2008

Towards Trajectory Anonymization: A Generalization-Based Approach

Mehmet Ercan Nergiz

Maurizo Atzori

Yucel Saygin

Report Number:
08-015

Towards Trajectory Anonymization: A Generalization-Based Approach

Mehmet Ercan Nergiz
Maurizio Atzori
Yucel Saygin

CSD TR #08-015
June 2008

Journal of Geographical Information Science
Vol. 00, No. 00, 00 Month 200x, 1–23

RESEARCH ARTICLE

Towards Trajectory Anonymization: a Generalization-Based Approach

Mehmet Ercan Nergiz[a] Maurizio Atzori[b] Yucel Saygin[c]

[a] *Purdue University, West Lafayette, IN, USA*;
[b] *DD Lab, ISTI-CNR, Pisa, Italy*;
[c] *Sabanci University, Istanbul, Turkey*

(*Received 00 Month 200x; final version received 00 Month 200x*)

Trajectory datasets are becoming more and more popular due to the massive usage of GPS and other location-based devices and services. In this paper, we address privacy issues regarding the identification of individuals in static trajectory datasets. We provide privacy protection by defining trajectory k-anonymity, meaning every released information refers to at least k users/trajectories. We propose a novel generalization-based approach that applies to trajectories and sequences in general. We also suggest the use of a simple random reconstruction of the original dataset from the anonymization, to overcome possible drawbacks of generalization approaches.

We present a utility metric that maximizes the probability of a good representation and propose trajectory anonymization techniques to address time and space sensitive applications. The experimental results over synthetic trajectory datasets show the effectiveness of the proposed approach.

Keywords: Spatio temporal, k anonymity, privacy

1. Introduction

Data publishing is essential for providing resources for research, and for the transparency of government institutions and companies. However, data publishing is also risky since published data may contain sensitive information. Therefore, the first step before data publishing is to remove the personally identifying information. In (*37*), it has been shown that removing personally identifying information is not enough to protect privacy. This is due to the fact that the released database can be linked to public databases through a set of common attributes which are called quasi-identifiers. For example in US the combination of zip code, and birth date is unique for 87% of the citizens (*37*). This figure increases as more attributes are added to the combination. Sweeney et al. showed that they could re-identify the supposedly anonymous health records via linking them to publicly available voters registration list. This striking result increased the concerns and research efforts for privacy and anonymization in published databases. The problem of linkage becomes even more complicated in our highly connected world as the number and variety of data sources increase.

Mobile service providers can now predict the location of mobile users via triangulation with a high precision. Coupled with applications such as location-based services (LBS) that are enabled by GPS equipped mobile devices, it is now very easy to track the location of individuals voluntarily or non-voluntarily over a period of time. The time and location information of a person (or a moving object in

ISSN: 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/1365881YYxxxxxxxx
http://www.informaworld.com

general) collected over a period of time forms a trajectory which can be thought of as a set of spatio-temporal data points spanning a time interval. Trajectory data sets contain valuable information which can be harvested by data mining tools to obtain models for applications such as city traffic planning or marketing.

However, time and location are sensitive information, therefore personally identifying information needs to be removed from trajectories before they can be released. But, even after de-identification, trajectory data sets are still prone to linkage attacks since space and time attributes are very powerful quasi-identifiers. For example, for a trajectory that starts at a specific location every weekday in the morning and reaches another location in an hour, it is very easy to infer that the starting location in the morning is home, and the location reached after an hour is the work place. What an adversary can do is to look at a phone directory to search for home addresses and work addresses to link the trajectories with their owners.

In general, the solution to prevent linkage attacks in de-identified data sets is anonymization (*37, 36*). k-Anonymity was proposed as a standard for privacy over relational databases. It can be summarized as “safety in numbers” and ensures that every entity in the table is indistinguishable from $k - 1$ other entities. Achieving optimal k-anonymity was proven to be NP-Hard, therefore heuristic algorithms have been proposed in the literature to k-anonymize data sets. In case of spatio-temporal trajectories the problem of anonymization is even harder since consecutive points in a trajectory are dependent on each other. Therefore anonymization should consider every trajectory as a whole when anonymizing.

In this paper, we concentrate on spatio-temporal trajectories. We first extend the notion of k-anonymity for trajectories and then describe a heuristic method for achieving k-anonymity of trajectories. Trajectories are then published by only releasing a representative trajectory to further protect the privacy of people whose trajectories refer to.

Outline of the paper. In Section 2 we give some motivating applications for anonymization of static trajectory and sequence datasets. In Section 3 related work on privacy over relational databases and spatio-temporal data is presented. We then describe the problem of trajectory anonymity in Section 4. Detailed algorithms on how to obtain generalized trajectories and results on the computational complexity are given in Section 5, while in Section 6 we propose a (optional in general but required in some applications) reconstruction step to release only a representative trajectory (instead of generalized trajectories). Finally, in Section 7 we discuss results of our empirical experimentation and then we sketch some conclusions.

2. Applications

A number of applications motivate our work. In this section, we give several examples of interesting and emerging applications where privacy over static trajectory or sequence dataset is of paramount importance:

2.1. *Data Analysis and Mining*

As the use of mobile devices grows rapidly, the value of storing spatio-temporal data is better understood. Business companies, governments, and science institutes are heavily collecting and storing spatio-temporal data to extract useful and relevant information (*35, 30, 39, 31, 27, 11*).

The applications over mobile data, such as GPS data, is no longer limited to location-based servicing or querying. Several spatio-temporal data mining techniques has been developed. Such techniques has been used by business companies to maximize employee efficiency (*30*), by governments to understand the infrastructure (*39*) and by research groups to observe human behavior (*31, 27, 35*).

We stress that the output of mining algorithms might fail to remove all individually identifying information In fact, work in (*5, 4*) shows that even simple statistics such as the most higher counts over column projections (same as frequent itemset mining in data mining literature) may violate anonymity based privacy definitions. In other words, information regarding very few people may be released, allowing possible linking/joining attacks through the use of some columns. Therefore, in order to release statistics over any dataset, provable anonymization techniques must be applied before computing statistics and/or mining. Our paper provides such techniques for spatio-temporal trajectory datasets, that can be used before any non anonymity-preserving mining or analysis algorithms.

2.2. *Trajectory Data Sharing and Outsourcing*

As in the case of conventional databases, storing of spatio-temporal data along with the variety and importance of applications necessitate the release of the data. Since most trajectory databases contain personal information, publicizing such databases is subject to privacy regulations and requires de-identification (*14, 15, 20*). One of the most effective and recognized technique for de-identification is anonymization (*20*).

Even though human data is subject to changes, most real world applications work on static data. The reason is the high cost of mining dynamic information in terms of both accuracy and efficiency. Most systems instead follow a trade-off. Changes in the system are captured by incremental mining up to date data periodically (e.g. monitor the traffic continuously but mine the data every week) or updating the existing data mining model with fresh data. In either cases, static databases are valuable. This is the case when sharing trajectory databases for outsourced trajectory analysis. As a typical example, we have municipalities willing to perform traffic data analysis but with limited internal skills.

2.3. *Web Analytics and other Logs-based Activities*

Web analytics, that is, analysis and mining of user traces, is not only becoming of fundamental importance for internet business, but also posing serious privacy concerns.

A notable event related to this privacy problem is American On-line's (AOL) release of massive amounts of log data. The data included queries done by those users in a three month period this year, as well as whether they clicked on a result, what that result was and where it appeared on the result page. Although there was no personally-identifiable data linked to these accounts, a number of attacks have been performed by using quasi-identifiers and other infrequent information contained in the logs.

The paper is focused on trajectories, i.e., sequences of spatio-temporal points. Nevertheless, the generalization-based approach we are proposing easily adapt to different kind of sequences, such as web server logs of page visits.

