Discovering Private Trajectories Using Background Information[☆]

Emre Kaplan[a,1], Thomas B. Pedersen[a,1], Erkay Savaş[a,1], Yücel Saygın[a,1]

[a] *Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, Turkey*

Abstract

Trajectories are spatio-temporal traces of moving objects which contain valuable information to be harvested by spatio-temporal data mining techniques. Applications like city traffic planning, identification of evacuation routes, trend detection, and many more can benefit from trajectory mining. However, the trajectories of individuals often contain private and sensitive information, so anyone who possess trajectory data must take special care when disclosing this data. Removing identifiers from trajectories before the release is not effective against linkage type attacks, and rich sources of background information make it even worse. An alternative is to apply transformation techniques to map the given set of trajectories into another set where the distances are preserved. This way, the actual trajectories are not released, but the distance information can still be used for data mining techniques such as clustering. In this paper, we show that an unknown private trajectory can be re-constructed using the available background information together with the mutual distances released for data mining purposes. The background knowledge is in the form of known trajectories and extra information such as the speed limit. We provide analytical results which bound the number of the known trajectories needed to reconstruct private trajectories. Experiments performed on real trajectory data sets show that the number of known samples is surprisingly smaller than the actual theoretical bounds.

Key words: Privacy, Spatio-temporal data, trajectories, data mining

1. Introduction

The spatio-temporal traces of individuals can now be collected with GPS devices, GSM phones, RFID tag readers, and by many other similar means. Banks register time and location information of the financial transactions we

[☆]This work was partially funded by the Information Society Technologies Programme of the European Commission, Future and Emerging Technologies under IST-014915 GeoPKDD project.

[1]{emrekaplan@su.,pedersen@,erkays@,ysaygin@}sabanciuniv.edu

perform using our credit cards. A growing number of RFID tags are being used to give us access to, e.g., parking spaces or public transportation. Collected spatio-temporal data could be used in many ways such as traffic management, geo-marketing and sometimes for geo-spamming. From the point of view of data-analysis, the availability of all this information gives us the ability to find new and interesting patterns about how people move in the public space. On the other hand, collection of all these time and location pairs of individuals enables anyone, who observes the data, to reconstruct the movements (the trajectory) of others with a very high precision. There is a growing concern about this serious threat to privacy of individuals whose whereabouts are easily monitored and tracked. Legal and technical aspects of such threats were highlighted at a recent workshop on mobility, data mining, and privacy [17].

Considering its variety of applications, there is no doubt that the amount of spatio-temporal data being collected will increase drastically in the future, and so will the privacy concerns. In order to protect the privacy of individuals, the first thing to do is to remove personally identifying information from the released data sets. However, this has been shown not to preserve privacy against linkage type attacks even for ordinary data sets[19]. For the case of spatio-temporal data sets, availability of rich background information makes the privacy issues even more complicated[20]. A safer approach would be to perturb the trajectories or apply transformations which preserve important properties of the data such as mutual distances[12]. However, there may still be privacy risks in such transformations. In this paper we consider distance preserving data transformations on trajectories, and show that with background information such as a set of known trajectories and speed limits, an attacker can reconstruct individuals trajectories with very high precision. In particular, we consider the following scenario: A malicious person wishes to reconstruct the movements (the "target trajectory") of a specific individual. Besides a released set of mutual distances between a data set of trajectories, which contains the target trajectory, the attacker has some background information, such as the average speed or maximum speed of the trajectory, and some of the other trajectories in the data set. We propose a concrete algorithm which can reconstruct the target trajectory from this information.

Contributions of this work can be summarized as follows: 1) We demonstrate that trajectories can be reconstructed very precisely with very limited information using relatively simple methods. In particular we apply our method to two real-world data-sets. In one data-set, containing the trajectories of private cars in Milan, we can reconstruct an unknown trajectory with 500 sample points by knowing its distance to only 60 known trajectories. This is in sharp contrast to the 1001 known distances which would be needed to solve the corresponding system of equations to find the unknown trajectory. 2) We propose a method which can reconstruct trajectories from a very wide range of continuous properties (cf. Section 4); the method of known distances is only a special case. Our method is optimal in the sense that it will eventually find a candidate which exhausts all the information available about the unknown trajectory.

2. Related Work

Research efforts on trajectory mining have been boosted by a recent EU funded research project called "Geographic Privacy-aware Knowledge Discovery and Delivery" (GeoPKDD) [5]. As the title of the project implies, privacy is an important aspect of GeoPKDD. In the context of GeoPKDD, many techniques were proposed to mine useful patterns from trajectories. Some of the recent results are [6, 14] where in [6] the authors mine for temporal patterns of the form $a \rightarrow^t b$ meaning that t is the typical time to travel from location a to location b. Their algorithm needs to know what points of interests the trajectories pass through, and at which time intervals. Trajectories attracted other research groups too[10, 11]. In [10], authors give a clustering algorithm which considers sub-trajectories. The main observation is that sub-parts of trajectories may follow interesting common patterns, while the trajectories as a whole may be very different from each other. In [11] authors give a method for finding "hot-routes" in a given road network, which can help officials in traffic management.

Previous work on spatio-temporal data privacy include anonymization in location based services. Some of the recent work include [13, 3]. However, they do not deal with trajectory data. Techniques for trajectory anonymization were recently proposed in [1] and [16], but privacy risks after data release were not considered. In another recent work, privacy risks due to distance preserving data transformations were identified [21], however spatio-temporal data was not addressed. The privacy risks in trajectory data was addressed in [20] where authors point out how parts of a trajectory could be used as quasi-identifiers to discover the rest of the trajectory. In this work, authors assume that the trajectories are distributed vertically across multiple sites where sites are curious to learn the rest of the trajectory, and the authors propose methods to prevent that by suppressing parts of the trajectories before they are published.

In all the algorithms mentioned above for trajectory mining, different properties of the trajectories are needed. Some methods only need the mutual distances between trajectories, some need the exact trajectories, and others only need to know at what times the trajectories pass through certain areas of interest. In this paper, we show how, even very little, information is enough to recover the movement behavior of an individual. In particular we demonstrate how an unknown trajectory can be almost entirely reconstructed from its distance to a few fixed trajectories.

3. Privacy in Trajectory Data

As more and more data mining techniques aimed at trajectory data are invented, researchers are forced to ask themselves which kinds of violations of the privacy of individuals may occur. Defining privacy in trajectory data has proven to be a very complicated task. In this paper we do not intend to give any new definition of privacy,[*but limit our attention to the simple case where an outsider can identify a small area where an individual has been.*]

Consider a car insurance company, who gives discounts to clients who volunteer to install a GPS device in their cars, and submit all GPS data to the insurance company (to decease the burden of prof in case of an accident). Besides being valuable to the insurance company, this GPS data has a considerable value in other applications such as advertisement placement. The insurance company may sell "anonymized" versions of the dataset for profit, if they are convinced (and can convince their clients) that the anonymized data cannot be abused.

