TRANSACTIONS ON DATA PRIVACY 2 (2009) 47–75

# Towards Trajectory Anonymization: a Generalization-Based Approach

**Mehmet Ercan Nergiz***, **Maurizio Atzori****, **Yücel Saygın***, **Barış Güç*****

*Sabanci University, Istanbul, Turkey.

**KDD Lab, ISTI-CNR, Pisa, Italy.

***Swiss Federal Institute of Technology, Zurich, Switzerland.

E-mail: ercann@sabanciuniv.edu, atzori@di.unipi.it, ysaygin@sabanciuniv.edu, baris@student.ethz.ch

**Abstract.** Trajectory datasets are becoming popular due to the massive usage of GPS and location-based services. In this paper, we address privacy issues regarding the identification of individuals in static trajectory datasets. We first adopt the notion of k-anonymity to trajectories and propose a novel generalization-based approach for anonymization of trajectories. We further show that releasing anonymized trajectories may still have some privacy leaks. Therefore we propose a randomization based reconstruction algorithm for releasing anonymized trajectory data and also present how the underlying techniques can be adapted to other anonymity standards. The experimental results on real and synthetic trajectory datasets show the effectiveness of the proposed techniques.

**Keywords.** Spatio-temporal data, k anonymity, privacy

## 1 Introduction

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 in the form of trajectories. Trajectories of individuals collected over months or years contain valuable information which can be harvested by data mining tools. Applications of data mining models obtained from trajectories include city traffic planning, and intelligent transportation [16, 21]. On the other hand, the trajectories of people contain many forms of sensitive information, therefore trajectories cannot be released for public use before they are properly anonymized. One may think that removing the personally identifying information from trajectories would be enough for anonymizing them. However, such a naive method does not work even for simple tabular data. This is due to the fact that the released database can

*This work was partially funded by the Information Society Technologies Programme of the European Commission, Future and Emerging Technologies under IST-014915 GeoPKDD project, and The Scientific and Technological Research Council of Turkey (TUBITAK) National Young Researchers Career Development Programme under grant number 106E116.

be linked to public databases through a set of common attributes which are called *quasi-identifiers*. For example, in [42], it has been shown that the combination of zip code, and birth date is unique for 87% of the citizens in US. This figure increases as more attributes are added to the combination. The problem of linkage becomes even more complicated in our highly connected world as the number, variety, and ubiquity of data sources increase.

In case of trajectory data, space and time attributes are very powerful quasi-identifiers which can be linked to various other types of data. Consider the trajectory of a worker who starts at a specific location every weekday in the morning and reaches another location in an hour. Even if there is no directly identifying information in this trajectory, 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 [42, 41], and $k$-anonymity was proposed as a standard for privacy over relational databases. We can summarize k-Anonymity as "safety in numbers" which 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 trajectory data sets, 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 for anonymization. In this paper, We first extend the notion of $k$-anonymity for trajectories and then describe a heuristic method for achieving $k$-anonymity of trajectories. We then propose a technique for publishing trajectories which is based on releasing a representative trajectory to further protect privacy. While $k$-anonymity provides indistinguishability between different entities, it does not necessarily protect against disclosure of sensitive information in some scenarios. This is the case especially when most members of a given indistinguishable group have common sensitive information. This issue has been addressed by several works [28, 30, 32] mostly by enforcing constraints on the groups formed by the anonymizer. In this paper, we take $k$-anonymity as the base and discuss how the proposed trajectory $k$-anonymity method can be extended to other standards for anonymity.

**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 reconstruction step to release only a representative trajectory instead of generalized trajectories. Empirical results on both synthetic and real data sets were presented in Section 7. Discussions on how the underlying methodology can be extended for other anonymity standards are in Section 8. Finally, in Section 9, we conclude the paper and outline future research directions.

## 2 Motivating 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 [40, 34, 45, 35, 31, 13].

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 have been used by companies to maximize employee efficiency [34], by governments to understand the infrastructure [45] and by research groups to observe human behavior [35, 31, 40].

We stress that the output of mining algorithms might fail to remove all individually identifying information In fact, work in [6, 5] shows that even simple statistics such as the highest 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 [17, 18, 25]. One of the most effective and recognized technique for de-identification is anonymization [25].

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 Log-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 Online's (AOL) release of massive amounts of log data. The data included queries done by those users in a three month period in 2006, as well as the search results clicked. Although there was no personally-identifiable data linked to these accounts, a number of attacks have been performed by intersecting queries and some domain knowledge.

In this paper we are focused on trajectories which are sequences of spatio-temporal points. Nevertheless, the generalization-based approach we are proposing can be easily adapted to different kinds 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 [39] 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 [41, 32, 4, 30, 37]. Simply removing uniquely identifying information (SSN, name) from the released data is not sufficient to prevent identification because partially identifying attributes called quasi-identifiers such as age, sex, and city can still be mapped to individuals by using external knowledge [42]. $k$-Anonymity is defined in [41], as a privacy standard to protect against identification of individuals in person specific datasets. A dataset is $k$-anonymous if each record over quasi-identifiers 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 via generalizations and suppressions preserves the truth of the data; explaining the data at a higher granularity.

Entities in trajectory datasets are more complex than those studied by classical $k$-anonymity approaches. Anonymization of complex entities was proposed in [39] 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 [39] 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 [32] authors also warn that, in each set of people with same values for the anonymized QI $\ell$-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 $\ell$-diversity for trajectory dataset in Section 8.

As done in previous work on LBS and trajectory privacy, we will not directly address $\ell$-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

A considerable amount of research has been conducted 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 [15, 26] used perturbation and obfuscation techniques to deidentify a given request or a location; they differ from our work in the privacy constraints they enforce. Anonymization based privacy protection was used in [19, 7, 22, 23, 36, 12]. In [23], anonymity was enforced on sensitive locations rather than user location points or trajectories. In [19, 22, 36, 12, 3], individual location points belonging to a user are assumed to be unlinked and points of the users are anonymized rather than the trajectories. In [7], anonymization process enforces points referring to same set of users to be anonymized together all the time. However their work assumes anonymization per request rather than whole trajectory anonymization and the heuristic to specify groups of users is restricted to a specific time frame. (Such an approach does not anonymize time.) In [20], location privacy is protected via cryptographic techniques based on the theoretical work on private information retrieval.

