TRANSACTIONS ON DATA PRIVACY 4 (2011) 103–126

Show Me How You Move and I Will Tell You Who You Are

Sébastien Gambs[1], **Marc-Olivier Killijian**[2,3], **Miguel Núñez del Prado Cortez** [2,3]

[1] Université de Rennes 1 - INRIA / IRISA ; Campus Universitaire de Beaulieu, 35042 Rennes, France

[2] CNRS ; LAAS ; 7 avenue du Colonel Roche, F-31077 Toulouse, France

[3] Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

E-mail: sgambs@irisa.fr, marco.killijian@laas.fr, mnpc@computer.org

Abstract. Due to the emergence of geolocated applications, more and more mobility traces are generated on a daily basis and collected in the form of geolocated datasets. If an unauthorized entity can access this data, it can use it to infer personal information about the individuals whose movements are contained within these datasets, such as learning their home and place of work or even their social network, thus causing a privacy breach. In order to protect the privacy of individuals, a sanitization process, which adds uncertainty to the data and removes some sensitive information, has to be performed. The global objective of GEPETO (for *GEoPrivacy Enhancing TOolkit*) is to provide researchers concerned with geo-privacy with means to evaluate various sanitization techniques and inference attacks on geolocated data. We describe our experiments conducted with GEPETO for comparing different inference attacks, and evaluating their efficiency for the identification of point of interests, as well as their resilience to sanitization mechanisms such as sampling and perturbation. We also introduce a mobility model that we coin as *mobility Markov Chain*, which can represent in a compact yet precise way the mobility behaviour of an individual. Finally, we describe an algorithm for learning such a structure from the mobility traces of an individual and we report on experimentations performed with real mobility data.

Keywords. Privacy, Geolocated data, Geo-privacy, Inference attacks, Sanitization, Clustering.

1 Introduction

A *geolocated system* is an object or device which has an associated location. For instance, it can be a smartphone or a GPS-equipped vehicle. Usually, a geolocated system belongs to an individual (or to a group of individuals, such as a family) and as such its location corresponds to the location of its owner(s). Geolocated data is already publicly available and sometimes easy to obtain. For instance, some people diffuse publicly, almost in real-time, their current location via social application such as Twitter which in turn can be collected to predict whether or not they are currently at home[1]. Other applications, such as Google Latitude[2], allow to track the movements of friends' cellphones and display their position on a map. Apart from these social applications, there are also other public sources of information that can be exploit by a potential adversary for causing a privacy breach,

[1] http://pleaserobme.com/

[2] http://www.google.com/latitude

such as free and easy access to geographic knowledge with Google Maps[3], Yahoo!Maps[4] and Google Earth[5].

We have started to explore, study and axiomatize the different types of inference attacks on geolocated data and basically our main finding is that among all the *Personal Identifiable Information* (PII), learning the location of an individual is one of the greatest threat against his privacy. For instance, the spatiotemporal data of an individual can be used to infer the location of his home and workplace, to trace his movements and habits, to learn information about his center of interests or even to detect a change from his usual behaviour. We provide a brief overview and classification of inference attacks on geolocated data in Section 2. Through the combination of several inference attacks, the adversary can potentially gather gradually more and more information about the mobility behaviour of an individual. We also describe some sanitization algorithms and methods for preserving geoprivacy in Section 2.4. In Section 3, we introduce a mobility model that we coin as *mobility Markov Chain*, which can represent in a compact yet precise way the mobility behaviour of an individual.

One of the main challenge for geoprivacy is to balance the benefit for an individual of participating to a geolocated application with the privacy risks he incurs by doing so. For example, if Alice's car is equipped with a GPS and she accepts to participate in the real-time computation of the traffic map, this corresponds to a task that is mutually beneficial to all the drivers but at the same time Alice wants to have some privacy guarantees that her individual locations will be protected and not broadly disclosed. In practice, we clearly advocate to follow the "privacy by design" paradigm which explicitly takes into account the privacy issues in the design process of a geolocated application, rather than simply deploying it and wait for the possible disastrous consequences.

We emphasize that simply removing the identifiers of individuals or replacing them by a pseudonym is usually not sufficient to protect their privacy. Instead, a *sanitization* process, which adds uncertainty to the data and removes some sensitive information, has to be performed. This loss of data, incurred by the sanitization process, comes with a dilemma: it certainly brings some privacy guarantees but at the cost of a decrease of utility due to the quality degradation of the data. For instance, the well-known k-anonymity procedure [28] ensures through generalization and suppression operations on attributes that each individual is in a group of at least $k-1$ other individuals sharing the same profile. However, this process also leads to a loss of information, thus almost surely hurting the utility of the application that will use this data. Therefore, there is often a trade-off between the utility of the global task and the privacy protection of individuals. In Section 4, we describe our ongoing work on GEPETO (for *GEoPrivacy Enhancing TOolkit*) [7], a flexible open source software which can be used to visualize, sanitize, attack and measure the utility of a particular geolocated dataset. Afterwards, in Section 5, we report on experimentations conducted on geolocated data through the GEPETO framework before finally concluding with a brief discussion in Section 6.

2 Inference Attacks on Geolocated Data

An *inference attack* is an algorithm that takes as input some geolocated data D, possibly with some auxiliary information aux, and produces as output some additional knowledge [12]. For example, an inference attack may consist of identifying the house or the place of work of an individual. The auxiliary information reflects any *a priori* knowledge that the adversary might have gathered (for instance through previous attacks and by accessing some public data source) and which may help him in conducting an inference attack. We propose to classify the inference attacks according to (at least) three dimensions such as the type of data it works on, the objective of the attack as well as the specific technique used. We also briefly review some geosanitization mechanisms that can be used to protect the privacy of geolocated data and limit the applicability and efficiency of inference attacks.

[3] http://maps.google.com/
[4] http://maps.yahoo.com/
[5] http://earth.google.com/

2.1 Geolocated Data

Nowadays, the rapid growth and development of geolocated applications has multiplied the potential sources of geolocated data. The geolocated data generated by these diverse applications varies in its exact form and content but it also shares some common characteristics. Regarding the type of data, we differentiate mainly between mobility traces and contact traces. A *mobility trace* is characterized by:

- An *identifier*, which can be the real identifier of the device (e.g. "Alice's phone"), a pseudonym or even the value "unknown" (when full anonymity is desired). A pseudonym is generally used when we want to protect the true identity of the system while still being able to link different actions performed by the same user.
- A *spatial coordinate*, which can be a GPS position (e.g. latitude and longitude coordinates), a spatial area (e.g. the name of a neighbourhood in a particular city) or even a semantic label (e.g. "home" or "work").
- A *time stamp*, which can be the exact date and time or just an interval (e.g. between 9AM and 12AM).
- Additional information such as the speed and direction for a vehicle, the presence of other geolocated systems or individuals in the direct vicinity or even the accuracy of the estimated reported position. For instance, some geolocated systems are able to estimate the precision of their estimated location as a function of the number of GPS satellites they are able to detect.

Contact traces are a specific form of mobility traces which consist in the recording of encounters between different devices. This kind of trace is composed of the identifiers of the devices and a time stamp. It may be recorded for instance by a device which has no integrated capacity for geopositioning but is capable of probing his neighbourhood to detect the presence of other devices (e.g. using Bluetooth neighbour discovery).

A *geolocated dataset* D is a dataset which contains mobility traces of individuals. Technically, this data may have been collected either by recording locally the movements of each geolocated system for a certain period of time, or centrally by a server which can track the location of these systems in real-time. A *trail of traces* is a collection of mobility traces that corresponds to the movements of an individual over some period of time. A geolocated dataset D is generally constituted by an ensemble of trails of traces from different individuals. The Crawdad project[6] is an example of a public repository giving access to geolocated datasets, which can be used for research purpose.

2.2 Objective of the Attack

An adversary attacking some geolocated data may have various objectives ranging from identifying the home of the target to reconstructing his social network, or even obtaining knowledge of his favourite jogging tracks. More precisely, the objective of an inference attack may be to:

- *Identify important places*, called *Points Of Interests* (POIs), which characterize the interests of an individual [17]. A POI may be for instance the home or place of work of an individual or locations such as a sport center, theater or the headquarters of a political party. Revealing the POIs of a particular individual is likely to cause a privacy breach as this data may be used to infer sensitive information such as hobbies, religious beliefs, political preferences or even potential diseases. For instance, if an individual has been visiting a medical center specialized in a specific type of illness, then it can be deduced that he has a non-negligible probability of having this disease. However, some POIs like the home or the place of work can sometimes be considered public data if they can be discovered by another mean such as googling the name of

[6] `http://crawdad.cs.dartmouth.edu/`

a particular individual or by browsing through the Yellow Pages. In this case, the main privacy risk is that this data can be used to *deanonymize* a particular individual whose name has been replaced by a pseudonym in a sanitized dataset but in which the combination of some of his POIs still uniquely characterizes him (see thereafter the linking attack for a similar argument).

