Privacy in spatio-temporal databases: A microaggregation-based approach

Rolando Trujillo-Rasua[1] and Josep Domingo-Ferrer[2]

[1] Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg, Luxembourg
rolando.trujillo@uni.lu

[2] Department of Computer Engineering and Mathematics
Universitat Rovira i Virgili, Catalonia
josep.domingo@urv.cat

Abstract. Technologies able to track moving objects such as GPS, GSM, and RFID, have been well-adopted worldwide since the end of the 20th century. As a result, companies and governments manage and control huge spatio-temporal databases, whose publication could lead to previously unknown knowledge such as human behavior patterns or new road traffic trends (*e.g.,* through Data Mining.) Aimed at properly balancing data utility with users' privacy rights, several microaggregation-based methods for publishing movement data have been proposed. These methods are reviewed in this book chapter. We highlight challenges in the three stages of the microaggregation process, namely clustering, obfuscation, and privacy and utility evaluation. We also address some of these challenges by presenting yet another microaggregation-based method for privacy-preserving publication of spatio-temporal databases.

1 Introduction

The already mature establishment of telecommunication and wireless technologies has boosted the collection of spatio-temporal data at a large scale. To fully exploit the analytical usefulness of these data, they eventually need to be released to researchers and/or analysts. By doing so, useful knowledge can be acquired and applied to, for example, intelligent transportation, traffic monitoring, urban and road planning, etc.

However, spatio-temporal data in the form of trajectories of individuals are likely to contain sensitive information that users expect to keep private. Consequently, publishing or outsourcing databases of trajectories should properly balance data utility against the users' privacy rights.

While data utility preservation solely depends on the data, privacy protection needs to consider, in addition, the potential of the adversary. The adversary's capability is normally defined as background knowledge learned from other public sources of information (*e.g.,* census data or social networks). Knowing the times at which an individual visited a few locations can help an adversary to identify the individual's trajectory in the published database, and therefore learn the individual's other locations at other times. All this makes simple de-identification realized by removing identifying attributes a naive protection mechanism. Hence, more sophisticated privacy-preserving techniques ought to be considered.

Contributions. In this book chapter we review the literature on microaggregation-based methods for privacy-preserving trajectory data publication. In particular, we focus on similarity measures for clustering trajectories and privacy models based on k-anonymity. Among those privacy models, we concentrate in (k, δ)-anonymity [5, 6] and prove that it does not preserve privacy in the sense of k-anonymity for $\delta > 0$. We also present a distance between trajectories able to compare trajectories that are not defined over the same time span. Based on this distance, a microaggregation-based approach that preserves original locations (*i.e,* it contains no fake, perturbed or generalized locations) is proposed and empirically evaluated by using a real-life dataset.

Organization. Section 2 reviews the k-anonymity concept applied to the trajectory anonymization problem and describes expected properties of the similarity measure used for microaggregation. A flaw in the (k, δ)-anonymity concept is shown in Section 3. Our method and distance between trajectories are presented in Section 4, which are empirically evaluated in Section 5. Section 6 summarizes and concludes the book chapter.

2 Related work

Samarati and Sweeney [1] proposed in 1998 a novel privacy model named k-anonymity. k-Anonymity is based on the concept of *quasi-identifiers*, which are defined as any set of attributes that can potentially appear in publicly available datasets that contain identifiers. A database is said to satisfy k-anonymity if each combination of values of quasi-identifier attributes is shared by at least k records. Therefore, k-anonymity ensures that an adversary (even provided with background knowledge) cannot pinpoint the identity behind a record with probability higher than $1/k$.

A popular and effective technique to achieve k-anonymity is microaggregation [2]. The microaggregation technique works in two stages:

1. *Clustering.* The original records are partitioned into clusters based on some similarity measure. Each cluster contains at least k records and typically no more than $2k-1$ [3].
2. *Obfuscation.* Each cluster is anonymized individually by obfuscation. The obfuscation may be based on an aggregation operator like the average or the median, or can also be achieved by replacing the records in the cluster with synthetic or partially synthetic data.

Microaggregation was proposed for location k-anonymity in location-based services [4], but achieving k-anonymity using microaggregation in spatio-temporal data is not straightforward. In a trajectory, any location can be regarded as a quasi-identifier attribute [5]. In this case, k-anonymity would require each anonymized trajectory to be equal to, at least, $k-1$ other anonymized trajectories. This undoubtedly causes a huge information loss.

To overcome this issue, several trajectory similarity measures and ad-hoc privacy models based on k-anonymity have been proposed [9, 12, 13, 5–8, 11]. Both aspects of the microaggregation process are discussed in detail next.

2.1 Distances between trajectories

In microaggregation, selecting the *best* distance is of paramount importance. However, what does *best* mean in the context of spatio-temporal data publication could have different, and sometimes contradictory, answers. For instance, some applications (*e.g.,* urban traffic monitoring) might need precise temporal information, while others (*e.g.,* evaluation of attractiveness of touristic places) can make do with coarse-grained temporal data. We thus list next a few desirable properties of a distance measure for trajectories.

Uncertain sampling rate: Trajectories can be recorded at different sampling rates either due to performance issues or technology singularity. The difference in the sampling rate, which typically leads to differences in the size of the trajectories, should have no effect on the result of the distance measure. Neither the Euclidean-based distances used in [5, 7, 8] nor the EDR or the Log-cost distances adopted in [6] and [9], respectively, meet this property.