Here we only sketch the idea on how to extend trajectory anonymity to web server logs. Suppose the trajectory anonymization algorithm recognizes the *second* page of a visit, namely:

```
session=8545634 page_sequence=2
servername.com/sect1/sect1.2/page1.html
```

as an infrequent "point" among the user web traces. In this case, the point may be generalized to, e.g.,

```
session=8545634 page_sequence=[2 OR 3]
servername.com/section1/*
```

by suppressing or using or a user-provided page hierarchy. Notice that this kind of generalization cannot be computed by relational k-anonymity algorithms (*34*) since the *sequence* information of the user trace would not be taken into account appropriately. Even if we ignore ordering among page visits, session or user pseudo-ID columns will force the anonymization process to consider pages of the same user. This will bring to possibly overestimating privacy protection (e.g., when a user has visited several pages) but, more often, reducing the effectiveness by suppressing unnecessary data (e.g., when a user visited less than k pages).

3. Related Work

3.1. *k-Anonymity and Privacy over Relational Databases*

Addressing privacy concerns when releasing person specific datasets is well studied in the literature (*36, 28, 3, 26, 32*). Simply removing uniquely identifying information (SSN, name) from data is not sufficient to prevent identification because partially identifying information (quasi-identifiers (QI); age, sex, city ...) can still be mapped to individuals by using external knowledge (*37*). k-Anonymity is defined in (*36*), to protect against identification of individuals in person specific datasets.

Definition 3.1 *k-Anonymity:* A table T^* is k-anonymous w.r.t. a set of attributes QI if each record in $T^*[QI]$ appears at least k times.

k-Anonymity property ensures that a given set of quasi identifiers can only be mapped to at least k entities in the dataset. The most common technique being used to anonymize a given dataset is value generalizations and suppressions. In multidimensional space, the counter part of these operations is replacing a set of points with the minimum bounding box that covers the points. It should be noted that k-anonymization preserves the truth of the data.

Entities in trajectory datasets are more complex than those studied by classical k-anonymity approaches. Anonymization of complex entities was proposed in (*34*) where data about private entities reside in multiple datasets of a relational database. Even though trajectory datasets can be represented in relational databases, order of points over a given trajectory matters due to the linear time property. Work in (*34*) does not assume any ordering between points. Also applications over trajectory databases are very specific and require different cost metrics and different anonymization techniques.

In (*28*) authors also warn that, in each set of people with same values for the anonymized QI ℓ-diversity must hold, i.e., sensitive attribute values must be diverse enough. Otherwise, it is possible to infer the exact sensitive value with arbitrarily high probability. We will discuss how to extend the concept of ℓ-diversity for trajectory dataset in Section 8.

As done in previous work on LBS and trajectory privacy, we will not directly address ℓ-diversity issues during the presentation, while we will sketch some possible approaches to this interesting issue as a future work in Section 9.

3.2. *Privacy-preserving LBS*

There has been a lot of work on privacy issues regarding the use of location based services (LBSs) by mobile users. Most work defined the privacy risk as linking of requests and locations to specific mobile users. Works in (*13, 21*) used perturbation and obfuscation techniques to deidentify a given request or a location; they differ from this work in the privacy constraints they enforce. Anonymization based privacy protection was used in (*16, 6, 17, 18*). In (*18*), anonymity was enforced on sensitive locations other than user location points or trajectories. In (*16, 17*), individual location points belonging to a user is assumed to be unlinked and points of the users are anonymized other than the trajectories. In (*6*), anonymization process enforces points referring to same set of users to be anonymized together always. However work assumes anonymization per request other than whole trajectory anonymization and heuristic to specify groups of users is restricted in a time frame. (Such an approach does not anonymize time.)

3.3. *Trace and Trajectory Anonymization*

All of the proposed privacy preservation methods on LBSs so far assume a dynamic, real-time environment and methodology being used is based on local decisions. We are also aware of very recent, independent work (*7, 22*) addressing the problem of preserving privacy in static trajectory databases. Both works rely on uncertainty in the spatio-temporal data in order to enforce anonymity. The first technique (*7*) protects privacy by shifting trajectory points (that are already close to each other in time) in space. Clusters of k trajectories are enforced to be close to each other to fall in the same area of uncertainty given by a user parameter representing the GPS precision. The second work (*22*) presents a subsampling-based algorithm, i.e., privacy is preserved by removing some points s.t. uncertainty between consecutive points is increased to avoid identification. Due to the inherit uncertainty assumption of both works on trajectories, the privacy constraints enforced and the cost metric do not match with those used in this work.

In this work, we address the privacy concerns when publishing static trajectory databases by extending the concept of k-anonymity to trajectories. We model trajectories in a general way (sequences of spatio-temporal points) such that the *same techniques can be possibly used in other context* (sequence events, strings, non-euclidean spaces, etc.) without much effort.

To the best of our knowledge, this is the first work that extends the concept of relational k-anonymity to trajectories without relying on data distortion and uncertainty. We instead remove information from the data by making use of *space and time generalizations*, point alignment both in space and time, point and trajectory suppressions. The basic methodology does not rely on uncertainty (as was the case in previous works). The cost metric being used is statistically derived and captures time and space sensitivity to address various applications. Also no previous work seems to have measured the level of distortion due to anonymization in the context of trajectory mining applications, which we consider one of the ultimate purpose in trajectory publishing.

In systems where freshness of the data is crucial (e.g., healthcare data, stream data), release (and anonymization) of data needs to be on the fly. An important example is *authenticated LBS*, where authenticated users send streams of queries to a service provider, and a trusted anonymizer filters the communication by applying anonymization techniques. To the best of our knowledge, no work on authenticated LBS studied space-time generalization, although it is considered a state-of-the-art technique for non-authenticated LBS. Our work make the assumption that all the data is static. Adapting trajectory k-anonymization framework given in this paper for such online systems is no different than adapting conventional k-anonymization for dynamic databases. The latter is already studied by the literature (*38, 9*) and such an extension to the framework seems not to be challenging, and currently left for future work.

4. Problem Formulation

4.1. *Notation*

We assume the space is discretized into $\epsilon_s \times \epsilon_s$ size grids and a *point* in our domain is actually a grid. All space measurements are in units of ϵ_s. We assume *time* is also discretized into buckets of size ϵ_t and domain of time is finite. So datasets act as the snapshots of the world in many time instances. Datasets with continuous time and space domains can be fit into this assumption by the use of interpolations. The level of granularity in discretization does not affect the efficiency of the proposed methodology.

We define a trajectory database in an object-oriented way. A trajectory dataset T is a set of private entities or trajectories (e.g., $T = \{tr_1, \cdots, tr_n\}$, $|T| = n$). Each private entity tr_i is an ordered set of spatio-temporal 3D volumes (e.g., points) composed of time, x, and y dimensions (e.g., $tr_i = \{p_1, \cdots, p_m\}$ where $p_k =< t_k, x_k, y_k >$, $|tr_i| = m$). We assume that the t_i, x_i and y_i components are range of values defined as $t_i : [t_i^1 - t_i^2]$, $x_i : [x_i^1 - x_i^2]$ and $y_i : [y_i^1 - y_i^2]$. Each tr_i is ordered by their subtime component t_i^1. tr_is refer to the individuals and each triplet specifies the area location of the individual at some time in the corresponding time interval. We use the following notation for components to express their length; $|x_i| = |x_i^1 - x_i^2|, |y_i| = |y_i^1 - y_i^2|, |t_i| = |t_i^1 - t_i^2|$. We also use '$\cdot$' operator to refer to a specific component of a bigger set. (E.g., $tr_i.p_j$: jth point of the ith trajectory)

We say a trajectory tr_1 is a *subset* of another trajectory tr_2 and write $tr_1 \subset tr_2$ if for each point $p_i \in tr_2$, we have some unique $p_j \in tr_1$ such that $t_i^1 \leq t_j^1$, $t_i^2 \geq t_j^2$, $x_i^1 \leq x_j^1$, $x_i^2 \geq x_j^2$, $y_i^1 \leq y_j^1$, $y_i^2 \geq y_j^2$. We say a trajectory tr is atomic if $|x_i| = |y_i| = |t_i| = 1$ for every $p_i \in tr$. We use the notation BB_P for the 3D point with minimum volume that covers all points inside set P (E.g., minimum bounding box).

We also assume S is the universal space (the maximum area possible in the space domain), T is the universal time (the maximum time interval in the time domain), and U is the universal volume ($U = S \cdot T$).