- *Predict the movement patterns of an individual* such as his past, present and future locations [12]. From the movement patterns, it is possible to deduce other PII such as the mode of transport, the age or even the lifestyle[7]. According to some recent work [10,26], our movements are easily predictable by nature. For instance in [26], the authors have explored the limits of predictability in human mobility by analyzing mobility patterns of 50000 individuals within an anonymized geolocated dataset obtained from a mobile phone company that has more than 10 million users. By measuring the entropy of individual's trajectories, these authors have found a 93% potential predictability in user mobility.
- *Learn the semantics of the mobility behaviour of an individual* from the knowledge of his POIs and movement patterns. For instance, some mobility models such as *semantic trajectories* [2,27] do not only represent the evolution of the movements of an individual over time but also attach a semantic label to the places visited. From this semantic information, the adversary can derive a clearer understanding about the interests of an individual as well as his mobility behaviour than simply from his movement patterns. For instance, the adversary might be able to infer that on a typical weekday the individual considered generally leaves his home (POI 1) to bring his kid to school (POI 2) before going to work (POI 3), which is a more deep knowledge than simply knowing the movement pattern "POI 1 $\Rightarrow$ POI 2 $\Rightarrow$ POI 3".
- *Link the records of the same individual*, which can be contained in different geolocated datasets or in the same dataset, either anonymized or under different pseudonyms. This is the geoprivate equivalent of the *statistical disclosure risk* where privacy is measured according to the risk of linking the record of the same individual in two different databases (e.g., establishing that a particular individual in the voting register is also a specific patient of an hospital [28]). In a geolocated context, the purpose of a linking attack might be to associate the movements of Alice's car (contained for instance in dataset A) with the tracking of her cell phone locations (recorded in another dataset B). As the POIs of an individual and his movement patterns constitute a form of fingerprinting, simply anonymizing or pseudonymizing the geolocated data is clearly not a sufficient form of privacy protection against linking or deanonymization attacks. Indeed, a combination of locations can play the role of a *quasi-identifier* if they characterize almost uniquely an individual in the same way as the combination of his firstname and last name. For example, Colle and Kartridge [9] have shown that even the pair home-work becomes almost unique per individual, and thus acts as a quasi-identifier, if the granularity is not coarse enough (e.g., if the street is revealed instead of the neighbourhood).
- *Discover social relations* between individuals by considering for instance that two individuals that are in contact during a non-negligible amount of time share some kind of social link (of course false positive may happen) [16]. This information can also be derived from mobility traces by observing that certain individuals are in the vicinity of each other on a frequent basis.

2.3 Inference Technique

We describe thereafter some learning algorithms and methods that can be used as inference technique:

- *Clustering* is a form of unsupervised learning that tries to group objects that are similar in the same cluster while putting objects that are dissimilar in different clusters. A clustering algorithm needs a *distance measure* (or a similarity metric) to quantify how far/similar are two objects relative to each other and to drive the clustering process. A natural distance between two

[7] See for instance `http://www.sensenetworks.com/`.

locations is simply the Euclidean distance but of course more complex metrics can be used, such as the length of the shortest path according to the existing roadmap. For instance, k-means is an iterative clustering algorithm that outputs k clusters as well as their respective centres (which are effectively the average of the locations within each cluster). This algorithm can be used straightforwardly to discover the POIs of one particular individual if it is fed only with his data [4], or the generic *hotspots* if it is given the geolocated data of a whole population. Hoh, Gruteser, Xiong and Alrabady have performed a study [15] on the geolocated data of vehicles within the Detroit area (Michigan, USA). The goal of their study was to automatically discover the home of the vehicles' drivers. The authors have used a clustering algorithm to automatically identify the houses and their findings is that among the 2 neighbourhoods and the 65 persons on which the authors have focused, the estimated houses correspond at 85% to the houses that a human would have recognized[8]. More complex techniques such as density-based clustering [6] can be used instead of k-means to overcome some of its shortcomings, such as k the predefined number of clusters and the constraint that the shape of the clusters has to be spherical.

- *Mobility models* can be learned from the geolocated data of individuals, and then used either to identify them among a geolocated dataset (even when they are anonymous) or to predict their next movements. For instance, Lio, Fox and Kautz [20] have shown that it is possible to train a relational Markov network, so that it can predict with a relatively good accuracy the next location of an individual or his current activities. Another possibility is to use an algorithm for tracking the movement of targets [25] to reconstruct the paths followed by several individuals in a geolocated dataset even if they are anonymous. Some mobility models [2, 27] also attach a semantic to the places visited, thus enriching the knowledge extracted.
- *Heuristics* gives also good results in practice [19] for identifying POIs at a relatively low cost. An heuristic can be as simple as choosing the last stop before midnight or the average (or median) of several stop locations for identifying the home or the most stable location during the day for finding the place of work.
- *Data coming from social applications* is a possible source of information that the adversary might draw on to attack the privacy of individuals. The website "Please Rob Me" is a striking example of how it is possible from publicly available information in the form of Twitter's posts (i.e. *tweets*) to build a classifier that can predict whether or not somebody is currently at home. Another example of social application is Google Latitude that offers the possibility of following in real-time on a map the movements of siblings and friends who have previously agreed to this service by confirming this on a SMS received on their phone[9]. However, some social applications such as Locaccino[10] tries to integrate explicitly the privacy issues in their design, by giving the possibility to a user to choose how he wants to disclose and share its location with its friends, and helping him understand what are the potential privacy risks that he might incur.
- *Data coming from public sources* is also a potential source of knowledge that can be exploited by the adversary. For instance, by using Google Maps and Yahoo!Maps the adversary can easily reconstruct the path followed by an individual between two consecutive mobility traces. Moreover, *reverse geocoding* tools exist that can transform a spatial coordinate into a physical address, which in turn can be cross-referenced with the corresponding entries in the Yellow Pages.

Often, the inference process does not consist only in the application of one inference attack, but rather is an incremental process in which the adversary gradually gathers more and more knowledge through the combination of several inference attacks.

[8] As the exact identity of the drivers have been kept secret it was not possible for the authors to compare directly the houses returned by the algorithm against the ground truth (i.e. the exact address of the drivers) which explained why this particular evaluation method was chosen.

[9] An infamous use of Google Latitude is known as the *shower attack* where a suspicious husband waits for his wife to take her shower, before sending the Google Latitude SMS to her cellphone, accepting this service on her behalf on the cellphone and then erasing it thus leaving not clue for her that she is now tracked.

[10] `http://locaccino.org/`

2.4 Geo-sanitization Mechanisms

A *sanitization algorithm* S takes as input a geolocated dataset D, introduces some uncertainty and removes some information from this dataset to increase the privacy of individuals whose movements are contained in the dataset. S produces as output D', a sanitized version of the original dataset D. The main idea behind sanitization is that, for a potential adversary, breaching the privacy of a particular user is harder when working on D' than with D. A sanitization procedure usually comes with some privacy guarantees. For instance, it can guarantee that at each time step, there is a minimum number of individuals in each spatial area. Possible sanitization techniques include:

- *Pseudonymization* replaces the common identifier of several mobility traces by either a randomly generated pseudonym (thus providing anonymity but not unlinkability) or by the $unknown$ value (thus theoretically granting full anonymity and unlinkability)[11]. Pseudonymization is generally performed as the first step of a sanitization process but as such is often not sufficient for protecting the privacy of individuals.
- *Perturbation* methods [3] modify the spatial coordinate of a mobility trace by adding some random perturbation. For example, this noise can be generated uniformly or using Gaussian noise within a sphere of radius r centered on the original coordinate. If the geography of the surrounding area is not taken into account, it may happen that the perturbed coordinate corresponds to a location which has no physical sense (e.g., in the middle of a river or on a cliff).
- *Aggregation* merges several mobility traces into a single spatial coordinate. For instance, this spatial coordinate can be a surrounding spatial area such as a neighbourhood or the average of the mobility traces. During data preprocessing, a *clustering algorithm* (such as k-means) can be used to group traces that are close together into the same cluster while putting tracloakingces that are significantly distant into distinct clusters. This can be used to detect which traces should be merged together during an *aggregation* step. Another possibility is to detect traces occupying the same spatial area (for instance the same neighbourhood) at a certain moment in time and to replace each one of these individual traces by the coordinate of this spatial area.
- *Sampling* can be seen as a form of temporal aggregation. A *sampling* mechanism summarizes several mobility traces into fewer traces, generally by representing an ensemble of traces, which have occurred within some time window, into one median or average trace. By decreasing the total number of traces, sampling has the additional benefit that it compresses the data and, therefore, reduces the computational resources needed to further sanitize the data.
- *Spatial cloaking* [11] is an extension of the concept of k-anonymity [28] to the spatiotemporal domain and a form of aggregation. The main idea is to ensure that at each time step, each individual is located within a spatial area that is shared by a least $k-1$ other individuals. This spatial area is disclosed instead of the exact location of these individuals, thus guaranteeing that even if an adversary can target the group where an individual is located, his behaviour will be indistinguishable from at least $k-1$ other individuals (k is a privacy parameter of the algorithm). A possible approach to achieve the property of spatial cloaking is to split recursively the space into areas of different sizes, until further splitting violates k-anonymity. It is worth noting that spatiotemporal cloaking and the k-anonymization of more structured movement patterns such as trajectories [29] is a very active research topic. For instance in [1], a k-anonymity approach has been proposed for the sanitization of trajectories. More precisely, similar trajectories that are within a certain uncertainty radius of each other are clustered together such that each cluster is of size at least k and only the representative trajectory of the cluster is published. Moreover, trajectories that are deemed too "problematic" are erased from the sanitized data published. Another method achieving trajectory anonymization has also been recently proposed [22]. This

[11] *Anonymity* can be defined as being able to perform a particular action without having to reveal his identity whereas *unlinkability* is a stronger notion that involves not being able to link two different actions that have been performed by the same user. Typically, performing different actions under a pseudonym (instead of using his real name) provides anonymity but not unlinkability. See [23] for more details.

method combines an approach ensuring k-anonymity on the trajectories with a reconstruction algorithm that samples the (k-)anonymized dataset of trajectories and releases only "atomic trajectories".