Noise resiliency: Several outlier detection mechanisms for spatio-temporal data exist. However, subtle differences might appear when comparing two trajectories, which could be regarded as a kind of "noise", but definitely not as outliers. See Figure 1 for an example. There, two identical (except in one location) trajectories are shown. However, distance measures like the Fréchet distance [10], do not deal well with this scenario. Others, such as the EDR distance, have mechanisms to ignore this "noise" and would consider both trajectories to be equal.

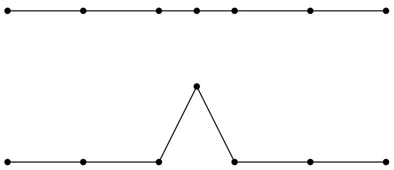

Fig. 1. Two trajectories that are equal except in the peak. They are represented in different planes for visualization purpose only.

Shape preservation: The *flow* of the two curves (trajectories) need also to be taken into account. In other words, a trajectory should not be treated as a set of locations (*e.g.,* see the Hausdorff distance) but as a sequence of locations.

Other properties: i) Combine the space and time dimensions (*e.g.,* [7, 8]). ii) Meet the triangle inequality (*e.g.,* the Euclidean distance). iii) Have low computational complexity (the Fréchet distance is an example of a computationally expensive distance).

In Section 4.1 we present our own similarity measure specifically designed for clustering trajectories that might not overlap in time.

2.2 Privacy models

Privacy models for trajectory anonymization heavily depend on the assumptions about the data and the adversary's knowledge. A trajectory can be downgraded to a sequence of locations (*e.g.,* as in [12]), which simplifies the model by removing the time dimension from the problem. Other approaches assume that the data owner anonymizing the database knows the set of quasi-identifiers used by the adversary. Consequently, those parts of the trajectories matching the adversary's knowledge are simply removed from the published data [11].

A conservative, yet common, assumption is that every location might be regarded as a quasi-identifier. These models then define privacy as the highest re-identification probability for all the users in the dataset. In order to achieve k-anonymity under this assumption, the obfuscation method should transform the trajectories in a cluster in such a way that they become indistinguishable. In this regard, different obfuscation methods for trajectory anonymization have been proposed (*e.g.,* generalization [9, 12, 13], spatial translation [5, 6], and permutation [7, 8].)

In 2008, the (k, δ)-anonymity concept [5], which exploits the spatial uncertainty in the trajectory recording process, was proposed. Parameter k has the same meaning as in k-anonymity, while δ is a lower bound on the uncertainty radius when recording locations. We show in the next section that, for any $\delta > 0$ (that is, whenever there is actual uncertainty), (k, δ)-anonymity does not offer trajectory k-anonymity[3]. As a result, the anonymization methods *Never Walk Alone* (NWA, [5]) and *Wait for Me* (W4M, [6]) preserve the claimed user privacy when $\delta = 0$ only.

3 Privacy analysis of (k, δ)-anonymity

The (k, δ)-anonymity privacy notion is based on the assumption that trajectories are imprecise by nature. Unlike records in traditional databases, trajectory data do not remain constant over time, because a moving object should report its position in real time. However, this is impractical due to performance and wireless-bandwidth overhead.

[3] The proof and analysis provided in Section 3 can also be found in the original paper [15].

For this reason, Trajcevski *et al.* [14] suggest that a moving object and the server should reach an agreement consisting on an uncertainty threshold δ, meaning that a position is reported only when it deviates from its expected location by δ or more. Considering so, a moving object does not draw a trajectory anymore, but an uncertain trajectory defined by a trajectory τ and an uncertainty threshold δ.

Definition 1 (Trajectory). *A* trajectory *is an ordered set of time-stamped locations*

$$\tau = \{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\} \ ,$$

where $t_i < t_{i+1}$ *for all* $1 \leq i < n$.

Notation. For any time-stamp $t_1 \leq t \leq t_n$, the function $\tau(t)$ outputs the location of τ at time t. If $t = t_i$ for some $i \in \{1, \cdots, n\}$ then $\tau(t) = (x_i, y_i)$, otherwise $\tau(t)$ is the linear interpolation of the poly-line τ at time t. Similarly, $\tau(t)[x]$ and $\tau(t)[y]$ denote the spatial coordinates of location $\tau(t)$.

Definition 2 (Uncertain trajectory). *An uncertain trajectory is a pair* (τ, δ) *where* τ *is a trajectory and* δ *is an uncertainty threshold. Geometrically, the uncertain trajectory is defined as the locus*

$$UT(\tau, \delta) = \{(t, x, y) | d((x, y), (\tau(t)[x], \tau(t)[y])) \leq \delta\} \ ,$$

where $d((x_1, y_1), (x_2, y_2))$ *represents the Euclidean distance between locations* (x_1, y_1) *and* (x_2, y_2).

As shown in Figure 2, an uncertain trajectory $UT(\tau, \delta)$ is the union of all the cylinders of radius δ centered in the lines formed by (x_i, y_i) and (x_{i+1}, y_{i+1}) for every $1 \leq i < n$. Then, any continuous function $PMC^{\tau} : [t_1, t_n] \rightarrow \mathbb{R}^2$ such that $PMC^{\tau}([t_1, t_n]) \subset UT(\tau, \delta)$ is said to be a *possible motion curve* of the uncertain trajectory $UT(\tau, \delta)$.

