

Trajectory Anonymization from a Time Series Perspective

Sergi Martínez-Bea

IIIA - Institut d'Investigació en Intel·ligència Artificial
CSIC - Consejo Superior de Investigaciones Científicas
Campus de la UAB, 08193 Bellaterra, Catalonia, Spain
Email: smartinez@iiia.csic.es

Vicenç Torra

IIIA - Institut d'Investigació en Intel·ligència Artificial
CSIC - Consejo Superior de Investigaciones Científicas
Campus de la UAB, 08193 Bellaterra, Catalonia, Spain
Email: vtorra@iiia.csic.es

Abstract—In a world of constant technological evolution, the expansion of location tracking technologies is a fact. Most of us may have a device which can tell us our location with more or less accuracy. Most of those devices collect our location information and, usually, the trajectories we make. Publishing this information is essential for companies to improve their marketing strategies, for the traffic department in order to have some control about the traffic, and for many other entities. With the publishment of this information an important problem arises, the privacy. On the literature, many approaches to trajectory protection are provided. We contribute to the scene providing a framework with a protection method and measures for both the computation of the perturbation added to the data, and the risk of linking a protected trajectory to its original.

Index Terms—Trajectory anonymization, location privacy, information loss, disclosure risk.

I. INTRODUCTION

Nowadays, it is very common to have devices equipped with technologies that can localize you, such as smartphones, GPS navigators, etc. Those devices can store data about the trajectories you made. Those trajectories might be very useful for companies, e.g. they can know which are the most common places that people visit, the most transited roads, etc. The problem here is that this data can easily compromise your privacy. If this data falls in the hands of the wrong person, he could identify an individual and have the information about all their moves, which can be highly dangerous. The rationale, in order to avoid this, is to protect the data. There are many approaches in the literature for this purpose such as [1], [2], [3], and [4] between others. Some of them, for example [1], [2], and [3], only focus on protecting the data. Others, like [4], prevent the disclosure and try to preserve the data utility as high as possible. There is literature that focus on using generalization as a protection method, like in [1]. Generalization receives n trajectories and it returns n undistinguishable trajectories. Therefore, the re-identification of an individual becomes more difficult. In this paper, we focus on using microaggregation as a protection method, like in [4], and we provide a framework to evaluate the information loss (i.e. the amount of perturbation added to the data) and the disclosure risk. In contrast to [4], our work aims to be as generic as possible for the information loss calculation. This is specially important as there is a wide range of uses that this data can be given. This paper is organized as follows. In Section 2 we present some preliminar information

that will be used across all the text. In Section 3 we talk about protection methods and we present our approach. In Section 4, the information loss measures are presented. In Section 5, we explain the disclosure risk measure we use. In Section 6, we present the experiments made in order to validate our framework and we present the results. Finally, Section 7 concludes and give some directions for future research.

II. PRELIMINARIES

In this section we review several definitions on trajectories that are needed in the rest of the paper.

A. Trajectories

A trajectory can be defined as a sequence of places or spots that a given object or person has gone through. We will define each place or spot with their coordinates x and y and a timestamp t that determines the time when the object or person was there. Therefore, we can define formally a trajectory as:

$$\{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)\} \text{ where } n \in \mathbb{N}$$

In this way, we can observe a trajectory as numerical data that varies over time, and thus we can find similarities with the definition of time series. Recall that they are defined as:

$$\{(x_1, t_1), (x_2, t_2), \dots, (x_n, t_n)\} \text{ where } n \in \mathbb{N}$$

Given this, we can easily adapt distances and statistics from time series to trajectories. The first part of this section includes a section for distances and another for the autocorrelation function. The original distances were taken from [5].

B. Distances between trajectories

As we have already pointed out, there is a similarity between trajectories and time series. Due to this, we take two distance functions from time series and adapt them to trajectories. Trajectories can be totally overlapped over time, partially overlapped over time, or non overlapped at all. The distances adapted should take into account all those cases. It is important, though, that when we apply a distance to a pair of trajectories, the two trajectories have the same number of points.

1) *Adapted Euclidean Distance*: The common Euclidean distance does not take into account the time. As the distance between two trajectories partially overlapped over time should be less than the distance between two trajectories non overlapped, we need to include the time factor in the distance calculation.