- *Mix-zones* [5] are inspired from the concept of mix-nets due to Chaum. Mix-zones are spatial areas where (1) no measurements about the locations of individuals are performed and (2) such that each individual entering a mix-zone will have a different pseudonym when he exits the mix-zone. The main purpose of a mix-zone is to make it more difficult to link the different actions of an individual. Areas or buildings with a high traffic are usually good candidates for mix-zones.
- *Swapping* consists in exchanging the mobility traces of two different individuals for a certain period of time. For example, by swapping Alice's and Bob's traces during one day, their behaviours become more atypical and less predictable.
- *Removing* the mobility traces that are deemed too sensitive can also be considered as a sanitization procedure. In the same spirit, it is also possible to *add fake records* (called *dummies*) [31] inside the sanitized dataset to blend the true movements of individuals inside artificial data.

As sanitization leads to a loss of information, it is important to have a *utility metric* in order to compare the utility of the original dataset D and the sanitized one D'. The utility measure can either be generic, for instance linked to some global statistical properties of the dataset, or application-dependent, in which case it evaluates how well a particular application can be performed by using D' instead of D.

3 Mobility Markov Chain

In this section, we introduce a form of mobility model that we coin as *mobility Markov chain* that can represent in a compact yet relatively precise way the mobility behaviour of an individual. Basically, a mobility Markov chain is a probabilistic automaton in which states represent POIs and transitions between states corresponds to a movement from one POI (i.e. state) to another POI. The automaton is probabilistic in the sense that a transition between one POI to another is not deterministic but rather that there are a probability distribution over the transitions leaving from the current POI representing the probability. Note that Markov networks are a popular technique used for the study of motion (see for instance [30] for a recent work using hidden Markov networks to extract POIs from geolocated data). We will describe in Section 4.3 an algorithm that can learn the mobility Markov chain of an individual from his trail of traces.

More formally, a mobility Markov chain is a transition system composed of:

- *A set of states* $P = \{p_1, \ldots, p_n\}$, in which each state p_i corresponds to a POI (or a set of POIs). These POIs may have been learned for instance by running a clustering algorithm on the trail of mobility traces from an individual or simply by collecting the locations that he has posted on a geolocated social network such as Foursquare or Gowalla. Each state (i.e. POI) is therefore associated with a physical location. In the mobility Markov chains considered in this paper, it will often happen that p_1 is the "home" of this individual and p_2 is his "work". Therefore, it is often possible to attach a *semantic label* to the states of the mobility Markov chain. The states are ordered by decreasing importance of the POIs they embody and the last state is often made of what we call the "infrequent POIs", which are POIs that have been visited several times by an individual but not on a frequent basis.
- *A set of transitions*, $T = \{t_{1,1}, \ldots, t_{n,n}\}$, where each transition $t_{i,j}$ represents a movement from the state p_i to the state p_j. Each transition $t_{i,j}$ has a probability assigned to it that corresponds to the probability of moving from state p_i to state p_j. Sometimes an individual can move from one POI, go somewhere else (but not to one of his usual POIs) and come back later to the same POI. For example, an individual might leave his house to go wash his car in a facility near his home and come back 30 minutes later. This type of behaviour is materialized in the mobility Markov

chain by a transition from one state to itself. For instance, according to the previous example if p_1 is the home of the individual then the transition $t_{1,1}$ would be assigned a non-null probability. The sum of the probabilities of the transitions leaving one state is equal to one, meaning that $\sum_{j=1}^{n} t_{i,j} = 1$. Note that the probability of going from state p_i to state p_j (for $i \neq j$) is generally different from the probability of going from stage p_j to state p_i (e.g. the probability of going from "home" to "grocery" is not symmetric with the probability of going from "grocery" to "home"), therefore in general $t_{i,j} \neq t_{j,i}$. Once the states of the mobility Markov chain have been learnt from a trail of traces, the transition probabilities can be easily estimated by simply counting for each state the number of movements leaving to each other state and then dividing by the total number of movements leaving from this state.

The mobility Markov chain can be represented either as a *transition matrix* or as a *directed graph* in which nodes correspond to states and there is a directed weighted edge between two nodes if and only if the transition probability between these two nodes is non-null. The sum of all the edges' weights leaving from a state is equal to 1.

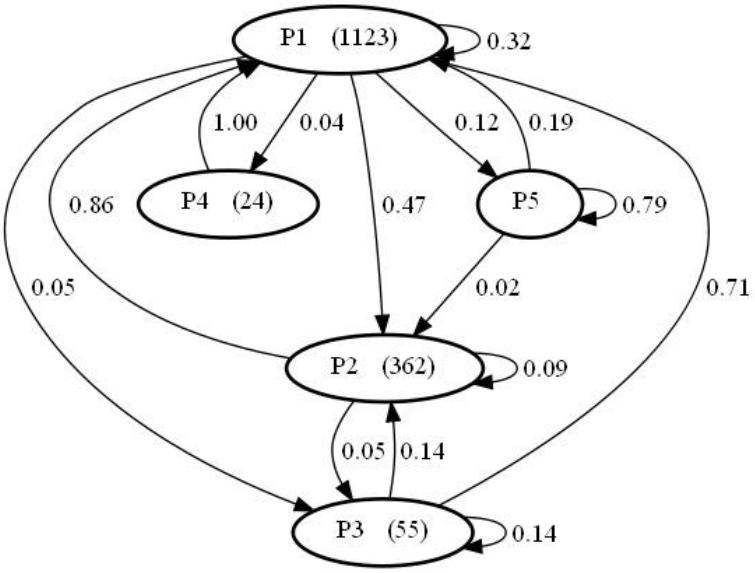

Figure 1: Mobility Markov chain from user 1.

For instance, consider for illustration purpose, an individual, that we refer thereafter as "user 1", who has a set of 4 important POIs that he visits often (extracted by a clustering algorithm) plus some other POIs that are less important to him. Therefore, we could define a mobility Markov chain composed of 5 states, one for each important POI plus a last one that will contain all the infrequent POIs. Thus, we have $P = \{p_1, p_2, p_3, p_4, p_5\}$. Suppose now that we have been able to learn the following mobility Markov chain (Figure 1) for this individual from his trail of traces. As additional information, in this Markov chain each state also has a weight associated to it in the form of an integer. We will see how to compute explicitly this weight in Section 4.3 but let suppose for now that this weight is given and is related (but not directly proportional) to the time spent by an individual in this state. As such the weight of a state gives an indication of the importance of a state and the states are ordered in decreasing importance of their weights, except the last state that is composed of all the infrequent POIs and his weight is not considered as meaningful.

Simply by looking at the structure of the Markov chain, it is easy to realize that the state p_1 is the only one that can be reached from all states. Moreover, this happens often with a relatively high probability

(except from state p_5). If we combine these observations with the high weight of state p_1, then we can infer with a relatively good confidence that p_1 might be the "home" of user 1. Afterwards, if we want to identify the place of "work" of user 1 and considering as a rule of thumb that "people often go from home to work and vice-versa", we can infer that state p_2 is a good candidate to be the "work" (which is also corroborate by the high weight of this state). Regarding state p_3, we can see that it is either reached from home or work and that a transition leaving this state is likely to take user 1 back to home. Applying now the rule of thumb that "at the end of the afternoon people often go to sport after work before coming back to home", we can infer that state p_3 is a place where user 1 plays sport on a regular basis. Finally, state p_4 can only be reached from home and can only lead back to home, therefore it makes a good candidate for an activity done on a regular basis during the week-end such as leisure or shopping to a nearby supermarket for instance. To summarize, we can attach the semantic label "home" to p_1, "work" to p_2, "sport" to p_3 and "leisure" to p_4 and by default "infrequent POIs" to p_5. Taking into account these labels and disregarding the weight of the states, we can now equivalently represent the mobility Markov chain of user 1 as the following transition matrix.

	P1 - Home	P2 - Work	P3 - Sport	P4 - Leisure	P5 - Inf. POIs
P1 - Home	0.321	0.469	0.049	0.037	0.124
P2 - Work	0.86	0.093	0.047	0.0	0.0
P3 - Sport	0.714	0.143	0.143	0.0	0.0
P4 - Leisure	1.0	0.0	0.0	0.0	0.0
P5 - Inf. POIs	0.2	0.02	0.0	0.0	0.78

Figure 2: Mobility Markov chain of user 1 represented as a transition matrix.

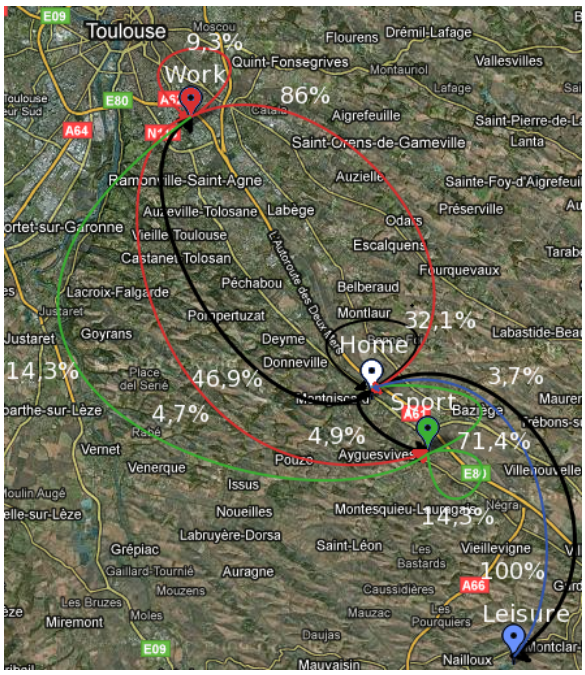

Figure 3: Mobility Markov chain of user 1 displayed on a real map.