If a trajectory τ_1 is a possible motion curve of the uncertain version (τ_2, δ) of another trajectory τ_2 and viceversa (τ_2 is a possible motion curve of (τ_1, δ)), then τ_1 and τ_2 are said to be *co-localized* with respect to δ [5, 6]. This relation is denoted as $Coloc_{\delta}(\tau_1, \tau_2)$ and provides the rationale behind (k, δ)-anonymity.

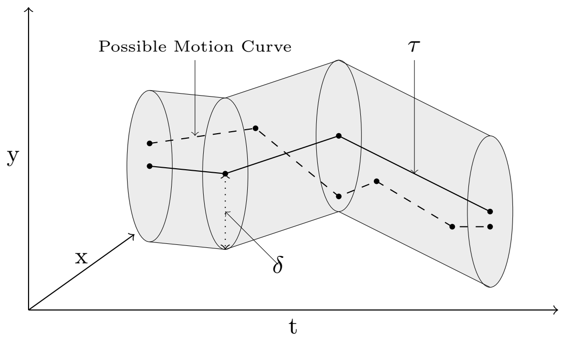

Fig. 2. A trajectory τ and its uncertain trajectory $UT(\tau, \delta)$. A possible motion curve within $UT(\tau, \delta)$ is also shown.

Definition 3 (**(k, δ)-anonymity set**). *Given an uncertainty threshold δ, a set of trajectories S is considered an anonymity set if and only if $Coloc_\delta(\tau_i, \tau_j)$ $\forall \tau_i, \tau_j \in S$.*

Then, (k, δ)-anonymity is defined as follows in [5, 6]:

Definition 4 (**(k, δ)-anonymity**). *Given a database of trajectories $\mathcal{D}$, an uncertainty threshold δ, and an anonymity threshold k, (k, δ)-anonymity is satisfied if, for every trajectory $\tau \in \mathcal{D}$, there exists a (k, δ)-anonymity set $S \subseteq \mathcal{D}$ such that $\tau \in S$ and $|S| \geq k$.*

In order to evaluate the privacy offered by (k, δ)-anonymity, we should rely on a second definition of trajectory k-anonymity under the same assumptions. We then use a privacy notion similar to the ones adopted in [7, 12, 9], which are less restrictive than (k, δ)-anonymity [5, 6] in the sense that the parameter δ is not required.

Definition 5 (**Trajectory k-anonymity**). *Let T^* be an anonymized set of trajectories corresponding to an original set of trajectories T. Let $\Pr_{\tau^*}[\tau|\sigma]$ denote the probability of the adversary's correctly linking the anonymized trajectory $\tau^* \in T^*$ with its corresponding original trajectory $\tau \in T$ given that the adversary knows a strict subset σ of the locations of τ. Then T^* satisfies trajectory k-anonymity if $\Pr_{\tau^*}[\tau|\sigma] \leq 1/k$ for every $\tau \in T$ and σ subset of the locations of τ.*

In Definition 5 above, the adversary's knowledge is represented as a *sub-trajectory* of an original trajectory, that is, as a subset of the set of time-stamped locations of the original trajectory. This background knowledge representation is appropriate for the trajectory anonymization schemes [7, 12, 9]. However, the uncertainty on the data under (k, δ)-anonymity does not permit to assume that the adversary knows a sub-trajectory in the above sense, except when $\delta = 0$ (no uncertainty). For $\delta > 0$, the adversary at best could know a possible motion curve PMC_τ of a trajectory τ contained in the original database $\mathcal{D}$. In other words, the adversary cannot be sure that her knowledge PMC_τ is exactly what was recorded in $\mathcal{D}$. It should be remarked that the adversary's knowledge was not explicitly defined in [5] or [6]. However, this is required here in order to provide formal privacy proofs.

Definition 6. *The adversary's knowledge in a database $\mathcal{D}$ of uncertain trajectories is defined as a random possible motion curve PMC_τ of some trajectory $\tau \in \mathcal{D}$.*

Definition 6 can be seen the other way round: the adversary is assumed to have the ability to acquire true actual locations about a user, such as home address or visited places, but the locations recorded in the database form a random possible motion curve of the adversary's knowledge due to the location uncertainty δ. Note that *not* considering the recorded trajectory as a random possible motion curve of the true original trajectory contradicts the (k, δ)-anonymity concept.

Theorem 1. *Let $\mathcal{D}$ be a database satisfying (k, δ)-anonymity. In general, $\mathcal{D}$ does not satisfy trajectory k-anonymity for any $\delta > 0$.*

Proof: We first give a counterexample which satisfies $(2, \delta)$-anonymity for any $\delta > 0$ but does not satisfy trajectory 2-anonymity; we will then generalize the argument for any k. Let τ_1 and τ_2 be two different but co-localized trajectories w.r.t. δ such that each of them consists of a single location. By the co-localization condition, the time stamp of both locations is the same and the distance d between the spatial coordinates of both locations satisfies $0 < d \leq \delta$.

Let $\mathcal{D}$ be the original dataset containing τ_1 and τ_2 only. Let us provide the adversary with a random possible motion curve PMC_{τ_i} where $i \in_R \{1, 2\}$ is randomly chosen. According to Definition 5, trajectory 2-anonymity is achieved if the adversary cannot guess with probability greater than $\frac{1}{2}$ whether $i = 1$ or $i = 2$.