## 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 research [8, 27, 43] 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 [8] protects privacy by shifting trajectory points in space that are already close to each other in time. Clusters of $k$ trajectories are enforced to be close to each other so that they fall in the same area of uncertainty given by a user parameter representing the GPS precision. The second work [27] 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. Work in [43] limits the probability of disclosing the tail of the trajectories given the head of the trajectories. The proposed technique is limited since it is suppression based and the protection is one way.

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 another context such as 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 work). The cost metric we 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 to be one of the ultimate goals of 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 makes 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 [44, 10] and such an extension to the framework is not theoretically challenging. However supporting dynamic trajectory databases may introduce additional loss in utility. We leave the practical evaluation of such an extension as future work.

# 4 Problem Formulation

## 4.1 Preliminaries and 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 $\epsilon_s$. We assume *time* is also discretized into buckets of size $\epsilon_t$ 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 $t_i^1$. $tr_i$s 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., $tr_i.p_j$: $j$th point of the $i$th 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 ($U = S \cdot T$).

$k$-Anonymity property for single tables can be formally defined as;

**Definition 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.

## 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.

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

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

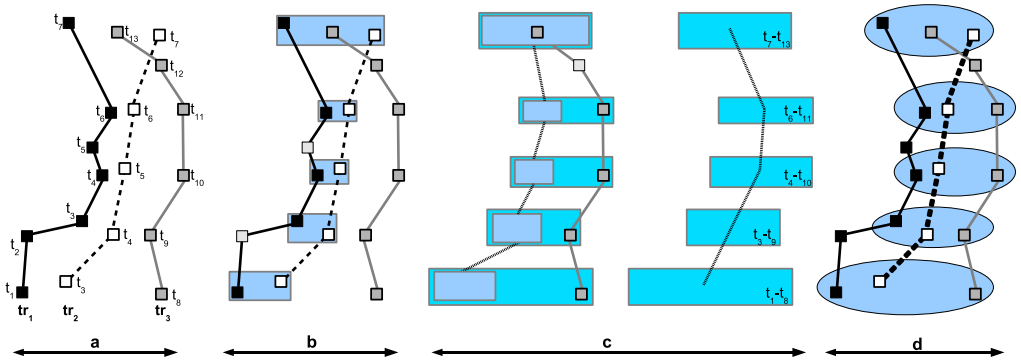



Figure 1: Anonymization Process
a. trajectories $tr_1$,$tr_2$, and $tr_3$; b. anonymization $tr^*$ of $tr_1$ and $tr_2$; c. anonymization of $tr^*$ and $tr_3$; d. point matching used in the anonymization of $tr_1$,$tr_2$, and $tr_3$. Matching contains five point links

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 Anonymization of Trajectories

In this section, we redefine the $k$-anonymity notion for sets of trajectories. Next, we propose a condensation based approach to form groups of *close* trajectories. Finally, we show how to $k$-anonymize trajectories in a given group.

## 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 can be used as the quasi identifiers in our application domain. $k$-anonymity requires 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 2** (*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.
- trajectories in $T$ and $T^*$ can be sorted in such a way that the $i$th trajectories $tr_i^* \in T^*$, $tr_i \subset tr^*$ satisfies $tr_i \subset tr_i^*$ for all $i$.

Following definitions are essential for the anonymization of a set of trajectories.

**Definition 3** (*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 $k$, $PL_i.t_k^1 < PL_j.t_k^1$.

**Example 4.** Figure 1.d shows a point matching between trajectories $tr_1$, $tr_2$, and $tr_3$. Note that point links are ordered, they do not overlap and there may be unmatched points in any of the trajectories.

**Theorem 5.** *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 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. □

**Example 6.** 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 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 resulting anonymization.

## 5.2 Trajectory Grouping

Although there are numerous $k$-anonymity algorithms proposed for single table datasets, a grouping based approach is shown to be more suitable for the anonymization of complex structures, due to the direct identification of private entities (trajectories in our case) being anonymized [39]. Most clustering algorithms can easily be modified for $k$-anonymity by enforcing that the size of the clusters should be more than $k$ [2, 38, 14, 1]. The only challenge at this stage is to define a distance metric for 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 of 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 $\sigma$ is given in Table 1. The log cost metric (LCM) is also incremental and defines $\sigma$ 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

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

Table 1: Optimal Alignment Optimizing Against Metric $\sigma$

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

The pseudocode to calculate the log cost distance between two trajectories is given in Algorithm 1. The output of the Algorithm 1 is the distance of the given two trajectories and the optimal point matching that minimizes the log cost.

**Algorithm 1** Dynamic programming algorithm that calculates the distance between two trajectories and returns a minimum cost point matching

**Require:** Trajectories $tr_1 = \{p_1, \cdots, p_m\}$, $tr_2 = \{p_1, \cdots, p_n\}$
**Ensure:** return the distance between $tr_1$ and $tr_2$ and the associated point matching $PM$.
1: $PM = \{\}$
2: Let $M$ be a $(m+1) \times (n+1)$ matrix.
3: $M[i][0] = i \cdot \log U$ for all $i \in [0 - m]$
4: $M[0][j] = j \cdot \log U$ for all $j \in [0 - n]$
5: $i = 1, j = 1$
6: **while** $i <= m$ **do**
7: **while** $j <= n$ **do**
8: $M[i][j] = min\{\ M[i-1][j-1] + \log BB_{tr_1.p_i, tr_2.p_j},\quad M[i][j-1] + \log U,$ $M[i-1][j] + \log U\}$
9: **if** $M[i][j] = M[i-1][j-1] + \log BB_{tr_1.p_i, tr_2.p_j}$ **then**
10: $PM+ = \{tr_1.p_i, tr_2.p_j\}$ //link $tr_i.p_i$ and $tr_2.p_j$
11: **end if**
12: $j+ = 1$
13: **end while**
14: $i+ = 1$
15: **end while**
16: Return the distance $M[m][n]$ and the point matching $PM$.