Though we do not intend to give a detailed study of possible abuses of trajectory data, it is clear that a dataset which enables an outsider (from the advertisement placement company) to identify a few points on a trajectory can be abused. [*Suppose for instance, the dataset reveals that a certain trajectory "stays" at location A during the night, then stays at location B from approximately 8am to approximately 10am and finally spends the rest of the day at location C. From this simple information it is not hard to guess that the trajectory belongs to a person living at A, who works at C, and that the person in question has visited location B for two hours before coming to work.*] Correlating this information with an address registry, the yellow pages, and possibly a small drive to locations A, B, and C, we may be able to identify the individual and learn that he is visiting the hospital at location B.

From the example above, we see that even relatively vague information about a trajectory is enough to reveal information which should be considered private. [*The attack presented in this paper is capable of approximating the movements of an individual, similar to the example above.*]

4. Trajectories and Continuous Properties

In their most general form trajectories are paths in space-time. In practice, however, trajectories are collected by moving objects with GPS devices, or other discrete sampling methods, and have to be stored in a format which is suitable for its intended use. There are many ways to represent and store a trajectory, but in this paper we focus on the intuitive and common approach of storing a trajectory as a polyline.

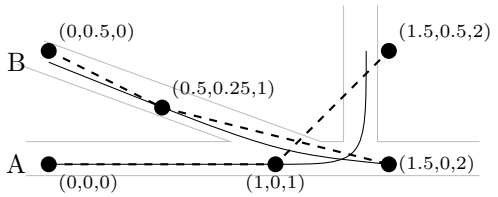


Figure 1: Two example trajectories with (x,y) and time coordinates.

A discrete trajectory is a polyline represented as a list of sample-points: $T = ((x_0, y_0, t_0), \ldots, (x_{n-1}, y_{n-1}, t_{n-1}))$. We write T_i to represent the ith sample-point (x_i, y_i, t_i). In most of this paper we think of a trajectory as a column-

vector in a large vector-space. We use calligraphic letters to refer to the vector representation of a trajectory. The vector representation of a trajectory T is: $\mathscr{T} = (x_0, y_0, t_0, \ldots, x_{n-1}, y_{n-1}, t_{n-1})^T \in \mathbb{R}^{3n}$. In this case $\mathscr{T}_i$ is the ith element of the vector (i.e. $\mathscr{T}_0 = x_0, \mathscr{T}_1 = y_0, \ldots, \mathscr{T}_{3n-1} = t_{n-1}$). [*Figure 1 shows two moving objects on a road map. The full lines are the actual movements of the two objects, and the dots are the sampled locations which makes the data set. Due to measurement inaccuracy some of the sample points are not on the actual paths of the moving objects. In this paper "trajectory" refers to the polyline connecting the sample points: the dashed lines in the figure. Algorithms used to remove measurement inaccuracy are out of the scope of this paper.*]

In this paper we assume that trajectories 1) are *aligned*[2] and 2) have constant sampling rate ($t_{i+1} - t_i = \Delta t$, for some constant Δt). Most of the distance measures defined below are most meaningful when trajectories satisfy these two conditions. Algorithms for ensuring these conditions can be found in [7]. In consequence we discard the time component and represent a trajectory as a list of (x, y) coordinates (or a vector in $\mathbb{R}^{2n}$). [*The two trajectories in Figure 1 are aligned; they are both sampled at the same times, and have a constant sample rate of one time unit. After discarding the time component from the trajectories we have:* $A = ((0,0),(1,0),(1.5,0.5))$*, and* $B = ((0,0.5),(0.5,0.25),(1.5,0))$*, or, in vector notation:* $\mathcal{A} = (0,0,1,0,1.5,0.5)$*, and* $\mathcal{B} = (0,0.5,0.5,0.25,1.5,0)$.]

A trajectory $\mathscr{T}$ can posses many properties which are of interest in different situations, such as maximum and average speed of a trajectory, closest distance to certain locations, duration of longest "stop", or percentage of time that $\mathscr{T}$ moves "on road". In this work we show how any property of $\mathscr{T}$ which can be expressed as a continuously differentiable function $f : \mathbb{R}^{2n} \to \mathbb{R}$ can be used to reconstruct $\mathscr{T}$. All the examples given above can be expressed as continuously differentiable properties of $\mathscr{T}$. In this paper we focus on known distances and known maximum and average speed. We will further explain these properties in the following subsections.

4.1. Known Distances

The first property of trajectories, which we consider, is the distance from an unknown trajectory $\mathscr{T}$ to a fixed trajectory, $\mathscr{T}'$. When using a continuously differentiable norm to compute the distance between $\mathscr{T}$ and $\mathscr{T}'$ we obtain a continuously differentiable property of $\mathscr{T}$; e.g. $\Delta_{\mathscr{T}'}(\mathscr{T}) = d(\mathscr{T}', \mathscr{T})$ is continuously differentiable.

A wide range of distance measures have been used for trajectories. Some commonly used measures of distance between two trajectories used in the literature [15] are:

[2]Two trajectories are aligned if they have the same sampling times and the same number of sample points.

Euclidean distance

$$\|\mathscr{T} - \mathscr{T}'\|_2 = \sqrt{\sum_{i=0}^{2n-1} |\mathscr{T}_i - \mathscr{T}_i'|^2}, \tag{1}$$

P-norm distance

$$\|\mathscr{T} - \mathscr{T}'\|_p = \left(\sum_{i=0}^{2n-1} |\mathscr{T}_i - \mathscr{T}_i'|^p\right)^{1/p}, \tag{2}$$

Average sample distance

$$d_2(T, T') = \frac{1}{n}\sum_{i=0}^{n-1} \|T_i - T_i'\|_2, \tag{3}$$

Average p-norm distance (More general form of average sample distance)

$$d_p(T, T') = \frac{1}{n}\sum_{i=0}^{n-1} \|T_i - T_i'\|_p. \tag{4}$$

Variance distance

$$d_v(T, T') = \frac{1}{n}\sum_{i=0}^{n-1} (\|T_i - T_i'\|_2 - d_2(T, T'))^2. \tag{5}$$

Area distance $d_A(T, T')$, which is the area of the region enclosed between the two trajectories[15].

With the exception of p-norm distance for odd p, all these distance measures are continuously differentiable.