However, let us consider the following adversarial strategy:

1. The adversary computes $d(PMC_{\tau_i}, \tau_1)$ and $d(PMC_{\tau_i}, \tau_2)$.
2. If $d(PMC_{\tau_i}, \tau_1) < d(PMC_{\tau_i}, \tau_2)$, the adversary's guess $i = 1$; otherwise, the adversary's guess is $i = 2$.

Now we will show that the previous strategy achieves a probability of success greater than $\frac{1}{2}$. To that end, let us compute the probability that $d(PMC_{\tau_1}, \tau_1) \geq d(PMC_{\tau_1}, \tau_2)$ for a random PMC_{τ_1}.

Let A and B be the two points of intersection of the uncertainty circles of τ_1 and τ_2 (see Figure 3). Then, $d(PMC_{\tau_1}, \tau_1) \geq d(PMC_{\tau_1}, \tau_2)$ only holds when PMC_{τ_1} lies in the arc segment area formed by the points A, B, and the uncertainty circle of τ_1 (shaded area in Figure 3). Since the line $\overline{AB}$ intersects the line formed by τ_1 and τ_2 in its middle point, it can be concluded that $0 \leq d(A, B) < 2\delta$. As $d(A, B)$ grows towards 2δ, the aforementioned arc segment area becomes asymptotically close to its maximum value $\pi\delta^2/2$. This means that:

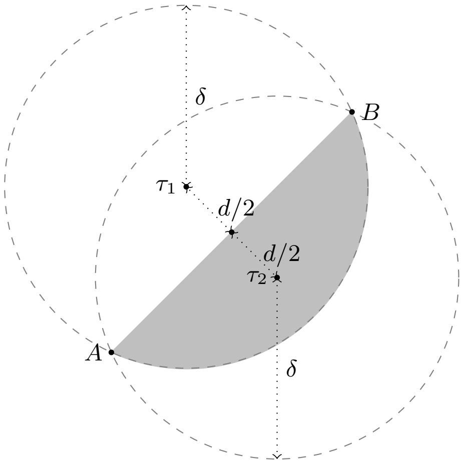

Fig. 3. Two trajectories τ_1 and τ_2 of size 1 such that $d(\tau_1, \tau_2) = d \leq \delta$. The two circles that intersect at A and B represent the uncertainty areas of both trajectories according to Definition 2.

$$\Pr(d(PMC_{\tau_1}, \tau_1) \geq d(PMC_{\tau_1}, \tau_2)) < \frac{1}{2}. \qquad (1)$$

From Expression (1), it can be concluded that the adversary's success probability is always greater than $\frac{1}{2}$ for any $\delta > 0$, which contradicts 2-anonymity.

The above reasoning can be generalized to any number k of trajectories. The generalized adversarial strategy is:

1. The adversary computes $d(PMC_{\tau_i}, \tau_j)$ for all $j \in \{1, \cdots, k\}$.
2. The adversary's guess is trajectory τ_g such that

$$g = \text{arg } \min_{1 \leq j \leq k} d(PMC_{\tau_i}, \tau_j)$$

By generalizing the geometric argument of Figure 3, it can be seen that the adversary's success probability with the above strategy is greater than $\frac{1}{k}$. This contradicts trajectory k-anonymity for any k and δ. □

Corollary 1. *The methods NWA [5] and W4M [6] can only offer trajectory k-anonymity for $\delta = 0$, that is, when all k trajectories in any (k, δ)-anonymity set are identical. In other words, trajectory k-anonymity is offered only when the set of anonymized trajectories consists of clusters containing k or more identical trajectories each.*

4 Our microaggregation-based method

In this section we present a heuristic method, named *SwapLocations*, for privacy-preserving publication of trajectories. SwapLocations is based on microaggregation of trajectories and permutation of locations. It first groups the trajectories into clusters of size at least k based on their similarity and then transforms via location permutation the trajectories inside each cluster to preserve privacy.

For clustering purposes, we present a distance for trajectories which naturally considers both spatial and temporal coordinates. Our distance is able to compare trajectories that are not defined over the same time span, without resorting to time generalization. It can also compare trajectories that are timewise overlapping only partially or not at all.

4.1 Our similarity measure

Clustering trajectories requires defining a similarity measure —a distance between two trajectories. Because trajectories are distributed over space and time, a distance that considers both spatial and temporal aspects of trajectories is needed. Many distance measures have been proposed in the past for both trajectories of moving objects and for time series but most of them are ill-suited to compare trajectories for anonymization purposes. Therefore we define a new distance which can compare trajectories that are only partially or not at all timewise overlapping. We believe this is necessary to cluster trajectories for anonymization. We need some preliminary notions.

Definition 7 (p%-contemporary trajectories). *Two trajectories*

$$T_i = \{(t^i_1, x^i_1, y^i_1), \ldots, (t^i_n, x^i_n, y^i_n)\}$$

and

$$T_j = \{(t^j_1, x^j_1, y^j_1), \ldots, (t^j_m, x^j_m, y^j_m)\}$$

are said to be p%-contemporary if

$$p = 100 \cdot \min(\frac{I}{t^i_n - t^i_1}, \frac{I}{t^j_m - t^j_1})$$

with $I = \max(\min(t^i_n, t^j_m) - \max(t^i_1, t^j_1), 0)$.