In this paper, we adopted and slightly modified the condensation based grouping algorithm given in [1] for trajectory $k$-anonymity. *multi TGA* given in Algorithm 2, 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, which is then removed from $TR$. Trajectories in every group are anonymized with each other (details are in Section 5.3.). 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 2 (*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 2** 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 Algorithm

Once the groups are formed, the trajectories inside each group need 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 proven to be NP-Hard [29]. 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 7** (*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 $DTA(TR) \leq c$?)

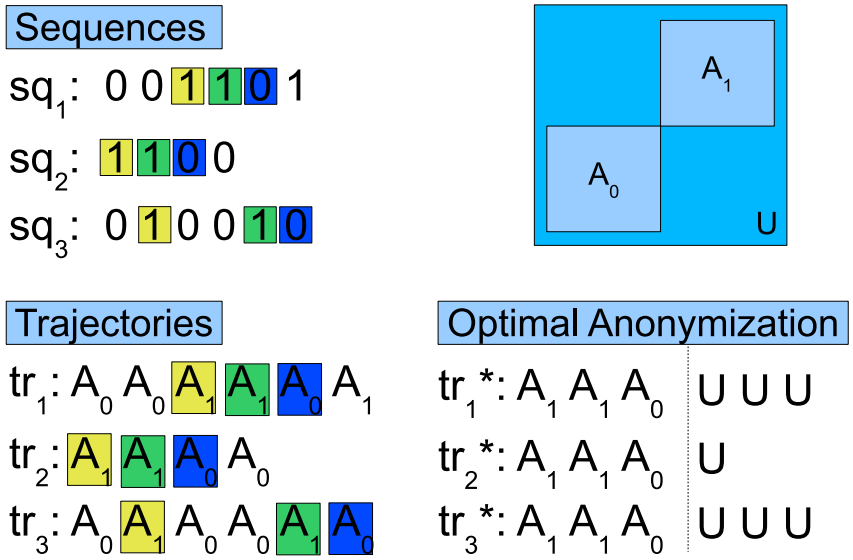



Figure 2: NP-Hardness Reduction Construction

We first assume the log cost function has parameters $w_s = 1$,$w_t = 0$. Extension of the proof for cost functions with arbitrary weight parameters will be discussed later. We prove that DTA is NP-Hard by reducing from the longest common subsequence problem (LCS) which is proven to be NP-Hard for a sequence alphabet of size 2 [33]:

**Definition 8** (*LCS Problem*)**.** Given an integer $\ell$ 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 $\ell$? (i.e., is $LCS(SQ) \geq \ell$?)

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

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

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

**Lemma 9.** *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.* ($\overset{only if}{\leftarrow}$) Suppose $sq' = \{s'_1, \cdots, s'_\ell\}$ is one common subsequence, and let $in^i_j$ return 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 $\stackrel{S}{=}$ as an equality operator for points having the same spatial components) $PL_i.p_1 \stackrel{S}{=} \cdots \stackrel{S}{=} PL_i.p_n$ 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 $\ell$ space-similar points. The final anonymization will have a unit ($1 \times 1$) area in $\ell$ positions. Assuming the worst anonymization (in this case, an area of $2 \times 2$) for the $t - n \cdot \ell$ points, we have a log cost at most $(t - n \cdot \ell) \log 4 + n \cdot \ell \log 1 = (t - n \cdot \ell) \log 4$.

($\stackrel{if}{\rightarrow}$) Let $PM = \{PL_1, \cdots, PL_r\}$ be the point matching resulting in at most $(t - n \cdot \ell) \log 4$ log cost. Let $PM^0 = \{PL_i \in PM \mid PL_i.p_1 \stackrel{S}{=} \cdots \stackrel{S}{=} PL_i.p_n\}$ 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 $2 \times 2$). The same cost applies also for points unmatched (suppressed). However the points in $PM^0$ will have unit ($1 \times 1$) 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 $\ell$ where the points linked to each other are space-similar. The reverse construction of the ($\stackrel{onlyif}{\leftarrow}$) proof states that such a matching implies a common subsequence of length at least $\ell$. □

**Theorem 10.** *DTA problem is NP-Hard.*

*Proof.* Theorem follows from the above construction when we ignore the effect of time component in the log cost function ($w_t = 0$). 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 [24] (where an upper bound on the total pairwise distance for the output alignment is guaranteed) for trajectory alignment problem. Algorithm *anonTraj* given in Algorithm 3 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 later sections, we show experimentally that alignment heuristic works in practice.

**Algorithm 3** anonTraj($G$)

**Require:** a (set) group of trajectories $G$.
**Ensure:** anonymize the trajectories inside $G$.
1: let $tr_m \in G$ be the trajectory whose total pairwise distance with other trajectories is minimum.
2: let set of trajectories $M$ contains initially $tr_m$.
3: **repeat**
4: let $tr^*$ be the anonymization of trajectories in $M$ through linked points.
5: let $tr_{new} \in G - M$ be a randomly chosen trajectory
6: run $OPT_{\sigma_{LCM}}$ to find a min cost matching between the points in $tr_{new}$ and $tr^*$
7: create links between the points matched by $OPT_{\sigma_{LCM}}$.
8: suppress all unmatched points and all points directly or indirectly linked to unmatched points.
9: $M = M + tr_{new}$
10: **until** $M = G$
11: **for all** unsuppressed point $p$ of each $tr \in M$ **do**
12: let $PL$ be the point link containing $p$.
13: $p = BB_{PL}$
14: **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.

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., an example of such applications is studied in [37]. The presence or absence of individuals in a given dataset is protected by bounding the existence probabilities.)

