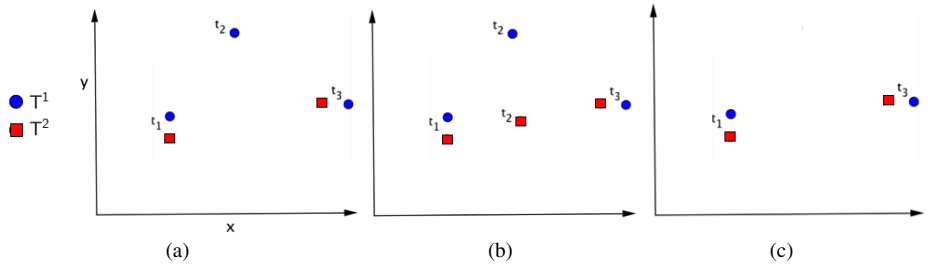


Fig. 4 Illustration of how different measures handle non-sampled points. (a) Two sample trajectories T^1 and T^2 with a non-sampled point in T^2 (b) DISSIM approximates the non-sampled point using interpolation (c) LCSS omits the sampled point in T^1 at time t_2

Since Trajectories do not necessarily have the same sampling rates and times, they may not share the same sampled points which results in *non-sampled points* in trajectories compared to each other.

DEFINITION 1. A point $p_i^1 \in T^1$ is a *non-sampled point* of trajectory T^2 at time $t_{p_i^1}$ iff $\nexists p_j^2 \in T^2$ so that $t_{p_j^2} = t_{p_i^1}$ while $t_{p_1^2} < t_{p_i^1} < t_{p_{n_2}^2}$.

EXAMPLE 1. Let $T^1 = ((1,2,1),(2,3,3),(2,6,4),(3,7,5),(5,8,7))$ and $T^2 = ((4,5,1),(10,10,5),(10,12,6),(12,13,7))$, then based on the definition, there are two non-sampled points at time 3 and 4 in T^2 in comparison to T^1. Vice versa, for the timestamp 6, there is not any coordinate in T^1, while $(10,12,6)$ exists in T^2.

To compare trajectories, a key problem is non-sampled points resulting from different sampling rates and times. Fig. 4(a) illustrates two trajectories T^1 and T^2 (sample trajectories of objects O^1 and O^2) in which T^2 has a non-sampled point at time t_2 in comparison to T^1. Some existing measures such as DISSIM, use interpolation and projection to approximate non-sampled points (Fig. 4(b)). On the other hand, some other measures like LCSS use time shifting and some thresholds to handle non-sampled points in trajectories by omitting the corresponding sampled points like Fig. 4(c) or by matching them using time shifting. However, using interpolation, projection or time shifting independent of movement characteristics of objects can result in imprecise similarity assessment. We address the non-sampled points problem through using movement characteristics of objects, including an estimate of their travelled maximum speed. In fact, the location of an object at non-sampled points is uncertain. Using movement characteristics of an object including their estimated maximum speed, we will address this uncertainty. A straightforward plug-in estimate of the maximum speed of an object is to consider all pairs of adjacent points in the trajectory, and for each pair, compute the speed and then select the maximum speed value across all point pairs. This simple method is very effective in practice, which we will show in our experiments. Also, since we are approximating the maximum allowed speed, we investigated the impact of increasing the estimated maximum speed (see the Appendix). The results show that there is no considerable impact, if we underestimate the maximum speed.

Using the estimated maximum speed of an object, we are able to compute the minimum and maximum distances between two trajectories, when there are non-sampled points in one or both of them. For the trajectories in Fig. 4(a), using the estimated maximum speed of O^2, the nearest location of object O^2 to O^1 at time t_2 can be computed (Fig. 5(a)). Computing the non-sampled points in this way, the overall distance of the trajectories will be the minimum possible distance. Also, the farthest location of O^2 to O^1 at time t_2 is illustrated in Fig. 5(b)

Table 2 Notations used for the problem statement

Notation	Description
DB	A set of m trajectories, $DB = (T^1, T^2, ..., T^m)$
T^i	The i^{th} trajectory of DB, $T^i = (p^i_1, p^i_2, ..., p^i_{n_i})$, $\|T^i\| = n_i > 0$
p^i_j	The j^{th} time-stamped coordinate of T^i, $p^i_j = (x_{p^i_j}, y_{p^i_j}, t_{p^i_j})$
n	The maximum size of the trajectories in DB, $n = max(n_1, n_2, .., n_m)$
t_{T^i}	The set of sampled point timestamps in trajectory T^i, $t_{T^i} = (t_{p^i_1}, t_{p^i_2}, ..., t_{p^i_{n_i}})$
$[t_{min}, t_{max}]$	The longest common time interval of two trajectories (see Definition 2)
$t_{T^i} \cup t_{Tj}$	The set of sampled point timestamps of trajectories T^i and T^j ($\forall t \in t_{T^i} \cup t_{Tj}$, $t \in t_{T^i}$ *or* $t \in t_{Tj}$)
$V^i_{\max}$	Estimated maximum speed in trajectory T^i
$D(Q,R)$	The distance between two points Q and R
$\|\|Q-R\|\|_2$	The Euclidean distance between points Q and R
$C(x_c, y_c, r)$	A circle with center (x_c, y_c) and radius r

and the result of distance computation in this situation is the maximum possible distance. Thus, the distance of two trajectories could be either the minimum possible distance in the best-case-scenario, the maximum possible distance in the worst-case-scenario, or any value between them.

Generally, there is no agreed definition when two trajectories are similar. However, we can define bounds that determine when two trajectories are highly similar or equal. Two trajectories are highly similar if they represent the same trip. In other words, we consider two trajectories to be highly similar if they can be derived from a single trajectory that has been sampled at a higher rate. Our key task is to develop a distance measure that ensures the different trajectories which originate from a finer sampled trajectory have a low distance. In summary, a distance measure for spatiotemporal trajectories has the following requirements:

R1 Both temporal and spatial dimensions must be considered in the distance computation.
R2 The distance must be computed using all the sampled points in the longest common time interval (see Definition 2) of two trajectories.
R3 It must be tolerant to different sampling rates and asynchronous sampling, i.e. the minimum distance between two trajectories must be low, when they are derived from a single trajectory.

DEFINITION 2. The *longest common time interval* of two trajectories T^1 and T^2 is the time interval $[t_{min}, t_{max}]$ in which:

$$t_{min} = max(t_{p^1_1}, t_{p^2_1}),\ t_{max} = min(t_{p^1_{n_1}}, t_{p^2_{n_2}})$$

In other words, we compare the similarity of the two trajectories over their common time period (an interval of time over which measurements for both trajectories are available). i.e. The intersection of the time periods of the trajectories.

Problem statement. Given two trajectories T^1 and T^2 and their longest common time interval $[t_{min}, t_{max}]$, our aim is to compute the distance interval of T^1 and T^2. We define function *TIDE* that computes the distance interval with the minimum and maximum distances of two trajectories as its bounds.

$$\begin{aligned} &TIDE(T^1, T^2, [t_{min}, t_{max}]) = \\ &[minDist_{T^1,T^2,[t_{min},t_{max}]}, maxDist_{T^1,T^2,[t_{min},t_{max}]}] \end{aligned} \tag{1}$$

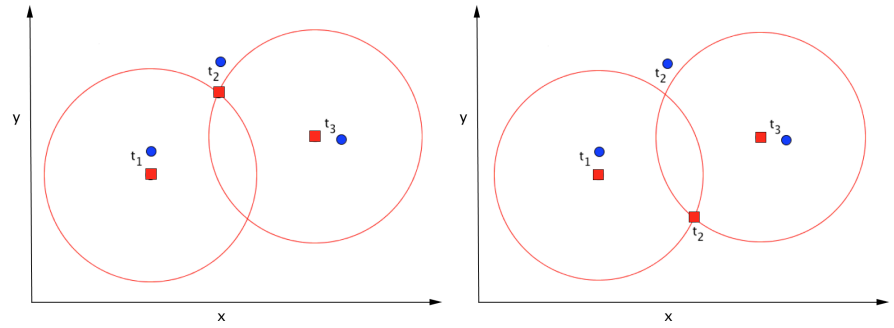


Fig. 5 Illustration of two trajectories T^1 and T^2 in space dimensions (time dimension is shown as label) (a) the minimum possible distance and (b) the maximum possible distance of the trajectories in Fig. 4(a) at time t_2

In Sect. 4, we will explain that how we compute the minimum and maximum distances of two trajectories. However, for the simplicity of representation and without loss of generality, we will have the assumption that the trajectories have the same starting and ending time and we will omit the time interval constraint $[t_{min}, t_{max}]$.

In order to compare our method with existing techniques, we focus on using the minimum distance. Minimum distance is applicable when we want to match a query trajectory to dataset trajectories. For example, in a carpooling application, when we want to find an everyday trajectory that matches the query trajectory, we search for nearest neighbour trajectories according to the minimum distance. In the general case though, the maximum distance is also important. E.g., when computing the distance between trajectories (with different sampling rates and times) and assigning a degree of uncertainty to the distance value. In a forensic scenario, where a low-sampled GPS trajectory from a suspicious person is being analysed, one could assign a degree of uncertainty to the distance value to quantify the level of confidence that their trajectory is a match.