[*Continuing the example in Figure 1 we compute:*

$$\begin{aligned} \|\mathcal{A} - \mathcal{B}\|_2 &= \sqrt{0^2 + 0.5^2 + 0.5^2 + 0.25^2 + 0^2 + 0.5^2} \\ &\approx 0.9 \\ d_2(A, B) &= (\|(0,0) - (0,0.5)\| + \|(1,0) - (0.5,0.25)\| \\ &\quad + \|(1.5,0.5) - (1.5,0)\|)/3 \\ &\approx 0.52 \end{aligned}$$

]

4.2. Trajectory Speed

Another property of trajectories, which is natural to consider, is the maximum or average speed at which the moving object is traveling. Since we only have discretized versions of the trajectories, with sample points taken at a fixed sample rate, we can only approximate the average and maximum speed:

$$\text{avgSpeed}(T) = \frac{1}{n-1}\sum_{i=0}^{n-1}\frac{\|T_i - T_{i+1}\|_2}{\Delta t}, \tag{6}$$

$$\text{maxSpeed}(T) = \max_i\left\{\frac{\|T_i - T_{i+1}\|_2}{\Delta t}\right\}, \tag{7}$$

where Δt is the known, constant sample rate (which we have discarded from the description of the trajectory itself). Note that the average/max speed in this case is approximated by the average/max speed of each *segment* of the discretized trajectory, where segment i is the line segment (T_i, T_{i+1}), or, when written in the vector notation: $((\mathscr{T}_{2i}, \mathscr{T}_{2i+1}), (\mathscr{T}_{2i+2}, \mathscr{T}_{2i+3}))$. [*The max speed of trajectory A from Figure 1 is* $\max\{\|(0,0)-(1,0)\|, \|(1,0)-(1.5,0.5)\|\} = 1$. *The average speed of A is approximately* $(1+0.7)/2 = 0.85$.]

The average speed is easily seen to be continuously differentiable. To compute the derivative of the maxSpeed, first note that the derivative of the maximum function can be approximated as:

$$\frac{\partial}{\partial x_i}\max\{x_0,\ldots,x_{n-1}\} = \begin{cases} 1 & \text{for } i \in \text{argmax}_i\{x_0,\ldots,x_{n-1}\} \\ 0 & \text{else,} \end{cases} \tag{8}$$

where $\text{argmax}_i\{x_0,\ldots,x_{n-1}\} = \{i_1\ldots,i_l\}$ is the set of indices such that x_{i_j} has the largest value of $\{x_0,\ldots,x_{n-1}\}$ (more than one element can have the maximum value).

When there is more than one largest argument to the max function, the partial derivatives with respect to those arguments are not well-defined (the right-derivatives are 1, while the left-derivatives are 0). However, in the following, we will use the convention that the partial derivatives of the largest arguments are 1 in those arguments.

Let S be the set of indices of the first sample points on the fastest segments of the trajectory: $S = \text{argmax}_i\{\|T_i - T_{i+1}\|_2/\Delta t\}$, and let $\mathcal{S}_t = \{2s + t | s \in S\}, t \in \{0,\ldots,3\}$ be the sets of the indices of the coordinates of the vector representation of the fastest segments ($\mathcal{S}_0$ is the set of x-coordinates on the first sample points, $\mathcal{S}_1$ is the set of y-coordinates on the first sample points, $\mathcal{S}_2$ is the set of x-coordinates on the second sample points, etc.). In the following we will use a generalization of Kronecker delta: $\delta_{i,\mathcal{S}}$, which is 1 if $i \in \mathcal{S}$, and 0

otherwise. The partial derivatives of the maximum speed is:

$$\begin{aligned}\frac{\partial}{\partial \mathscr{T}_i}\text{maxSpeed}(T) &= \frac{\partial}{\partial \mathscr{T}_i}\max_j\left\{\frac{\|T_j - T_{j+1}\|_2}{\Delta t}\right\}\\ &= \sum_{k=0}^{3}\delta_{i,\mathcal{S}_k}\frac{1}{\Delta t}\frac{\partial}{\partial \mathscr{T}_i}\|(\mathscr{T}_{i-k},\mathscr{T}_{i-k+1}) - (\mathscr{T}_{i-k+2},\mathscr{T}_{i-k+3})\|_2\\ &= \sum_{k=0}^{3}\delta_{i,\mathcal{S}_k}\frac{1}{\Delta t}\frac{\mathscr{T}_i - (-1)^{\delta_{k,\{0,1\}}}\mathscr{T}_{i+2} - (-1)^{\delta_{k,\{2,3\}}}\mathscr{T}_{i-2}}{2\|(\mathscr{T}_{i-k},\mathscr{T}_{i-k+1}) - (\mathscr{T}_{i-k+2},\mathscr{T}_{i-k+3})\|_2}.\end{aligned}$$

This partial derivative is not continuous. However, as we argue in Section 8, it is still suitable for the reconstruction of trajectories.

5. Reconstructing Trajectories

In this paper we consider how a malicious person can find an unknown trajectory, X, with as little information as possible. Any information we have about X may improve our ability to reconstruct X; a car does not drive in the ocean, and rarely travels at a speed of more than 200 km/h. The information which the malicious person has about a trajectory can be divided into two kinds: 1) data which has been released into the public domain by a data holder (in some anonymized format), and 2) background information which the malicious person already had about the trajectory. In this paper the only kind of released information we address are the mutual distances between trajectories. This data may be released in order for a third party to perform clustering on the trajectories. Speed limit is an example of background information of trajectories, since any it is well-known.

With a sufficient number of known properties of X, the trajectory can be fully reconstructed. If, for example, $2n$ linear properties of X are known, we have a system of $2n$ linear equations. Solving these $2n$ equations gives us the exact unknown trajectory. The number of linear properties we need to know, however, is at least as large as the number of coordinates in the trajectory itself. If only $m \ll 2n$ linear properties are known, the solution will be in a $(2n-m)$-dimensional subspace, at best. When the candidate can only be restricted to a subspace, it can be arbitrarily far away from X. If the known properties are non-linear, finding a solution to the corresponding equations, even if enough properties are known, may become computationally infeasible.

As an example, consider m known trajectories, $\mathscr{T}^1,\ldots,\mathscr{T}^m$, and m corresponding positive real values δ_i, where

$$\delta_i = \|\mathscr{X} - \mathscr{T}^i\|_2, \tag{9}$$

for unknown trajectory $\mathscr{X}$. Our task is to find an approximation $\mathscr{X}'$ which minimizes the distance $\|\mathscr{X} - \mathscr{X}'\|_2$. This can be done by hyper-lateration, a generalization of trilateration. By squaring the known distances we obtain a system of n quadratic equations: $\delta_i^2 = \sum_{i=0}^{2n-1}|\mathscr{T}_i - \mathscr{T}_i'|^2$, for $i \in \{1,\ldots,n\}$.