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 [38, 1] where an atomic dataset is recreated from the anonymized dataset by uniformly selecting atomic points from anonymized regions. It is experimentally shown in [38] 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

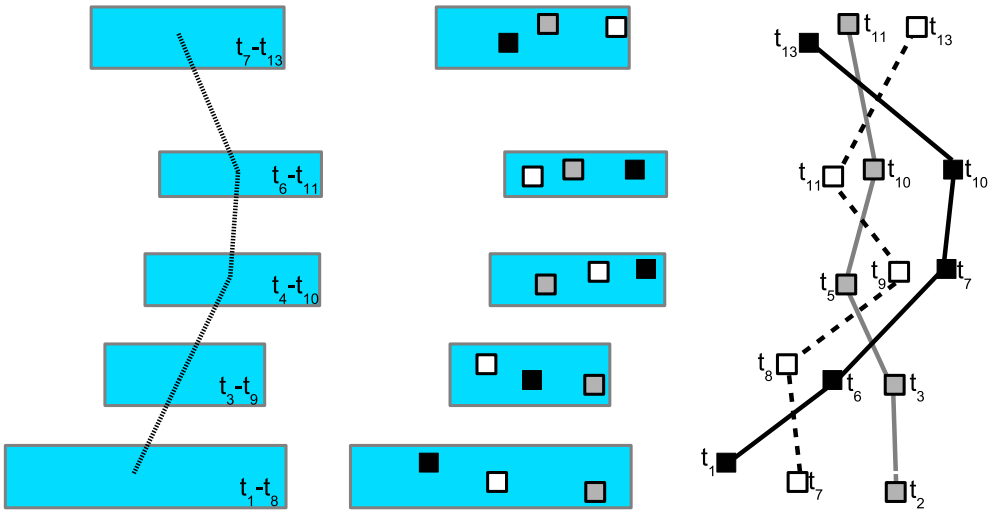



Figure 3: Reconstruction Process

Table 2: Reconstruction $tr^R$ of $tr^*$

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

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.

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

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;

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

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.[2] 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 implemented the proposed anonymization technique in Java. In order to assess the effectiveness of the proposed techniques, we tested them on both synthetic and real datasets.

*Real Dataset*: We used the GPS traces of taxis in Milano obtained in the context of the GeoPKDD project [16]. Dataset contains the traces collected over a month from GPS devices installed in taxis. We compiled 1000 trajectories from the real data set with a total of 98544 points. In Figure 4, we show the anonymization of two trajectories that are grouped by our algorithm for $k = 2$. This is an example of how log distance captures the similarities of trajectories in space.

*Synthetic Dataset*: We generated a synthetic dataset by using the state-of-the-art Brinkhoff generator[3]. 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 5. 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, (2) time performance, and (3) accuracy in query answering.

## 7.1 Utility

We compared the anonymized datasets (by varying $k$ and the anonymization heuristics) against the original ones and measured how different they are according to a number of metrics.

[2] In fact, any fixed number could be assigned as the penalty of suppression.

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

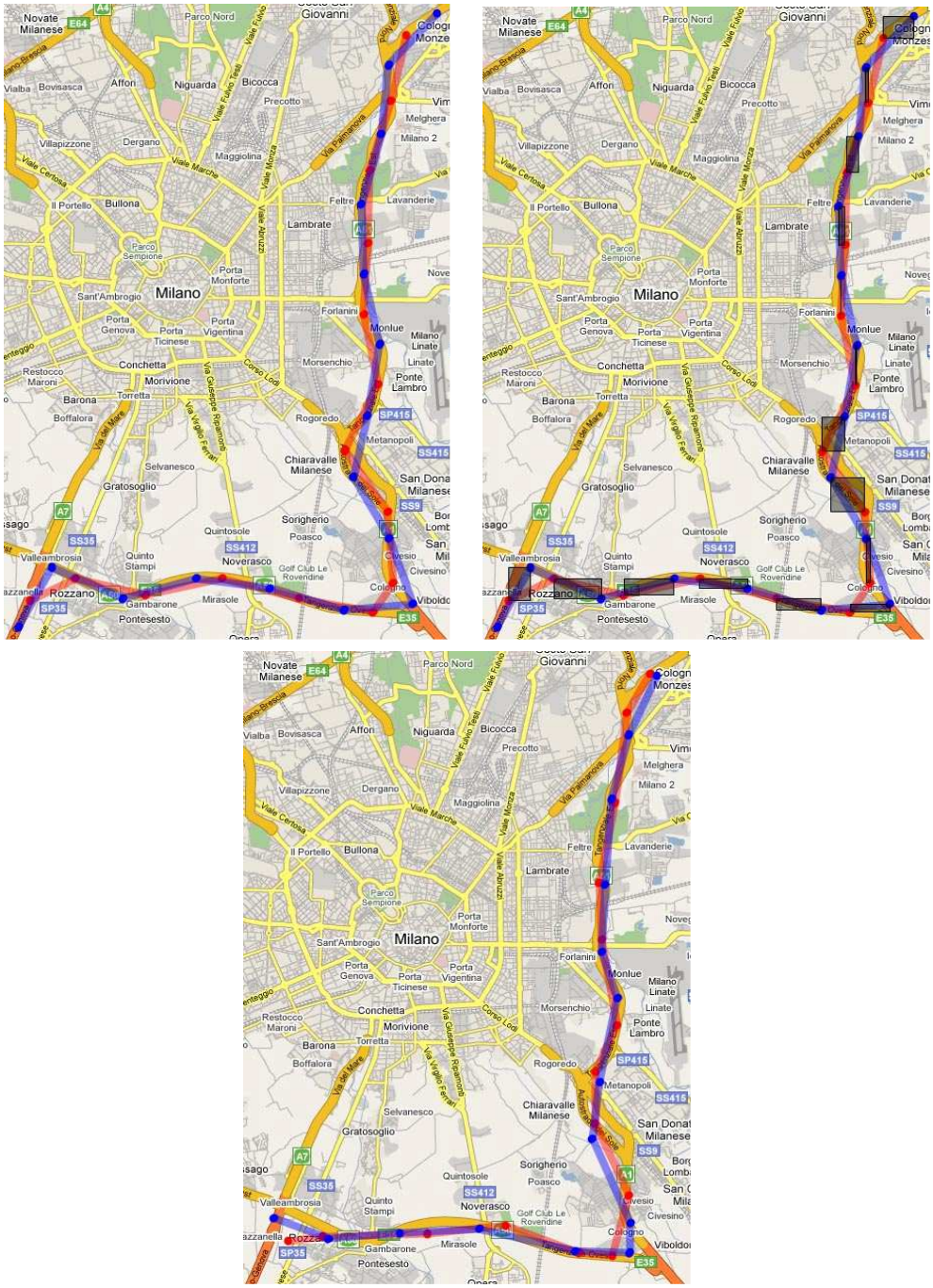



Figure 4: Original trajectories, anonymized trajectories, and reconstructed trajectories