4 Trajectory Interval Distance Estimation

In this section, we will describe our proposed measure Trajectory Interval Distance Estimation (TIDE) in detail. First, we will describe the fundamental concepts of our approach including how we compute the minimum and maximum distances between two trajectories. Then we will state that the computation of the minimum and maximum distances is an computationally expensive convex optimisation problem and then we will introduce an approximation approach for this optimisation problem.

The first step to approximate the location of an object at a timestamp without location data is to find the possible area that object must be there at that time. Using Lemma 1, we can confine the location of an object for a non-sampled point to a bounding area.

Lemma 1: Given segment (p_i^1, p_{i+1}^1) of trajectory T^1 and the estimated maximum speed of $V_{\max}^1$, the location of the object (x_{np}, y_{np}) at time t_{np} $(t_{p_i^1} < t_{np} < t_{p_{i+1}^1})$ is bounded by the

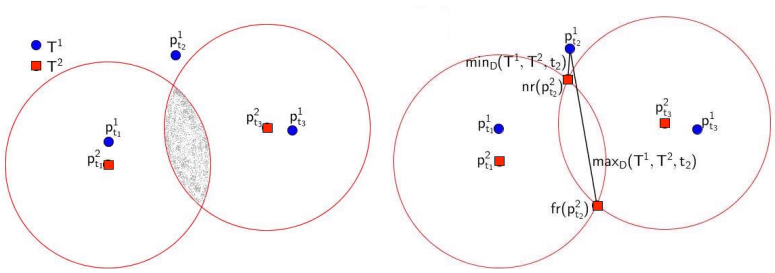


Fig. 6 (a) The bounding area for the non-sampled point of the trajectory T_2 at time t_2. (b) The minimum and maximum possible distance between T^1 and T^2 at time t_2

intersection of two circles, while p_i^1 and p_{i+1}^1 are the centres:

$$\begin{gathered}(x_{np},y_{np}) \in C(x_{p_i^1},y_{p_i^1},r_1) \wedge (x_{np},y_{np}) \in C(x_{p_{i+1}^1},y_{p_{i+1}^1},r_2) \\ \text{where:} \\ (x,y) \in C(x_c,y_c,r) \Longleftrightarrow (x-x_c)^2+(y-y_c)^2 \leq r^2 \\ \wedge\, r_1 = |t_{np}-t_{p_i^1}| \times V_{\max}^1 \wedge r_2 = |t_{np}-t_{p_{i+1}^1}| \times V_{\max}^1\end{gathered} \tag{2}$$

This area is called the bounding area (*BA*) for the non-sampled point at time t_{np}.

Proof Starting from p_i^1, the object can travel only by the distance $|t_{np}-t_{p_i^1}| \times V_{\max}^1$ until time t_{np}. Also, for being in the p_{i+1}^1 at time $t_{p_{i+1}^1}$, the object's distance from $(x_{p_{i+1}^1},y_{p_{i+1}^1})$ at time t_{np} should not be farther than $|t_{p_{i+1}^1}-t_{np}| \times V_{\max}^1$. Then, the combination of these two conditions restricts the spatial position of the moving object at time t_{np}.

EXAMPLE 2. Given the estimated maximum speed for trajectory T^2 in Fig. 4(a), the bounding area for the non-sampled point at timestamp t_2 is the intersection of the circles in Fig. 6(a).

When we compare two trajectories which have non-sampled points in comparison to each other, based on Lemma 1, we compute a bounding area for every non-sampled point in the trajectories. Thus, for every timestamp $t \in t_{T^1} \cup t_{T^2}$, there is either (1) a pair of spatial coordinates including p_i^1 and p_j^2 ($\exists p_i^1 \in T^1$ and $\exists p_j^2 \in T^2$ so that $t_{p_i^1} = t_{p_j^2} = t$) or (2) a spatial coordinate p_i^1 and a bounding area BA_i^2 ($\exists p_i^1 \in T^1$ while $t_{p_i^1} = t$ and $\nexists p_j^2 \in T^2$ so that $t_{p_j^2} = t$). Then, for every timestamp, we compute the distance of two coordinates or distance of a coordinate and a bounding area.

DEFINITION 3. Given a timestamp $t \in t_{T^1} \cup t_{T^2}$ and a distance function D, the minimum distance between trajectories T^1 and T^2 at time t is computed as follows:

a) If p_i^1 is a sampled point of T^1 and p_j^2 is a sampled point of T^2 at time t:

$$min_D(T^1,T^2,t) = D(p_i^1,p_j^2) \tag{3}$$

b) If p_i^1 is a sampled point of T^1 and BA_i^2 is a bounding area for a non-sampled point of T^2 at time t:

$$min_D(T^1,T^2,t) = \begin{cases} 0, & \text{if } p_i^1 \in BA_i^2 \quad (4a) \\ \min\limits_{x \in BA_i^2} D(p_i^1,x) & \text{if } p_i^1 \notin BA_i^2 \quad (4b) \end{cases}$$

DEFINITION 4. The minimum distance between BA_i^2 and p_i^1 occurs at $nr(p_i^2)$ which is the nearest point in BA_i^2 to p_i^1.

$$nr(p_i^2) = \underset{x \in BA_i^2}{\operatorname{argmin}} D(p_i^1, x) \tag{5}$$

DEFINITION 5. Given a timestamp $t \in t_{T^1} \cup t_{T^2}$ and a distance function D, the *maximum distance* between trajectories T^1 and T^2 at time t is computed as follows:

a) If p_i^1 is a sampled point of T^1 and p_j^2 is a sampled point of T^2 at time t:

$$max_D(T^1, T^2, t) = D(p_i^1, p_j^2) \tag{6}$$

b) If p_i^1 is a sampled point of T^1 and BA_i^2 is a bounding area for a non-sampled point of T^2 at time t:

$$max_D(T^1, T^2, t) = \max_{x \in BA_i^2} D(p_i^1, x) \tag{7}$$

DEFINITION 6. The maximum distance between BA_i^2 and p_i^1 occurs at $fr(p_i^2)$ which is the farthest point in BA_i^2 to p_i^1.

$$fr(p_i^2) = \underset{x \in BA_i^2}{\operatorname{argmax}} D(p_i^1, x) \tag{8}$$

The distance function D can be replaced by different distance functions. However, here, we use Euclidean distance which is the most common distance function for trajectory comparison applications.

DEFINITION 7. The minimum and maximum possible distances between trajectories T^1 and T^2 are computed as follows:

$$minDist_{T^1,T^2} = \sum_t min_D(T^1, T^2, t) \tag{9a}$$

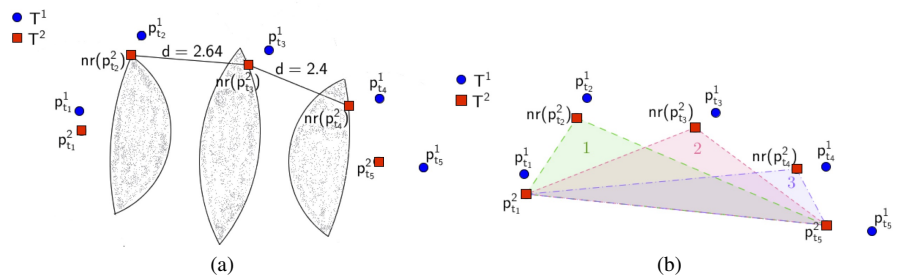
$$maxDist_{T^1,T^2} = \sum_t max_D(T^1, T^2, t) \tag{9b}$$

EXAMPLE 3. Fig. 6(b) shows the minimum and maximum distances at time t_2 between two trajectories T^1 and T^2 (for time t_1 and t_3, the minimum and maximum possible distance both are equal to the Euclidean distance between points).

However, computing $nr(p)$ $(fr(p))$ for consecutive non-sampled points independently, does not always result in a feasible combination of them. When we compute them for every timestamp independently, the resulting points do not meet the estimated maximum speed requirements, if it is not possible to travel between the independently computed $nr(p)$ $(fr(p))$ points with considering the estimated maximum speed during the time interval.

EXAMPLE 4. Fig. 7(a) illustrates two trajectories T^1 and T^2 with three non-sampled points in T^2. When for every non-sampled point, min_D is computed independently, the points $nr(p_{t_2}^2)$, $nr(p_{t_3}^2)$ and $nr(p_{t_4}^2)$ are the nearest possible locations in trajectory T^1 to their equivalent points in T^2. However, with the given estimated maximum speed of 2 per timestamp, it is not possible that the red object travels the distance between $nr(p_{t_2}^2)$ and $nr(p_{t_3}^2)$ from time t_2 to t_3 (or between $nr(p_{t_3}^2)$ and $nr(p_{t_4}^2)$ from time t_3 to t_4).