Given two time series r and s as follows

$$r = \{p_1, p_2, \dots, p_n\}$$

$$s = \{q_1, q_2, \dots, q_n\}$$

with their elements p_i and q_i represented as

$$p = (p_x, p_y, p_t)$$

$$q = (q_x, q_y, q_t)$$

we can define the Adapted Euclidean Distance between r and s as:

$$d_{EU}(r, s) = \sqrt{\sum_{i=1}^n d_e(r_i, s_i)}$$

where

$$d_e(p, q) = (1 + (p_t - q_t)^2) \cdot ((p_x - q_x)^2 + (p_y - q_y)^2)$$

2) *Adapted Short Time Series Distance*: Adapting the Short Time Series distance to trajectories is quite straightforward. It is enough to add the y coordinate to the distance formula. Therefore, we can define the Adapted Short Time Series Distance as:

$$d_{STSA}(r, s) = \sqrt{\sum_{i=1}^n d_s(r_i, s_i)}$$

where

$$d_s(p, q) = \left[\left(\frac{q_{x_{i+1}} - q_{x_i}}{q_{t_{i+1}} - q_{t_i}} - \frac{p_{x_{i+1}} - p_{x_i}}{p_{t_{i+1}} - p_{t_i}} \right)^2 \right] + \left[\left(\frac{q_{y_{i+1}} - q_{y_i}}{p_{t_{i+1}} - p_{t_i}} - \frac{p_{y_{i+1}} - p_{y_i}}{p_{t_{i+1}} - p_{t_i}} \right)^2 \right]$$

C. Adapted Autocorrelation Function

The Autocorrelation function applied to time series is used in [5] as part of the computation of data utility. In the same way as the distances, we adapt this function to work with trajectories. Again, it is quite straightforward. It is enough to add the y coordinate to the formula.

Given the time series r represented by

$$r = \{p_1, p_2, \dots, p_n\}$$

with the elements p_i with the following components

$$p = (p_x, p_y, p_t)$$

we can define the Adapted Autocorrelation Function as:

$$\rho(h) = \frac{\gamma'_A(h)}{\gamma'_A(0)}$$

where

$$\gamma'_A(h) = n^{-1} \cdot \sum_{i=1}^{n-|h|} (p_{x_{t+|h|}} - \mu_x) \cdot (p_{x_t} - \mu_x) + (p_{y_{t+|h|}} - \mu_y) \cdot (p_{y_t} - \mu_y)$$

and

$$\mu_x = \frac{1}{n} \sum_{i=1}^n p_x$$

$$\mu_y = \frac{1}{n} \sum_{i=1}^n p_y$$

III. PROTECTION METHODS

As stated before, the objective of trajectory anonymization is to preserve the privacy of the individuals whose trajectory data has been collected. In order to achieve this, we need a protection method.

Given a data set X a protection method transforms X into a protected data set X' . In general, X can be seen as a matrix with n rows (records) and m columns (attributes). We can classify the attributes in two types: identifiers and quasi-identifiers. A given attribute is considered an identifier when it unambiguously identifies an individual (e.g the social security number), in contrast an attribute is considered a quasi-identifier when it can not identify an individual by itself but a set of them can. Note that a concrete group of quasi-identifier attributes can identify an individual depending on the scenario. E.g. age and sex can be quasi-identifiers in a small company but not in a class. Quasi-identifier attributes can be divided in two classes: confidential and non-confidential depending on whether they contain private information. For instance, let us imagine that an attacker A has some quasi-identifier attributes of an individual, as e.g. age, birth place, sex and medical history. The first three attributes are non-confidential while the last one is confidential. In a very large database those attributes would not give the attacker the necessary information to identify an individual as lots of people of the same sex may have been born in the very same place at the same year. But if we have a different scenario, e.g. a database with the employees of a company, where those quasi-identifier attributes are unique in the whole data set, the attacker could clearly identify the individual.

Our scenario is the same as the scenario presented in [5]. In the standard scenario (see eg. [6] [7] [8] [9] [10] [11]) before releasing a data set X with confidential attributes, a protection method ρ is applied, leading to a protected data set X' . Indeed, we will assume the following: (i) identifier attributes in X are either removed or encrypted, therefore we will write $X = X_{nc} || X_c$; (ii) confidential quasi-identifier attributes X_c are not modified, so we have $X'_c = X_c$; (iii) the protection method itself is applied to non-confidential quasi-identifier

attributes in order to preserve the privacy of the individuals whose confidential data is being released. Therefore, we have $X'_{nc} = \rho(X_{nc})$.

In the remaining part of this section we discuss microaggregation, one of the most effective data protection methods. We describe our approach for applying microaggregation to trajectories.

A. Microaggregation

Microaggregation is a protection method that ensures k -anonymity by building small clusters of at least k elements and replacing the original values with the centroid of the cluster to which the record belongs to.