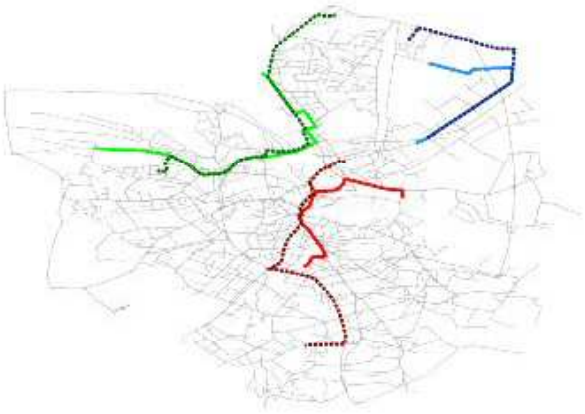


Figure 5: Map of the city with 3 groups each containing 2 trajectories

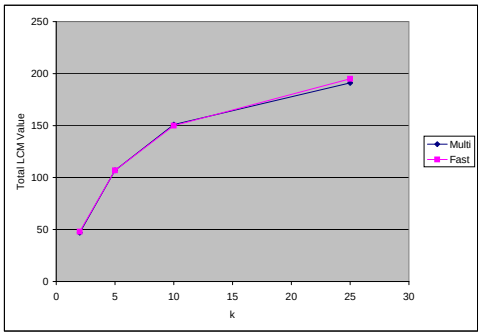
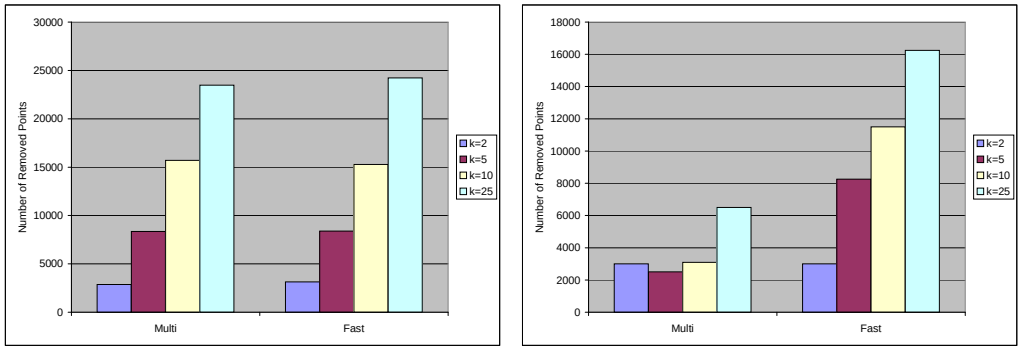



Figure 6: LCM for anonymizations - Milano Dataset




(a) Milano Dataset (b) Synthetic Dataset

Figure 7: Points removed in the anonymized dataset

### 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 7 shows the results on two heuristics used in our experiments: *multi*, i.e. log distance was computed on multiple trajectories; and *fast*, where log distance was computed only on trajectory pairs (see Section 5.3). As expected, the number of removed points generally increases with $k$ for both datasets. For Milano data, *multi* and *fast* have showed little difference w.r.t. no of suppressed points. But for synthetic dataset, *multi* has a low distortion, with less than $9\%$ of points removed even with $k = 25$ while *fast* heuristic needs to remove nearly twice or three times the number of points removed by *multi*. For $k = 2$ the two heuristics are equals, and the only small difference is due to the randomization in the reconstruction of trajectories.

### 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 [11], which has been specifically developed for trajectories.

As the algorithm requires to specify the number of clusters as input, we experimented with a range of 2 to 60 clusters. 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). Note that due to the large number of experiments and the complexity of the clustering algorithm we used 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 therefore have four cases, namely: true positive