DEFINITION 8. If $(p_i^1, p_{i+1}^1, ..., p_{i+k-1}^1)$ is a sub-trajectory of T^1 and (p_j^2, p_{j+1}^2) is a segment of T^2 so that $t_{p_j^2} < t_{p_i^1} < t_{p_{i+1}^1} < ... < t_{p_{i+k-1}^1} < t_{p_{j+1}^2}$, when k approaches n, computing



(a) (b)

Fig. 7 (a) The impossible combination of the nearest possible points ($nr(p)$) with the estimated maximum speed of 2 for the trajectory T^2 (b) The resulting triangles when p^2_j and p^2_{j+1} are connected to every $nr(p)$ point

$minDist_{T^1,T^2}$ ($maxDist_{T^1,T^2}$) is the following optimisation problem:

$$\text{Minimize} \sum_{l=i}^{i+k-1} ||p^1_l - nr(p^2_l)||_2 \tag{10a}$$

Subject to

$$g_1(nr(p^2_l)) = ||nr(p^2_l) - p^2_l||_2 \leq |t_{nr(p^2_l)} - t_{p^2_l}| \times V^2_{\max} \tag{10b}$$

$$g_2(nr(p^2_l)) = ||nr(p^2_l) - p^2_{l+1}||_2 \leq |t_{nr(p^2_l)} - t_{p^2_{l+1}}| \times V^2_{\max} \tag{10c}$$

$$g_3(nr(p^2_l)) = ||nr(p^2_l) - nr(p^2_{l-1})||_2 \leq |t_{nr(p^2_l)} - t_{nr(p^2_{l-1})}| \times V^2_{\max} \tag{10d}$$

In other words, for consecutive non-sampled points, a $nr(p)$ ($fr(p)$) point should meet the following constraints:

1. A $nr(p)$ ($fr(p)$) point must be inside its corresponding bounding area according to Lemma 1 (10(b) and 10(c)).
2. Traveling from a $nr(p)$ ($fr(p)$) point to its immediate neighbour points must be feasible with its estimated maximum speed consideration (10(d)).

The objective function for this optimisation problem is to minimise (maximise) sum of the distances between corresponding points in the trajectories which is considered as the minimum (maximum) distance between trajectories T^1 and T^2.

Lemma 2: The optimisation problem in Definition 8 is a *convex* optimisation problem.

To prove the convexity of our optimisation problem, based on the definition of convex optimisation problem [25], we need to prove that all the functions 10(a) to (d) are convex. In the following, we prove the convexity of all the function in our optimisation problem.

Lp-norms for $p \geq 1$ are convex [25]. Also, 1) non-negative weighted sum, and 2) composition with an affine mapping preserve the convexity [25]. Thus, functions 10(a) to (c) are convex.

To prove convexity 10(d), we use the following Lemma.

Lemma 3: If $\{x_1, x_2, ..., x_k\}$ and $\{y_1, y_2, ..., y_k\}$ are two answer sets in which every two consecutive points satisfy the estimated maximum speed constraint in 10(d) as the following:

$$\forall\, i \in \{2, ..., k\}:$$
$$||x_i - x_{i-1}||_2 \leq (t_i - t_{i-1}) \times V_{\max},$$
$$||y_i - y_{i-1}||_2 \leq (t_i - t_{i-1}) \times V_{\max}$$

then the convex combination of them also satisfies the estimated maximum speed constraint:

$$||(\lambda x_i + (1-\lambda) y_i) - (\lambda x_{i-1} + (1-\lambda) y_{i-1})||_2 \leq (t_i - t_{i-1}) \times V_{\max}$$

Proof Proof is immediate:

$$\begin{aligned}
&||(\lambda x_i + (1-\lambda) y_i) - (\lambda x_{i-1} + (1-\lambda) y_{i-1})||_2 \\
&= ||\lambda (x_i - x_{i-1}) + (1-\lambda)(y_i - y_{i-1})||_2 \\
&\leq ||\lambda (x_i - x_{i-1})||_2 + ||(1-\lambda)(y_i - y_{i-1})||_2 && \textit{triangular inequality} \\
&= \lambda ||(x_i - x_{i-1})||_2 + (1-\lambda)||(y_i - y_{i-1})||_2 && \textit{homogeneity} \\
&\leq (t_i - t_{i-1}) \times V_{\max} && \textit{since } 0 \leq \lambda \leq 1
\end{aligned}$$

for all $x_i, y_i \in \mathbb{R}^n$ and $0 \leq \lambda \leq 1$.

All functions 10(a) to (d) are convex, thus, finding the minimum distance between two trajectories is a convex optimisation problem.

We study two approaches to solve this optimisation problem: one optimal (TIDE) and one approximate version (TIDE*). For the optimal solution, we use a solver to find the global minimum (maximum) as the least (greatest) possible distance between two trajectories. However, solving the optimisation problem to find the exact minimum distance between two trajectories incurs a high computation time (Table 4). Since the computation time is important for fast trajectory matching, we propose an approach that approximates a feasible combination of $nr(p)$ $(fr(p))$. A naïve approximation computes a $nr(p)$ $(fr(p))$ for a single non-sampled point and bounds the other non-sampled point relative to this point, so that the value of the objective function is greater in comparison to the situation where other non-sampled points were selected. However, the time complexity of this approach is $O(n^2 logn)$ in average and $O(n^3)$ for worst case, while our solution ensures a lower computation time.

Assume that trajectory T^2 has k non-sampled points between p_j^2 and p_{j+1}^2 in comparison to T^1 and the independently approximated $nr(p_l^2)$ $(fr(p_l^2))$, $l \in \{i, i+1, ..., i+k-1\}$, do not meet the maximum speed requirement. Our solution to approximate a feasible set of $nr(p_l^2)$ $(fr(p_l^2))$ for the non-sampled points is as follows:

1. For every independently-computed $nr(p_l^2)$, compute the area of the triangle when p_j^2, p_l^2 and p_{j+1}^2 are vertices.
2. Compare the triangles' areas ofs Step 1 and choose the $nr(p)$ that maximises (minimises) the area.
3. Re-calculate for every other non-sampled point, a $nr(p)$ by assuming that the selected $nr(p)$ is a sampled point.

The theoretical insight for this approximation is based on the Douglas-Peucker (DP) algorithm [40]. DP "takes a curve composed of line segments and finds a similar curve with fewer points" [40]. The idea of DP is to evaluate the points that are furthest for each line segment. Our insight is similar: we maximise the triangle area and the area of a triangle

is half of its base multiplied by its height. As Fig. 7 shows, the base is constant across all the triangles, i.e., the triangle area is linearly dependent on the height. Thus, the point that maximises the triangle area has the longest height, i.e., is furthest away. In the end, we use an approach that has a similar intuition to DP, where the red points represent a simplified curve of the curve of the blue points.

Algorithms 1 and 2 show the details of minimum distance computation using measure TIDE*. Algorithm 1 processes trajectories to find non-sampled points in them. Then it calls Algorithm 2 for approximation of non-sampled points.

Algorithm 1 TIDE*: Minimum Distance Computation

Input: $T^1, T^2, V^1_{\max}, V^2_{\max}, D$ ▷ D is the given distance function
Output: $minDist_{T^1,T^2}$ ▷ the minimum distance between T^1 and T^2

$PT^1 = \{\}, PT^2 = \{\}$
while $i < n_1$ **do**
  if $(is-non-sampled(t_{p_i^1}, t_{p_{i+1}^1}, T^1, T^2) == False)$ **then**
    $PT^1 = PT^1 + \{p_i^1\}$
  else
    $PT^1 = PT^1 + (p_i^1) + Approx(p_i^1, p_{i+1}^1, V^1_{\max}, T^2, D) + (p_{i+1}^1)$
  end if
  i = i+1
end while
$PT^1 = PT^1 + \{p_{n_1}^1\}$
while $j < n_2$ **do**
  if $(is-non-sampled(t_{p_j^2}, t_{p_{j+1}^2}, T^2, T^1) == False)$ **then**
    $PT^2 = PT^2 + \{p_j^2\}$
  else
    $PT^2 = PT^2 + (p_j^2) + Approx(p_j^2, p_{j+1}^2, V^2_{\max}, T^2, D) + (p_{j+1}^2)$
  end if
  j = j+1
end while
$PT^2 = PT^2 + \{p_{n_2}^2\}$
$minDist_{T^1,T^2} = D(PT^1, PT^2)$

THEOREM 1. The time-complexity for the approximation of k consecutive non-sampled points, when k approaches n, is $(O(nlogn))$ in average and $O(n^2)$ in worst-case.