4.2. *Problem Definition*

We assume that prior to release, the trajectory database is complete and static. No uniquely identifying information is released. However we assume that we have adversaries that may

(1) already know some portion of the trajectory of an individual in the dataset and may be interested in the rest. (e.g., adversary knows that a particular person lives in a particular house. He also knows that she leaves the house and comes back home at specified times. He is interested in finding the locations she visited.)

(2) already know the whole trajectory of an individual but be interested in some sensitive information about the individual. This is a concern if some sensitive info is also released, as part of the database, for some of the spatio-temporal triplets or for some individuals. Sensitive info, for example, could be

the requests done by the individual to location based services.

We protect privacy of the individuals against the above adversary by using the following techniques

- k-Anonymity: anonymize the dataset so that every trajectory is indistinguishable from $k-1$ other trajectories.
- Reconstruction: release atomic trajectories sampled randomly from the area covered by anonymized trajectories.

k-Anonymity limits the adversary's ability to link any information to an individual. Reconstruction further prevents leakage due to anonymization. Both techniques are discussed in Sections 5 and 6.

Since reconstruction is just sampling from anonymized data, expectation on the amount of privacy-utility depends only on the anonymization. As an anonymization is required to satisfy the privacy constraints, it also needs to maximize the utilization. An anonymization with a reconstruction that better explains the data is considered to be highly utilized. However the amount of utilization also depends on the target applications. Although there may be many classes of target applications, in this work, we consider two of them:

Time Sensitive Applications: This class covers the applications in which the time component is crucial compared to space components. Trajectories that have similar paths in space, but occur in different time periods are considered to be far away from each other. Such applications include mining traffic data to monitor traffic jams, anomaly detection when timely access control constraints are in place, etc.

Space Sensitive Applications: Similarities are calculated w.r.t. space. *Time shifted* trajectories or trajectories with different velocities can be considered to be close. Target applications include mining the world for region popularity to make business decisions, measuring road erosion caused by vehicles for maintenance, etc.

Section 6.2 discusses that some anonymization tr^* of tr minimizing the following equation (*log cost metric*[1]) also maximizes the probability of generating the exact dataset.

$$\begin{aligned} LCM(tr^*) = \sum_{p_i \in tr^*} [w_s(\log|x_i| + \log|y_i|) + w_t \log|t_i|] \\ +(|tr| - |tr^*|) \cdot (w_s \log S + w_t \log T) \end{aligned} \tag{1}$$

where w_s and w_t are weights to adjust sensitivity to space and time respectively.

From now on, our objective is to minimize Equation 1 while respecting k-anonymity in anonymizations. In later sections to ease the discussion, we assume, without loss of generality, $w_s = w_t = 1$ unless noted otherwise.

5. k-Anonymity in Trajectory Databases

In this section, we redefine the k-anonymity notion for sets of trajectories. Next, we use a condensation based approach to form groups of *similar* trajectories. Last, we show how to k-anonymize trajectories in a given group. Anonymization process will be dependent on the selection of metric parameters being used for grouping.

[1] We postpone the discussion on the reasoning behind using the log cost as a metric until Section 6.2

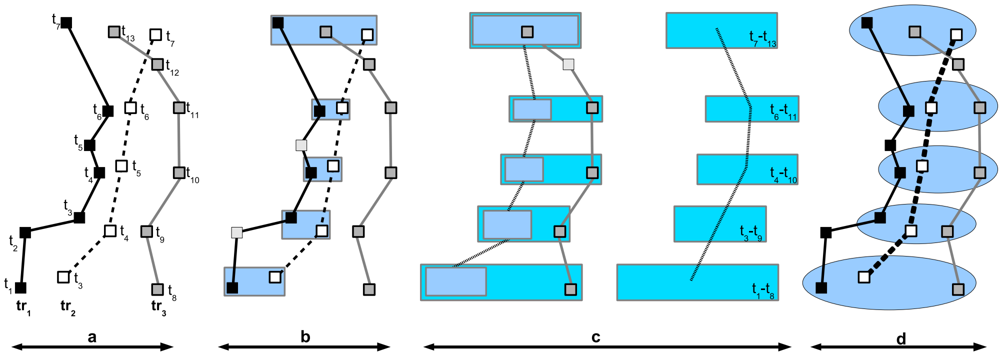


Figure 1. Anonymization Process
a. trajectories tr_1,tr_2, and tr_3; b. anonymization tr^* of tr_1 and tr_2; c. anonymization of tr^* and tr_3; d. point matching used in the anonymization of tr_1,tr_2, and tr_3. Matching contains five point links

5.1. *k-Anonymity for Trajectory Databases*

Original k-anonymity prevents an adversary from identifying a given QI to be in a set with less than k elements in the anonymized dataset. Since we assume adversaries know about all or some of the spatio-temporal points about an individual, the set of all points corresponding to a trajectory become the quasi identifiers in our domain. To enforce k-anonymity against such an adversary, we require the following property to hold in a given anonymization T^* of trajectory dataset T:

$$|\{tr^* \in T^* \mid tr \subset tr^*\}| \geq k \qquad \forall\ tr \in T$$

This implies that a given trajectory in the original dataset can at best be linked to at least k trajectories in the anonymized dataset. It can be shown easily that the following definition for k-anonymity satisfies the requirement and also preserves the truth of the original dataset:

Definition 5.1 *Trajectory k-Anonymity*: A trajectory database T^* is a k-anonymization of a trajectory dataset T if

- for every trajectory in T^*, there are at least $k-1$ other trajectories with exactly the same set of points.
- there is a one to one relation between the trajectories $tr \in T$ and trajectories $tr^* \in T^*$ such that $tr \subset tr^*$.

Following definitions show how to create anonymization of a set of trajectories.

Definition 5.2 *Point Link and Matching*: A point link between a set of trajectories $TR = \{tr_1, \cdots, tr_n\}$ is an ordered set of points $PL = \{p_1, \cdots, p_n\}$ such that $p_i \in tr_i$. An ordered set of point links between trajectories in TR, $PM = \{PL_1, \cdots, PL_m\}$, is a point matching between the trajectories if for all $i < j$ and all possible k, $PL_i.t^1_k < PL_j.t^1_k$.

Figure 1.d shows a point matching between trajectories tr_1, tr_2, and tr_3. Note that point links are ordered, do not overlap and there may be unmatched points in any of the trajectories.

Theorem 5.3: *Let $TR = \{tr_1, \cdots, tr_n\}$ be a set of trajectories and $PM = \{PL_1, \cdots, PL_m\}$ be a valid point matching between them. Let $TR^* = \{tr^*_1, \cdots, tr^*_n\}$ be another set such that $tr^*_1.p_i = \cdots = tr^*_n.p_i = BB_{PL_i}$. Then TR^* is an n-anonymization of TR.*

Proof: Since all the n elements in TR^* are the same, the first requirement of

Table 1. Optimal Alignment Optimizing Against Metric σ

$$OPT_\sigma(tr_1, tr_2) = \begin{cases} \sum\limits_{p_i \in tr_1} \sigma(p_i, \perp), & |tr_2| = 0; \\ \sum\limits_{p_i \in tr_2} \sigma(p_i, \perp), & |tr_1| = 0; \\ min\{OPT_\sigma(tr_1 - tr_1.p_1\ ,\ tr_2 - tr_2.p_1) + \sigma(tr_1.p_1, tr_2.p_1), & \\ OPT_\sigma(tr_1\ ,\ tr_2 - tr_2.p_1) + \sigma(tr_2.p_1, \perp), & \\ OPT_\sigma(tr_1 - tr_1.p_1\ ,\ tr_2) + \sigma(tr_1.p_1, \perp)\}, & |tr_1|, |tr_2| > 0. \end{cases}$$

anonymity trivially holds. Since each point in tr_j^* is a bounding box for some point in tr_j; $tr_j \subset tr_j^*$. The second requirement also holds. □

Figure 1.c shows the 3-anonymization of tr_1, tr_2, and tr_3 through the point matching in d. Unmatched points are suppressed in the anonymization.