However, by subtracting each of these equations from the first equation we obtain $n-1$ linear equations:

$$\delta_1^2 - \delta_i^2 = \|\mathscr{X} - \mathscr{T}^1\|_2^2 - \|\mathscr{X} - \mathscr{T}^i\|_2^2 \quad (10)$$

$$\Rightarrow \delta_1^2 - \delta_i^2 = \sum_{j=1}^{2n} 2\mathscr{X}_j(\mathscr{T}_j^i - \mathscr{T}_j^1) + (\mathscr{T}_j^1)^2 - (\mathscr{T}_j^i)^2, \quad (11)$$

for $i \in \{2, \ldots, 2n+1\}$. [*To uniquely identify trajectory A from Figure 1, we need to know at least 7 other trajectories, and their distances to A. In the example, we only know trajectory B, which is at distance* 0.9 *to A.*] As previously argued, this approach is unsatisfactory since we need to know at least $(2n+1)$ distances[3], and the method is too sensitive to noise.

The discussion above reveals a need to find a method which can approximate the unknown trajectory with considerably fewer known properties than coordinates. However, the best we can hope for is to find a candidate trajectory which has the same properties as the properties we know about $\mathscr{X}$. If, for instance, the only information we have about $\mathscr{X}$ is that it is a car driving at an average speed of 50 km/h in Athens, then any $\mathscr{X}'$ which moves along the roads of Athens at 50 km/h is a possible solution. We thus want to minimize the difference between the given properties of $\mathscr{X}$, and the corresponding properties of the candidate $\mathscr{X}'$; in the case above, the distances to the known trajectories. To this end, we define the "error" of a candidate $\mathscr{X}'$ as

$$E(\mathscr{X}') = \frac{1}{2}\sum_{i=1}^{m} \big(P_i(\mathscr{X}') - P_i(\mathscr{X})\big)^2, \quad (12)$$

where P_i are the properties which are known about the target trajectory (in other words: $P_i(\mathscr{X})$ are known values). Clearly the error function is 0 if the candidate is equal to the unknown trajectory. Furthermore, the error function is positive, and differentiable as long as the properties are differentiable.

A natural way to solve this problem is to see it as an optimization problem. [*By squaring the differences in Equation 12 the derivative in a point is the "distance" to a solution. This trick makes the gradient descent optimization algorithm converge fast when it is far from a solution, and slow when it is close to a solution, making the algorithm more robust. Gradient descent is the algorithm used in our method described in detail in Section 8.*]

6. Erroneous Knowledge

The information available to the attacker about the unknown trajectory may not always be precise — it can be subject to noise. This noise can be either

[3]Considering that a trajectory may have thousands of sample points obtaining $(2n+1)$ distances is infeasible in many cases.

a deliberate attempt from the data holder to anonymize the released data, or simply errors in the background knowledge of the attacker.

It is not in the scope of this paper to evaluate the effectiveness of anonymization techniques based on data perturbation, such as the techniques presented in [2]. Indeed, several other papers have treated this topic, showing that data perturbation techniques are not always effective in protecting privacy [9]. We will, however, study the robustness of our attack in face of errors in the information available to the attacker.

It is important to note the difference between noise in the original measurement of trajectories, and noise which is added before data is released. There is always an unavoidable amount of noise in the measurement of a trajectory. GPS devices, for instance, can only measure location to a certain accuracy. [*In Figure 1 noise in the measurement process placed some of the sample points slightly off the actual path of the moving object. One sample point was even off the street.*] This *pre-storage* noise, however, does not introduce any inconsistencies in the data stored in the database, it only reduces the accuracy of the data. Noise can also be added *post-storage* when data is released from the database in an attempt to prevent breaching the privacy of individuals. As mentioned above, data perturbation is a well-studied field in data privacy. Contrary to pre-storage noise, post-storage noise may give a slightly inconsistent view of the data in the database. As an example of post-storage noise, consider a trajectory database which releases the mutual distances between all trajectories it contains. The distances of these trajectories (when thought about as vectors) have to satisfy the triangle inequality. If, however, noise is added independently to each of the released distances, the distances will no longer satisfy the triangle inequality. [*Consider once more the example from Figure 1: The Euclidean distance between the two stored trajectories A and B is approximately 0.9. When the distance data is released, however, the owner of the data may add a deliberate error of 0.1 and say that the distance is 1.*]

[*The aim of this paper is to reconstruct the stored trajectory (the dashed polyline in Figure 1). In this case pre-storage noise is irrelevant, so we concentrate on the effects of post-storage noise.*]

We consider the case of known distances, where the attacker knows m trajectories, $\mathscr{T}^1, \ldots, \mathscr{T}^m$, and m corresponding distances:

$$\delta_i = \|\mathscr{X} - \mathscr{T}^i\|_2 + \epsilon_i, \tag{13}$$

where ϵ_i are *noise terms*.

When the equations known to the attacker have errors as above, reconstruction based on solving the system of equations by, for instance, hyper-lateration as described in Section 5 does not work well. On the other hand, if the noise follows a distribution with an expected value of 0, a reconstruction method based on optimization should still perform well, since the real solution is likely to be close to the solution of the erroneous equations. In Section 9 we show that our method can handle additive noise which follows a Gaussian distribution up to a certain standard deviation. While we only demonstrate that our attack can

handle additive Gaussian noise, we are aware that there are many other models of noise which an attacker may face. However, a full study of noise is out of the scope of this work.

7. Measuring Success

Before describing our technique for finding an unknown trajectory, a discussion about the measure of success of such reconstruction algorithm is in place. In essence the success depends on how well the candidate represents the target.

In [8] an unknown target trajectory was reconstructed from knowledge of the distance from the target trajectory to each trajectory in a set of known trajectories. To evaluate the success of the reconstruction the following success rate was used:

$$SR(\mathscr{X}') = 1 - \frac{\|\mathscr{X} - \mathscr{X}'\|_2}{\delta_{min}}, \tag{14}$$

where $\delta_{min} = \min_i(\delta_i)$ is the smallest known distance. This success-rate is 1 if the method finds $\mathscr{X}$ precisely, 0 if it returns the closest known trajectory, and negative if it performs worse than just returning the closest known trajectory. This measure has a number of shortcomings, which makes it difficult to compare the success of different algorithms, or even the same algorithm, but applied to different datasets. One obvious problem is that the success rate cannot be applied to reconstruction methods which do not use the distance to known trajectories. Furthermore, it is very difficult to obtain a high success rate for a dataset with many close trajectories (since δ_{min} is likely to be a very small number). Another problem is that this success does not take the "resolution" of the target trajectory into account: For fixed length target trajectories the success rate does not depend on the number of sample points. If the target trajectory has a high sample rate (high resolution) it is likely that the quality of the reconstruction is more sensitive to noise than if the same target trajectory only has a low resolution.