Proof Based on Algorithm 2, if the combination of independently computed $nr(p)$ points is impossible (regarding the estimated maximum speed), when k approaches n, the time-complexity in average and worst case is as follows:

a) In worst case, a single call of $Approx$ takes $O(n)$ in addition to two recursive calls of size 0 and $n-1$. The recurrence relation for worst case is as follows:

$$T(n) = O(n) + T(0) + T(n-1) \tag{11}$$

To solve Equation 11 results in $T(n) = O(n^2)$ which is the worst case time-complexity.

b) In a balanced case, a single call of $Approx$ takes $O(n)$ plus two recursive calls of size $n/2$ and the recurrence relation is as follows:

$$T(n) = O(n) + 2T(n/2) \tag{12}$$

To solve Equation 12 results in $T(n) = O(nlogn)$. Thus, the time-complexity of $Approx$ in average is $O(nlogn)$.

Algorithm 2 Approx: Computation of non-sampled points

Input: p_i^1, p_{i+1}^1, $V_{\max}^1$, T^2, D
Output: A ▷ the approximated nearest points set for non-sampled points between p_i^1 and p_{i+1}^1

$S1 = sampled(t_{p_i^1}, t_{p_{i+1}^1}, T^2)$
$maxArea = 0$
$A = \{\}$
for (every $p_j^2 \in S1$) **do**
  $r_1 = (|t_{p_j^2} - t_{p_i^1}|) \times V_{\max}^1$
  $r_2 = (|t_{p_j^2} - t_{p_{i+1}^1}|) \times V_{\max}^1$
  $BA_j^1 = Intersection(C(x_{p_i^1}, y_{p_i^1}, r_1), C(x_{p_{i+1}^1}, y_{p_{i+1}^1}, r_2))$
  $nr_j^1 = \text{argmax}(p_j^2, x),\ x \in BA_j^1$
  $area_j^1 = triangleArea(nr_j^1, p_i^1, p_{i+1}^1)$
  if ($area_j^1 > maxArea$) **then**
    $maxArea = triangleArea_j^1$
    $baseline = nr_j^1$
  end if
  if ($j > 1$) **then**
    if $D(nr_j^1, nr_{j-1}^1) > (t_{nr_j^1} - t_{nr_{j-1}^1}) \times V_{\max}^1$ **then**
      $impossible = True$
    end if
  end if
end for
if ($impossible == False$) **then**
  $A = A + (nr_1^1, ...nr_{|S1|}^1)$
else
  $A = A + Approx(p_i^1, baseline, V_{\max}^1, T^2) + baseline + Approx(baseline, p_{i+1}^1, V_{\max}^1, T^2)$
end if

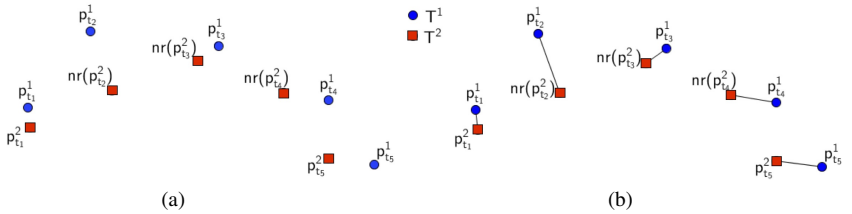


Fig. 8 (a) $nr(p_{t_3}^2)$ is selected as the baseline (because triangle 2 has the maximum area) and others are re-computed. (b) The minimum distance of trajectories T^1 and T^2 in Fig. 7(b) using our proposed measure TIDE

EXAMPLE 5. Take the trajectories of Example 4 in which T^2 has three non-sampled points in comparison to T^1. First, the areas of triangles are computed as described in step 1 (Fig. 7(b)). Comparing the areas of triangles 1, 2 and 3 the triangle 2 has the maximum area. Therefore, $nr(p_{t_3}^2)$ is selected as the baseline and $nr(p_{t_2}^2)$ and $nr(p_{t_4}^2)$ will be re-computed as illustrated in Fig. 8(a).

After approximating or exact computing of $nr(p)$ ($fr(p)$) points in two trajectories that are subject to comparison, we use Euclidean distance to compute the distance of corresponding points in trajectories (Fig. 8(b)).

Table 3 The datasets used in the experiments

Dataset	Number of trajectories	$(Avg. \pm Std.)$ of sampled points in trajectories
Truck	1000	(411 ± 222)
Cabspotting	32000	(257 ± 180)

5 Experiments

In this section, we present an empirical study on our measure using two real datasets. We verify the effectiveness and efficiency of similarity comparison using the following measures: TIDE, TIDE*, DISSIM, LCSS, time-constrained DTW [22], Discrete Fréchet distance [34] (DFréchet), Continuous Fréchet distance [33] (CFréchet), EDwP, MMAD [36] and STLIP [37]. TIDE and TIDE* represent the exact and approximated version of our approach, respectively. For TIDE, we use IBM ILOG CPLEX Optimisation Studio.

Since we consider the time dimension to compute trajectory similarity, only trajectories that have a common time interval with the query trajectory are comparable to the query trajectory. As the original DTW is a spatial similarity measure, we use the time-constrained DTW to compare it to the spatiotemporal similarity measures. The time-constrained DTW has a warping window size, similar to the parameter δ in LCSS. in all the experiments, we assigned 5% of the trajectory length to both parameters. Also, we use linear interpolation (similar to [5]) to compute non-sampled points for LCSS and DTW. As a preliminary step, we use interpolation to improve DTW's performance for this task (results are omitted). Hence, we use DTW+interpolation as one of our baselines. Since both the continuous and the discrete Fréchet distance are also spatial similarity measures, we use again a warping window (set to 5% of the trajectory length) to make it time dependent.

Similar to [5], to compute non-sampled points in DTW, we use linear interpolation. For example, we have two trajectories $T^1 = \{(4,3,1),(14,8,6)\}$ and $T^2 = \{(2,1,1),(4,3,3),(2,3,6)\}$. T^1 has a non-sampled point at time 3, in comparison to T^2. To do linear interpolation, we compute the average speed in both x and y dimension for two consecutive sampled points in T^1 ($V_x = \frac{(14-4)}{(6-1)} = 2$ and $V_y = \frac{(8-3)}{(6-1)} = 1$). Then, we compute the distance that the object can travel in each dimension given those computed average speeds from time 1 to time 3 as $(4,2)$ (using the formula $(V_x \times (3-1), V_y \times (3-1))$). Adding those distances to the location of the object at time 1, gives the location of the object at time 3 $((4+4,3+2))$. We add the approximated location of the object at time 3, to the trajectory T^1. Then, as a result of interpolation, we have T^1 as $T^1 = \{(4,3,1),(8,5,3),(14,8,6)\}$ which does not have non-sampled points in comparison to T^2.

Datasets: We use two traffic datasets, Athens truck dataset and Cabspotting [2]. Athens truck dataset contains trajectories of 50 trucks during 33 days in Athens. Cabspotting contains one-month trajectories of 536 taxis in San Francisco and has about 15 millions of points. Like [17], we partition the trajectories of these datasets when the moving objects are stationary for more than 15 minutes. Table 3 illustrates a summary of the datasets including the number of sampled points in trajectories and the average and the standard deviation of number of sampled points in trajectories.

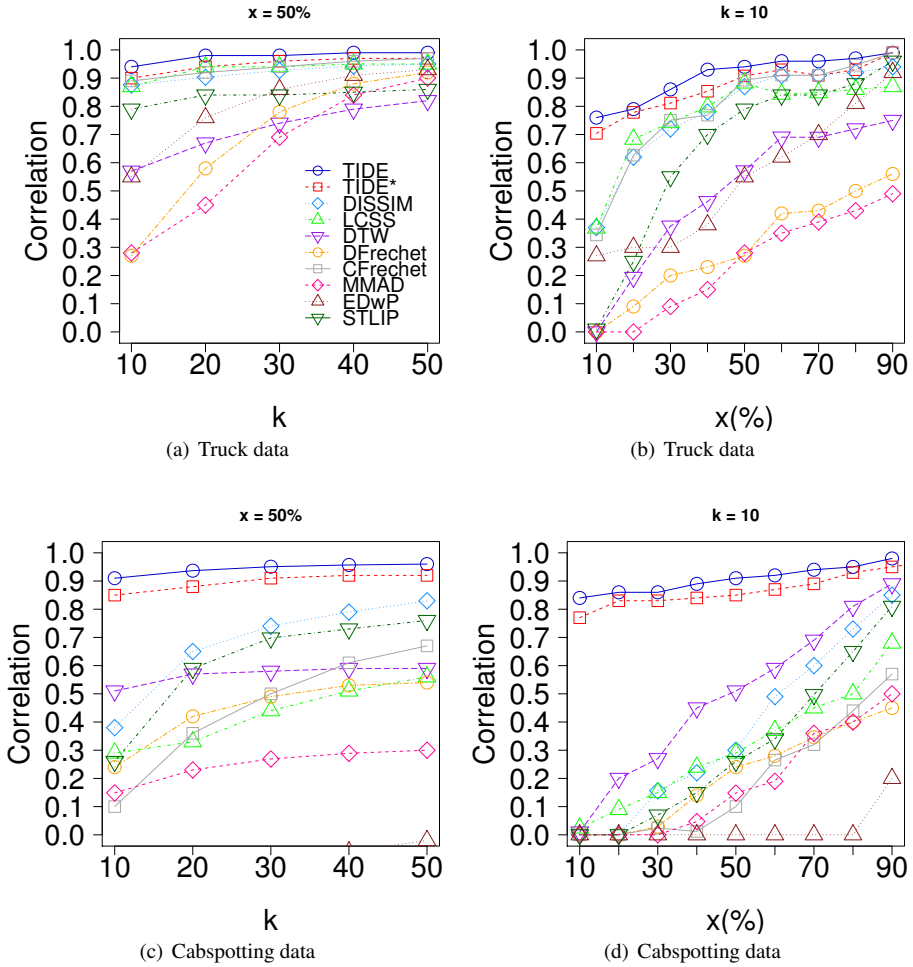
Comparison criteria: We use a similar approach to [17] for the effectiveness assessment of the similarity measures. As discussed before, different sampling rates and asynchronous sampling is a key problem when we consider the time dimension in similarity comparison of trajectories. In this paper, *we want to assess the robustness of the measures to different*