In order to solve the information loss problem of the basic microaggregation method, multivariate microaggregation is used at the cost of increasing the disclosure risk. Specifically, after dividing attributes into different blocks and applying the basic microaggregation technique to each block separately. This means that the k records which fall in the same cluster for the first block of attributes, may fall in a different cluster for any of the other block of attributes. So, the resulting protected records will not be equal and no k -anonymity is ensured.

In order to preserve information loss as low as possible, microaggregation methods try to minimize the total sum of distances between all the elements to be protected and the centroid of the cluster where an element belongs to, i.e. to minimize the total Sum of Square Errors (SSE):

$$SSE = \sum_{i=1}^C \sum_{x_{ij} \in C_i} (X_{ij} - \bar{X}_i)^T (X_{ij} - \bar{X}_i),$$

where c is the total number of clusters, C_i is the i th cluster and $\bar{x}_i$ is the centroid of C_i . The restriction is $|C_i| \geq k$, for all $i = 1, \dots, c$. In general, the larger the value of k the lower the disclosure risk. Therefore, in order to parametrize microaggregation methods, k has to be large enough to ensure confidentiality but not too large to compromise the statistical utility of the protected information.

The rationale of this process is to make the protected data as similar as possible to the original one. In any case, methods should provide clusters with at least k elements. Finding the optimal multivariate microaggregation has been proven to be an NP-Hard problem. For this reason, heuristic methods have been proposed in the literature.

In our work, we will adapt for trajectories one of the most used methods for heuristic multivariate microaggregation: the MDAV (Maximum Distance to Average Vector) algorithm.

B. MDAV microaggregation

The MDAV (Maximum Distance to Average Vector) algorithm is an heuristic algorithm for clustering records in a data set X so that each cluster is constrained to contain at least k records. You can see the MDAV algorithm description in Algorithm 1.

The generic MDAV algorithm can be used for different data types, but then it requires the appropriate definitions for distance

and average. When we want to apply the MDAV algorithm to a particular type of data, we need to define what the most distant record means, which the closest records of a given record are, and the average record of a set of records. The average vector will be used in step (3) and to mask the original data in steps (4) and (8).

Algorithm 1: MDAV

Input: X : original microdata, k : integer
Output: X' : protected microdata
1: **while** $|X| > k$ **do**
2: Compute the average record $\bar{x}$ of all records X
3: Consider the most distant record x_r to the average record $\bar{x}$
4: Form a cluster around x_r . The cluster contains x_r together with the $k - 1$ closest records to x_r .
5: Remove these records from microdata file X
6: **if** $|X| > k$ **then**
7: Find the most distant record x_s from record x_r .
8: Form a cluster around x_s . The cluster contains x_s together with the $k - 1$ closest records to x_s .
9: Remove these records from microdata file X
10: **end if**
11: **end while**
12: Form a cluster with the remaining records

C. Trajectory microaggregation

In order to apply the MDAV algorithm to trajectories we need to determine which distance and which average functions we are going to use in order to implement the general MDAV algorithm described in Algorithm 1.

- **Distance functions.** We use our *Adapted Euclidean Distance* and our *Adapted Short Time Series Distance*, which were already explained in section 2.
- **Average.** As we are working with trajectories, using an average function would not give realistic results. Given a cluster, when doing the aggregation process, that is, the step when you replace the original values with the centroid of the cluster, if the centroid is computed with an average function, the microaggregated trajectories could have points that fall inside a building, which would be correct if the individual is a pedestrian, but not if it is an automobile. In the latter case, the trajectory would not have any utility. We assume that there can not be points inside a building. In order to avoid that, we propose two approaches based on the median function.
 - The first approach, which we call (*X median, Y median*), is to calculate the median for the X coordinate of points of different trajectories that have the same timestamp t and fall into the same cluster. Then, do the same with the Y coordinate, so you end up having a trajectory that represents all the trajectories in that cluster.

E.g. Given the trajectories r , s and u defined as

$$\begin{aligned} r &= (r_1, r_2, r_3) \\ s &= (s_1, s_2, s_3) \\ u &= (u_1, u_2, u_3) \end{aligned}$$

where

IV. INFORMATION LOSS MEASURES

$$\begin{aligned} r_1 &= (3, 6, 1), r_2 = (1, 7, 2), r_3 = (8, 5, 3) \\ s_1 &= (2, 9, 1), s_2 = (7, 8, 2), s_3 = (2, 4, 3) \\ u_1 &= (0, 3, 1), u_2 = (6, 2, 2), u_3 = (9, 2, 3) \end{aligned}$$

we calculate the median for the X and Y coordinate of the points with the same timestamp t , resulting in a single trajectory v . That is,