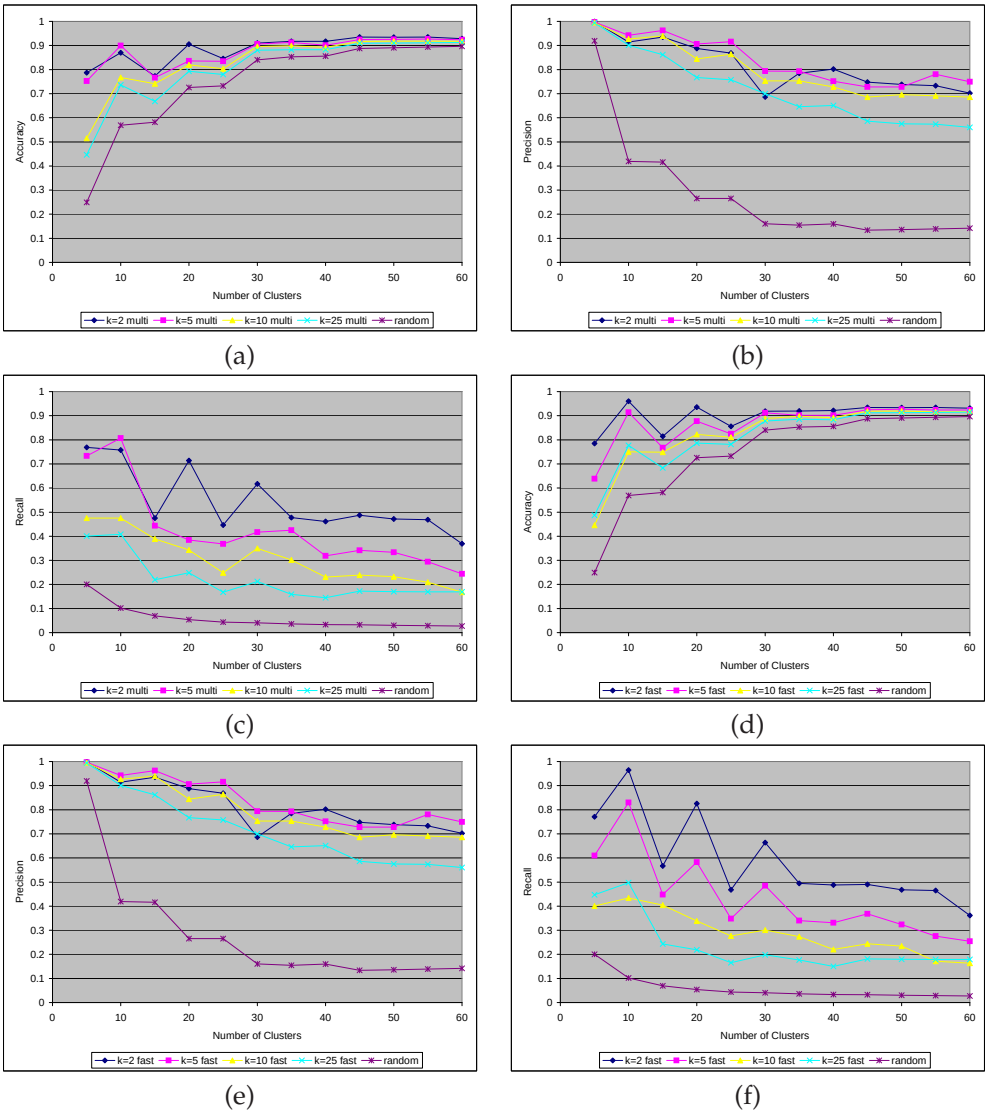



Figure 8: Clustering results - Milano Dataset

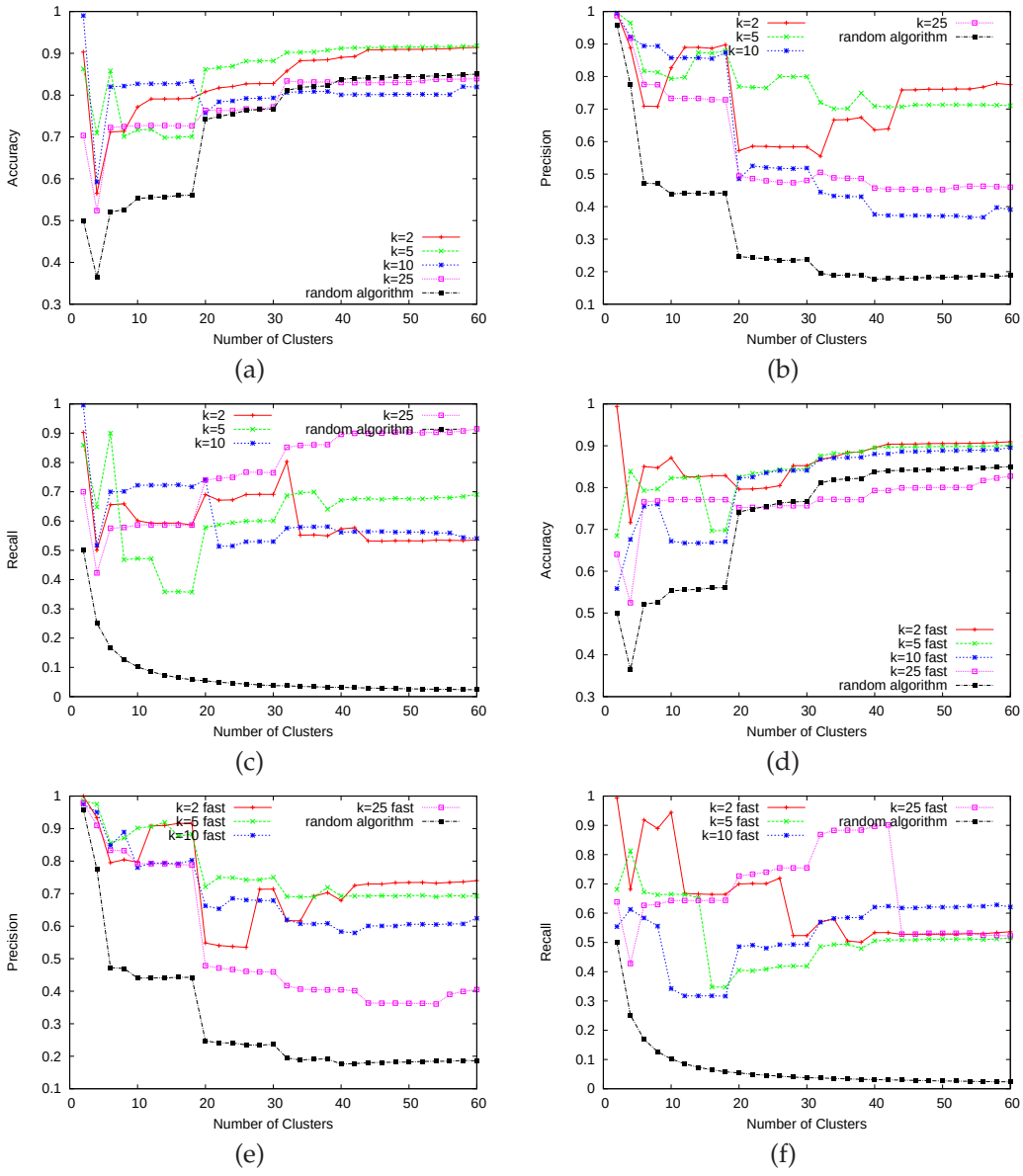
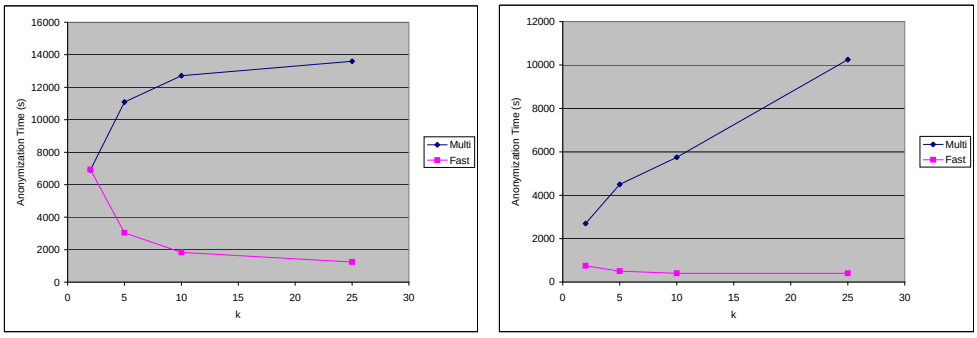