Intuitively, two trajectories are 100%-contemporary if and only if they start at the same time and end at the same time; two trajectories are 0%-contemporary if and only if they occur during non-overlapping time intervals. Denote the overlap time of two trajectories T_i and T_j as $ot(T_i, T_j)$.

Definition 8 (Synchronized trajectories). *Given two p%-contemporary trajectories T_i and T_j for some $p > 0$, both* trajectories are said to be synchronized *if they have the same number of locations timestamped within $ot(T_i, T_j)$ and these correspond to the same timestamps. A* set of trajectories is said to be synchronized *if all pairs of p%-contemporary trajectories in it are synchronized, where $p > 0$ may be different for each pair.*

If we assume that between two locations of a trajectory, the object is moving along a straight line between the locations at a constant speed, then interpolating new locations is straightforward. Trajectories can be then synchronized in the sense that if one trajectory has a location at time t, then other trajectories defined at that time will also have a (possibly interpolated) location at time t. This transformation guarantees that the set of new locations interpolated in order to synchronize trajectories is of minimum cardinality. Algorithm 1 describes this process. The time complexity of this algorithm is $O(|TS|^2)$ where $|TS|$ is the number of different timestamps in the data set.

Algorithm 1 Trajectory synchronization

```
Require: $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of trajectories to be synchronized, where each $T_i \in \mathcal{T}$
  is of the form:
      $T_i = \{(t^i_1, x^i_1, y^i_1), \ldots, (t^i_{n^i}, x^i_{n^i}, y^i_{n^i})\};$
Let $TS = \{t^i_j \mid (t^i_j, x^i_j, y^i_j) \in T_i \ : \ T_i \in \mathcal{T}\}$ be all timestamps from all locations of
  all trajectories;
for all $T_i \in \mathcal{T}$ do
    for all $ts \in TS$ with $t^i_1 < ts < t^i_{n^i}$ do
        if location having timestamp $ts$ is not in $T_i$ then
            insert new location to $T_i$ having the timestamp $ts$ and coordinates inter-
              polated from the two timewise-neighboring locations;
        end if
    end for
end for
```

Definition 9 (Distance between trajectories). *Consider a set of synchronized trajectories $\mathcal{T} = \{T_1, \ldots, T_N\}$ where each trajectory is written as*

$$T_i = \{(t^i_1, x^i_1, y^i_1), \ldots, (t^i_{n^i}, x^i_{n^i}, y^i_{n^i})\} \ .$$

The distance between trajectories *is defined as follows. If $T_i, T_j \in \mathcal{T}$ are $p\%$-contemporary with $p > 0$, then*

$$d(T_i, T_j) = \frac{1}{p} \sqrt{\sum_{t_\ell \in ot(T_i, T_j)} \frac{(x^i_\ell - x^j_\ell)^2 + (y^i_\ell - y^j_\ell)^2}{|ot(T_i, T_j)|^2}} \ .$$

If $T_i, T_j \in \mathcal{T}$ *are* 0%*-contemporary but there is at least one subset of* $\mathcal{T}$

$$\mathcal{T}^k(ij) = \{T_1^{ijk}, T_2^{ijk}, \ldots, T_{n^{ijk}}^{ijk}\} \subseteq \mathcal{T}$$

such that $T_1^{ijk} = T_i$, $T_{n^{ijk}}^{ijk} = T_j$ *and* T_ℓ^{ijk} *and* $T_{\ell+1}^{ijk}$ *are* $p_\ell\%$*-contemporary with* $p_\ell > 0$ *for* $\ell = 1$ *to* $n^{ijk} - 1$*, then*

$$d(T_i, T_j) = \min_{\mathcal{T}^k(ij)} \left(\sum_{\ell=1}^{n^{ijk}-1} d(T_\ell^{ijk}, T_{\ell+1}^{ijk}) \right)$$

Otherwise $d(T_i, T_j)$ *is not defined.*

The computation of the distance between every pair of trajectories is not exponential as it could seem from the definition. Polynomial-time computation of a distance graph containing the distances between all pairs of trajectories can be done as follows.

Definition 10 (Distance graph). *A* distance graph *is a weighted graph where*

(i) Nodes represent trajectories,
(ii) two nodes T_i *and* T_j *are adjacent if the corresponding trajectories are* $p\%$*-contemporary for some* $p > 0$*, and*
(iii) the weight of the edge (T_i, T_j) *is the distance between the trajectories* T_i *and* T_j*.*

Now, given the distance graph for $\mathcal{T} = \{T_1, \ldots, T_N\}$, the distance $d(T_i, T_j)$ for two trajectories is easily computed as the minimum cost path between the nodes T_i and T_j, if such path exists. The inability to compute the distance for all possible trajectories (the last case of Definition 9) naturally splits the distance graph into connected components. The connected component that has the majority of the trajectories must be kept, while the remaining components represent outlier trajectories that are discarded in order to preserve privacy. Finally, given the connected component of the distance graph having the majority of the trajectories of $\mathcal{T}$, the distance $d(T_i, T_j)$ for *any two* trajectories on this connected component is easily computed as the minimum cost path between the nodes T_i and T_j. The minimum cost path between every pair of nodes can be computed using the Floyd-Warshall algorithm with computational cost $O(N^3)$, *i.e.*, in polynomial time.

4.2 The SwapLocations method

Algorithm 2 describes the process followed by the SwapLocations method in order to anonymize a set of trajectories. First, the set of trajectories is partitioned into several clusters. Then, each cluster is anonymized using the SwapLocations function in Algorithm 3.