Theorem 5.3 states that any matching between the points of a given set of trajectories can be used to anonymize the trajectories. Although there are many possible matchings, the aim of the anonymization is to find the one that will minimize the log cost of the output anonymization.

5.2. *Trajectory Grouping*

Although there are numerous k-anonymity algorithms proposed for single table datasets, a grouping based approach is discussed to be more suitable for the anonymization of complex structures, due to the direct identification of private entities (trajectories in our case) being anonymized (*34*). Most clustering algorithms can easily be modified for k-anonymity by enforcing that the size of the clusters should be more than k (*2, 33, 12, 1*). The only challenge at this stage is to define a distance metric between trajectories. Since our objective is to minimize the log cost metric, we can define the distance of two trajectories as the cost of their *optimal* anonymization. Having said that the problem reduces to finding the cost optimal anonymization given two trajectories.

Finding the optimal anonymization of two trajectories is the same as finding the point matching between the two trajectories such that anonymizing the trajectories through the matching minimizes the log cost. A similar *alignment* problem is well studied for strings (where the goal is to find an alignment of strings such that total pairwise edit distance between the strings is minimized) in the context of DNA comparisons. Alignment problem for two trajectories is polynomial and can be solved by using a dynamic programming approach. The equation that solves the alignment problem for optimizing against a given incremental function σ is given in Table 1. The log cost metric (LCM) is also incremental and defines σ as follows:

$$\sigma_{LCM}(p_1, p_2) = \begin{cases} \log U, & p_2 = \perp; \\ \log BB_{\{p_1, p_2\}}, & \text{otherwise.} \end{cases}$$

So the distance between two trajectories tr_1 and tr_2 is given by

$$DST(tr_1, tr_2) = OPT_{\sigma_{LCM}}(tr_1, tr_2)$$

In this work, we adapted and slightly modified the condensation based grouping algorithm given in (*1*) for trajectory k-anonymity. Algorithm *multi TGA* given in

Algorithm **1**, in each iteration, creates an empty group G, randomly samples one trajectory $tr \in TR$, puts tr into G, sets the group representative $rep_G = tr$. Next, the closest trajectory $tr' \in TR - G$ to rep_G is specified (line 6). tr' is added into G and group representative rep_G is updated as the anonymization of rep_G and tr' (line 8). Update of rep_G and G with new trajectories continues until G contains k trajectories. At the end of each iteration, a new group of k trajectories is formed, and is removed from TR. Trajectories in every group are anonymized with each other (details are in the next subsection.). Iteration stops when there are less than k trajectories remaining in TR.

The costly operation in the grouping algorithm is finding the closest trajectory to the group representative (line 6). This nearest neighbor operation needs to be done $|TR|$ times and it is difficult to speed up each operation by indexing. (This is because our distance metric does not satisfy triangular inequality.) To decrease the number of operations, we also try another version of algorithm **1** (*fast TGA*) by skipping the update of group representative (e.g., skipping of line 9). In this case, $k-1$ closest trajectories to the group representative can be found in one pass so the number of nearest neighbor operations will be $\frac{|TR|}{k}$. The resulting algorithm is faster by a factor of k but expected to have less utility since it does not directly optimize against log cost function. Experiments on the time/utility relations between fast and multi TGA algorithms are provided in Section 7.

Algorithm 1 multi & fast TGA(TR, k)

```
Require: Set of trajectories $TR$, integer $k > 1$, the log distance metric
Ensure: return $k$-anonymization of the trajectories in $TR$.
 1: repeat
 2:   Let $G$ be an empty group with group representative $rep_G$
 3:   Let $tr \in TR$ be a randomly selected trajectory.
 4:   $G = \{tr\}$, $rep_G = tr$.
 5:   repeat
 6:     Let $tr' \in TR - G$ be the closest trajectory to $rep_G$.
 7:     $G+ = tr'$,
 8:     if multi TGA then
 9:       $rep_G = anonTraj(rep_G, tr')$.
10:     end if
11:   until $|G| = k$
12:   $anonTraj(G)$
13:   $TR- = G$
14: until $|TR| < k$
15: Suppress remaining trajectories in $TR$.
```

5.3. *Anonymization of Trajectories*

Once the groups are formed, the trajectories inside each group needs to be anonymized. As mentioned before, the anonymization process needs to specify the optimal point matching that will minimize the log cost. Finding the optimal matching between two trajectories is easy. Algorithm specifies the point pairs between the trajectories by tracing $OPT_{\sigma_{LCM}}$ and anonymizes the paired points w.r.t. each other (by replacing the points with the minimum bounding box that covers the points). Any unmatched points are suppressed.

The real challenge is to find the optimal point matching between $n > 2$ trajectories. Similar versions of the problem on strings were proved to be NP-Hard (*24*).

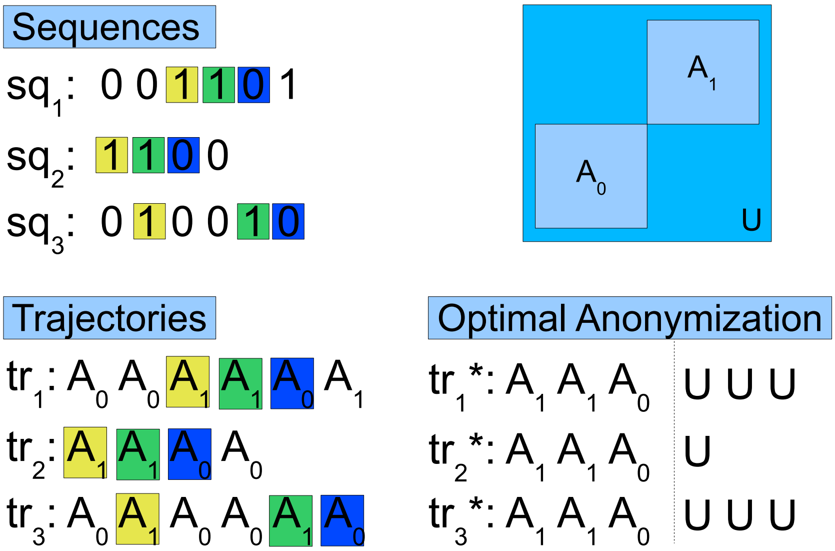


Figure 2. NP-Hardness Reduction Construction

Trajectory alignment and its complexity is not yet studied. Now, we formalize and prove the NP-Hardness of the *decision trajectory alignment* problem (DTA):

Definition 5.4 *DTA Problem*: Given a set of trajectories $TR = \{tr_1, \cdots, tr_n\}$ for arbitrary $n > 2$, is there a point matching PM between the trajectories in TR such that the log cost (with arbitrary weights w_s and w_t) of anonymizing TR through PM is at most c?(i.e., is $DTA(TR) \leq c$?)

Theorem 5.5 : *DTA problem is NP-Hard*

Proof : We first assume the log cost function has parameters $w_s = 1$,$w_t = 0$. We extend the proof for cost functions with arbitrary weight parameters at the end. We proof that DTA is NP-Hard by reducing from longest common subsequence problem (LCS) which is proved to be NP-Hard for a sequence alphabet of size 2 (*29*):

Definition 5.6 *LCS Problem*: Given an integer ℓ and a set of sequences $SQ = \{sq_1, \cdots, sq_n\}$ where each $sq_i = \{s_1, \cdots, s_m\}$ is an ordered set of strings from the alphabet $\sum = \{0, 1\}$; is there a common subsequence of sequences in SQ with length at least ℓ? (i.e., is $LCS(SQ) \geq \ell$?)

For an instance (ℓ, SQ) of LCS, we create the set of input trajectories $TR_{SQ} = \{tr_1, \cdots, tr_n\}$ for DTA, as follows: setting $|tr_i| = |sq_i|$

$$tr_i.p_j = \begin{cases} < [j \text{ - } j+1], [\text{0-1}], [\text{0-1}] >, & sq_i.s_j = 0; \\ < [j \text{ - } j+1], [\text{1-2}], [\text{1-2}] >, & sq_i.s_j = 1. \end{cases}$$

Figure 2 shows an example trajectory construction for a given set of sequences.