In this paper we overcome some of the shortcomings of the old success measure by defining a new *success rate* $SR(\mathscr{X}')$ of a candidate trajectory. The *success rate* should satisfy the following properties:

- $SR(\mathscr{X}') \in [0, 1]$
- $SR(\mathscr{X}) = 1$
- Depend only on the target and candidate trajectories.
- Be independent of the magnitude of coordinates.

Intuitively the quality of a candidate trajectory depends on how far away the candidate trajectory is from the target trajectory at any given time. In our case, since we assume that trajectories are aligned, the average distance of the

candidate trajectory to the target trajectory over time is the average sample distance:

$$ASD_T(X) = \frac{1}{n}\sum_{i=0}^{n-1} \|X_i - T_i\|_2. \tag{15}$$

The average sample distance alone, however, is not a good measure of success, since it depends highly on the magnitude of the coordinates. To factor out this dependency on the magnitude of the coordinates, we divide the average sample distance with the total length of the target trajectory, which can be computed as:

$$\|T\|_l = \sum_{i=0}^{n-2} \|T_i - T_{i+1}\|_2. \tag{16}$$

The fraction $ASD_T(X)/\|T\|_l$ is a non-negative real number, which is 0 when $X = T$. We define the success rate from this fraction as follows:

$$SR(X') = \mathrm{e}^{-\alpha ASD_T(X)/\|T\|_l}, \tag{17}$$

where α is a sensitivity factor which decides how steep the success rate goes to 1 as the candidate approaches the target. The new success rate satisfies the criteria listed above: $SR(T) = \mathrm{e}^{-\alpha ASD_T(T)/\|T\|_l} = \mathrm{e}^0 = 1$, and as $ASD_T(X)$ tends to infinity, $SR(X)$ tends to $\mathrm{e}^{-\infty} = 0$.

[*As an example, suppose that trajectory B from Figure 1 is an attempted reconstruction of trajectory A. The success rate of B is* $SR(B) \approx 0.74$ *(with* $\alpha = 1$*), whereas the trajectory* $((0,1),(1,1),(1.5,1))$*, running on a parallel street, has success rate* 0.61*, and the trajectory* $((0,2),(1,2),(1.5,2))$*, two streets over, has success rate* 0.34.]

More research in a proper way to measure how well a candidate trajectory represents a target trajectory is needed. We are aware that the success measure defined above is not appropriate in all situations. Trajectories may be laying on top of each other, thus giving the visual impression of a perfect match, but may be very far apart in time: Even though all sample points overlap, the chronological ordering may be reversed, this situation will give a very poor success rate with the measure defined above, but will appear as a perfect match and, indeed, it will identify exactly where the moving object has been. Furthermore, the context of a trajectory has a great influence on how well we perceive the reconstruction to be. A very coarse reconstruction of a trajectory which moves in a rural area with only few roads may be better than even a very accurate reconstruction of a trajectory moving in a urban area with very small and close roads.

8. Our Reconstruction Method

We adopt the steepest descent (gradient descent search) algorithm to find a candidate with minimum error.

The error-function (12) has value 0 exactly when the candidate trajectory has the same properties as the known trajectory $\mathscr{T}_i$, for all properties $P_i, i \in$

$\{1, \ldots, m\}$. Furthermore, since (12) is a positive valued function, the target trajectory is a global minimum. There may, however, be more than one global minimum, as well as several local minima; but any zero of the error-function exhausts the knowledge we can possibly have about the unknown trajectory, given the known properties. Recall that the gradient descent algorithm finds a zero of a positive and continuously differentiable function E as follows

1. Choose a random point, x_0, in the domain of E.
2. Iteratively define $x_{i+1} = x_i - \gamma \nabla E(x_i)$, for some step-size $\gamma > 0$.
3. When $x_{i+1} = x_i$ ($\nabla E(x_i) = 0$) a (local) minimum has been reached. If $E(x_i) = 0$ we have a global minimum (since E is non-negative), and we stop. Otherwise, we go back to step 2.

Note that the size of the steps taken in the direction of the gradients are determined by the step size, γ. Ideally, the attack should neither underestimate nor overestimate the step size. If the step size is too small, the attack will converge very slowly, thus yielding poor success rate, whereas if the step size is too large the attack takes big steps and possibly overshoots the target, which again yields a poor success rate. Also note that gradient descent is not the most efficient algorithm for solving this kind of optimization problem. However, the aim of this paper is to demonstrate potential dangers in data disclosure. A formal analysis of the efficiency of the attack is out of the scope of this work.

The gradient, $\nabla E(\mathscr{X}')$, depends on the differentiable properties $P_i, i \in \{1, \ldots, m\}$:

$$\frac{\partial}{\partial \mathscr{X}'_i} E(\mathscr{X}') = \sum_{j=1}^{m} \big(P_i(\mathscr{X}') - P_i(\mathscr{X})\big) \frac{\partial}{\partial \mathscr{X}'_i} P_j(\mathscr{X}'). \tag{18}$$

If all properties are continuously differentiable, then the gradient is a continuous function in the candidate trajectory.

Recall that not all partial derivatives of the maximum speed property are continuous. The discontinuity happens when more than one segment are equally fast, and are the fastest segments. However, since we defined the derivative to be one in this case, the gradient descent will still change the speed of these segments until they satisfy the known maximal speed.

Even though an attacker cannot know the final success rate of his attack, there are situations where he can give a lower bound on the success rate. Since the success rate is defined in terms of the average sample distance, he can get the following bound in the situation where he knows the average sample distance to a set of known trajectories.

Theorem 1. *Let $T^1, \ldots, T^m$ be known trajectories, and let $\delta_i = d_2(T^i, X)$ be the average sample distances to the unknown trajectory X. Then, for any trajectory X' with $E(X') = 0$ the success rate is:*

$$SR(X') \geq \mathrm{e}^{-2\alpha\delta_{max}/(n\|X\|)}, \tag{19}$$

where $\delta_{max} = \max_i(\delta_i)$ is the largest given distance, and E is the error function defined in Eq. 12.

While the attacker does not know the length of X, he may be able to estimate it from his background knowledge.

Proof. We first observe that since E is a sum of the non-negative terms $1/2(d_2(T^i, X') - \delta_i)^2$, and since $E(X') = 0$, necessarily $d_2(T^i, X') = \delta_i$.