We limit ourselves to clustering algorithms which try to minimize the sum of the intra-cluster distances or approximate the minimum and such that the cardinality of each cluster is k, with k an input parameter; if the number of trajectories is not a multiple of k, one or more clusters must absorb the up to $k-1$ remaining trajectories, hence those clusters will have cardinalities between $k+1$ and $2k-1$. This type of clustering is precisely the one used in microaggregation [3]. The purpose of minimizing the sum of the intra-cluster distances is to obtain clusters as homogeneous as possible, so that the subsequent independent treatment of clusters does not cause much information loss. The purpose of setting k as the cluster size is to fulfill trajectory k-anonymity. We employ any microaggregation heuristic for clustering purposes.

Algorithm 2 Cluster-based trajectory anonymization($\mathcal{T}, R^t, R^s, k$)

```
Require: i) $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of original trajectories such that $d(T_i, T_j)$ is
    defined for all $T_i, T_j \in \mathcal{T}$, ii) $R^t$ a time threshold and $R^s$ a space threshold;
1: Use any clustering algorithm to cluster the trajectories of $\mathcal{T}$, while minimizing
   the sum of intra-cluster distances measured with the distance of Definition 9 and
   ensuring that minimum cluster size is $k$;
2: Let $C_1, C_2, \ldots, C_{n_\mathcal{T}}$ be the resulting clusters;
3: for all clusters $C_i$ do
4:     $C_i^\star$ = SwapLocations($C_i, R^t, R^s$);                    // Algorithm 3
5: end for
6: Let $\mathcal{T}^\star = C_1^\star \cup \cdots \cup C_{n_\mathcal{T}}^\star$ be the set of anonymized trajectories.
```

The SwapLocations function (Algorithm 3) begins with a random trajectory T in C. The function attempts to cluster each unswapped triple λ in T with another $k-1$ unswapped triples belonging to different trajectories such that: i) the timestamps of these triples differ by no more than a time threshold R^t from the timestamp of λ; ii) the spatial coordinates differ by no more than a space threshold R^s. If no $k-1$ suitable triples can be found that can be clustered with λ, then λ is removed; otherwise, random swaps of triples are performed within the formed cluster. Randomly swapping this cluster of triples guarantees that any of these triples has the same probability of remaining in its original trajectory or becoming a new triple in any of the other $k-1$ trajectories. Note that Algorithm 3 guarantees that every triple λ of every trajectory $T \in C$ will be swapped or removed.

Algorithm 3 SwapLocations(C, R^t, R^s)

```
Require: i) $C$ a cluster of trajectories to be transformed, ii) $R^t$ a time threshold and
    $R^s$ a space threshold;
 1: Mark all triples in trajectories in $C$ as "unswapped";
 2: Let $T$ be a random trajectory in $C$;
 3: for all "unswapped" triples $\lambda = (t_\lambda, x_\lambda, y_\lambda)$ in $T$ do
 4:     Let $U = \{\lambda\}$; // Initializing $U$ with $\{\lambda\}$
 5:     for all trajectories $T'$ in $C$ with $T' \neq T$ do
 6:         Look for an "unswapped" triple $\lambda' = (t_{\lambda'}, x_{\lambda'}, y_{\lambda'})$ in $T'$ minimizing the
            intra-cluster distance in $U \cup \{\lambda'\}$ and such that:
                $|t_{\lambda'} - t_\lambda| \leq R^t$ ,
                $0 \leq \sqrt{(x_{\lambda'} - x_\lambda)^2 + (y_{\lambda'} - y_\lambda)^2} \leq R^s$ ;
 7:         if $\lambda'$ exists then
 8:             $U \leftarrow U \cup \{\lambda'\}$;
 9:         else
10:             Remove $\lambda$ from $T$;
11:             Goto line 3 in order to analyze the next triple $\lambda$;
12:         end if
13:     end for
14:     Randomly swap all triples in $U$;
15:     Mark all triples in $U$ as "swapped";
16: end for
17: Remove all "unswapped" triples in $C$;
18: return $C$.
```

The method SwapLocations meets trajectory k-anonymity in the sense of Definition 5. Refer to the original work [7] for details on the privacy analysis of SwapLocations.

5 Empirical results

In this section we evaluate the SwapLocations method by using a real-life data set of cab mobility traces that were collected in the city of San Francisco [16][4]. We consider three utility measures: i) percentage of removed trajectories, ii) percentage of removed locations, iii) and spatio-temporal range queries as proposed in [14]. The latter are described in more detail next.

5.1 Spatio-temporal range queries

Trajcevski et al. proposed in [14] six spatio-temporal range queries. For the sake of simplicity, we just keep the two more relevant for our experiments: *Sometime Definitely Inside* (SI) and *Always Definitely Inside* (AI).

- $SI(T, R, t_b, t_e)$ is *true* if and only if there exists a time $t \in [t_b, t_e]$ at which every possible motion curve PMC^T of an uncertain trajectory $U(T, \sigma)$ is inside region R. For a non-uncertain T, the previous condition can be adapted as: if and only if there exists a time $t \in [t_b, t_e]$ at which T is inside R.
- $AI(T, R, t_b, t_e)$ is *true* if and only if at every time $t \in [t_b, t_e]$, every possible motion curve PMC^T of an uncertain trajectory $U(T, \sigma)$ is inside region R. For a non-uncertain T, the previous condition becomes: if and only if at every time $t \in [t_b, t_e]$, trajectory T is inside R.