Moreover, we can also take the abstract structure of the mobility Markov chain and put it on a real map (disregarding the "Infrequent POIs" state), which gives the following result (Figure 3). The mobility Markov chain is a data structure representing the mobility behaviour in a compact yet accurate manner and as such it can be used to perform several inference attacks. For instance, the states themselves directly represent the most significant POIs of an individual and therefore they can be used to derive information about his center of interests. Moreover, if the adversary knows the current position of the individual and if this position corresponds to a state of the Markov chain, he can predict the next movement of the individual by randomizing over the transition probabilities leaving from the current state. The same kind of reasoning can be used to predict the past locations visited by an individual or even guess his actual position. If a semantic label can be attached to some states of the mobility Markov chain (obtained for instance with the help of a reverse geocoding tool), then the mobility behaviour can be analyzed in a much deeper way. Imagine for instance that the adversary has been able to learn the mobility Markov chain of an individual and that he knows that this individual is also contained inside a geolocated dataset that has been pseudonymized. The pseudonymity of the individual can be lift fairly easily by finding inside the dataset the individual whose mobility Markov chain is the most similar to the one learnt previously. Of course, this either requires to compute a metric measuring how two different mobility Markov chains are similar to each other or to evaluate the likelihood that a specific trail of traces is compatible with the Markov chain. It is even possible to imagine to use the Markov chain as a generative model for synthetizing artificial data of trail of traces.

4 GEoPrivacy Enhancing TOolkit (GEPETO)

In this section, we report on our ongoing work towards building a generic toolkit for evaluating both sanitization methods and inference attacks on geolocated data. In particular, we introduce the architecture of the toolkit and we describe some clustering algorithms, which can be used as inference attack, and evaluate their efficiency for the identification of POIs, even after the application of sanitization mechanisms such as sampling and perturbation. We also describe an algorithm for learning mobility Markov chains. All these algorithms are currently implemented within the toolkit.

4.1 Architecture of GEPETO

The global objective of GEPETO (for *GEoPrivacy Enhancing TOolkit*) [7] is to provide researchers concerned with geo-privacy with means to evaluate various sanitization techniques and inference attacks on geolocated data. GEPETO has an interface for the management of geolocated data and offers several ways to manipulate this data such as sanitization mechanisms, inference attacks and a visualisation tool to display this data on a world map. The main idea is to offer a generic and flexible tool so that anyone can easily plug a new sanitization technique or inference attack. Moreover, the utility and visualization components provide means to evaluate the benefits of sanitization with regard to the success of inference attacks. To the best of our knowledge, there is almost no previous work that have tried to integrate all these features into a unified approach, with the exception of tools developed within the GeoPKDD project (and now the subsequent MODAP project) [8]. Another notable exception is [21] that tries to model formally the knowledge of the adversary with respect to the locations of individuals, and the possible counter-measures that these individuals might apply.

GEPETO is designed following a multi-layer architecture with the intended goal of making the system functional, efficient, scalable, easily modifiable and reliable. First, the data layer is a set of classes managing the communication with the database server for inserting, updating and deleting geodata. A control layer is in charge of the presentation, the local management and control of the data and provides a model of the data. The application layer is where the utility functions, the inference attacks and sanitization techniques are implemented. Finally, the visualization layer constitutes the

graphical user interface of GEPETO in which the user can load geolocated data, apply inference attacks and sanitization mechanisms and visualize the corresponding results. This layered architecture is targeted to provide a clear separation between data access and data presentation, so that it is easy to implement new algorithms in the application layer without worrying about how the control and the presentation layers will access and visualize data. In GEPETO, the presentation layer uses external web-services for the visualization of the data such as Google Maps or Yahoo Maps. The design choices behind this architecture imply both benefits and drawbacks; GEPETO cannot be used offline as it needs access to the database server as well as to the Internet in order to visualize data, but the implementation and maintenance are handled more easily this way, with a clear separation between the database and the visualization parts.

4.2 Description of the Clustering Algorithms

We describe thereafter succinctly the clustering algorithms that we are currently implemented within GEPETO and that we have evaluated during our experiments.

- *Density-Joinable cluster* (DJ Cluster) [24] is a clustering algorithm taking as input a minimal number of points $minpts$, a radius r and a trail of mobility traces M. This algorithm works in three phases. First, the preprocessing phase discards all the moving points (whose speed is above ϵ, for ϵ a small value) and then, squashes series of repeated static points into a single occurrence for each series. The speed of each point is computed by measuring its Euclidian distance divided by its time difference with its predecessor. Then, the second phase clusters the remaining points based on neighbourhood density. More precisely, the number of points in the neighbourhood must be equal or greater than $minpts$ and these points must be within radius r from the centroid of a set of points. Finally during the last phase, the algorithm merges the clusters which share at least one common point.
- *Density-Time cluster* (DT Cluster) [14] is an iterative clustering algorithm taking as input a distance threshold d, a time threshold t and a trail of mobility traces M. First, the algorithm starts by building a cluster C composed of all the consecutive points within distance d from each other. Afterwards, the algorithm checks if the accumulated time of mobility traces within range is greater than the time threshold t and created a cluster added to the list of POIs outputted if it is the case. Finally as a post-processing step, DT Cluster merges the clusters whose centroids are less than $d/3$ far from each other.
- *Time-Density cluster* (TD cluster) is a novel clustering algorithm inspired from DT Cluster. The main motivation behind this algorithm was to design a clustering algorithm inspired from DT clustering algorithm but more resilient to distortion. The TD clustering algorithm takes as input parameters a radius r, a time window t, a tolerance rate τ, a distance threshold d and a trail of mobility traces M. The algorithm starts by building iteratively clusters from a trail M of mobility traces that are located within the time window t. Afterwards, for each cluster, if a fraction of the points (above the tolerance rate τ) are within radius r from the centroid, the cluster is integrated to the list of clusters outputted, whereas otherwise it is simply discarded. Finally, as for DT Cluster, the algorithm merges the clusters whose centroids are less than d far from each other. See Algorithm 1 for a brief description of this method.

4.3 Learning Mobility Markov Chains

In this section, we describe an algorithm for learning mobility Markov chains that we have implemented within GEPETO. At a high level, the algorithm (Algorithm 2) starts (line 1) by applying a clustering algorithm on a trail of traces of an individual in order to identify clusters of locations that are significant. Then, in order to reduce the number of resulting clusters, the algorithm merges clusters whose medoids are within a predefined distance d of each other (line 2). This merging is not

Algorithm 1 TD clustering algorithm

Require: Trail of (mobility) traces M, time window t, radius r, tolerance rate τ, distance threshold d

Initialize N has being the number of records in the trail of traces M (i.e. $M.length$) and $cumulTime = 0$
Set L, the list of POIs found, has being the empty list
Create a empty cluster C
for $i = 0$ to $N - 1$ **do**
  $cumulTime = cumulTime + (M[i+1].time - M[i].time)$
  if $cumulTime \leq t$ **then**
    Add the mobility trace $M[i]$ to cluster C
  else
    Compute the centroid of C
    $nbPtsOut = 0$
    for $j = 0$ to $C.nbPts$ **do**
      if $distance(C[j], C.centroid) > r$ **then**
        $nbPtsOut = nbPtsOut + 1$
      end if
    end for
    if $nbPtsOut/N < \tau$ **then**
      Add the cluster C to L
    end if
    Reset $cumulTime$ to 0 and create a new empty cluster C
  end if
end for
Merge clusters of L whose distance between centroids is less than d
return L, the list of POIs discovered (which are effectively the centres of the clusters)

Algorithm 2 Mobility Markov chain learning algorithm

Require: Trail of (mobility) traces M, merging distance d, speed threshold ϵ, time interval threshold $mintime$

Run a clustering algorithm on M to learn the most significant clusters
Merge all the clusters that are within d distance of each other
Let $listPOIs$ be the list of all remaining clusters
for each cluster C in $listPOIs$ **do**
  Compute the $time\_interval$ and the $density$ of C
end for
for each cluster C in $listPOIs$ **do**
  if $C.time\_interval > mintime$ **then**
    Add C to $freqPOIs$ (the list of frequent POIs)
  else
    Add C to $infreqPOIs$ (the list of infrequent POIs)
  end if
end for
Sort the clusters in $freqPOIs$ by decreasing order according to their densities
for each cluster C_i in $freqPOIs$ (for $1 \leq i \leq n-1$) **do**
  Create a state p_i in the mobility Markov chain
end for
Create a state p_n representing all the clusters within $infreqPOIs$
Let M' be the trail of traces obtained from M by removing all the traces whose speed is above ϵ
for each mobility trace in M' **do**
  if the distance between the trace and the state p_i is less than d and the state p_i is the closest state **then**
    labelled the trace with "p_i"
  else
    labelled the state with the value "unknown"
  end if
end for
Squash all the successive mobility traces sharing the same label into a single occurrence
Compute all the transition probabilities between each pair of states of the Markov chain
return the mobility Markov chain computed

performed in an agglomerative manner but rather a first pass is make on the clusters to determine which clusters are within d-distance from each other and then they are merged within a single global step. Each resulting cluster can be considered as a POI, for instance by taking the centroid or the medoid of the cluster to be the physical location of this POI. For each cluster (lines 4 to 6), we compute the number of mobility traces inside the cluster (which we call the *density* of the cluster) and the *time interval* (measured in days) between the earliest and the latest mobility traces of the cluster (line 5). The POIs (i.e. clusters) are then split (lines 7 to 13) into two categories; the *frequent POIs* that correspond to POIs whose time interval is above or equal to a certain threshold $mintime$ and the *infrequent POIs* whose time interval is below this threshold $mintime$. In the set of frequent POIs (line 14), we sort the POIs by decreasing order according to their densities. Therefore, the first POI will be the denser and the last POI the less dense.