sampling rates and asynchronous sampling in comparing trajectories. for every measure, the ground truth is built by computing k nearest neighbours (k-NN) for a given query trajectory from a given set of original reference trajectories, DB. In each set of experiments, we build a new version of the original reference trajectories, for example lower-sampled trajectories, and extract a new set of k nearest neighbours (k-NN') for the same query trajectory from the new dataset. Then, we compute Spearman's rank correlation between k-NN and k-NN'. An ideal correlation between these two sets is 1. It means that for a given query trajectory, the ideal is to extract the same nearest neighbours from the lower-sampled version of reference trajectories, as the ones from the original reference trajectories. For example, take T_1 to T_{10} as reference trajectories and Q as a given query trajectory. Using a given measure 4-NN set for Q is $< T_2, T_4, T_9, T_7 >$ as the most similar trajectories with the similarity rank of $< 3, 1, 2, 4 >$. It means that T_4 has the rank 1 and it is the most similar trajectory to Q and T_9 is the second most similar trajectory to Q and so on. Similarity rank is based on the similarity (distance) value, calculated by the measure (for example, similarity values for those four trajectories might be as $< 0.6, 0.95, 0.8, 0.4 >$). Also, we generate a lower sampled version of T_1 to T_{10} (for example, by choosing 50% of their sampled points randomly) and build trajectories T'_1 to T'_{10}. Then we extract 4-NN' for the query trajectory Q from T'_1 to T'_{10}. The ideal situation is to extract the same set of similar trajectories with the same ranking so that 4-NN and 4-NN' have the rank correlation of 1. However, if a measure extracts the same similar trajectories in a different ranking such as $< 2, 1, 3, 4 >$, then Spearman's rank correlation between $< 3, 1, 2, 4 >$ and $< 2, 1, 3, 4 >$ is 0.8. Similar to [17], if the 4-NN and 4-NN' sets do not overlap completely, we compute the rank correlation for their union set.

Platform: All measures are implemented in C++ and run on a computer with an Intel Core i5 CPU (2.7 GHz) and 8 GB memory.

5.1 Accuracy for different sampling rates

In this experiment, we wish to verify the effectiveness of similarity measures when they compare trajectories with different sampling rates. We build two datasets DB_1 and DB_2. Trajectories in DB_1 are the same as trajectories in DB ($T'_i = T_i$, $T_i \in DB$ and $T'_i \in DB_1$), while trajectories in DB_2 are a lower-sampled version of trajectories in DB. To generate a lower-sampled version of a trajectory $T_i \in DB$, we randomly select $x\%$ of its points and add them as the new trajectory T''_i to DB_2. In this way, trajectories in DB_2 have a lower sampling rate in comparison to those in DB_1. As discussed before, we select a random trajectory from DB_1 as the query trajectory. Then, we extract k-NN for the query trajectory from DB and k-NN' from DB_2 and we compare Spearman's rank correlation between k-NN and k-NN'. Fig. 9(a) and Fig. 9(c) show the average Spearman's rank correlation results for the Truck and Cabspotting datasets when we increase k from 10 to 50, while x is 50%. Also Fig. 9(b) and 9(d) show the results when k is 10 and we increase x from 10% to 90%. The results of this experiment for the Cabspotting data illustrates that TIDE and TIDE* are more accurate than the other measures. Also, TIDE* is almost as accurate as TIDE which means the approximation does not have an adverse effect on the effectiveness of the approximated approach in this experiment. However, comparing the accuracy of other measures including DTW and Fréchet for two datasets, they have a lower accuracy for the Cabspotting dataset in comparison to Truck dataset. The reason is that the number of trajectories that have finite distance (or non-zero similarity) is higher for Cabspotting dataset which makes it harder to have the same ranking for similar trajectories.



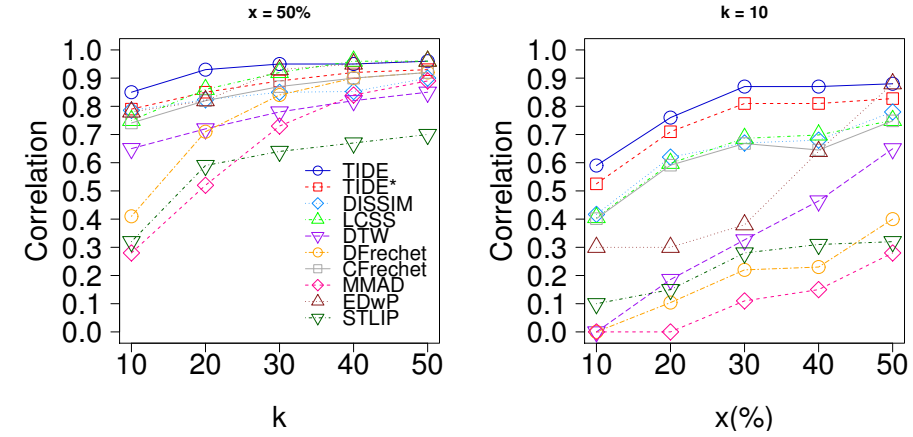
(a) Truck data (b) Truck data

(c) Cabspotting data (d) Cabspotting data

Fig. 9 The average Spearman's rank correlation for different sampling rates

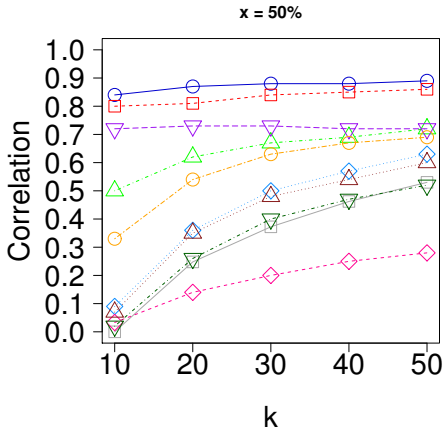
5.2 Accuracy for asynchronous sampling

In this experiment, we verify the effectiveness of the measures in comparing trajectories that are asynchronously sampled. We build two datasets DB_1 and DB_2. For every trajectory $T_i \in DB$, we randomly choose $x\%$ of its sampled points and insert it in DB_1 as trajectory T_i', then we add the rest of the points $(\{T_i\} - \{T_i'\})$ to DB_2 as T_i''. In this way, trajectories of DB_1 and DB_2 are sampled asynchronously. Similar to Sect. 5.1, we select a random trajectory from DB_1 as the query trajectory. Then, we extract k-NN for the query trajectory from DB and k-NN' from DB_2 and we compare Spearman's rank correlation between k-NN and k-NN'. Fig. 10 illustrates the results of this experiment for the truck and Cabspotting datasets. The accuracy of TIDE and TIDE* are higher than the other measures for both datasets. Comparing the results in Fig. 10(c) and 10(d) with Fig. 9(c) and 9(d) shows that the accuracy of all measures decreases for asynchronous sampling. The reason is that in