We accumulate the number of trajectories in a set of trajectories $\mathcal{T}$ that satisfy the SI or AI range queries using the SQL style code below.

- Query $\mathcal{Q}_1(\mathcal{T}, R, t_b, t_e)$:
  `SELECT COUNT (*) FROM` $\mathcal{T}$ `WHERE SI(`$\mathcal{T}$`.traj, R, t`$_\mathtt{b}$`,t`$_\mathtt{e}$`)`
- Query $\mathcal{Q}_2(\mathcal{T}, R, t_b, t_e)$:
  `SELECT COUNT (*) FROM` $\mathcal{T}$ `WHERE AI(`$\mathcal{T}$`.traj, R, t`$_\mathtt{b}$`,t`$_\mathtt{e}$`)`

Then, we define two different *range query distortions*:

- $\text{SID}(\mathcal{T}, \mathcal{T}^\star) = \frac{1}{|\xi|} \sum_{\forall <R,t_b,t_e> \in \xi} \frac{|\mathcal{Q}_1(\mathcal{T},R,t_b,t_e) - \mathcal{Q}_1(\mathcal{T}^\star,R,t_b,t_e)|}{\max(\mathcal{Q}_1(\mathcal{T},R,t_b,t_e), \mathcal{Q}_1(\mathcal{T}^\star,R,t_b,t_e))}$ where ξ is a set of SI queries.
- $\text{AID}(\mathcal{T}, \mathcal{T}^\star) = \frac{1}{|\xi|} \sum_{\forall <R,t_b,t_e> \in \xi} \frac{|\mathcal{Q}_2(\mathcal{T},R,t_b,t_e) - \mathcal{Q}_2(\mathcal{T}^\star,R,t_b,t_e)|}{\max(\mathcal{Q}_2(\mathcal{T},R,t_b,t_e), \mathcal{Q}_2(\mathcal{T}^\star,R,t_b,t_e))}$ where ξ is a set of AI queries.

[4] A more comprehensive empirical evaluation can be found in the original paper where SwapLocations is introduced [7].

5.2 Results on real-life data

The San Francisco cab data set [16] we used consists of several files each of them containing the GPS information of a specific cab during May 2008. Each line within a file contains the space coordinates (latitude and longitude) of the cab at a given time. However, the mobility trace of a cab during an entire month can hardly be considered a single trajectory. We used big time gaps between two consecutive locations in a cab mobility trace to split that trace into several trajectories.

For our experiments we considered just one day of the entire month given in the real-life data set, but the empirical methodology described below could be extended to several days. In particular, we chose the day between May 25 at 12:04 hours and May 26 at 12:04 hours because during this 24-hour period there was the highest concentration of locations in the data set. We also defined the maximum time gap in a trajectory as 3 minutes; above 3 minutes, we assumed that the current trajectory ended and that the next location belonged to a different trajectory. This choice was based on the average time gap between consecutive locations in the data set, which was 88 seconds; hence, 3 minutes was roughly twice the average. In this way, we obtained 4582 trajectories and 94 locations per trajectory on average.

The next step was to filter out trajectories with strange features (outliers). These outliers could be detected based on several aspects like velocity, city topology, etc. We focused on velocity and defined 240 km/h as the maximum speed that could be reached by a cab. Consequently, the distance between two consecutive locations could not be greater than 12 km because the maximum within-trajectory time gap was 3 minutes. This allowed us to detect and remove trajectories containing obviously erroneous locations; Figure 4 shows one of these removed outliers where a cab appeared to have jumped far into the sea probably due to some error in recording its GPS coordinates. Altogether, we removed 45 outlier trajectories and we were left with a data set of 4547 trajectories with an average of 93 locations per trajectory. Figure 5 shows the ten longest trajectories (in number of locations) in the final data set that we used.

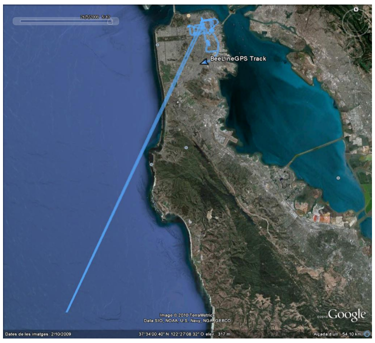

Fig. 4. Example of an outlier trajectory in the original real-life data set

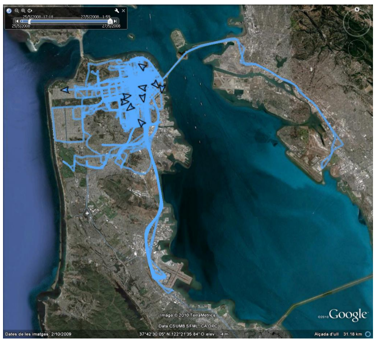

Fig. 5. Ten longest trajectories in the filtered real-life data set

We first consider the percentage of removed trajectories and the percentage of removed locations as utility measures. Table 1 shows how SwapLocations performs in terms of both.

Finally, Table 2 reports the performance of SwapLocations regarding spatio-temporal range queries. We picked random time intervals of length at most 20 minutes. Also, random uncertain trajectories with uncertainty threshold of size at most 7 km were chosen as the regions, which is roughly a quarter of the average distance of all trajectories. It can be seen that the SwapLocations method provides lower range query distortion for every value of k when the space threshold is small, *i.e.* when the total space distortion is also small. However, the smaller the space threshold, the larger the number of removed trajectories and locations (see Table 1). This illustrates the trade-off between the utility properties considered.