Figure 9: Clustering results - Synthetic Dataset




(a) Milano Dataset (b) Synthetic Dataset

Figure 10: Time performance

(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 8 and 9 show the results computed from the Milano and synthetic datasets, by using various number of clusters. Figures 9(a,b,c), 8(a,b,c) show the behavior of the *multi* heuristic, while on the second row Figures 9(d,e,f), 8(d,e,f) show a similar behavior for the *fast* heuristic. 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 for synthetic dataset and nearly all parameters tested for Milano dataset) all the measures reported high clustering performance. and both *fast* and *multi* resulted in similar levels of distortion. We also notice that smaller $k$ values result in less distortion, although there is not a tight monotonicity due to the randomization steps.

## 7.2 Time Performance

In Figure 10, we show results on time performance. As we can see for both datasets, execution time grows almost linearly with increasing $k$ for *multi* algorithm, while execution time for *fast* is decreasing. As mentioned in Section 5.2, this is because *multi* performs $k$ nearest neighbor queries per group while *fast* requires only one. 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. Recall that for Milano dataset, *fast* and *multi* creates anonymizations of similar utility. This justifies the adequacy of *fast* heuristic.

## 7.3 Query Answering

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

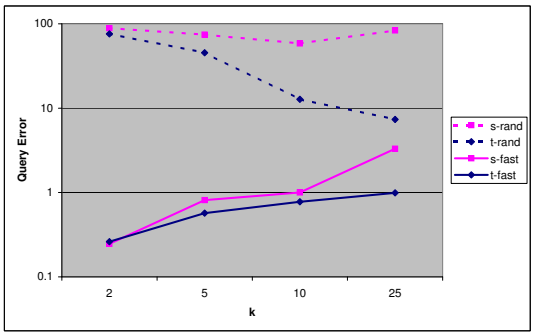



Figure 11: Query error over increasing $k$ constraint

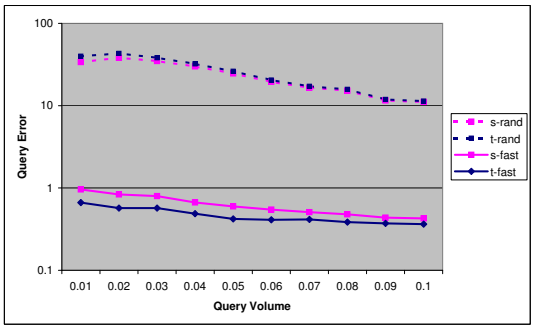



Figure 12: Query error over queries with increasing volume

We will be using our synthetic dataset, since we observed for Milano dataset that space and time distances per pairs of points correlate with each other (e.g., points are close in time iff they are also close in space).

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,

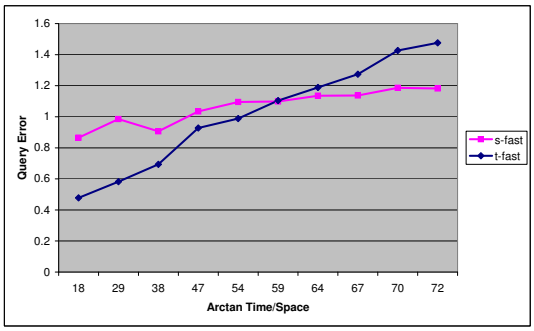



Figure 13: Query error over queries with increasing time component

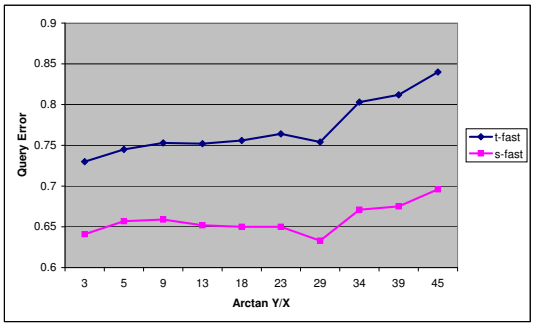



Figure 14: Query error over queries with increasing diagonal slope

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 11 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 12 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 11 and 12 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 13 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 13, 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.

Additional experiments shown in Figure 14 and 15 evaluate the effect of varying the shape of the query in space. We fixed the time component of the query to be the highest range

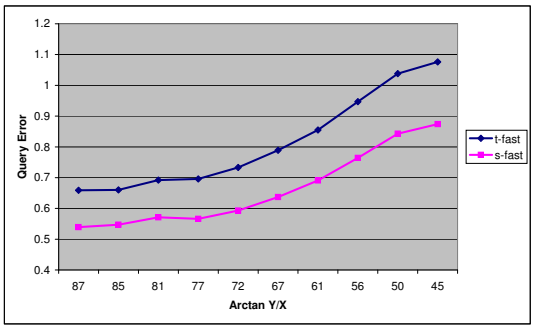



Figure 15: Query error over queries with decreasing diagonal slope

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 14 and 15 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. [4]

# 8 Extension to Other Anonymity Standards

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 [28, 30, 32]. Such extensions on $k$-anonymity are generally independent of the anonymization process; the inherit grouping mechanisms are modified to enforce additional constraints. As mentioned in the paper, trajectory anonymization can work with any clustering based grouping mechanism, thus any extension working on clustering based $k$-anonymity also works in the domain of trajectories. (This is also true for anonymization of dynamic databases [44, 10].) For example, it was stated in [39] that $\ell$-diversity[5] can be achieved:

- by applying a higher (or infinite) weight between the entities with similar sensitive values as stated in [9].

[4] 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 the log cost metric difficult to use for visualization purposes.