Now, note that for all $k \in \{1, \ldots, m\}$

$$\begin{aligned} ASD_X(X') &= \frac{1}{n}\sum_{i=0}^{n-1} \|X'_i - X_i\|_2 \\ &= \frac{1}{n}\sum_{i=0}^{n-1} \|X'_i - T_i^k + T_i^k - X_i\|_2 \\ &\leq \frac{1}{n}\sum_{i=0}^{n-1} (\|X'_i - T_i^k\|_2 + \|T_i^k - X_i\|_2) \\ &= \frac{1}{n}(d_2(T^k, X') + d_2(T^k, X)) \\ &= \frac{2\delta_k}{n}. \end{aligned}$$

Inserting this in the definition of the success rate gives us:

$$SR(X') = \mathrm{e}^{-\alpha ASD_X(X')/\|X\|_l} \geq \mathrm{e}^{-2\alpha\delta_k/(n\|X\|_l)}. \tag{20}$$

Since Eq. 20 is true for all δ_k, it is true for δ_{max}. □

9. Experimental Results

To validate our reconstruction method, we have designed three different tests, and applied them on two datasets of real-world GPS data. In the first test, we let the recunstruction method run for many iterations to see how the success-rate evolves over time. The second test consists of several executions of the reconstruction algorithm on the same dataset, but with a varying number of known trajectories and background information. The aim of the second test is to verify the claim that an attacker can reconstruct a target trajectory with only a few known trajectories. In the third test, we apply Gaussian noise to the released distance data to see how fast the success-rate diminishes in the face of errors.

The first dataset contains trajectories of school busses in Athens[4, 18]. This dataset contains 145 trajectories each with 1096 (x, y) sample points. The trajectories are aligned with samples approximately every half minute on 108 different days.[*The sampling frequency of this dataset is so high that consecutive sample points are very close; often three or more consecutive sample points lie on a near straight line. This gives a lot of redundancy in the data, which our reconstruction algorithm will benefit from.*] This dataset is chosen because of the high redundancy, which enables us to test our reconstruction algorithm in a near best-case scenario.

The second dataset contains trajectories of private cars in Milan[5]. The dataset contains 135 trajectories recorded with sample points at irregular intervals over a period of time of one week. The density of sample points of the Milan dataset is lower than the dataset from Athens[*, giving a much lower redundancy than in the Milan data set*]. Even though the trajectories in the Milan dataset are not aligned, for the purpose of these tests, we assume that they are. This assumptions only means that we are not working with the original trajectories, but trajectories which follow the same routes, but at different speeds. The Milan dataset is [*much less redundant*] than the Athens dataset, and is chosen to test our reconstruction algorithm in a scenario which is much more realistic (and relevant) than the Athens dataset.

For the purpose of testing the reconstruction method described in Section 8 we implemented a limited version. In the implementation the step-size γ is set to one, and the implementation does not restart if a local maxima, or saddle point is reached. Furthermore, we assume that the two datasets are aligned, so that we can discard time. In all tests in this section we report the success rate as defined in Equation 17. We have chosen the smoothness parameter $\alpha = 20$ based on visual impression from several tests.

Even though efficiency is not a primary concern in this work, we remark that it takes approximately 8 minutes to run the reconstruction method with 50 known trajectories from the Athens dataset for 60.000 iterations on a 1.7 GHz laptop.

9.1. Success-rate over Time

In the first test, we run the reconstruction method on the Athens dataset for one million iterations to see how the success-rate evolves over time. Figure 2 shows the convergence speed of our reconstruction method. The success-rate is an average value obtained from 5 runs of the reconstruction algorithm on the Athens dataset with 50 known trajectories, where the target trajectory is selected at random in each of the 5 runs. The x-axis shows the number of iterations in log-scale.

Figure 3 shows the evolution of candidates in one experiment with the Athens dataset and one with the Milan dataset. The test uses 60 known trajectories from the Athens dataset, and 90 known trajectories from the Milan dataset. Notice that a success-rate of 0.6 allows us to determine the general area in which the target trajectory is moving, but not specific streets. With a success-rate of 0.85 it is possible to identify some, but not all, streets.

9.2. Distance Measures and Background Information

In the second test, we fix the number of iterations used in the reconstruction to 60.000, and measure the success-rate as a function of the information available to the attacker. We run the reconstruction with a different number of known trajectories, ranging from 10 to 140. We also run the reconstruction both with and without background information about average and maximum speed in the dataset. And finally we run the reconstruction with two different distance measures: Euclidean distance, and average sample distance.

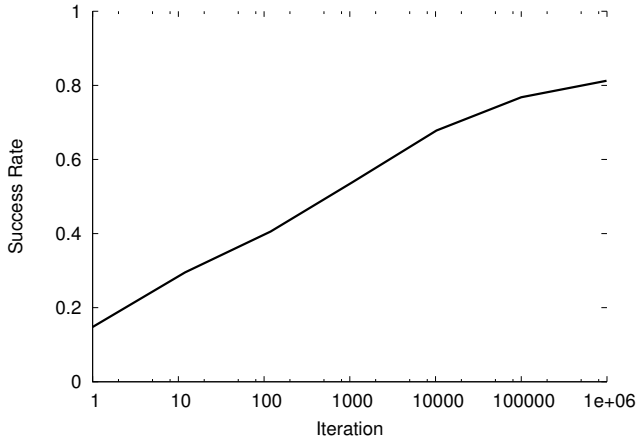
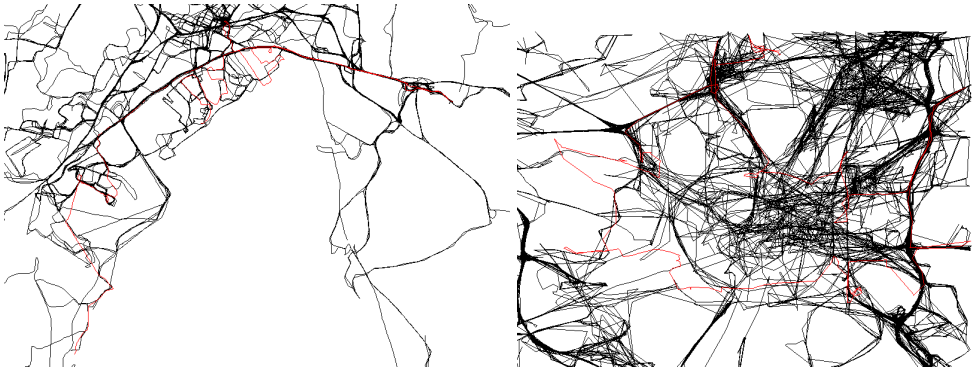


Figure 2: Success-rate vs. number of iterations for the Athens dataset. The x-axis is in log-scale (Average of 5 experiments with 50 known trajectories).