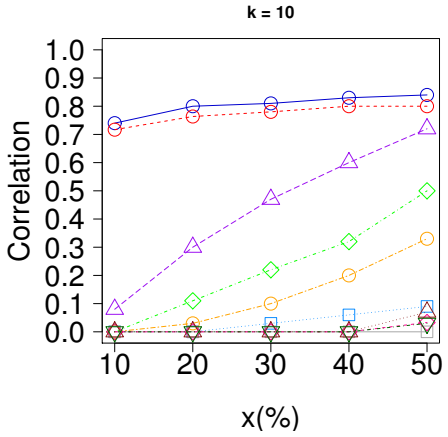


(a) Truck data

(b) Truck data



(c) Cabspotting data



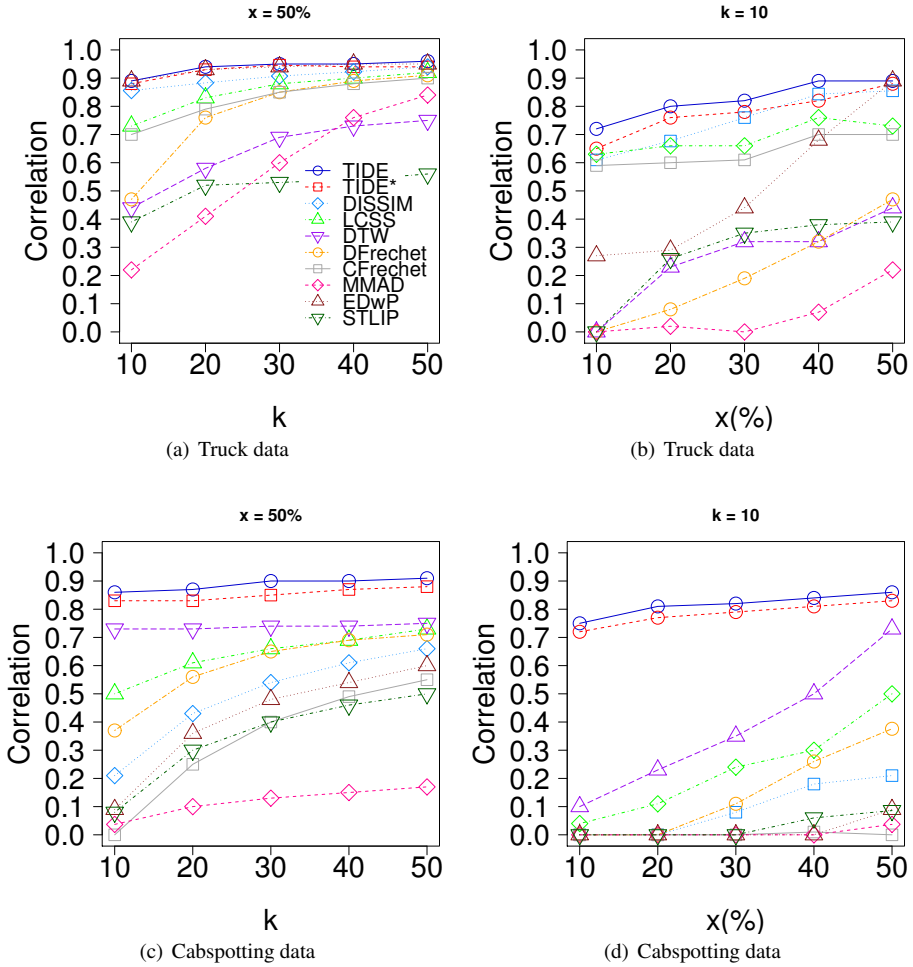
(d) Cabspotting data

Fig. 10 The average Spearman's rank correlation for asynchronous sampling

this experiment, trajectories do not have any sampled points in common which increases the uncertainty of the distance computation. Also, the accuracy of DISSIM is much lower in comparison to the previous experiment for Cabspotting data. This is because it does not warp and only uses interpolation to compute non-sampled points.

5.3 Accuracy for different sampling rates and asynchronous sampling

We examined the measures for trajectories that have different sampling rates (Sect. 5.1) and are sampled asynchronously (Sect. 5.2). As discussed before, trajectories that have been sampled using different devices, might have different sampling rates and times. In this experiment, we compare the effectiveness of the measures when they have both different sampling rates and times. For a trajectory $T_i \in DB$, we randomly choose $x\%$ of the points as T_i' to add



(a) Truck data (b) Truck data

(c) Cabspotting data (d) Cabspotting data

Fig. 11 The average Spearman's rank correlation for different sampling rates and asynchronous sampling

it to DB_1. To build T_i'', we choose $(100-x)\%$ of the sampled points from T_i (in the previous experiment $T_i'' = \{T_i\} - \{T_i'\}$, while here, the points in T_i'' are randomly selected from T_i). In this way, trajectories T_i' and T_i'' have some sampled points in common. Similar to Sect. 5.1 and 5.2, we select a random trajectory from DB_1 as the query trajectory. Then, we extract k-NN for the query trajectory from DB and k-NN' from DB_2 and we compare Spearman's rank correlation between k-NN and k-NN'. As shown in Fig. 11, similar to the results of different sampling rates in Sect. 5.1 (Fig. 9), TIDE and TIDE* outperform other measures. However, despite the similarities of DTW and Fréchet distances, DTW has a higher accuracy in comparison to Fréchet distances for Cabspotting data. The reason is likely because in DTW, the distance between trajectories is computed as the sum of the distances between the points, while in Fréchet, the distance is computed as the infimum over the distances between points. Also, in most of the experiments, MMAD has the lowest accuracy. The reason

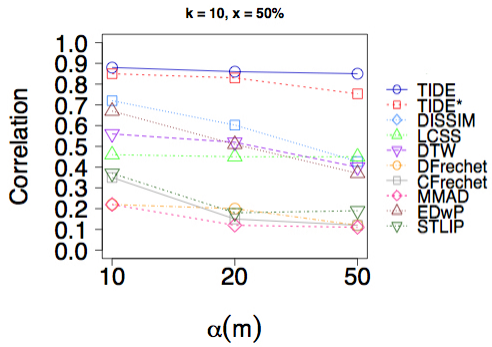


Fig. 12 The average Spearman's rank correlation in different sampling rates and asynchronous sampling for noisy Truck data

is that, it is a REMO pattern similarity measure and when we remove some of the sampled points, the motion pattern of a trajectory changes.

5.4 Accuracy for noisy data with different sampling rates and asynchronous sampling

GPS receivers generally have a level of noise in their provided data [23]. First, we generate DB_1 and DB_2 in the same way as the previous experiment. Then, we add α meters (10, 20 and 50 meters) of noise to the sampled points (p_j^i) of every trajectory T_i' in DB_1 (we apply the same procedure to every trajectory (Truck data) T_i'' in DB_2) so that:

1. The distance between the noisy point $\phi(p_j^i, \alpha)$ and original point p_j^i is α meters ($||\phi(p_j^i, \alpha) - p_j^i||_2 = \alpha$).
2. The angle that points $\phi(p_j^i, \alpha)$ and p_j^i make with a fixed point C is between 0 and 180 ($0 < \measuredangle\phi(p_j^i, \alpha)\ p_j^i\ C \leq 180$).

Comparing Fig. 12 with Fig. 11(a) when k is 10 shows that having 10 meters of noise does not have an important effect on the accuracy of all the measures. However, when we increase the amount of noise to 50 meters, because of the higher precision in non-sampled points computation, TIDE has a higher accuracy than TIDE*. Also, adding noise has a greater impact on DISSIM and EDwP measures. The reason is these measures use interpolation and projection to approximate the non-sampled points and then compute the area between two trajectories as the distance of them and in both steps, noise can affect the accuracy.

5.5 Efficiency of the similarity comparison

In this experiment, we wish to ascertain the time efficiency of our proposed measure. For this purpose, we measure the computation time of a similarity comparison between a given query trajectory against trajectories of the truck dataset to find the most similar trajectories to it. We randomly choose trajectory $T_i \in DB$ and we measure the time that we need to compare it to all the trajectories $T_j \in DB$. Table 4 illustrates the average computation time for the measures. As shown in Table 4, TIDE, LCSS and DTW have tremendously higher

computation time than TIDE* and DISSIM. The reason is that original LCSS and DTW have quadratic time complexity. Also, TIDE is computing the exact minimum distance between two trajectories which is an optimisation problem. In addition, the results show that the computation time of TIDE* is comparable to the computation time of DISSIM, while its accuracy is significantly higher than DISSIM.

If we use faster versions of DTW and LCSS, their computation time will decrease. However, in this experiment, our purpose is to show that providing a highly accurate similarity comparison using TIDE* does not incur a high computation time and its efficiency is comparable to DISSIM which has linear time complexity.

Table 4 The average computation time (milliseconds) of the measures for trajectory retrieval

Approach	TIDE	TIDE*	DISSIM	LCSS	DTW	Fréchet	MMAD	EDWP	STLIP
Time (milliseconds)	73000	200	30	65000	64000	53000	53000	58000	6700

5.6 Other Experiments

In the appendix, we provide some results on effectiveness of the various measures, using a different metric - NDCG. This metric is popular in the information retrieval community for comparing ranked lists. We also include an experiment assessing variation in performance of TIDE as the maximum speed is varied by up to 25%.

6 Discussion

Our focus is on *similarity comparison of trajectories* when we consider *both spatial and temporal dimensions*. There are the following considerations regarding our proposed method:

C1 Given a query trajectory and a set of reference trajectories, we propose a similarity measure that enables us to find k most similar trajectories to the query trajectory. The focus is not on finding the most likely reference trajectory taken by a moving object.
C2 Trajectories may belong to different types of moving objects including those that are not moving along road network such as animals.
C3 Trajectories that are sampled using different devices may be sampled asynchronously with different sampling rates. Taking into account both spatial and temporal dimensions, we need to handle *trajectories that are sampled asynchronously or at variable rates*.
C4 Our proposed measure uses the maximum speed of an object to handle different sampling rates and asynchronous sampling. If the maximum speed is not given, we can estimate the maximum speed using the sampled points of the trajectory (see Section 3).