Lemma 5.7: *For a sequence* $SQ = \{sq_1, \cdots, sq_n\}$, $LCS(SQ) \geq \ell$ *if and only if* $DTA(TR_{SQ}) \leq (t - n \cdot \ell) \cdot \log 4$ *where* $t = \sum_i |tr_i|$

Proof: ($\overset{onlyif}{\leftarrow}$) Suppose $sq' = \{s'_1, \cdots, s'_\ell\}$ is one common subsequence, and let in^i_j returns the index of s'_i in sq_j. Observe that $PM = \{PL_1, \cdots, PL_\ell\}$ where $PL_i.p_j = tr_j.p_{in^i_j}$ is a valid point matching for TR_{SQ}. Since $sq_1.s_{in^i_1} = \cdots = sq_n.s_{in^i_n} = s'_i$; we have, using the notation $\overset{S}{=}$ as an equality operator for points having the same spatial components, $PL_i.p_1 \overset{S}{=} \cdots \overset{S}{=} PL_i.p_n$ for every $1 \leq \; i \leq \ell$. This implies that every point in a point link has the same spatial components. So anonymizing TR_{SQ} through PM will match ℓ space-similar points. The final anonymization will have a unit (1×1) area in ℓ positions. Assuming the worst anonymization (in this case, an area of 2×2) for the $t - n \cdot \ell$ points, we have a log cost at most $(t - n \cdot \ell) \log 4 + n \cdot \ell \log 1 = (t - n \cdot \ell) \log 4$.

($\overset{if}{\rightarrow}$) Let $PM = \{PL_1, \cdots, PL_r\}$ be the point matching resulting in at most $(t - n \cdot \ell) \log 4$ log cost. Let $PM^0 = \{PL_i \in PM \mid PL_i.p_1 \overset{S}{=} \cdots \overset{S}{=} PL_i.p_n\}$ and $PM^1 = PM - PM^0$. (PM^0 contains the point links that connect space similar points. Every link in PM^1 contains at least two spatially different points.) Since we have only two points in our domain, the points in PM^1 will add a log cost of the whole space (an area of 2×2). The same cost applies also for points unmatched (suppressed). However the points in PM^0 will have unit (1×1) area. Since the total number of points in PM^0 is $n|PM^0|$, we have;

$$\begin{aligned} LCM(TR^*_{SQ}) &\leq (t - n \cdot \ell) \log 4 \\ n|PM^0| \log 1 + (t - n|PM^0|) \log 4 &\leq (t - n \cdot \ell) \log 4 \\ (t - n|PM^0|) \log 4 &\leq (t - n \cdot \ell) \log 4 \\ |PM^0| &\geq \ell \end{aligned}$$

This means that we have a possible matching of size at least ℓ where the points linked to each other are space-similar. The reverse construction of the ($\overset{onlyif}{\leftarrow}$) proof states that such a matching implies a common subsequence of length at least ℓ. □

We ignored the effect of time component in the log cost function ($w_t = 0$) in the above construction. However, the proof can be modified to prove NP-Hardness of any fixed log cost function with any selection of weight parameters. The intuition is to prevent the effect of time component on finding the optimal matching. (The same matching needs to be optimal regardless of the value of w_t.) This can be done by adjusting the domains of space and time components such that increase in cost due to time generalizations will be negligible compared to the cost due to space generalizations. ($w_s \log S >> n \cdot w_t \log T$ where S and T are the universal space and time respectively.) □

Given the similar nature of the string and trajectory alignment problems, we adopted the string alignment heuristic given in (*19*) (where an upper bound on the total pairwise distance for the output alignment is guaranteed.) for trajectory alignment problem. Algorithm *anonTraj* given in Alg. **2** uses the following heuristic to come up with a possible alignment of points. Algorithm first identifies the trajectory tr_m whose total pairwise log cost distance with other trajectories is minimum and marks tr_m as done. At each step, $OPT_{\sigma_{LCM}}$ finds the optimal matching between the points of one unmarked trajectory tr_{new} and the current anonymization of the marked trajectories, and marks tr_{new}. Each matching creates links between the points. Point suppressions and generalizations are applied according to the matching. (Figure 1 shows an example anonymization of three trajectories.) In

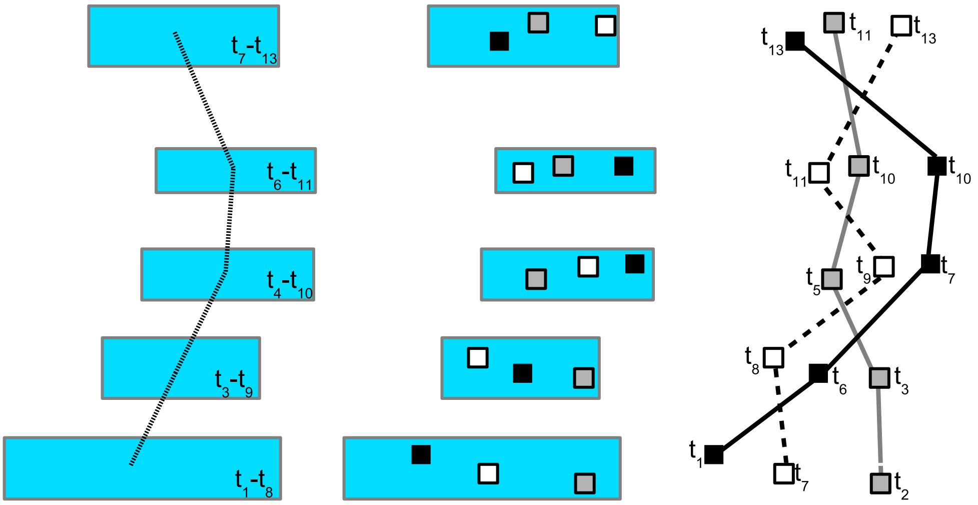


Figure 3. Reconstruction Process

later sections, we show experimentally that alignment heuristic works in practice.

```
Algorithm 2 anonTraj($G$)
Require: a (set) group of trajectories $G$.
Ensure: anonymize the trajectories inside $G$.
let $tr_m \in G$ be the trajectory whose total pairwise distance with other trajectories is minimum.
let set of trajectories $M$ contains initially $tr_m$.
repeat
    let $tr^*$ be the anonymization of trajectories in $M$ through linked points.
    let $tr_{new} \in G - M$ be a randomly chosen trajectory
    run $OPT_{\sigma_{LCM}}$ to find a min cost matching between the points in $tr_{new}$ and $tr^*$
    create links between the points matched by $OPT_{\sigma_{LCM}}$.
    suppress all unmatched points and all points directly or indirectly linked to unmatched points.
    $M = M + tr_{new}$
until $M = G$
for all unsuppressed point $p$ of each $tr \in M$ do
    let $PL$ be the point link containing $p$.
    $p = BB_{PL}$
end for
```

6. Randomized Reconstruction

6.1. *Reconstruction as a Privacy Method*

Trajectory anonymization techniques preserve the truth of the data while providing protection against certain adversaries. However the approach suffers from the following shortcomings.

Table 2. Reconstruction tr^R of tr^*

$$Pr(tr^R = tr') = \begin{cases} \dfrac{1}{\prod\limits_{p_i \in tr^*} (|x_i| \cdot |y_i| \cdot |t_i|)}, & \text{atomic } tr' \subset tr^*; \\ 0, & \text{otherwise.} \end{cases}$$

(1) Use of minimum bounding boxes in anonymization discloses uncontrolled information about exact locations of the points. (E.g., in the case of two trajectories, two non-adjacent corners give out the exact locations.) This information may be critical for applications where existence of a trajectory in a dataset is sensitive (e.g., δ-presence (*32*)).

(2) It is challenging to take full advantage of information contained in anonymizations. Most data mining and statistical applications work on atomic trajectories

The first problem can be weakened by applying some cloaking on the sides of the rectangle or by partitioning the space into grids and returning set of grids covering all points.