Now, we can start to build the mobility Markov chain by creating a state for each POI within the set of frequent POIs (lines 15 to 17) and also a last state representing the set of infrequent POIs (line 18). As evoked in Section 3, each state is then assigned a weight that we set to its density. Afterwards (line 19), we come back to the trail of traces that have been used to learn the POIs and we remove all the moving points (whose speed is above ϵ, for ϵ a small value). Then, we traverse the trail of traces in a chronological order (lines 20 to 26) labeling each of the mobility traces either with the tag of closest state (POI) of the mobility Markov chain (line 22) or with the tag "unknown" if the mobility trace is not within d-distance of one of the frequent or infrequent POIs (line 24). From this labeling, we can extract sequences of locations that have been visited by the individual in which all the successive mobility traces sharing the same label are merged into a single occurrence (line 27). For example, a typical day could be summarized as the following sequence "$p_1(home) \Rightarrow p_2(work) \Rightarrow p_3(sport) \Rightarrow$ "$unknown$" $\Rightarrow p_1(home)$", which is quite similar in spirit to the concept of semantic trajectory [2,27]. From the collection of sequences extracted, we can estimate the transition probabilities between the different states of the mobility Markov chain by counting the number of transitions between each pair of states and then normalizing these probabilities (line 28). If we observe a subsequence in the form of "$p_i \Rightarrow$ "$unknown$" $\Rightarrow p_i$" then we increment the count from the state p_i to itself (which translates in the graph representation by a self-arrow).

5 Experimentations

For the sake of demonstration, we begin by illustrating how GEPETO can be easily used to infer some private data about the taxi drivers of San Francisco, such as their home address for example. This geolocated data is available on the Crawdad repository. At first, GEPETO can simply be used to visualize the various mobility trails, and to characterize the geolocated data. When visualizing the data on the San Francisco map, one can easily recognize some hotspots, such as the San Francisco International Airport or various train and taxi stations. These hotspots being places where the taxi drivers usually wait for customers during some period of time, many traces correspond to plots on these spots. GEPETO can thus be used to "manually" perform inference attacks by visualizing and mining geolocated data.

5.1 Playing with Heuristics

The first step was to explore the use of heuristics. For instance, by considering that the beginning and ending locations of the taxi drivers, for each working day, might convey some meaningful information. This is the purpose of the *begin and end location finder* inference attack [7]. This attack is a simple heuristic assuming that the first and last recorded locations in a working day correspond to the departure and arrival points from a POI. The intuition is that when there is no mobility trace measured during a period longer than a given time threshold τ, this means that the individual had a "mobility break" and the place where he took this break is likely to be a POI. If τ is chosen sufficiently large

(e.g., 6 hours), this POI may be the home of an individual where he went to sleep after his work. First, we parse the mobility trails by looking for such breaks and extract the mobility traces that occurred right before and right after.

We must say that this attack has been very fruitful. A first interesting inference was the identification of location of the taxi company main parking. Indeed, many taxi drivers depart and arrive to this location after their working day, as they park their cab at the company headquarters. We were able to formally verify this statement simply by using the San Francisco Yellow Pages. The second category of statements that could be inferred from this attack directly concerns private information of the individual taxi drivers[12]. During this study, we examined the trails of 90 individual taxi drivers chosen at random in the dataset. We used GEPETO to visualize the data of these 90 taxi drivers after applying the *begin and end location finder* inference attack, manually picking those whose geolocated data seemed the most vulnerable. For 20 of these 90 taxi drivers, the visualization of the resulting data result in a narrow neighbourhood for their homes with a pretty high confidence. Note however that, as we do not have the real addresses of the taxi drivers, we were unable to formally validate these statements. However, we were able to some public data sources, such as Google Maps and Street View, to validate some of the inferred data. Indeed, for 10 of the 90 taxi drivers checked, the attack resulted in an address (or a small portion of a street) where the taxi was parked during most of the breaks. This address is most probably the home address of the taxi driver. Figure 4 shows the result of a successful attack, together with a Google Maps view and a StreetView of the address. For the remaining 70 taxi drivers examined, the *begin and end location finder* inference attack simply already identified hotspots (taxi stations, ...) that were already known.

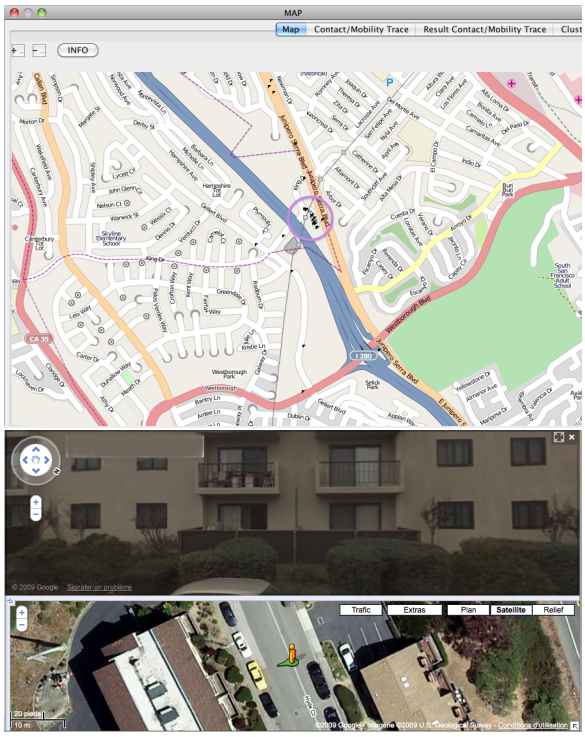

Figure 4: A successful *begin and end location finder* inference attack.

[12]It is worth noting that for protecting their privacy, we blurred their address. However, the interested reader can obviously find the actual information by applying the same algorithms we did on the original dataset.

5.2 Experimenting with Clustering

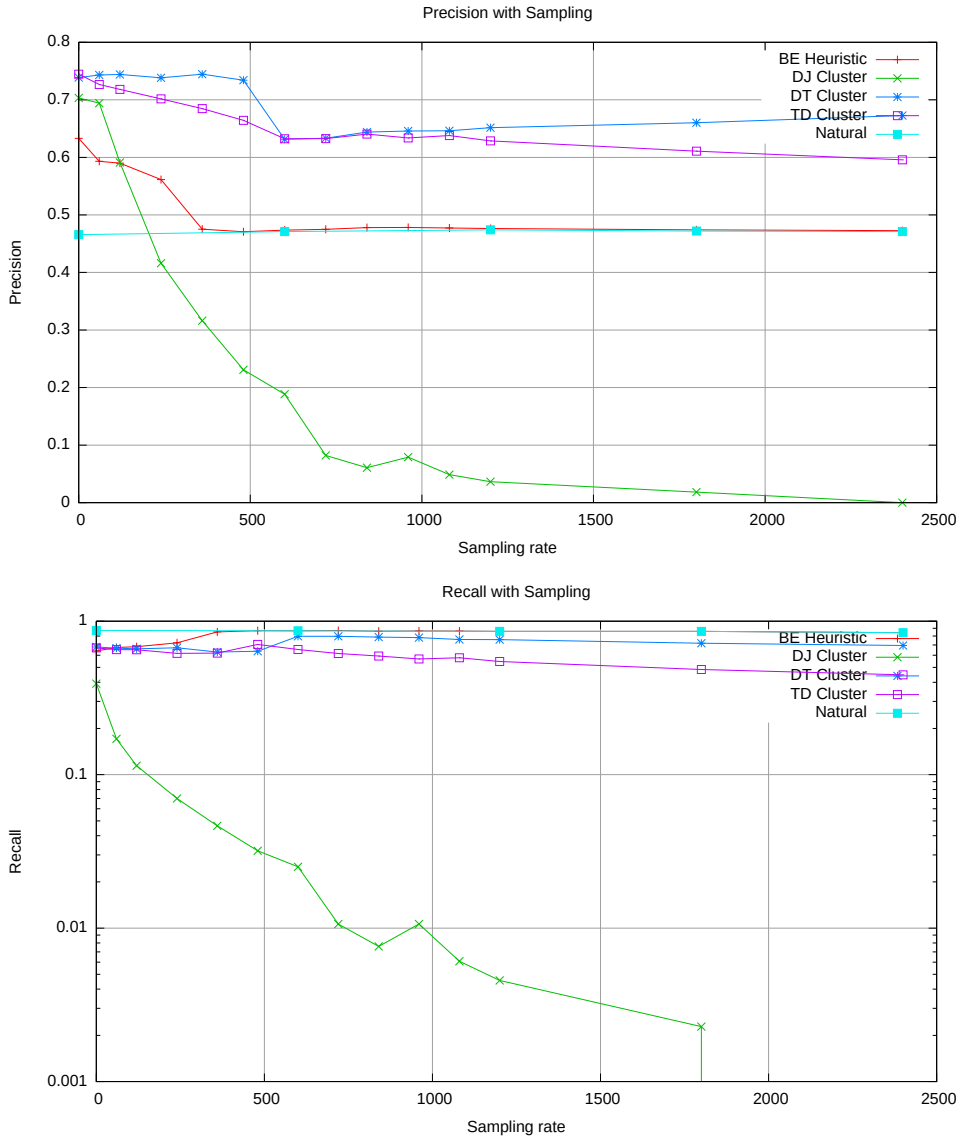

Figure 5: Precision-recall with sampling.