As pointed out in C1 and C2, we do not focus on (i) assigning a probability to each of k most similar trajectories to a query trajectory; (ii) finding shortest path and easiest to navigate path for a given trajectory; (iii) computing transition probabilities at intersections. However, our method can be combined with complementary methods to address such problems. For example, the approach in [28] handles spatiotemporal range queries in road

network and assigns a probability to each possible path. As another example, in [29], a history based route inference system is proposed to infer possible routes for trajectories in road network with low-sampling rates.

Regarding C3: The lack of reliable GPS measurements can often happen due to different reasons such as travelling in tunnels and subways, next to skyscrapers, and under foliage. This is exacerbated by low sampling rates resulting from energy conservation of battery-dependent devices. People use different devices and applications to record their location data, which results in trajectories that have different numbers of sampled GPS points. In this paper, we use two datasets: the Truck dataset and Cabspotting dataset. The sampling period for Truck dataset trajectories is 30 seconds and for Cabspotting is 60 seconds. In experiments, we down-sampled trajectories of both datasets by different rates to simulate trajectories that have low and different sampling rate and/or are asynchronously sampled. For example, in experiments, for every trajectory in the dataset, we randomly sample 50% of its points and generate a new dataset using new trajectories. As a result of down-sampling, the average sampling period for the new Truck dataset is 70 ± 10 and for new Cabspotting is 122 ± 16. We showed that our similarity measure outperforms other measures up to 7 times. Its accuracy is significantly higher than other measures for smaller numbers of sampled points in big datasets.

Regarding the maximum speed computation (C4): There are of course theoretical limits for any approach that estimates the maximum speed of an object. In general, it is not possible to compute the exact maximum speed without knowing initial and final speeds at the points of measurement, and the maximum acceleration and deceleration of an object. However, our experiments show that our approach is very effective in practice. Furthermore, we experimentally investigated the impact of increasing the estimated maximum speed by 25 percent on the accuracy (see the Appendix). The results show that there is no considerable impact.

7 Conclusion

In this paper, we introduced the distance interval concept for spatiotemporal trajectories to address the challenge of different sampling frequencies and asynchronous sampling in trajectory comparison. We illustrated that different sampling rates and times result in uncertainty in distance computation of trajectories. To tackle this uncertainty, we used the estimated maximum speed of objects to define a distance interval rather than a single distance value. Since the computation of the minimum and maximum distances between trajectories is an optimisation problem that incurs significant processing times, we proposed a highly efficient approximation method to overcome this problem. The experimental results on two real datasets demonstrate the effectiveness of both the exact and the approximation method in comparison to important existing measures. The efficiency study shows that the computation time of our approximation method, TIDE*, is very low and is close to the computation time of DISSIM, which has a linear time complexity.

References

1. Piorkowski M, Sarafijanovic-Djukic N, Grossglauser M (2009) A parsimonious model of mobile partitioned networks with clustering. In: First international communication systems and networks and workshops, IEEE, pp. 1-10.

2. Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun G, Huang Y (2010) T-drive: driving directions based on taxi trajectories. In: Proceedings of 18th international conference on advances in geographic information systems, ACM, pp. 99-108.
3. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of the international conference on management of data (ACM SIGMOD), ACM, pp. 593-604.
4. Piciarelli C, Foresti GL (2006) On-line trajectory clustering for anomalous events detection. In: Pattern recognition letters, 27(15):1835-42.
5. Frentzos E, Gratsias K, Theodoridis Y (2007) Index-based most similar trajectory search. In: Proceedings of the 23rd international conference on data engineering (ICDE), IEEE, pp. 816-825
6. Mamoulis N, Cao H, Kollios G, Hadjieleftheriou M, Tao Y, Cheung DW (2004) Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the 10th international conference on knowledge discovery and data mining (ACM SIGKDD), ACM, pp. 236-245
7. Laube P, Imfeld S (2002) Analyzing relative motion within groups of trackable moving point objects. In: International conference on geographic information science, Springer, Heidelberg, Berlin, pp. 132-144
8. Lin B, Su J (2005) Shapes based trajectory queries for moving objects. In: Proceedings of the 13th annual ACM international workshop on geographic information systems, ACM, pp. 21-30
9. Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. In: Proceedings of the 18th international conference on data engineering (ICDE), IEEE, pp. 673-684
10. Laube P, van Kreveld M, Imfeld S (2005) Finding REMO-detecting relative motion patterns in geospatial lifelines. In: Developments in spatial data handling, Springer, Heidelberg, Berlin, pp. 201-215
11. Su H, Zheng K, Wang H, Huang J, Zhou X (2013) Calibrating trajectory data for similarity-based analysis. In: Proceedings of the ACM international conference on management of data (ACM SIGMOD), ACM, pp. 833-844
12. Faloutsos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. ACM
13. Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. In: KDD workshop, Vol. 10, No. 16, pp. 359-370
14. Vlachos M, Hadjieleftheriou M, Gunopulos D, Keogh E (2003) Indexing multi-dimensional time-series with support for multiple distance measures. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (ACM SIGKDD), ACM, pp. 216-225
15. Chen L, Ng R (2004) On the marriage of lp-norms and edit distance. In: Proceedings of the 30th international conference on very large data bases-Volume 30, VLDB Endowment, pp. 792-803
16. Chen L, zsu MT, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM international conference on management of data (ACM SIGMOD), ACM, pp. 491-502
17. Ranu S, Deepak P, Telang AD, Deshpande P, Raghavan S (2015) Indexing and matching trajectories under inconsistent sampling rates. In: Proceeding of IEEE 31st international conference on data engineering (ICDE), IEEE, pp. 999-1010
18. Agrawal R, Faloutsos C, Swami A (1993) Efficient similarity search in sequence databases. In: International conference on foundations of data organization and algorithms, Springer, Heidelberg, Berlin, pp. 69-84
19. Esling P, Agon C (2012) Time-series data mining. In: ACM Computing Surveys (CSUR), 45(1):12.
20. Keogh EJ, Pazzani MJ (1998) An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Kdd, Vol. 98, No. 1, pp. 239-243
21. Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E (2008) Querying and mining of time series data: experimental comparison of representations and distance measures. In: Proceedings of the VLDB Endowment, 1(2):1542-52.
22. Keogh E, Ratanamahatana CA (2005) Exact indexing of dynamic time warping. In: Knowledge and information systems, 7(3):358-86.
23. Biagioni J, Eriksson J (2012) Map inference in the face of noise and disparity. In: Proceedings of the 20th international conference on advances in geographic information systems, ACM, pp. 79-88
24. Shi Y, Hou YT (2009) Optimal base station placement in wireless sensor networks. In: ACM Transactions on Sensor Networks (TOSN), 5(4):32.
25. Boyd S, Vandenberghe L (2004) Convex optimization. In: Cambridge university press
26. Meratnia N, Rolf A (2004) Spatiotemporal compression techniques for moving point objects. In: International conference on extending database technology, Springer, Heidelberg, Berlin, pp. 765-782
27. Ramamohanarao K, Xie H, Kulik L, Karunasekera S, Tanin E, Zhang R, Khunayn EB (2017) Smarts: scalable microscopic adaptive road traffic simulator. In: ACM transactions on intelligent systems and technology (TIST), 8(2):26.
28. Zheng K, Trajcevski G, Zhou X, Scheuermann P (2011) Probabilistic range queries for uncertain trajectories on road networks. In: Proceedings of the 14th international conference on extending database technology, ACM, pp. 283-294

29. Zheng K, Zheng Y, Xie X, Zhou X (2012) Reducing uncertainty of low-sampling-rate trajectories. In: IEEE 28th international conference on data engineering (ICDE), IEEE, pp. 1144-1155
30. Paparrizos J, Gravano L (2015) k-shape: Efficient and accurate clustering of time series. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, ACM, pp. 1855-1870
31. Quicksort. https://en.wikipedia.org/wiki/Quicksort. Accessed 20 Jun 2018.
32. Kuijpers B, Moelans B, Othman W, Vaisman A (2009) Analyzing trajectories using uncertainty and background information. In: International symposium on spatial and temporal databases, Springer, Heidelberg, Berlin, pp. 135-152
33. Alt H, Godau M (1995) Computing the Fréchet distance between two polygonal curves. In: International journal of computational geometry and applications, 5(01n02):75-91.
34. Eiter T, Mannila H (1994) Computing discrete Fréchet distance. In: Tech. Report CD-TR 94/64, information systems department, technical university of Vienna
35. Tang B, Yiu ML, Mouratidis K, Wang K (2017) Efficient motif discovery in spatial trajectories using discrete fréchet distance. In: EDBT
36. Trajcevski G, Ding H, Scheuermann P, Tamassia R, Vaccaro D (2007) Dynamics-aware similarity of moving objects trajectories. In: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, ACM, p. 11
37. Pelekis N, Kopanakis I, Marketos G, Ntoutsi I, Andrienko G, Theodoridis Y (2007) Similarity search in trajectory databases. In: 14th international symposium on temporal representation and reasoning, IEEE, pp. 129-140
38. Spearman's rank correlation. `https://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient`. Accessed 20 Jun 2018.
39. Normalized Discounted Cumulative Gain. `https://en.wikipedia.org/wiki/Discounted_cumulative_gain`. Accessed 20 Jun 2018.
40. Douglas Peucker algorithm. `https://en.wikipedia.org/wiki/RamerDouglasPeucker_algorithm`. Accessed 20 Jun 2018.