6 Conclusions

Several microaggregation-based methods for privacy-preserving spatio-temporal data publication have been proposed up to date. They mostly differ in the similarity measure, the obfuscation method, and the privacy model considered. In this book chapter we highlighted relevant properties for trajectory similarity measures that should be taken into account for microaggregation. We also described different privacy models based on k-anonymity in terms of the assumptions on the data and the adversary capabilities. In particular, we pro-

Table 1. Percentage of trajectories (columns labeled with **T**) and locations (columns labeled with **L**) removed by SwapLocations for several values of k and several space thresholds R^s on the real-life data set. Percentages have been rounded to integers for compactness.

$R^s \backslash k$	2		4		6		8		10		15	
	T	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**	**T**	**L**
1	23	43	40	64	49	71	58	74	62	77	71	81
2	19	29	34	47	42	54	50	58	54	60	50	66
4	14	17	27	29	35	35	40	40	45	41	54	49
8	9	10	19	19	25	25	31	29	34	31	42	38
16	5	7	11	16	17	22	20	27	23	30	32	38
32	1	7	2	15	3	22	4	27	5	30	8	38
64	0	6	0	15	0	22	0	27	0	30	0	38
128	0	6	0	15	0	22	0	27	0	30	0	38

Table 2. Range query distortion caused by SwapLocations in terms of SID (columns labeled with **S**) and AID (columns labeled with **A**), for several values of k and several space thresholds R^s. A range query distortion x is represented as the integer rounding of $x * 100$ for compactness.

$R^s \backslash k$	2		4		6		8		10		15	
	S	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**	**S**	**A**
1	13	22	18	27	20	29	19	29	24	31	25	34
2	16	24	25	34	26	35	24	35	27	37	27	37
4	18	25	30	37	33	41	34	42	38	46	38	45
8	21	27	34	40	38	44	40	46	44	50	48	54
16	20	26	36	42	42	47	45	50	50	54	53	58
32	21	26	39	44	45	49	48	53	53	57	58	62
64	20	25	39	44	46	50	51	54	54	57	61	64
128	21	26	39	44	48	50	51	56	54	58	61	64

vided a proof that invalidates the (k, δ)-anonymity concept for $\delta > 0$. Finally, we presented a similarity measure and a microaggregation-based approach that together deal with non-overlapping trajectories and preserve original locations. The method was evaluated by using a real-life dataset of trajectory data.

Acknowledgments and disclaimer

The second author is partially supported by the Government of Catalonia through an ICREA Acadèmia Prize. The following partial supports are also gratefully acknowledged: the Spanish Government under projects TIN2011-27076-C03-01 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and the European Commission under FP7 projects "DwB" and "Inter-Trust". The second author is with the UNESCO Chair in Data Privacy, but the views expressed in this paper neither necessarily reflect the position of UNESCO nor commit that organization.

References

1. P. Samarati, L. Sweeney, Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Tech. Rep. SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.
2. J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogenerous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.
3. J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.
4. J. Domingo-Ferrer. Microaggregation for database and location privacy. In *Proc. of Next Generation Information Technologies and Systems-NGITS'2006*, LNCS 4302, Springer, pp. 233-242, 2006.
5. O. Abul, F. Bonchi, and M. Nanni. Never walk alone: uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008*, Cancun, Mexico, 7-12 April 2008, pages 376–385. IEEE, 2008.
6. O. Abul, F. Bonchi, and M. Nanni. Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, 35(8):884–910, 2010.
7. J. Domingo-Ferrer, R. Trujillo-Rasua, Microaggregation- and permutation-based anonymization of movement data, *Information Sciences* 208:55–80, 2012.
8. J. Domingo-Ferrer, M. Sramka, and R. Trujillo-Rasua. Privacy-preserving publication of trajectories using microaggregation. In *Proceedings of the SIGSPATIAL ACM GIS 2010 International Workshop on Security and Privacy in GIS and LBS, SPRINGL 2010*, San Jose, California, USA, 2 November 2010. ACM, 2010.
9. M. E. Nergiz, M. Atzori, Y. Saygin, and B. Guc. Towards trajectory anonymization: a generalization-based approach. *Transactions on Data Privacy*, 2(1):47–75, 2009.
10. H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves., *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.
    `http://dblp.uni-trier.de/db/journals/ijcga/ijcga5.html`
11. M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proceedings of the 9th International Conference on Mobile Data Management, MDM 2008*, Beijing, China, 27-30 April 2008, pages 65–72. IEEE, 2008.
12. A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.
13. R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *Proceedings of the 12th International Conference on Extending Database Technology, EDBT 2009*, Saint Petersburg, Russia, 24-26 March 2009, volume 360 of *ACM International Conference Proceeding Series*, pages 72–83. ACM, 2009.
14. G. Trajcevski, O. Ouri, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems*, 29(3):463–507, 2004.
15. R. Trujillo-Rasua and J. Domingo-Ferrer. On the privacy offered by (k, δ)-anonymity. *Information Systems*, 38(4):491–494, 2013.

16. M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser. A parsimonious model of mobile partitioned networks with clustering. In *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, Bangalore, India, January 2009.