$$v = (v_1, v_2, v_3)$$

where

$$v_1 = (2, 6, 1), v_2 = (6, 7, 2), v_3 = (8, 4, 3)$$

- The second approach, which we call (X median, Y) is not very different. Again, calculate the median for the X coordinate of the same points of the different trajectories that fall into the same cluster. Then instead of doing the same with the Y coordinate, search through all the trajectories in the same cluster for that X value, and get the corresponding Y value. This gives a trajectory with real points (i.e. points that will not fall inside a building), that represents all the trajectories in that cluster. E.g. Given the trajectories r , s and u that fall into the same cluster, defined as

$$\begin{aligned} r &= (r_1, r_2, r_3) \\ s &= (s_1, s_2, s_3) \\ u &= (u_1, u_2, u_3) \end{aligned}$$

where

$$\begin{aligned} r_1 &= (3, 6, 1), r_2 = (1, 7, 2), r_3 = (8, 5, 3) \\ s_1 &= (2, 9, 1), s_2 = (7, 8, 2), s_3 = (2, 4, 3) \\ u_1 &= (0, 3, 1), u_2 = (6, 2, 2), u_3 = (9, 2, 3) \end{aligned}$$

we calculate the median for the X coordinate of the points with the same timestamp t and get the Y coordinate corresponding to that X value, resulting in a single trajectory v . That is,

$$v = (v_1, v_2, v_3)$$

where

$$v_1 = (2, 9, 1), v_2 = (6, 2, 2), v_3 = (8, 5, 3)$$

In section 6 we show how one approach or the other, affect the results.

Trajectories information can have a wide range of uses. It can be used by companies to improve their advertisement strategy, for marketing purposes, for traffic control, etc. In our scenario, we assume no concrete use can be specified before its protection.

When applying any perturbative methods to protect a data set, there is always some kind of information loss. I.e., the perturbation inflicted to the data decreases their utility. As we would like to minimize this loss, we need a measure to compare different perturbations due to different parameters within a method or even different methods.

Information loss clearly depends on the uses that will be given to the protected data. Nevertheless, as stated above we focus on generic uses. In this case, it is desirable to have a generic way to measure this information loss. It should reflect the amount of perturbation that is being added with a given protection method and the decay of data utility for a range of data uses.

In order to achieve this goal, we define information loss in terms of three components: IL_1 , IL_2 , IL_3 . They are partial information loss measures described below.

- IL_1 . It is defined as the average value of the measures $IL_{1.1}$ and $IL_{1.2}$ which are defined below. That is,

$$IL_1 = \frac{IL_{1.1} + IL_{1.2}}{2}$$

$IL_{1.1}$ is the average of the difference between the means of both original and protected trajectories, and $IL_{1.2}$ is the average of the difference between the autocorrelation functions of both original and protected trajectories. They are defined by:

$$IL_{1.1} = \frac{1}{2s} \left(\sum_{i=1}^s \frac{|\mu_{x_i} - \mu'_{x_i}|}{\text{Max}(|\mu_{x_i}|, |\mu'_{x_i}|)} + \frac{|\mu_{y_i} - \mu'_{y_i}|}{\text{Max}(|\mu_{y_i}|, |\mu'_{y_i}|)} \right)$$

$$IL_{1.2} = \frac{1}{4} \sum_{h=0, n/4, n/2, 3n/4} \left(\frac{1}{s} \sum_{i=1}^s \frac{|\rho_i(h) - \rho_i(h)'|}{\text{Max}(|\rho_i(h)|, |\rho_i(h)'|)} \right)$$

where s is the number of trajectories in the given data set, μ_x the average of the X coordinate, μ_y the average of the Y coordinate, and the $\rho_i(h)$ is the *Adapted Autocorrelation Function* explained on the preliminaries.

- IL_2 . It is defined as the absolute differences between original and protected data. That is,

$$IL_2 = \frac{1}{2 \times s \times n} \sum_{i=1}^{s \times n} \left(\frac{||x_i| - |x'_i||}{\text{Max}(|x_i|, |x'_i|)} + \frac{||y_i| - |y'_i||}{\text{Max}(|y_i|, |y'_i|)} \right)$$

where s is the number of trajectories in the given data set, n the number of points of a trajectory (i.e. its length), x and y the coordinates of a point that belongs to a non protected trajectory and x' and y' the coordinates of a point that belongs to a protected trajectory. Each pair of $x, y - x', y'$ has the same timestamp t .