[5] A set of sensitive values is $\ell$-diverse if the entropy of the set is more than $\log(\ell)$. An anonymization satisfies $\ell$-diversity if the sets of sensitive attributes within each equality group are $\ell$-diverse

Table 3: Constraint Enforcing TGA

**Require:** Same as in TGA (Algorithm 2)
**Ensure:** return an anonymization of the trajectories in $TR$ where each group satisfies the given constraint.
1: run TGA with $k = 2$. Let $C^-$ and $C^+$ be the set of clusters where the constraint is violated and not violated respectively.
2: **repeat**
3: let $c \in C^-$ be a cluster
4: let $c_{\texttt{closest}} \in C^-$ be the closest cluster to $c$
5: merge $c_{\texttt{closest}}$ and $c$ into $c_{\texttt{merged}}$.
6: **if** $c_{\texttt{merged}}$ satisfies the constraint **then**
7: Put $c_{\texttt{merged}}$ in $C^+$.
8: **else**
9: Put $c_{\texttt{merged}}$ in $C^-$.
10: **end if**
11: $C^- = C^- - \{c, c_{\texttt{closest}}\}$
12: **until** $|C^-| \leq 1$
13: anonymize trajectories in each cluster of $C^+$ w.r.t. each other.
14: Suppress remaining trajectories in $TR$.

- by using a bottom-up [top-down] hierarchical clustering approach (note that the methodology presented in this paper is independent of the clustering algorithm) and merge [partition] clusters until [only if] diversity requirement is not violated, or
- by simply suppressing those clusters violating the constraints. This approach has the advantage of being resistant against minimality attacks [46].

We present an example bottom-up algorithm in Table 3 which uses the same methodology in [39]. Any constraint on the equality groups can be enforced by using the given algorithm. Such constraints include;

- constraints on the distribution of sensitive attributes: Such constraints are enforced by many $k$-anonymity extensions on micro data. [28, 30, 32, 37].
- constraints on the size of the bounding boxes: If $k$ cars have the same spatio-temporal points, they are likely to be grouped together, and no generalization would be applied. This is not always desired especially when location information is sensitive.
- constraints on the coverage of the bounding boxes: For trajectories that traverse over a mapping, enforcing constraints on the size of the bounding boxes will not be sufficient. Constraints should be enforced over the number of sensitive nodes that are being covered by bounding boxes.

# 9 Conclusions and Future Work

In this paper, we address privacy issues regarding the identification of individuals while sharing trajectory datasets. We redefine the notion of k-anonymity for sequences of spatio-temporal points, and further enforce privacy by releasing only a randomly generated set

of representative trajectories. To the best of our knowledge, this is the first generalization based approach for trajectory anonymization which exploits previous results on string alignment. In order to further protect privacy against boundary-based attacks, we also propose an additional, simple reconstruction step. Experiments on real and synthetic data show that the log distance and the heuristics proposed are effective for publishing trajectory datasets.

We also show how the techniques given in this paper can be adapted to many other anonymity standards. However we leave the practical evaluation of enforcing such standards to trajectories as a future study.

When multiple $k$-anonymizations of the same private entities are released, a privacy attack known as intersection attack becomes possible (where two equality groups containing a specific individual are intersected to identify an individual). So releasing anonymizations of trajectories in a fixed region per period may be subject to this type of 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 specific to trajectories but could be pursued as a future study in general.

## References


[1] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In *EDBT'04: 9th International Conference on Extending Database Technology*, 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: Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 153–162, Chicago, IL, USA, June 26-28 2006.

[3] V. S. V. Aris Gkoulalas-Divanis. A free terrain model for trajectory k-anonymity. In *DEXA'08: 19th International Conference on Database and Expert Systems Applications*, pages 49–56, 2008.

[4] M. Atzori. Weak k-anonymity: A low-distortion model for protecting privacy. In S. K. Katsikas, J. Lopez, M. Backes, S. Gritzalis, and B. Preneel, editors, *ISC*, volume 4176 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2006.

[5] 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.

[6] 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.

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

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

[9] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *DASFAA07: The 12th International Conference on Database Systems for Advanced Applications*, Apr. 2007.

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

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

[12] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *6th Workshop Privacy Enhancing Technology Workshop*, pages 393–412. Springer, 2006.

[13] 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.

[14] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[15] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *The International Conference on Pervasive Computing*, pages 152–170, 2005.

[16] EU FP6. Geographic privacy-aware knowledge discovery and delivery (GEOPKDD). `http://www.geopkdd.eu`, 2005.

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

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

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

[20] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: Anonymizers are not necessary. In *SIGMOD'08: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 121–132, New York, NY, USA, 2008. ACM.

[21] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy*. 2008.

[22] 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.

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

[24] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. In *Bulletin of Mathematical Biology.*, pages 141–154, 1993.

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

[26] 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*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.

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

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

[29] T. Jiang and L. Wang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1:337–348, 1994.

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

[31] D. Luper, D. Cameron, J. A. Miller, and H. R. Arabnia. Spatial and temporal target association through semantic analysis and GPS data mining. In *IKE'07: The 2007 World Congress in Computer*

*Science, Computer Engineering, & Applied Computing*, Las Vegas, USA, June 25-28 2007.

[32] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. $\ell$-diversity: Privacy beyond $k$-anonymity. In *ICDE'06: Proceedings of the 22nd IEEE International Conference on Data Engineering*, Atlanta Georgia, Apr. 2006.

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

[34] T. McGhee. GPS technology tracks employees. `http://www.denverpost.com/headlines/ci_4800440`, 2006.

[35] MIT SENSEable City Lab. Real time Rome. `http://senseable.mit.edu/realtimerome/`, 2006.

[36] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new CASPER: Query processing for location services without compromising privacy. In *VLDB'06: Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774. VLDB Endowment, 2006.

[37] 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.

[38] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. *Data and Knowledge Engineering*, 63(3):622–645, Dec. 2007.

[39] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. *IEEE Transactions on Knowledge and Data Engineering*, 99(1), 5555.

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

[41] P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, Nov./Dec. 2001.

[42] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness Knowledge-Based Systems*, 10(5):557–570, 2002.

[43] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *MDM'08: Proceedings of the The Ninth International Conference on Mobile Data Management*, pages 65–72, Washington, DC, USA, 2008. IEEE Computer Society.

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

[45] 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.

[46] 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.