The next step was to implement two clustering algorithms described in the literature (Density-Joinable Cluster [24] and Density-Time Cluster [14]) and then to compare them with the Begin-end heuristic and to our own novel clustering algorithm (Time-Density clustering).

These four algorithms (the Begin-End heuristic and the DJ, DT and TD clustering algorithms) were implemented within GEPETO and applied to the taxi dataset for identifying POIs. We used both the original and sanitized versions of the taxi dataset to evaluate the resilience of the inference attacks against sanitization. More precisely, we applied both sampling and perturbation techniques (cf. Sec-

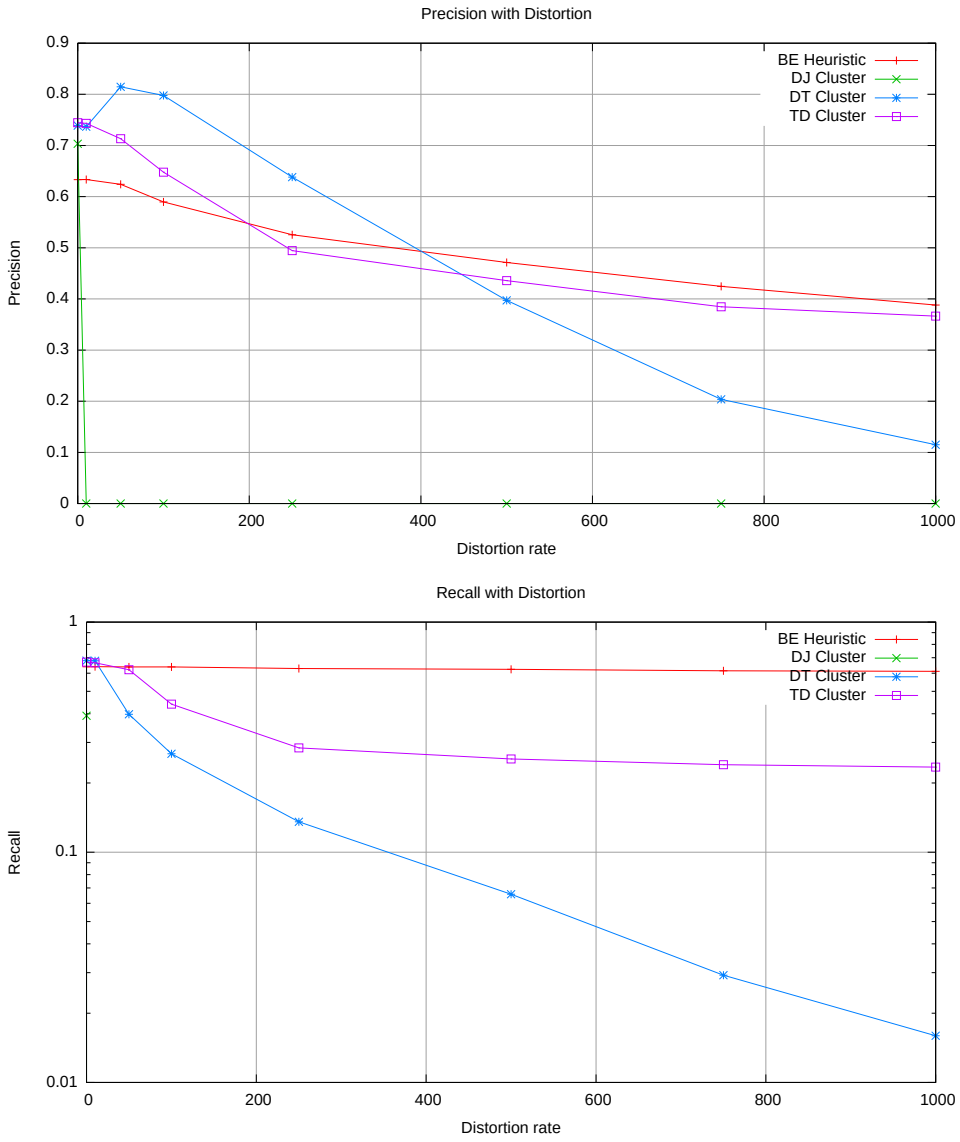

Figure 6: Precision-recall with perturbation.

tion 2.4) with various ranges of parameters. In each situation, we evaluate the recall and precision of the produced POIs.

This recall-precision evaluation requires to be able to judge whether or not a POI is "correct". To automatize this process, we defined 6 areas in San Francisco that make good candidates for real POIs and, which are at the same time generic enough: the taxi company parking lot, the main train station, the airport, the city center and three entertainment areas (the Castro district, Fisherman's Wharf and the Golden Gate recreational park). The *precision* is defined as the ratio between the number of correct POIs and the total number of POIs returned by an algorithm. In our experiments, a POI is considered "correct" if it falls inside one of the 6 ground truth areas. The *recall* is the ratio between

the number of area detected (i.e. hit by at least one POI) and the total number of areas. According to these definitions, an algorithm randomly generating many POIs would have a high recall and a low precision, as it would probably identify all the areas but many POIs would fall outside many of them. An "ideal" algorithm, displaying a high recall ad high precision, would generate 6 POIs, one for each area.

Figure 5 measures the recall-precision trade-off of the 4 algorithms against a sampling technique. We also evaluated "Natural", a naïve algorithm that directly outputs all the points of the dataset as POIs, which results in a low precision but a high recall. In Figure 6, the evaluation is performed for the 4 algorithms against random perturbation. These experiments have shown that the Begin-End heuristic has an excellent recall, due to the high number of POIs generated and an average precision but is sensitive to sampling. Indeed, when the sampling rate reaches the size of the time window of the begin-end heuristic, it considers all the traces as POIs and is as precise as the "Natural" algorithm. On the other hand, the begin-end heuristic is not too impacted by perturbation and even under large distortion, it stays one of the more precise algorithm. DJ Cluster displays a terrible behaviour in the presence of sampling, and even a worst one with respect to distortion. Indeed, the first phase of the algorithm removes the moving traces to focus on those where the individual is not moving. With sampling, the probability is high that static traces are removed by the sanitization process. Moreover, under the action of perturbation, every single trace implies some movement. Henceforth, all the traces are removed during the first phase of the algorithm and DJ Cluster does not output any POI. DT Cluster is highly resilient against sampling, with a high recall and the best precision, but displays a bad recall against distortion. However, the precision of the remaining POIs is still good under moderate distortion. Finally, TD Cluster seems to be a good compromise. For instance, its behaviour is comparable or just below DT Cluster in the presence of sampling with average to good recall and precision. Moreover, under distortion, it seconds the Begin-end heuristic with an average recall and a high precision.

To summarize, the efficiency of the inference attack depends strongly on the sanitization process that has been performed on the target data. For instance, in the presence of sampling, the DT cluster algorithm offers a high recall and a good precision, but its performance degrades significantly with respect to distortion. Therefore, if the sanitization process is only based on sampling, then the adversary can directly choose the DT cluster algorithm for performing an efficient inference attack. On the other hand, TD cluster seems to be a reasonable alternative for both sampling and distortion as as its performance remains good under these two type of perturbations.

5.3 Semantic Analysis of Mobility Behaviours

In principle, any clustering algorithm might be a valid candidate for building the initial clusters during the first part of the algorithm but in practice we have observed that out of the 3 clustering algorithms mentioned in Section 5.2, DJ cluster was the one that leads to the most meaningful results. In the rest of the section, we report on experiments conducted on mobility data gather through the Phonetic project [18]. The aim of this project is to build realistic mobility models out of real data as well as to study the privacy risks associated with this type of data. Therefore, the goals of this project are closely related to the ones of GEPETO. In this project, Nokia 5800 smartphones have been distributed to registered participants. These smartphones are equipped with a a GPS chip, an accelerometer, a compass, a WiFi and a bluetooth interface. The Phonetic software installed on the smartphones measures every minute the GPS position of the owner of the smartphone as well as the bluetooth neighbourhood. Actually, the example used in Section 3 to illustrate the concept of mobility Markov chain was learnt from the mobility data collected from one of the user of Phonetic. In the rest of this section, we discuss the mobility Markov chains learnt from three other users of Phonetic that we refer thereafter as "user 2", "user 3" and "user 4". Contrary to the previous quantitative experiments conducted on the mobility traces of the taxi drivers of San Francisco, the following evaluation has a more "qualitative flavour" in the sense that we really focus on the study of the mobility of a few individuals through their mobility Markov chains. The mobility Markov chains have been

learnt with a merging distance d of 200 meters and a value of the time interval threshold $mintime$ of 25 days. Previously, we have also tried to learn the mobility Markov chains of taxi drivers but as we expect the Markov chains we obtained were quite complex and difficult to interpret, with a big number of states mainly corresponding to hotspots in the city of San Francisco frequently visited by tourists.