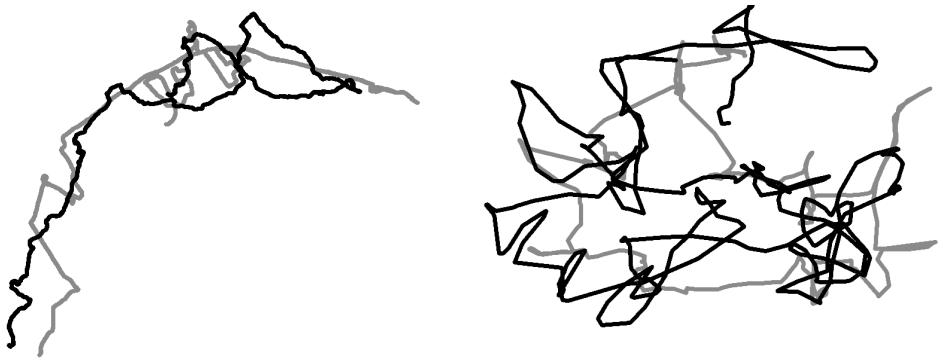
Figure 4 shows the success-rate attainable for different numbers of known trajectories in the Athens dataset. Each sample is the average success-rate of 20 tests each running for 60.000 iterations. Both target and known trajectories are chosen at random in each test. The solid line shows the success rate of the attack, when the attacker only uses the Euclidean distances between the target and the known trajectories as the continuously differentiable properties. The dashed line shows the success rate, when the attacker assumes that the target trajectory moves with an average and maximum speed similar to the average and maximum speed of his known trajectories (See Sec. 4.2). The graph shows that for the case of the Athens dataset, using knowledge about the average speed does not give extra success to the attack. However, Figure 5 shows the same experiment for the Milan dataset, and here it is clear that, for a low number of known trajectories, using knowledge about the average speed gives a success rate up to 0.05 higher (for 20–40 known trajectories). From the result, we see that simple background information, such as average and maximum speed, improves the accuracy of the reconstruction when [*the trajectory data has low redundancy*] (as in the Milan dataset), or when an insufficient number of known trajectories are available. However, for trajectory data [*which has high redundancy*], the impact of simple background information is not significant. We have only tested speed information, but other kinds of background information may give a higher impact.

Figure 6 shows the success-rate attainable for different numbers of known trajectories in the Athens dataset when the attacker knows the *average sample distance* to his known trajectories. Each sample is the average success-rate of 20 tests each running for 60.000 iterations. Both target and known trajectories are chosen at random in each test. Figure 7 shows the same result for the Milan dataset. The success rate attained from these tests shows that for our attack,



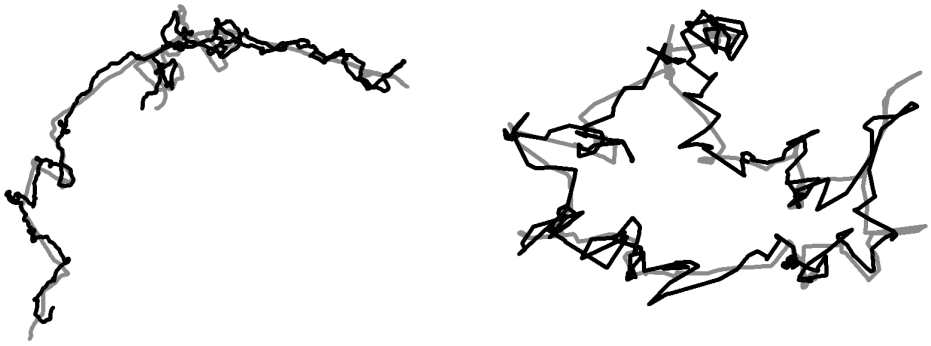
(a) The 60 known trajectories for Athens.

(b) The 90 known trajectories for Milan.



(c) Athens, Success-rate 0.60

(d) Milan, Success-rate 0.60



(e) Athens, Success-rate 0.85

(f) Milan, Success-rate 0.85

Figure 3: Evolution of the candidate trajectory in the Athens and Milan datasets.

knowing the mutual Euclidean distance is stronger than knowing the mutual average sample distances.

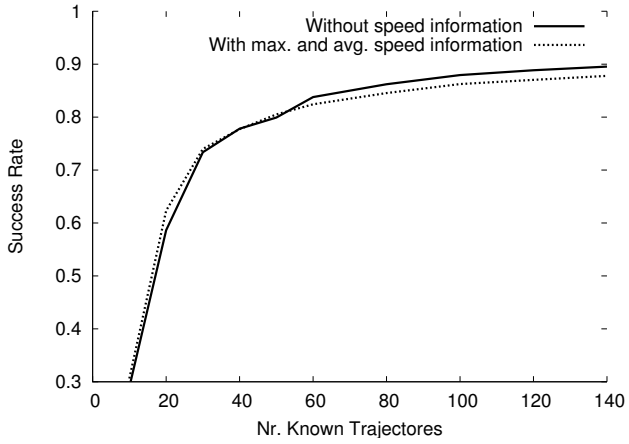


Figure 4: Success-rate vs. number of known trajectories in the Athens dataset with known Euclidean distances. With and without known average and maximum speed.

9.3. Noise

Figure 8 shows the success-rate attainable in the face of errors in the known distances. Independent and identically distributed Gaussian noise with a mean value of 0 has been added to each distance known to the attacker. The Gaussian x-axis of the figure shows the deviation of the noise as a fraction of the average value of the distances. This means that for $x = 1$ approximately 32% of the distances are subject to noise with the same magnitude as the distance itself.

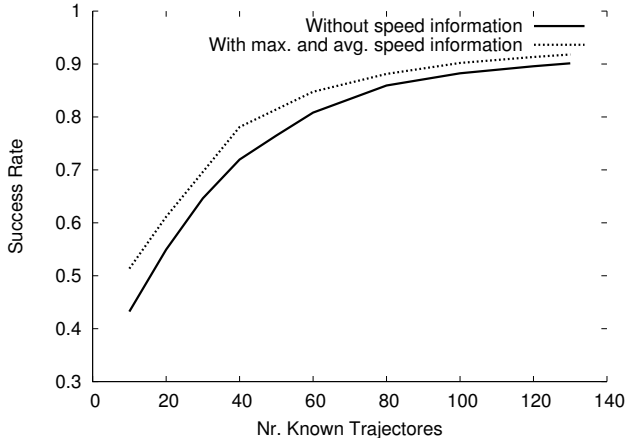


Figure 5: Success-rate vs. number of known trajectories in the Milan dataset with known Euclidean distances. With and without known average and maximum speed.