The second problem is more tricky as it is a common problem for heterogenous anonymizations with large output domain. (most clustering based anonymity algorithms suffer from the same problem.) One proposed technique to solve this issue is reconstruction (*33, 1*) where an atomic dataset is recreated from the anonymized dataset by uniformly selecting atomic points from anonymized regions. It is experimentally shown in (*33*) that reconstruction is sufficiently successful in learning from anonymized data.

In this work, we adapt the reconstruction approach as a means for privacy protection (as in (*1*)) and release reconstructed data rather than anonymized data. The intuition behind is that reconstruction not only serves as a solution to learn from the heterogeneous anonymized datasets but also greatly weakens the first problem without requiring a user input. We define the reconstruction tr^R of trajectory tr^* in Table 2.

An example reconstruction is shown in Figure 3. The output after reconstruction is atomic and suitable for any trajectory application.

6.2. *Maximizing Utility: The Log Cost Metric*

The success of the anonymization heavily depends on the success of the reconstructed data in explaining the original data. Since we have $tr \subset tr^*$ between original trajectory tr and its anonymization tr^*, the probability of generating the original trajectory is non-zero and given by the constant denominator in Table 2 case 1. A good anonymization would maximize this probability.

$$\begin{aligned} &\arg\max_{tr^*} \prod_{p_i \in tr^*} \frac{1}{|x_i|} \cdot \frac{1}{|y_i|} \cdot \frac{1}{|t_i|} \\ &= \arg\min_{tr^*}\left(\sum_{p_i \in tr^*} \log|x_i| + \log|y_i| + \log|t_i|\right) \qquad (2) \end{aligned}$$

The Equation 2 equally weights the effects of time and space on the reconstruction. This is not desirable if we have the class of target applications given in Section 5. So instead, we weight the log cost metric;

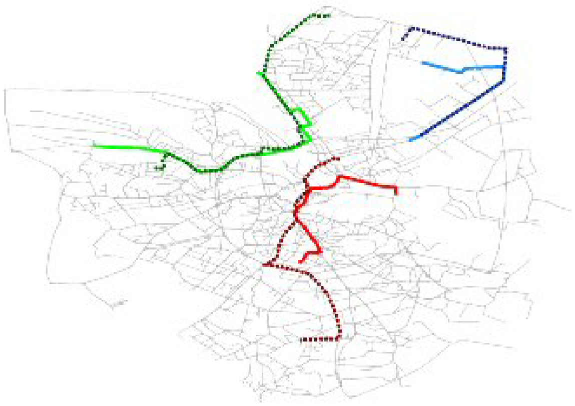


Figure 4. Map of the city with 3 groups each containing 2 trajectories

$$\sum_{p_i \in tr^*} w_s(\log|x_i| + \log|y_i|) + w_t \log|t_i| \tag{3}$$

Since a given anonymization tr^* of tr does not contain the points suppressed in tr, Equation 3 does not add any log cost regarding those suppressed points. However a suppressed point can be safely thought as a point covering the whole universal space. The final weighted log cost function is given by;

$$\begin{aligned} LCM(tr^*) = &\sum_{p_i \in tr^*} [w_s(\log|x_i| + \log|y_i|) + w_t \log|t_i|] \\ &+(|tr| - |tr^*|) \cdot (w_s \log S + w_t \log T) \end{aligned} \tag{4}$$

7. Experiments

We run a set of experiments on a trajectory dataset generated by using the state-of-the-art Brinkhoff generator[1]. It contains 1000 spatio-temporal trajectories with an average length of 70 points, for a total of 70118 spatio-temporal points. The spatial projection of the dataset is shown in Figure 4. For a qualitative understanding of the log distance behavior, we also show 3 randomly-chosen groups of trajectories obtained by using $k = 2$. Trajectories in the same group are clearly close in space and also similar in length (although not shown, also time intervals are similar.)

Experiments focus on (1) measuring the amount of utility preserved after anonymization and perturbation processes, and (2) time performance.

7.1. *Utility*

We compared the anonymized datasets (by varying k and the anonymization heuristics) against the original one, measuring how much different they are according to a number of metrics.

[1] `http://fh-oow.de/institute/iapg/personen/brinkhoff/generator/`

7.1.1. Number of Removed Points

The anonymization step allows suppression of points or trajectories, depending on the cost associated to suppression. We used a high cost for suppressions, but notice that since trajectories may have different lengths, suppression may be required to enforce k-anonymity. Figure 5 shows the results on two heuristics used in our experiments: *multi*, i.e. logdistance computed on multiple trajectories; and *fast*, where logdistance has been always computed only on trajectory pairs (see Section 5.3). As expected, the number of removed points generally increases with k. Notice that *multi* has a low distortion, with less than 9% of points removed even with $k = 25$. On the contrary, *fast* heuristic needs to remove nearly twice or three times the number of points removed by *multi*[2]

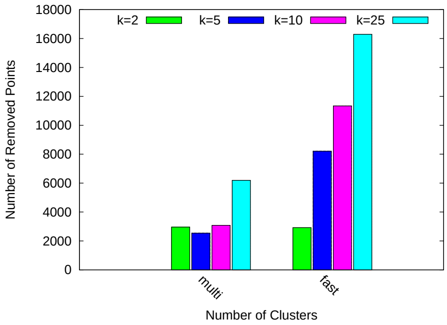


Figure 5. Points removed in the anonymized dataset

7.1.2. Distortion on Clustering

We also analyzed the utility of the anonymized datasets for mining purposes. We measured the deviation from the original clustering results,i.e., we compare clusters obtained from the original trajectory dataset (reference partition) against the clusters obtained from the sanitized dataset (response partition). For the evaluation, we used a bottom-up complete-link agglomerative clustering algorithm, coupled with the ERP distance metric (*10*), which has been specifically developed for trajectories.

As the algorithm requires to specify the number of clusters as input, we ranged from 2 to 60 clusters. Note that due to the large number of experiments and the final complexity of the clustering algorithm we used[1] the whole comparison process required days of computation.

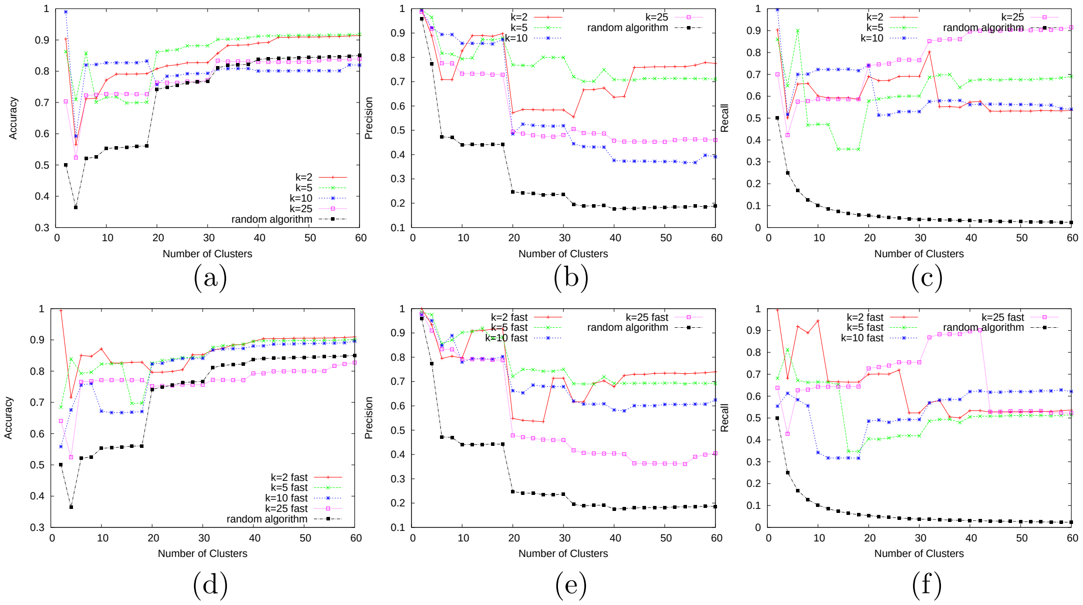
We used a standard approach to evaluate clusters. We considered every pair of trajectories and verified whether both are in the same cluster in the reference partition and whether they are in the response partition. We have therefore four case, namely: true positive (TP), true negative (TN), false positive (FP), false negative (FN). Then we computed the following standard measures:

- $accuracy = (TP + TN)/(TP + FN + FP + TN)$;
- $precision = TP/(TP + FP)$;
- and $recall = TP/(TP + FN)$.