- IL_3 . This measure was taken from [4] and is called *Space Distortion*. The aim of this measure is to capture the distortion of the shape of an anonymized trajectory. The definition on [4], takes into account that the distances used for the protection process are not defined for all kind of trajectories (i.e. totally overlapped over time, partially overlapped over time or non overlapped over time). As our distances can be applied to all of those trajectories we can simplify the Space Distortion measure presented in [4]. The *Simplified Space Distortion measure (SSD)* is defined as

$$SSD = \sum_{i=1}^s \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}$$

where s is the number of trajectories of a given data set, x and y the coordinates of a point that belongs to a non protected trajectory and x' and y' the coordinates of a point that belongs to a protected trajectory. Each pair of $x, y - x', y'$ has the same timestamp t .

After the SSD calculation, a min-max range normalization is applied as it is desirable to have this value in the range $[0, 1]$.

V. DISCLOSURE RISK MEASURE

When protecting trajectories or any data in general, the main goal is to maintain the privacy of the individuals. As it is desirable to keep the data useful, the protection method has to try to achieve a minimum information loss. The problem is that while the information loss decreases, it resembles more the original data and, therefore, the disclosure risk increases. Then a way to measure this disclosure risk is needed.

A. Record Linkage method

One direct approach to measure the disclosure risk is to get both the original and protected data and try to establish some links between the individuals in both data sets. That is what an attacker would do in case he gets the original and protected data sets. This method is based on the standard *Distance Based Record Linkage (DB-RL)* which can only be applied when a distance measure exists between a pair of records. This method links every protected record to the closest original one. Here, we use two different distances. Those distances are the same that were used in the microaggregation process, i.e. *Adapted Euclidean Distance* and *Adapted Short Time Series Distance*. Consider *AEULD (Adapted Euclidean distance Linkage Disclosure)* and *ASTSLD (Adapted Short Time Series distance Linkage Disclosure)*, then, we compute the disclosure risk (DR) as

$$DR = \text{Max}(AEULD, ASTSLD)$$

The DB-RL process for trajectories is detailed below, in Algorithm 2.

Algorithm 2: Trajectory Record Linkage

Input: X : original data set, k : protected data set
Output: X' : LP: linked pairs
1: **foreach** $a \in X$ **do**
2: $b' = \arg\min_{b \in X'} d_{tr}(a, b)$
3: $LP = LP \cup (a, b')$
4: **end foreach**

VI. EXPERIMENTS

In this section we present the experiments we made, in order to empirically analyze our framework. The experiments consist in applying our protection method to some data sets (those data sets are explained in detail later in this section). Once the data sets are protected, the information loss and the disclosure risk is computed. As stated before, as information loss decreases, disclosure risk increases, and viceversa. Due to this, a trade-off between the two measures is needed to evaluate the method. There are several approaches in the literature, some of them use a graphical representation, for instance R-U maps [12], others are analytical. For example, in [13], the authors use a combination of both information loss and disclosure risk, which they call *Score*, using the arithmetic mean between these two values. In this way, we can have a numerical value to determine the effectiveness of the method in different scenarios. The score is defined as

$$\text{score} = \frac{IL + DR}{2}$$

where IL is the information loss measure and DR the disclosure risk measure. Note that the lower the score, the best is the protection method.

A detailed explanation of the data sets we used as well as the process applied for the experiments and the commented results are found below.

A. Data sets

When doing experiments it is desirable that the data sets are real data from the real world instead of random generated data. In our case, in order to get data sets as real as possible, we used Brinkhoff's generator [14] to obtain trajectories of moving entities. The trajectories generated this way, have the same behaviour that would have real trajectories. That is because moving objects in a city usually follow a network (e.g. the street map) made of interconnected nodes (e.g. cross-streets), and also because of the large ammount of trajectories generated as well as their variable length. The synthetic trajectories are precisely generated this way. This generator has also been used for evaluation of approaches like [15] and [16]. We generated 1049 trajectories separated in 10 different data sets. The generated trajectories are over the map of Oldenburg in Germany, with 5 moving entities (object classes). As the object classes can represent different moving entities such pedestrians or automobiles, the microaggregation process is applied to each object class separately.

B. Process

In the clustering part of the microaggregation process, there is a parameter k that constraints each cluster to have at least k trajectories. Due to this, varying this parameter affects the final results as the microaggregated trajectories will not be the same for different values of k . Therefore, in our experiments, we considered applying the protection process for different values of k to each data set. That is, $k \in \{2, 3, 6, 9, 12, 14\}$. As we stated before, in the microaggregation we have two ways of doing the aggregation process. Recall that $(X \text{ median}, Y \text{ median})$ is the process where the median for the X coordinate and the median for the Y coordinate is calculated, and $(X \text{ median}, Y)$ the process where the median for the X coordinate is calculated and its corresponding Y coordinate is taken. We also stated before that we use two different distances, the *AEUD* (*Adapted Euclidean Distance*) and the *ASTSD* (*Adapted Short Time Series Distance*). This gives us two more ways of applying the microaggregation. The experiments take into account those two distances as well as the $(X \text{ median}, Y \text{ median})$ and $(X \text{ median}, Y)$ methods for doing the aggregation process.