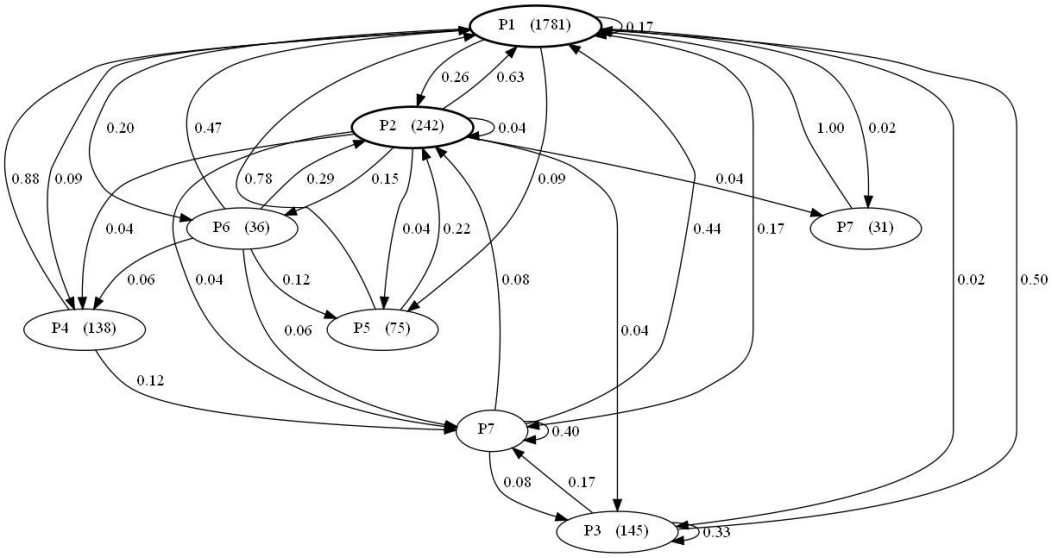

Figure 7: Mobility Markov chain from user 2.

Figure 7 shows the mobility Markov chain learnt from the trail of traces of user 2. Contrary to user 1, the number of frequent POIs is much higher, thus indicating potentially a more complex mobility behaviour. However, it remains fairly easy to identify the home (state p_1) as the POI that has the biggest number of arrows pointing to it. The work (state p_2) can also be inferred straightforwardly by looking at the transition leaving from the home that has the heaviest weight. Some states such as p_3, p_4, p_5, p_6 and p_7 are more difficult to interpret although they both have a high density. However, it is still possible to use a reverse geocoding tool such as Googlemaps to find the name of the closest physical address associated with the coordinates of this states. For state p_3, we obtain the address of a house in a small village approximately 150 kilometers from the home of user 2, which could be indicative of the home of some relative. The confidence in this guess could be strengthen if state p_3 is mainly visited during the week-end or holidays periods. For state p_4, the physical address corresponds to a plaza in the middle of the city in which user 2 lives, which could be for instance a frequent rendezvous where user 2 regularly meets with his friends. States p_5 and p_7 are located inside the university and can be accessed from home or work, which could be an indication that user 2 is either a student or a professor. Finally, state p_6 corresponds to the entrance of a park in a residential area close to which there are a few shops and schools and therefore is more ambiguous and difficult to interpret.

As shown by Figure 8, user 3 seems *a priori* to have a very complex mobility behaviour. We can start the analysis of this mobility Markov chain by labeling the state p_1 as "home" but then we are faced with the dilemma that state p_2 does not seem to be a valid candidate for "work" as the transition probability from state p_1 to p_2 is very low. Rather, it seems that state p_3 is a more likely candidate although, it is less dense that p_2, as the transition probability from p_1 to p_3 is greater than the transition probability from p_1 to p_2. To clarify this situation, we have used the reverse geocoding tool and observed that actually the states p_1 and p_2 are located in two different countries. Therefore, instead of being considered as "home" and "work", they should be labelled as "home from country 1"

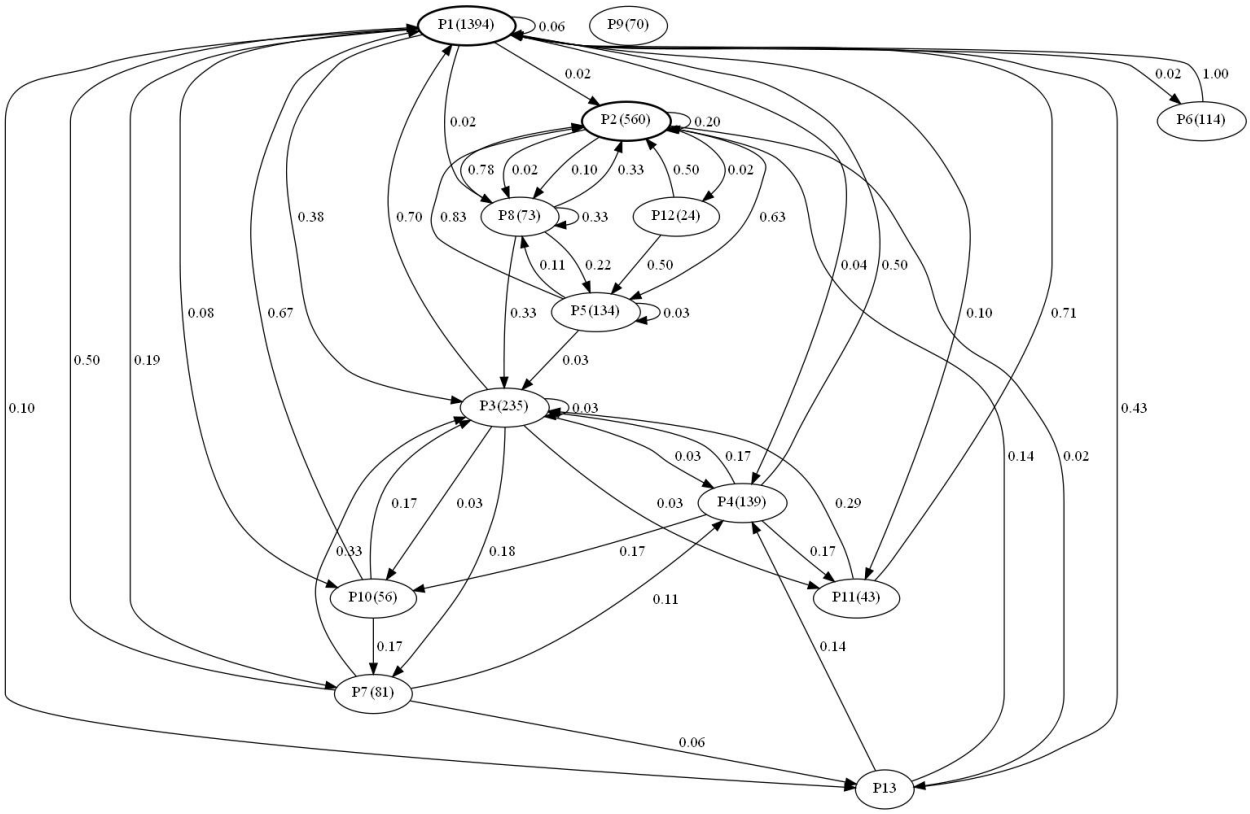

Figure 8: Mobility Markov chain from user 3.

and "home from country 2". Taking this new knowledge into account, we can label as "work from country 1" and "work from country 2", respectively the states p_3 and p_5 as they can be reached by the heaviest transitions leaving from states p_1 and p_2. We can now separate the states depending on whether or not they correspond to POIs of "country 1" or "country 2". Of course, this can be done straightforwardly with the help of a reverse geocoding tool but instead this could be learn also directly from the structure of the mobility Markov chain. For instance, if we start a random walk from state p_1 (home of country 1) for a few steps then we are likely to end up in one of the following states: p_3, p_4, p_6, p_7, p_{10} or p_{11}. On the other hand, if we were to begin the random walk on state p_2, after walking for a few steps we have a high probability of ending in state p_5, p_9 or p_{12}. This means that from the structure of the graph we could potentially infer the existence of two highly connected components, one for "country 1" composed of states p_1, p_3, p_4, p_6, p_7, p_{10} and p_{11} and the other for "country 2" composed of states p_2, p_5, p_9 and p_{12}. State p_8 seems to be in none of the two components and indeed a query to the reverse geocoding tool reveals it to be a house in a small village located in "country 2" but quite far from the "home of country 2". As for user 2, this may be the home of a relative or close friend of user 3 that he visits either before or after going to "home of country 1" or "home of country 2". By using the reverse geocoding tool and combining it with the results of the mobility analysis, we can find a name for each state of the mobility Markov chain of user 3 (except state p_6) as illustrated by the following table.

State	Label	Country
p_1	Home of country A	A
p_3	Work of country A	A
p_4	Sport	A
p_6	???	A
p_7	Parking	A
p_{10}	Restaurant	A
p_{11}	Nightclub	A
p_2	Home of country B	B
p_5	Work of country B	B
p_9	Shopping mall	B
p_{12}	Plaza in city center	B
p_8	House of relative or close friend	B
p_{13}	Unf. POIs	

Finally, we finish by analysing the mobility Markov chain of user 4 (Figure 9). From the weight of the different states of the Markov chain, it is easy to see that user 4 is an individual that has contributed so far very seldom to the Phonetic project and thus displays a very simplified mobility behaviour. Therefore, there is not much to be inferred from the his Markov chain except that state p_1 is likely to be his "home" and state p_2 should be his "work". If we query the reverse geocoding tool with the coordinates of p_3, we obtain a street in the town center around which there is a high number of restaurants.