Figure 6 shows the results computed from the sanitization datasets, by using different (prearranged) number of clusters. Figures 6(a,b,c) show the behavior of the *multi* heuristic, while on the second row Figures 6(d,e,f) show a similar behavior

[2] for $k = 2$ the two heuristics are equals, and the only small difference is due to the randomization in the reconstruction of trajectories

[1] Our hierarchical clustering implementation requires $O(n^3)$ distance computations (where n is the number of trajectories), and each ERP computation requires, by using dynamic programming, $O(l^2)$ (where l is the longest trajectory).



(a) (b) (c)

(d) (e) (f)

Figure 6. Distortion evaluation on clustering results.

for the *fast* heuristic. In this set of experiments, we notice therefore that for clustering purposes, the fast heuristic has nice behavior. In order to better understand the values of each measure used in the plots, we also show results of a "random algorithm", i.e. a randomly-selected reference partition of uniformly distributed clusters. For a reasonable number of clusters (e.g., up to 20) all the measures reported good results. We can also notice that smaller k's result in less distortion, although there is not a tight monotonicity due to the randomization steps.

7.2. *Time Performance*

In Figure 7, we show a plot on time performance. As we can see, execution time grows linearly with increasing k for *multi* algorithm, while *fast* is almost constant. Also notice that multi required almost 3 hours for $k = 25$; for datasets larger than 3K-4K trajectories, running time may be infeasible for *multi*, while *fast* scales well.

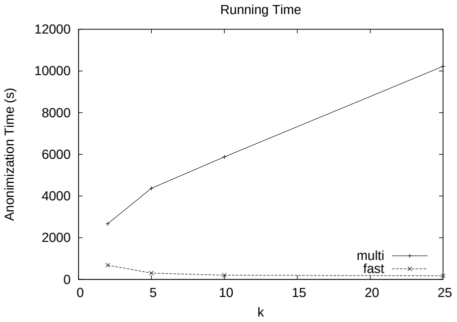


Figure 7. Time performance.

7.3. *Query Answering*

We stress that most common uses of static datasets are statistical analysis and data mining other than querying. However querying the anonymizations is still valuable

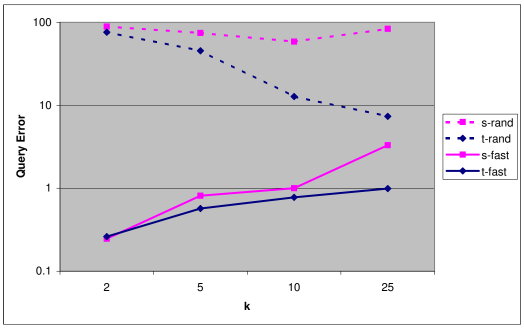


Figure 8. Query error over increasing k constraint

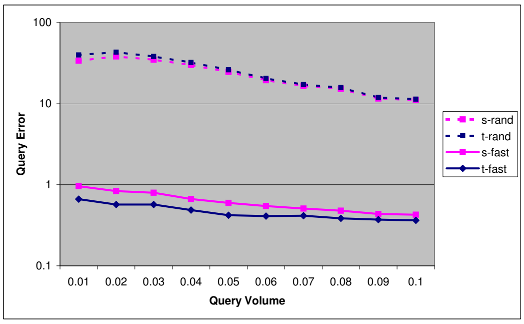


Figure 9. Query error over queries with increasing volume

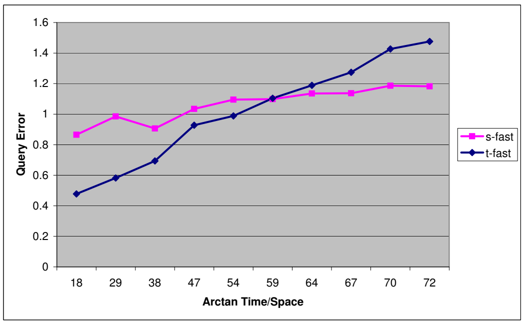


Figure 10. Query error over queries with increasing time component

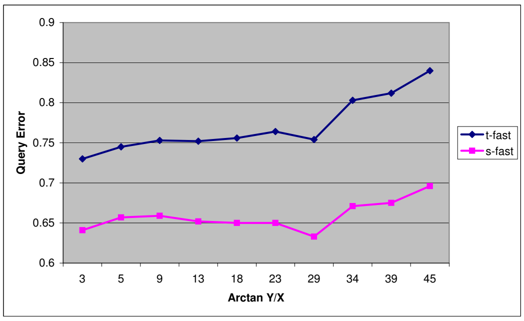


Figure 11. Query error over queries with increasing diagonal slope

to understand the behavior of the cost metric and the anonymization process. In this section, we make use of spatio-temporal queries in order to

- compare time and space sensitive anonymizations.
- and observe how anonymizations respond to queries of different shapes.

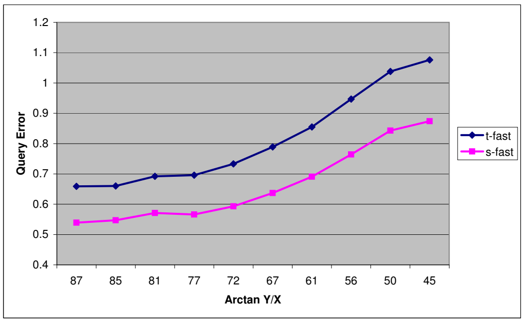


Figure 12. Query error over queries with decreasing diagonal slope

Our queries are basically, 3D rectangular volumes drawn inside the space of the spatio-temporal dataset. For an anonymization T^* of a dataset T, and a query volume Q, we are interested in two measures:

$Q(T)$: The number of trajectories passing through (having at least one point in) Q in T.

$Q(T^*)$: The expected number of trajectories passing through Q in a reconstruction T^R of T^*.

In a good anonymization, we would like the two measures be *close* to each other. Given this, we define the query error E_Q as

$$E_Q = \frac{|Q(T) - Q(T^*)|}{Q(T)}$$

By itself, E_Q is not very descriptive since it heavily depends on the volume and position of the query and the number of suppressed points. (Even a bad anonymization might give an E_Q close to 0, for a Q with large volume.) However it can be used to compare two anonymizations with similar number of suppressed points. Our aim is to measure the behavior of time and space sensitivity.

We first created 1000 queries of varying size, shape, and location; and measured an average E_Q value for both time-sensitive ($w_t = 1$) and space-sensitive ($w_t = 0$) anonymization, t-fast and s-fast respectively. (t-fast preserves time better, s-fast preserves space better.) We also created two totally random anonymizations, t-rand and s-rand, out of t-fast and s-fast by using the same number of points and trajectories. These random anonymizations serve as a lower bound on the utility of query answering. Figure 8 shows the average query error of t-fast, s-fast, t-rand, and s-rand for varying k. As desired, both fast algorithms has much lower error rates than their counter random versions. The difference decreases as k increases since anonymizations gets closer to randomization (and loses utility).

Figure 9 shows a similar scenario for $k = 5$ but this time for 1000 queries of varying volumes. The volumes are listed in the multiples of the whole space. This time error rate drops with larger queries since the $Q(T)$ becomes bigger. (Error for the largest possible query would be 0.) Figure 8 and 9 together show that t-fast is slightly better. The comparison is trustworthy since the number of suppressed points are similar for both algorithms.

Next, we fixed the volume to be 0.05 and k=5, and created 1000 queries for each different shapes. Figure 10 shows how anonymizations respond when we increase the length of the time component of the queries. (Horizontal axis lists arctan of the slope of the query diagonal with the space diagonal in celsius degrees.) A query with a low range time component would be very sensitive to distortion of time information. This means that in Figure 10, query sensitivity to time decreases along the horizontal axis. As expected for low time range queries, t-fast performs better. As the time range increases, s-fast outperforms t-fast. Even though it is not shown here, same behavior persists for queries of different volumes.