C. Results

This section presents the results on our experiments. In order to present them as clearly as possible, we only present the common behaviours detected from the analyzed results.

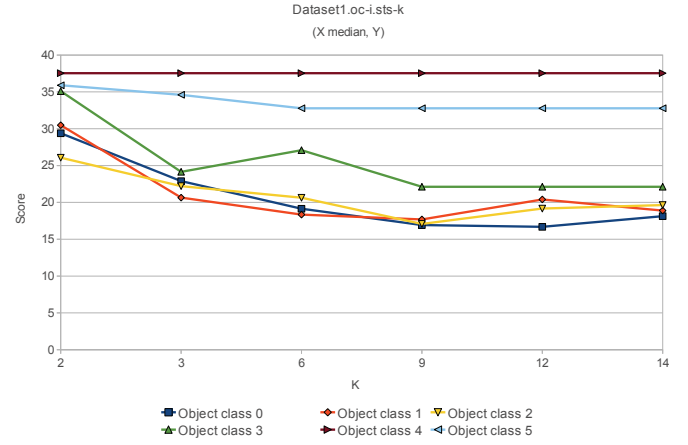
Figures 1(a) - 1(b) and 2(a) - 2(b) present four graphics as a representation of all the results. The graphics for the other data sets have a similar behaviour so we omit them.

In these figures, in the X axis we have the different values for the k parameter, on the Y axis we have the *Score* value (the lower, the better). Each graphic represents the *Score* for each k value, for the different object classes.

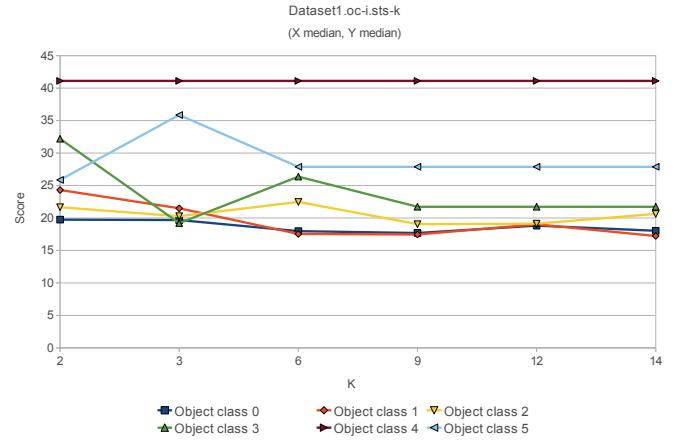
In general all the graphics have a similar shape. The *Score* value starts decreasing until it reaches a minimum value and then, they increase or they keep equal. The common behaviour of the *Score* value is precisely this. We stated that information loss and disclosure risk are contrary measures (i.e. when one increases the other decreases) and they both have the same relevance when computing the *Score*. This means that the best result for a given k value will be the one that has the lower *Score*. Thinking in this way, we observe that, among the different object classes, the lowest values for the *Score* are at $k = 9$, Except for the case of Figure 1(b) where the best k value is $k = 12$.

In all the results there are two object classes (4 and 5) that stand out of the others, not for they high *Score* values but for they shape: The *Score* is monotonically stable. This is because the information loss and the disclosure risk measures have the same relevance on the *Score* computation so, when they equilibrate each other, the *Score* does not change.

We stated before that we use two different distances for the microaggregation process as well as two different ways of doing the aggregation process, so it is interesting to compare the obtained results. Focusing at the k value that gives, in general, the best result (i.e. $k = 9$) and comparing the



