Processing Uncertain Spatial Data Resulting from Differentially-Private Sanitization

Mihai Maruseac, Gabriel Ghinita
University of Massachusetts, Boston, USA
Email: {mmarusea,gghinita}@cs.umb.edu

Abstract

The unprecedented revolution in mobile computing provides users with the ability to participate in applications that are customized to their spatial coordinates. Location-based social media and location-based recommendations are only two examples of popular scenarios where the use of geographical location can significantly improve user experience. However, serious privacy concerns arise when sharing users’ locations, as an adversary may be able to derive sensitive personal details from one’s whereabouts, such as health status, political or religious orientation, alternative lifestyles, etc. Privacy models such as differential privacy (DP) are commonly employed to protect individuals’ whereabouts before sharing. Typically, privacy is achieved by introducing uncertainty with respect to a user’s location. In this setting, uncertain data processing techniques become a natural choice for processing user location data that have been previously sanitized to protect privacy. In this article, we discuss some prominent scenarios where it is important to protect location data, and we explain how the de-facto standard of differentially-private location protection can be used in conjunction with uncertain data processing. We also look at a highly promising use case scenario of interest, namely privacy-preserving spatial crowdsourcing, and provide an overview of how DP and uncertain data processing are combined to address this problem.

1 Introduction

The spectacular advent of smartphone technology coupled with the prevalence of broad-coverage Wi-Fi networks has ushered the world into a new era which enables people to find personalized information customized to their current locations. This is made possible by the emergence of applications which are either completely centered around locations, or deeply enhanced by accessing information pertaining to the whereabouts of users. For example, a mobile application can use GPS sensor data in order to provide more accurate information to the user. This is the case of applications such as Google Maps and Waze for navigation and routing. Other applications use location data to offer meaningful social interactions for users (e.g., FourSquare, Facebook); to allow access to better search results (e.g., Google Search, Bing); to provide metadata tagging of events (e.g., Instagram), or even mobile entertainment applications (e.g., Pokemon Go). The majority of such applications will not function at all if the user does not allow the software to access her location data.

However, there are serious concerns that arise with respect to user privacy when location data are published or shared. An adversary with access to users’ whereabouts may be able to derive sensitive personal details about individuals, such as health status, political or religious orientation, alternative lifestyles, etc. If not properly addressed, such privacy breaches may have dire consequences for unsuspecting users, and in turn may lead to a backlash that risks to hinder development and adoption of location-based services.
To address privacy concerns, several privacy models and technical approaches have been proposed, ranging from spatial cloaking with $k$-anonymity and $\ell$-diversity [5, 2], to cryptographic approaches [4] and differential privacy (DP) [3], the latter representing currently the most popular approach. With the exception of cryptographic approaches, which are limited in query scope and are very expensive for most practical uses, all other approaches involve some sort of transformation that decreases the accuracy of location reporting. For instance, spatial cloaking techniques generate rectangular regions, and the user’s location is assumed to be uniformly distributed within the region’s area. DP-based approaches allow access to data through a statistical interface: the data processing algorithms are provided with noisy counts of the number of data points that fall within the requested region. The outcomes of these existing location protection approaches are naturally suited for processing using uncertain data techniques, as ultimately they reduce the amount of information that an adversary may gain on the actual users’ whereabouts.

In this paper, we explore this connection between privacy protection for location data and uncertain data processing. Specifically, we focus on the DP model, which is currently the de-facto standard of location protection. We give an overview of a framework that enhances location-based applications in such a way that the privacy of the users is protected while taking into account accuracy and performance requirements of applications. We begin by illustrating several common use-case scenarios where the privacy of individuals is at risk, and continue with a description of the state-of-the-art privacy preserving model of differential privacy (DP). To illustrate how the uncertainty introduced by DP can be handled to keep the utility of the application high, we present a specific use-case scenario, based on research results from [6]. We then conclude with a short presentation of future challenges and interesting issues pertaining to using differential privacy in spatio-temporal data processing.

2 Protecting Privacy of Location Data

First, we illustrate several examples of how data about individuals can be leaked through processing of spatio-temporal information. Next, we give an overview of differential privacy, which represents the state-of-the-art approach for privacy protection of spatial (as well as other types of) data.

2.1 Privacy Threats in Location-Based Services

An adversary with access to location data can infer sensitive details about an individual, and often the end users are not even aware of the privacy breaches that may occur. First, consider the Instagram application, which allows sharing photos after applying some image processing filters to improve quality. The application has a page on each user’s profile where photos are organized according to their associated geotags. This is a feature aimed at improving user experience by allowing other users to know which are the places where they can take interesting shots.

Nevertheless, as most users are unaware of the privacy risks, they are posting photos which are located near their home or workplace. Since the map is publicly available on the user’s profile, an attacker is able to use this information to determine an accurate estimation of the location of the user’s home and/or workplace address. As an example, Figure 1(a) shows a region of a map of a user visiting a city in Belgium. The density of photos and the moment these were posted allow a third party to determine the likely location of the hotel where the user stayed during the trip.

Although Instagram’s Photo Map FAQ suggests ways to remove photos from the map, and also recommends users to not publish images near relevant personal places, very few users are aware of the severity of such privacy threats.