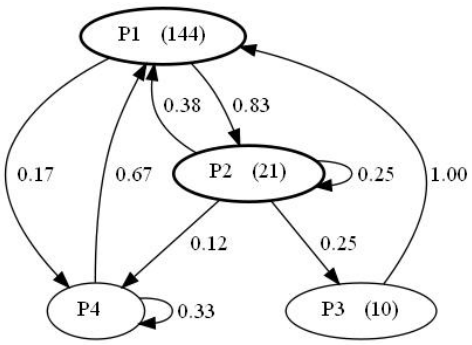

Figure 9: Mobility Markov chain from user 4.

We have also tested conducted some preliminary experiments for evaluating the behaviour of the mobility Markov chain under some sanitization procedures such as perturbation and downsampling. Basically, we have observed that the mobility Markov chain is relatively robust and that even under significant perturbation or a low rate of sampling, it is still possible to identify the home and the place of work of an individual simply by looking at the structure of the chain. However, it can happen that the less dense states (i.e. POIs) are not preserved under high perturbation and also that the transition probabilities are slightly different from the original ones. This is especially true in the situation in which some states are not preserved and the probability mass of the transitions pointing to/leaving from them is redistributed over the other existing transitions.

6 Conclusion

From the point of view of the adversary, these experiments show that the behaviour of the clustering algorithms can diverge significantly depending of the circumstances, for instance when sanitization is applied. On the other hand from the point of view of the data curator looking for the best sanitization method to protect privacy while preserving some utility in the geolocated datasets published, the conclusion is different. For instance, both the DT and TD clustering algorithms are quite resilient to sampling. Moreover, regarding perturbation, it seems that no clustering algorithm (among those we evaluate) performs a better precision than 50% under a distortion of magnitude 400 meters. A fundamental interrogation is whether or not the data remains useful with such a high level of distortion. As proof of concept, these experiments have demonstrated the usefulness of GEPETO as a tool to evaluate various algorithms for attacking or sanitizing geolocated data, but of course this is only a first step and more exhaustive experiments, with more sophisticated inference attacks and sanitization methods, remain to be done. Moreover as briefly highlighted previously, there is a strong interplay between the geolocated data of an individual and its social network in the sense that knowledge about one can help infer new information about the other (and *vice-versa*). We plan to investigate the inference attacks combining location and social knowledge and integrate them in GEPETO.

To summarize, rather than simply applying a particular sanitization mechanism with the hope the privacy guarantees offered will be sufficient, curators can used GEPETO to guide them on how to sanitize geolocated data before they release it publicly and to quantify how the sanitized data will be robust in the face of particular inference attacks. Although this method is not perfect as it does not capture all the possible inference attacks, GEPETO can be used to assess the privacy risks incurred by releasing a particular sanitized dataset as well as the utility remaining in this data. An analogy could be with be made with the security community in which usually when a particular information system is deployed (or even before), its security level is evaluated by using methods such as software for automatically detecting vulnerabilities or the use of red team. In order to avoid a "privacy by obscurity" approach, we should assume that the adversary knows the particular sanitization process used by the curator but that he does not have access to the internal random coins used by the curator during the randomization process.

We have also seen that the mobility Markov chain is a highly compact yet relatively precise representation of the mobility behaviour of an individual. By analyzing the structure of the Markov chain (and this even without knowing the coordinates of its states), it is sometimes possible to derive non trivial information about an individual such as his home (i.e. the state that can be reached from almost all the states) and his work (i.e. the state that can be reached with the heaviest transition from the home). Moreover, more advanced knowledge might also be derived by looking for particular patterns in the graph such as the presence of a particular cycle or the existence of two different highly connected components that can indicate two different geographical areas. In the future, we plan to investigate in a more systematic and theoretical manner the knowledge that can be inferred from the mobility Markov chain of an individual. We also want to integrate the time dimension directly into account in the design of the mobility Markov chain (and not indirectly just by looking at the transitions between states). For instance, more information about time could be integrated on the edge in the form of time interval during which the user is likely to take this transition or on the states to set a probability distribution of the different periods of the day representing how likely the user will be located on this state.

Being able to *quantify privacy* with respect to a particular geolocated dataset is another fundamental issue as it can be used for instance to measure the privacy gained by using protection mechanisms (such as sanitization algorithms). Despite several propositions that can be found in the literature, the problem of finding relevant privacy metrics for geolocated data is still open for now. For instance, is an individual hidden inside a crowd gathered in a small area really more protected in terms of privacy than an individual alone in the middle of a large area such as a desert? Or should we rather define privacy according to how much the behavior of an individual is indistinguishable from the behaviors of other (or a group of) users? One possibility is to study how anonymity is defined in

anonymous communication (e.g. the notion of anonymity set) [23] and how this applies to geolocated data. Taking into account unlinkability in the privacy metric seems to be particularly crucial in this context. Indeed, if the adversary can gather and link the movements of an individual during some period, he can build a complete profile of his behaviour if combined with other inference attacks.

In this paper, we have mainly focused on describing how to protect geoprivacy with sanitization procedures but of course other approaches are also possible. For instance by using *cryptographic primitives*, ubiquitous systems can perform computations which depend on their geolocated data in a secure manner such that only the output of the global computation is learnt (and nothing else). Moreover, *access-control mechanisms* can be used to control how an external entity accesses the geolocated data of individuals within a system. By auditing queries, it also can decide whether or not it should disclose more information since this could cause a privacy breach. In some sense, these two approaches are complementary to the sanitization one.

References

[1] O. Abul, F. Bonchi and M. Nanni, "Never walk alone: uncertainty for anonymity in moving objects databases," in *Proceedings of the of the 24th International Conference on Data Engineering (ICDE'08)*, 2008, pp. 376–385.

[2] L.O. Alvares, V. Bogorny, B. Kuijpers, J. Antônio Fernandes de Macêdo, B. Moelans and A.A. Vaisman, "A model for enriching trajectories with semantic geographical information," *ACM-GIS*, pp. 22, 2007.

[3] M. P. Armstrong, G. Rushton, and D. L. Zimmerman, "Geographically masking health data to preserve confidentiality," *Statistics in Medicine*, vol. 18, pp. 497–525, 1999.

[4] D. Ashbrook and T. Starner, "Learning significant locations and predicting user movement with *GPS*," in *Proceedings of the 6th IEEE International Symposium on Wearable Computers*, 2002, pp. 101–109.

[5] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, pp. 46–55, 2003.

[6] Z. Changqing, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personal gazetteers: an interactive clustering approach," in *Proceedings of the ACM International Workshop on Geographic Information Systems*, 2004, pp. 266–273.

[7] S. Gambs, M.-O. Killijian, and M. N. del Prado, "GEPETO: a GEoPrivacy Enhancing Toolkit," in *Proceedings of the International Workshop on Advances in Mobile Computing and Applications: Security, Privacy and Trust, held in conjunction with the 24th IEEE AINA conference, Perth, Australia*, April 2010.

[8] F. Giannotti and D. Pedreschi, *Mobility, Data Mining and Privacy Geographic Knowledge Discovery*, 2008.

[9] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," *Pervasive Computing*, pp. 390–397, May 2009.

[10] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, June 2008.

[11] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," *Proceedings of the ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003.

[12] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[13] J. Krumm and E. Horvitz, "Predestination: inferring destinations from partial trajectories," in *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp'06)*, 2006, pp. 243–260.

[14] R. Hariharan and K. Toyama, "Project lachesis: Parsing and modeling location histories," *Lecture notes in computer science - Geographic information science*, vol. 3, pp. 106–124, October 2004.

[15] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 38–46, 2006.

[16] L. Jedrzejczyk, B. A. Price, A. K. Bandara, and B. Nuseibeh, "I know what you did last summer: risks of location data leakage in mobile and social computing," *Department of Computing Faculty of Mathematics, Computing and Technology The Open University*, November 2009.

[17] J. H. Kang, B. Stewarta, G. Borriello, and W. Welbourne, "Extracting places from traces of locations," in *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, 2004, pp. 110–118.

[18] M.O. Killijian, M. Roy and G. Trédan, "Beyond San Francisco Cabs: building a *-lity Mining Dataset," *Workshop on the Analysis of Mobile Phone Networks (NetMob)*, 2010.

[19] J. Krumm, "Inference attacks on location tracks," *Pervasive Computing*, pp. 127–143, 2007.

[20] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition using relational Markov networks," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005, pp. 773–778.

[21] D. Matt, K. Lars, and B. Athol, "A spatiotemporal model of obfuscation strategies and counter strategies for location privacy," *Lectures Notes in Computer Science*, vol. 4197, no. 4, pp. 47–64, 2006.

[22] M.E. Nergiz, M. Atzori and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," in *Proceedings of the SIGSPATIALACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS (SPRINGL'08)*, 2008, pp. 52–61.

[23] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology," February 2008.

[24] B. A. Price, K. Adam, and B. Nuseibeh, "Keeping ubiquitous computing to yourself: A practical model for user control of privacy," *Int. J. Human-Computer Studies*, pp. 228–253, 2005.

[25] D. Reiter, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[26] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[27] S. Spaccapietra, C. Parent, M.L. Damiani, J. Macedo, F. Porto and C. Vangenot, "A conceptual view on trajectories," *DKE Journal*, vol. 65, no. 1, pp. 126–146, 2008.

[28] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[29] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM'08)*, 2008, pp. 65–72.

[30] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra and K. Aberer, "SeMiTri: a framework for semantic annotation of heterogeneous trajectories," in *Proceedings of the 14th International Conference on Extending Database Technology (EDBT'11)*, 2011, pp. 259–270.

[31] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Proceedings of the 2007 International Conference on Mobile Data Management*. IEEE Computer Society, 2007, pp. 278–282.