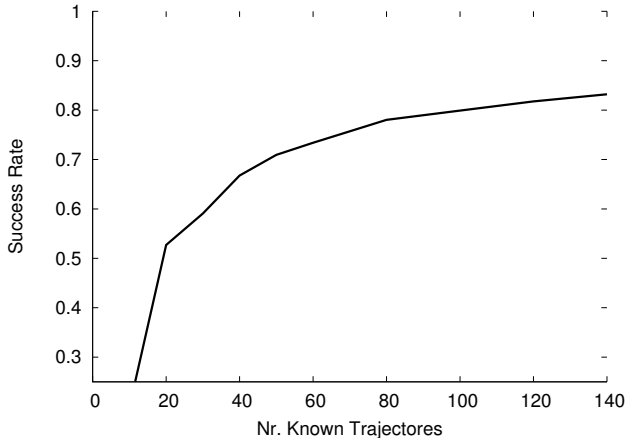


Figure 6: Success-rate vs. number of known trajectories for the Athens dataset with known average sample distance.

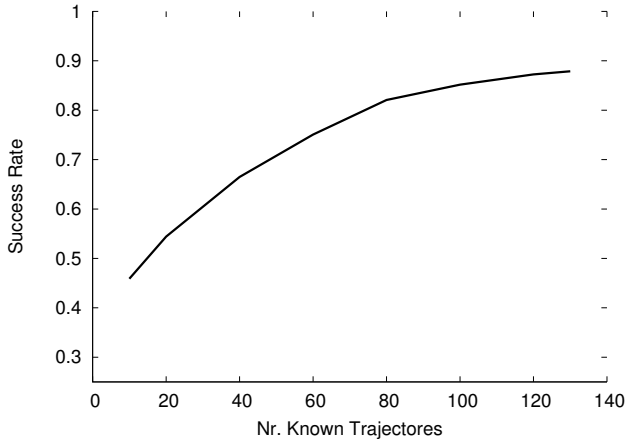


Figure 7: Success-rate vs. number of known trajectories for the Milan dataset with known average sample distance.

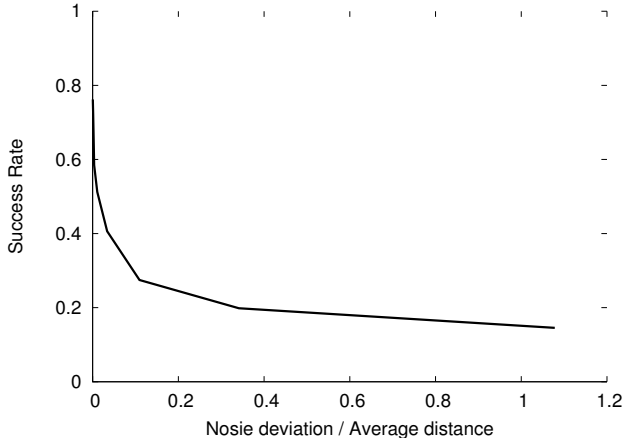


Figure 8: Success-rate for 40 known Euclidean distances subject to noise.

10. Conclusion

Privacy risks in trajectory data publishing are high due to rich sources of background information. In this paper we consider distance preserving data transformations, and assume that the mutual distances of trajectories are released rather than the actual trajectories. We show that, even in such a scenario, the individual trajectories can be identified using background information such as known samples and speed limits. We use the speed limit as background information, but the attack model we propose is general enough so that any kind of background information about trajectories with continuous properties could be the input. Our method is optimal in the sense that it will eventually find a candidate which exhausts all the information available about the unknown trajectory.

We implemented the proposed methods on two real data sets. One data set consists of routes of school busses in Athens and it represents a more predictable data set since busses will usually follow the same routes. The second data set is obtained in the context of the GeoPKDD project and it consists of the GPS tracks of a set of cars in the city of Milan in Italy. GPS tracks of cars are definitely less predictable since there are many routes that they can follow. Experiments performed on these real data sets show that unknown private trajectories with 1096 sample points can be reconstructed with an expected success-rate of 0.8 by knowing the distance to only 50 known trajectories. Reconstructing the trajectory perfectly with "tri-lateration" would require 2193 known trajectories.

[*We would like to thank the anonymous reviewers for valuable comments and suggestions.*]

References

[1] O. Abul and F. Bonchi. Never walk alone: Uncertainty for anonymity in moving objects databases. In *The 24th International Conference on Data Engineering (ICDE 2008)*, 2008.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *SIGMOD Conference*, pages 439–450, 2000.

[3] C. Bettini, S. Mascetti, X. S. Wang, and S. Jajodia. Anonymity in location-based services: Towards a general framework. In *MDM*, pages 69–76, 2007.

[4] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis. Nearest neighbor search on moving object trajectories. In *SSTD05: Advances in Spatial and Temporal Databases*, pages 328–345, 2005.

[5] http://www.geopkdd.eu/.

[6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD '07: Proceedings of the 13th ACM SIGKDD international*

conference on Knowledge discovery and data mining, pages 330–339. ACM, 2007.

[7] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bulletin of Mathematical Biology*, 55(1):141–154, January 1993.

[8] E. Kaplan, T. B. Pedersen, E. Savaş, and Y. Saygın. Privacy risks in trajectory data publishing: Reconstructing private trajectories from continuous properties. In *KES 2008: Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science, pages 642–649. Springer, 2008.

[9] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *ICDM*, pages 99–106, 2003.

[10] J. Lee, J. Han, and K. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[11] X. Li, J. Han, J.-G. Lee, and H. Gonzalez. Traffic density-based discovery of hot routes in road networks. In *SSTD 2007: 10th International Symposium on Advances in Spatial and Temporal Databases*, Lecture Notes in Computer Science, pages 441–459. Springer, 2007.

[12] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Trans. Knowl. Data Eng.*, 18(1):92–106, 2006.

[13] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: A privacy-aware location-based database server. In *ICDE*, pages 1499–1500, 2007.

[14] M. Nanni and D. Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, November 2006.

[15] C. J. Needham and R. D. Boyle. Performance evaluation metrics and statistics for positional tracker evaluation. In *Third International Conference on Computer Vision Systems, ICVS 2003*, pages 278–289, 2003.

[16] E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of ACM GIS Workshop on Security and Privacy in GIS and LBS, Irvine, CA, USA*, November 2008.

[17] First interdisciplinary workshop on mobility, data mining and privacy, rome, italy. http://wiki.kdubiq.org/mobileDMprivacyWorkshop/, February 2008.

[18] http://www.rtreeportal.org/.

[19] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188, 1998.

[20] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, pages 65–72, 2008.

[21] E. O. Turgay, T. B. Pedersen, Y. Saygın, E. Savaş, and A. Levi. Disclosure risks of distance preserving data transformations. In *SSDBM 2008: Scientific and Statistical Database Management Conference*, 2008.