Next experiments shown in Figure 11 and 12 evaluates the behavior varying the shape of the query on space. We fixed the time component of the query to be the highest range possible (e.g., factor out the time dimension). We again fixed the volume to be 0.05. We created different shapes with different emphasis on x component and perform 1000 queries for each shape. Figure 11 and 12 shows the behavior of the algorithms when query shapes turn from rectangles to squares. (Horizantal axis shows arctan of the slope of the space diagonal.) As expected, s-fast has consistently less query error. We also observed that query errors increase for square queries. This is due to the log cost behavior. Given a rectangle (high on one axis) and a square both with a diagonal of the same size, a rectangle has a smaller volume, so statistically such anonymization is less costly. [1]

8. Discussion and Future Work

As we define and enforce k-anonymity for spatio-temporal databases, there are still issues not addressed explicitly in this work.

k-anonymity provides de-identification for individually identifiable data. However as mentioned before, when sensitive information is present, k-anonymity does not necessarily prevent the disclosure of the sensitive information. (As mentioned in item 2 of Section 4.2; for trajectory datasets, sensitive information could be the requests done by the individual to location based services.) This is mainly because k-anonymity does not enforce diversity on the sensitive info within each equality groups. Such issues have been addressed with alternative privacy definitions (*23, 26, 28*). Extension of this work for such privacy definitions is not challenging. Once we know how to group and anonymize trajectories, we can also enforce other anonymity constraints on the groups. For example, ℓ-diversity can be achieved:

- by applying a higher (or infinite) weight between the entities with similar sensitive values as stated in (*8*).
- by using a top-down hierarchical clustering approach (note that the methodology presented in this paper is independent of the clustering algorithm) and partition clusters only if diversity requirement is not violated to achieve ℓ-diversity (*25*).
- by simply suppressing those clusters violating the constraints. This approach has the advantage of being resistant to against minimality attacks (*40*).

However we leave the practical evaluation of enforcing other privacy definitions on trajectory databases as a future study.

When multiple k-anonymizations of the same private entities is released, a privacy attack known as intersection attack becomes possible (where two equality groups containing a specific individual is intersected to identify an individual). So releasing anonymizations of trajectories in a fixed region per period may be subject to such attack. However such an attack is possible only if the quasi-identifiers (and the sensitive attributes) do not change over time, and as for the trajectories, this is generally not the case. Designing intersection resistant k-anonymization is not a specific problem to trajectories but could be pursued as a future study.

[1] This may not be a desired property always. Even though such an approach would preserve statistical properties better, human mind tends to view world in an Euclidean space. This makes it difficult to use the log cost metric for visualization purposes.

9. Conclusions

We addressed privacy issues regarding the identification of individuals while sharing trajectory datasets. We redefined the notion of k-anonymity for sequences of spatio-temporal points, and further provided privacy by releasing only a randomly generated set of representative trajectories. A novel generalization-based approach, which exploits previous results on string alignment, has been successfully applied to trajectory data for the first time. We also propose an additional, simple reconstruction step for applications where generalized trajectories are not effective, but true k-anonymity must be provably preserved.

Experiments show that the log distance and the heuristics proposed are effective for trajectory dataset sharing.

References

[1] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In *EDBT'04*, pages 183–199, Heraklion, Crete, Greece, Mar. 14 2004.

[2] G. Agrawal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *PODS '06: Proc. of the 25th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 153–162, Chicago, IL, USA, June 26-28 2006.

[3] M. Atzori. Weak -anonymity: A low-distortion model for protecting privacy. In *ISC*, pages 60–71, 2006.

[4] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Blocking anonymity threats raised by frequent itemset mining. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 561–564, Washington, DC, USA, 2005. IEEE Computer Society.

[5] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Anonymity preserving pattern discovery. *The VLDB Journal The International Journal on Very Large Data Bases*, Nov. 2006.

[6] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Secure Data Management*, pages 185–199, 2005.

[7] F. Bonchi, O. Abul, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24nd International Conference on Data Engineering (ICDE '08)*, Cancun, Mexico, Apr. 7 2008.

[8] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *In Proc. of 12th Intl Conf. on Database Systems for Advanced Applications (DASFAA07)*, Apr. 2007.

[9] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li. Secure anonymization for incremental datasets. In *Third VLDB Workshop on Secure Data Management (SDM'06)*, Seoul, Korea, Sept. 18 2006.

[10] L. Chen and R. Ng. The marriage of lp-norms and edit distance, 2004.

[11] M. Diomo and S. Ayman. Potential use of gps data for calibrating travel demand models. In *10th National Conference on Transportation Planning for Small and Medium-Sized Communities*, Nashville Tennessee, USA, Sept. 13-15 2006.

[12] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min. Knowl. Discov.*,

11(2):195–212, 2005.

[13] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive*, pages 152–170, 2005.

[14] European Parliament. Directive 95/46/ec. http://www.cdt.org/privacy/eudirective/EU_Directive_.html, 1995.

[15] European Parliament. Regulation (ec) no 45/2001. http://ec.europa.eu/justice_home/fsj/privacy/docs/application/286_en.pdf, Dec. 18 2000.

[16] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *The 25th International Conference on Distributed Computing Systems (ICDCS'05)*, 2005.

[17] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services*, 2003.

[18] M. Gruteser and X. Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 02(2):28–34, 2004.

[19] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. In *Bull. Math. Biol.*, pages 141–154, 1993.

[20] Standard for privacy of individually identifiable health information. *Federal Register*, 66(40), Feb. 28 2001.

[21] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.

[22] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via density-aware path cloaking. In *ACM Conference on Computer and Communications Security (CCS)*, VA, USA, Oct. 29 2007.

[23] A. O. hrn and L. Ohno-Machado. Using boolean reasoning to anonymize databases. *Artificial Intelligence in Medicine*, 15(3):235–254, Mar. 1999.

[24] T. Jiang and L. Wang. On the complexity of multiple sequence alignment. *J. Computer Biologyy*, 1:337–348, 1994.

[25] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pages 25–35, Atlanta, GA, Apr. 3-7 2006.

[26] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 23nd International Conference on Data Engineering (ICDE '07)*, Istanbul, Turkey, Apr. 16-20 2007.

[27] D. Luper, D. Cameron, J. A. Miller, and H. R. Arabnia. Spatial and temporal target association through semantic analysis and gps data mining. In *The 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing (IKE'07)*, Las Vegas, USA, June 25-28 2007.

[28] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proc. of the 22nd IEEE Int'l Conf. on Data Engineering (ICDE 2006)*, Atlanta Georgia, Apr. 2006.

[29] D. Maier. The complexity of some problems on subsequences and supersequences. *J. ACM*, 25(2):322–336, 1978.

[30] T. McGhee. Gps technology tracks employees. http://www.denverpost.com/headlines/ci_4800440, 2006.

[31] MIT SENSEable City Lab. Real time rome. http://senseable.mit.edu/realtimerome/, 2006.

[32] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals

in shared databases. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, Beijing, China, June 11-14 2007.

[33] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. In *ICDEW '06: Proc. of the 22nd Int'l Conf. on Data Engineering Workshops*, page 96, Washington, DC, USA, 2006. IEEE Computer Society.

[34] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. In *Proceedings of the 23nd International Conference on Data Engineering (ICDE '07)*, Istanbul, Turkey, Apr. 16-20 2007.

[35] Norwich Union. Pay as you drive. http://www.norwichunion.com/pay-as-you-drive/, 2007.

[36] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[37] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[38] T. M. Truta and A. Campan. k-anonymization incremental maintenance and optimization techniques. In *ACM Symposium on Applied Computing (SAC2007)*, page 380 387, Seoul, Korea, 2007.

[39] US Department of Transportation. Measuring day-to-day variability in travel behavior using gps data. http://www.fhwa.dot.gov/ohim/gps/conclusion.html, 2006.

[40] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei. Minimality attack in privacy preserving data publishing. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 543–554. VLDB Endowment, 2007.