(a) Adapted Euclidean distance, $X \text{ median} - Y$

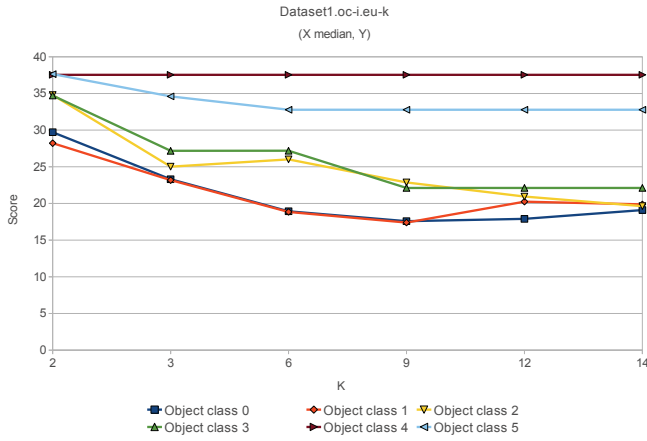


(b) Adapted Euclidean distance, $X \text{ median} - Y \text{ median}$

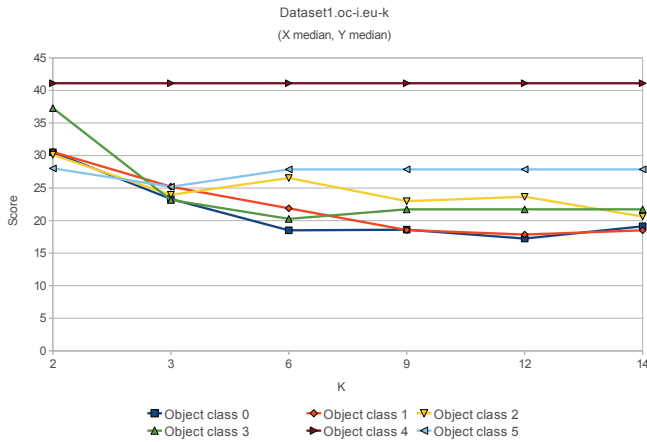
Fig. 1. Adapted Euclidean distance results

AEUD microaggregation with the *ASTSD* microaggregation in combination of the two different aggregation methods the $(X \text{ median}, Y \text{ median})$ and the $(X \text{ median}, Y)$, the differences between them are not remarkable but the combination of the *ASTSD* microaggregation with the $(X \text{ median}, Y)$ aggregation process seems to be the best. That is because the value of the *Score* is the lowest at each object class, except for the object class 5. This behaviour is due to a high value of the information loss measure for the *ASTSD* microaggregation with the $(X \text{ median}, Y)$ aggregation process, which come from the microaggregation process, where, in this case, the anonymity achieved is not high enough to equilibrate the information loss. Therefore, the *Score* value is higher for this object class. This can be appreciated on Figure 3.

Figure 4 represents the *Disclosure Risk* and *Information Loss* measures for each pair consisting of the microaggregation of an object class using a particular aggregation process. In the figure, the nearer a pair to $(0, 0)$, the better. Note that $(0, 0)$ stands for data with no information loss and no disclosure risk. Also, the exceptional cases commented above can be appreciated.



(a) Adapted Short Time Series distance, $X \text{ median} - Y$



(b) Adapted Short Time Series distance, $X \text{ median} - Y \text{ median}$

Fig. 2. Adapted Short Time Series Results

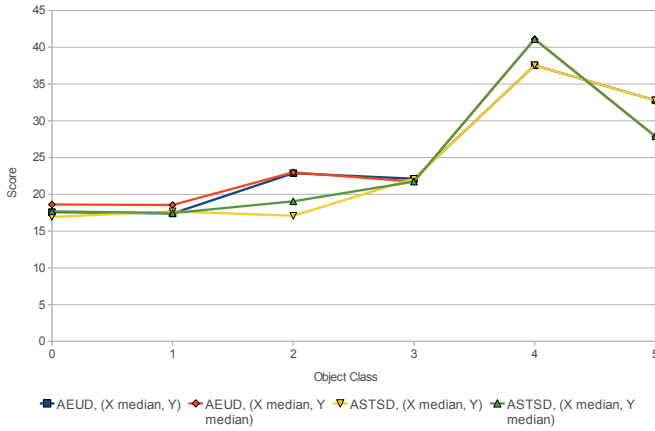


Fig. 3. Score comparison for $k = 9$

VII. CONCLUSIONS AND FURTHER WORK

In this paper, we have presented a new framework for trajectory anonymization. We adapted the microaggregation protection method in order to apply it to trajectory anonymization. We have introduced a set of information loss measures which try to cover the general uses of trajectories. We adapted the distance based record linkage for trajectories. We adapted

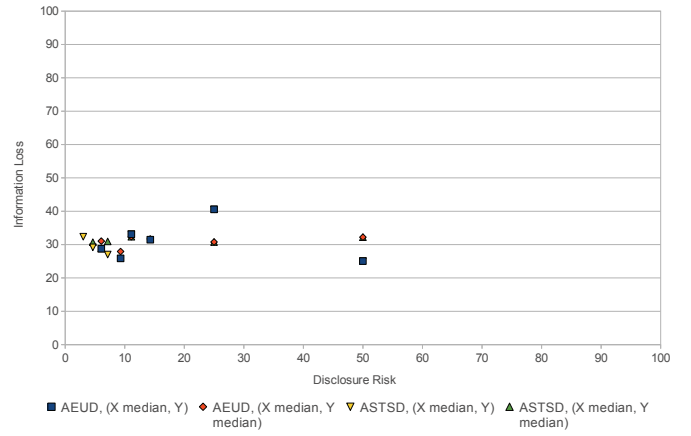


Fig. 4. $R-U$ Map for $k = 9$

a couple of existing distances so that they can be applied to trajectories, no matter what their overlapping degree is. Finally, we have presented some results from applying the microaggregation method to different data sets in order to get the best parameter combination for this method when working with trajectories.