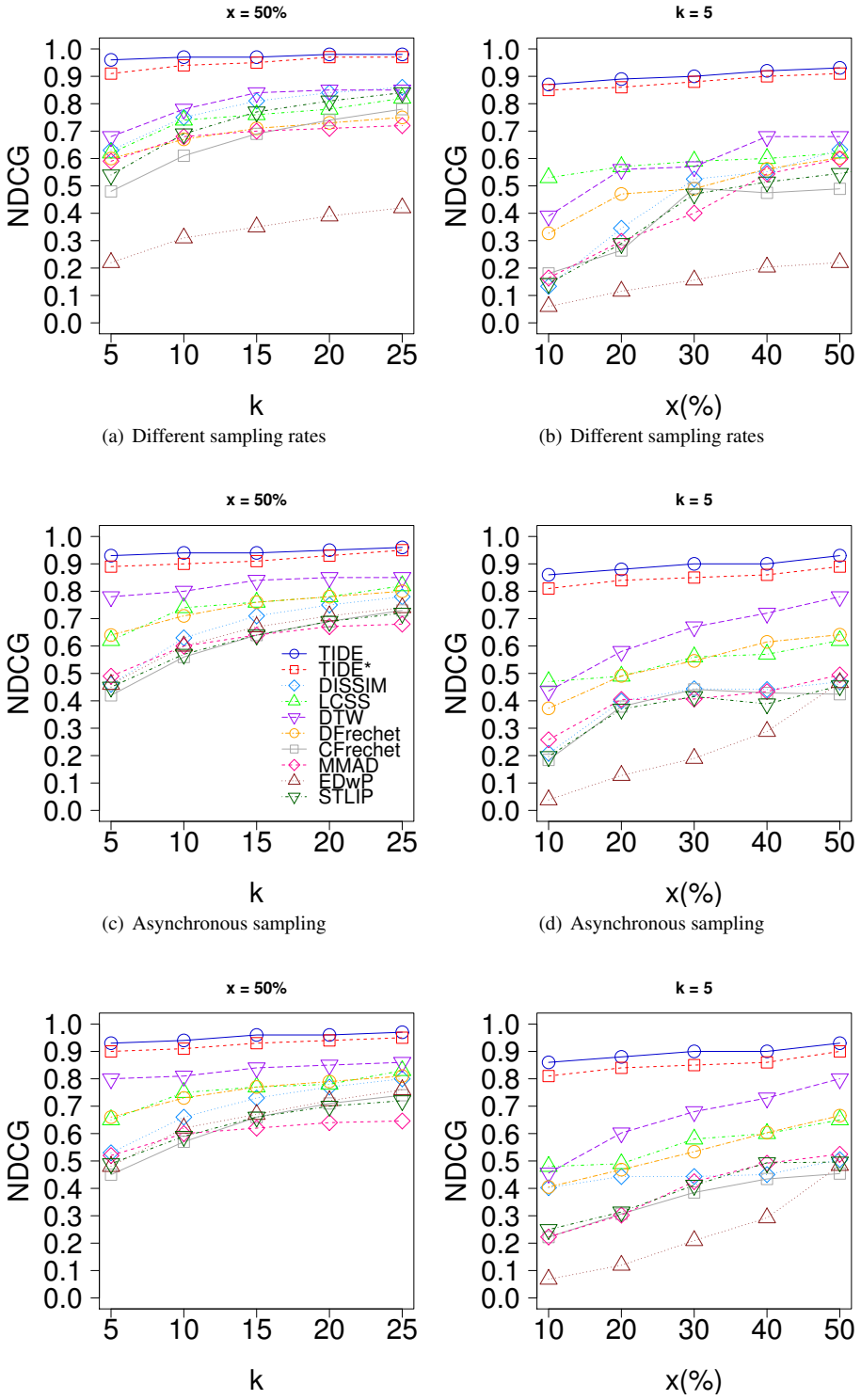
A Appendix

A.1 Accuracy using NDCG measure

In this section, we use Normalized Discounted Cumulative Gain [39] to examine the accuracy of different measures that we mentioned in Sect. 5 for Cabspotting dataset.

For every measure, similar to Sect. 5, the ground truth is built by computing k nearest neighbours (k-NN) set for a given query trajectory from a given set of original reference trajectories *DB*. The trajectories in k-NN are ordered from most similar to less similar trajectory. Then we give the relevance value of k to the first trajectory (the most similar trajectory) and relevance value of 1 to the last trajectory. We compute k-NN' for the same query trajectory from the dataset with lower sampling rate. We eliminate all the trajectories in set k-NN' - k-NN (k-NN' = k-NN' $\cap$ k-NN). Then, we assign a relevance to each of them based on their rank in k-NN'. For example, take T_1 to T_{10} as reference trajectories and Q as a given query trajectory. Using one of measures 4-NN set for Q is $< T_4, T_9, T_2, T_7 >$ as the most similar trajectories in order (it means that T_4 is the most similar trajectory to Q and T_9 is the second most similar trajectory to Q and so on.). We give a relevance value based on their similarity rank ($< 4, 3, 2, 1 >$). It means that relevance value for T_4 is 4 which means it has highest relevance to Q. Also, we generate a lower sampled version of T_1 to $T_1 0$ by choosing 50% of their sampled points randomly and build trajectories T_1' to $T_1' 0$. Then we extract 4-NN' for the query trajectory Q from T_1' to T_{10}'. The ideal situation is to extract the same set of similar trajectories with the same order so that 4-NN and 4-NN' have the NDCG of 1. However, if a measure extracts $< T_9, T_4, T_5, T_7 >$the same similar trajectories in order, we eliminate T_5 as it is not in k-NN. Then, the relevance values for the set $< T_4, T_9, T_2, T_7 >$ using k-NN' is $< 3, 4, 0, 1 >$ (T_4 has the relevance of 3 in k-NN' and T_9 has the relevance of 4, T_2 is not in k-NN' and T_7 has the relevance of 1. Indeed, the ideal relevance for the set $< T_4, T_9, T_2, T_7 >$ using that given measure is $< 4, 3, 2, 1 >$, however, for the lower sampled version of trajectories, it is $< 3, 4, 0, 1 >$. Then we compute DCG for $< 4, 3, 2, 1 >$ as the ideal DCG (IDCG) and for $< 3, 4, 0, 1 >$ as given DCG (GDCG). Then, we divide GDCG by IDCG.

Fig. 13 shows the results for cabspotting dataset. Similar to the results of Spearman's rank correlation, TIDE and TIDE* have higher accuracy in comparison to other measures. However, since NDCG does not penalize for "bad" trajectories in the results, we see better results in comparison to Spearman's rank correlation.



(a) Different sampling rates

(b) Different sampling rates

(c) Asynchronous sampling

(d) Asynchronous sampling

(e) Different sampling rates and asynchronous sampling

(f) Different sampling rates and asynchronous sampling

Fig. 13 NDCG for Cabspotting data

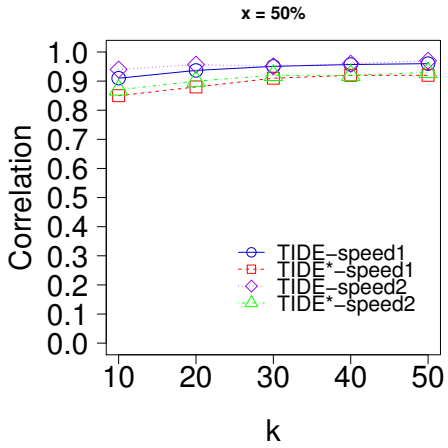


Fig. 14 The average Spearman's rank correlation with the original (speed1) and increased (speed2) estimated speed for Cabspotting data

A.2 The impact of estimated maximum speed

As discussed before, when we do not have information about speed limits of an object, we estimate the maximum speed of the object using sampled points of its trajectories (Sect. 3). In this experiment, we want to verify the impact of "estimating" the maximum speed. In other words, we may underestimate the maximum speed and we want to see the impact of increasing the estimated maximum speed on the accuracy. The ground truth is the same as the previous, however, we increase the estimated maximum speed for the lower-sampled version of the Cabspotting dataset. The outcome is that, there is not a considerable impact on the accuracy. As an example, in Fig. 14, we show the results for different sampling rates experiment in Fig. 9(c). TIDE-Speed1 and TIDE*-Speed1 show the results for the original speed estimation (Fig. 9(c)) and TIDE-Speed2 and TIDE*-Speed2 show the results for the increased speed (by 25 percent).