As future work, new protection methods can be found or adapted as well as new information loss and disclosure risk measures.

ACKNOWLEDGEMENTS

Partial support by the Spanish MICINN (projects eAEGIS TSI2007-65406-C03-02 and ARES - CONSOLIDER INGENIO 2010 CSD2007-00004) is acknowledged.

REFERENCES

- [1] G. Andrienko, N. Andrienko, F. Giannotti, A. Monreale, and D. Pedreschi, "Movement data anonymity through generalization," in *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, ser. SPRINGL '09. New York, NY, USA: ACM, 2009, pp. 27–31. [Online]. Available: <http://doi.acm.org/10.1145/1667502.1667510>
- [2] G. Ghinita, "Private queries and trajectory anonymization: a dual perspective on location privacy," *Trans. Data Privacy*, vol. 2, pp. 3–19, April 2009. [Online]. Available: <http://www.tdp.cat/issues/tdp.a018a09.pdf>
- [3] D. Lin, E. Bertino, R. Cheng, and S. Prabhakar, "Location privacy in moving-object environments," *Trans. Data Privacy*, vol. 2, pp. 21–46, April 2009. [Online]. Available: <http://www.tdp.cat/issues/tdp.a019a09.pdf>
- [4] J. Domingo-Ferrer, M. Sramka, and R. Trujillo-Rasúa, "Privacy-preserving publication of trajectories using microaggregation," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, ser. SPRINGL '10. New York, NY, USA: ACM, 2010, pp. 26–33. [Online]. Available: <http://doi.acm.org/10.1145/1868470.1868478>
- [5] J. Nin and V. Torra, "Towards the evaluation of time series protection methods," *Inf. Sci.*, vol. 179, no. 11, pp. 1663–1677, 2009.
- [6] J. Domingo-Ferrer and V. Torra, "A quantitative comparison of disclosure control methods for microdata," in *Confidentiality, disclosure, and data access : Theory and practical applications for statistical agencies*. Elsevier, 2001, pp. 111–113.
- [7] J. Nin, J. Herranz, and V. Torra, "Rethinking rank swapping to decrease disclosure risk," *Data Knowl. Eng.*, vol. 64, pp. 346–364, January 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.datak.2007.07.006>
- [8] J. Nin, J. Herranz, and V. Torra, "How to group attributes in multivariate microaggregation," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 16, no. Supplement-1, pp. 121–138, 2008.

- [9] V. Torra, J. M. Abowd, and J. DomingoFerrer, "Using mahalanobis distance-based record linkage for disclosure risk assessment," in *Lecture Notes in Computer Science*, 2006, p. 233242.
- [10] W. E. Winkler, "Re-identification methods for masked microdata," in *In Proc. Privacy in Statistical Databases 2004, Springer LNCS 3050*. Springer, 2004, pp. 216–230.
- [11] W. E. Yancey, W. E. Winkler, and R. H. Creecy, "Disclosure risk assessment in perturbative microdata protection," in *Inference Control in Statistical Databases, From Theory to Practice*. London, UK, UK: Springer-Verlag, 2002, pp. 135–152. [Online]. Available: <http://portal.acm.org/citation.cfm?id=645539.657842>
- [12] G. T. Duncan, S. A. Keller-mcnulty, and S. L. Stokes, "Disclosure risk vs. data utility: The r-u confidentiality map," *Chance*, Tech. Rep., 2001.
- [13] V. Torra and J. Domingo-Ferrer, "Disclosure control methods and information loss for microdata," in *Confidentiality, disclosure, and data access : Theory and practical applications for statistical agencies*. Elsevier, 2001, pp. 91–110.
- [14] T. Brinkhoff, "Generating traffic data," *Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society*, no. 26, p. 2003.
- [15] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *InProceedings ICDE*, 2008, pp. 376–385.
- [16] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," in *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, ser. SPRINGL '08. New York, NY, USA: ACM, 2008, pp. 52–61. [Online]. Available: <http://doi.acm.org/10.1145/1503402.1503413>