As another service provider that uses extensively location information consider the example of Google. Anyone with a Google account (that is, anyone who uses any Google product including an Android device

\[^1\text{https://help.instagram.com/502180853174568}\]
or Gmail) can access a timeline of all her locations. The image in Figure 1(b) shows a set of interactions with Google’s servers during the work hours of one day. By analysing this information and cross-correlating it with geographical information about the places shown in the map, an adversary may determine that during that time interval, the user was mostly located in the area corresponding to University of Massachusetts Boston. By correlating information from similar days, the attacker may determine that the user works or studies at the university, hence he can start targeting him with related advertising.

Fortunately, the information presented in this map is only available to the user and Google’s machine learning algorithms. Nevertheless, if an attacker is able to log in into a user’s profile, he can then access this information at will and use it to stage further attacks on the user’s privacy.

Major web applications providers are publicly taking steps in limiting the privacy breaches that stem from location information. As an example, Facebook recently removed the location reporting facility from Facebook Messenger because an intern constructed a proof of concept application showing how people could abuse that to cause a significant privacy leak\(^2\). Using the GPS coordinates stored on each message sent through the Messenger mobile interface, anyone was able to accurately determine where people lived, worked, and spent their free time.

However, despite this step, just like in the case of Google, each account owner can still access her own location information by accessing a link on her profile. As a result, she is shown an interface like the one in Figure 2. This image shows all places where the user accessed a Facebook service during a trip to Hong Kong. Again, the densest areas can be used to determine information about the user: one corresponds to a popular attraction and the other is the place of a hotel. Thus, an adversary is able to determine which hotel the user owning the data lodged at, and then target her with similar advertisements.

The Google and Facebook scenarios are exacerbated by the fact that these service providers, among others, are selling user data to third-parties or are using the data to derive additional information about users through machine learning. Furthermore, even if users are not directly releasing their location data, their privacy can be harmed through cross-correlating data from multiple providers. As an example, consider the case of the Taxi and Limousine Commission (TLC) of New York: due to a Freedom of Information Act based request they are required to annually publish all data about all taxi trips around the city. As a result, an enormous amount of data

\(^2\)http://mic.com/articles/119526/here-s-how-anyone-can-stalk-your-location
is released annually and all data regarding taxi and Uber trips since January 2009 are publicly available on their website\(^3\).

While these data have been used for insightful analyses\(^4\) such as determining the frequency of cab rides from each area of the city, the distribution of tips based on fare and trip length, the average trip length to NY airports, or the impact of the advent of Uber, it can also be used by an attacker to find particular details about drivers or passengers. Knowing this, in order to offer some anonymization, the NY TLC hashed the taxi medallions and license numbers. However, it turns out that hashing is not sufficient to protect data. After some hours of de-anonymization, an attacker was able to accurately identify each taxi trip\(^5\).

2.2 A Primer on Differential Privacy

Early approaches for protecting location privacy relied on syntactical transformations of the data, e.g., spatial \textit{k-anonymity}[1,5] and \textit{diversity}[2]. The main idea of such privacy-preserving transformations was to construct a cloaking region that would satisfy some constraint (e.g., a minimum number of enclosed users, or a certain type of enclosed features). Due to the constraint being satisfied, an adversary’s ability of associating a user with a certain sensitive query or sensitive location is kept below a threshold probability. However, such approaches lack a formal model on their security guarantees, and, as exemplified by the NY taxi dataset, simple syntactic transformations are insufficient to protect the privacy of users, especially when the adversary is able to cross-correlate with other datasets.

As a solution to the lack of adequacy of the syntactic models, Dwork [3] proposed \textit{differential privacy} (DP). DP represents a complete \textit{paradigm shift} in the way location data is processed: while most classical algorithms were deterministic, always mapping the same database to the same output, DP is a non-deterministic approach. Any DP algorithm can be considered as a function from the dataset to a distribution over the outputs, from which one of the outputs is sampled. To ensure that any attacker, no matter how powerful and with how much additional information, cannot determine from the output of a DP algorithm if data from a user was included in the dataset or not, DP requires that the distributions obtained through running the same DP algorithm on two neighboring datasets (i.e., datasets differing by just one tuple) should be similar, as shown in Figure 3. The closeness of these two distributions is captured by the pointwise inequality

\[
Pr\{A(D_1) = o\} \leq e^\varepsilon Pr\{A(D_2) = o\}
\]

where \(A\) is the DP algorithm running on the neighboring datasets \(D_1\) and \(D_2\). The parameter \(\varepsilon\) is the \textit{privacy budget}: higher values of \(\varepsilon\) imply bigger differences in the probability distributions, hence less privacy.

\(^3\)http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
\(^4\)http://www.hubcab.org/#13.00/40.7219/-73.9484
\(^5\)https://medium.com/@vijayp/of-taxis-and-rainbows-f6bc289679a1#.h32lezam6
One simple way to achieve differential privacy is to add noise from the Laplace distribution to the output of the non-private algorithm. The probability density of this noise is given by $Lap(\Delta A/\varepsilon)$, that is:

$$f(x) = \frac{\varepsilon}{2\Delta A} \exp\left(-\frac{\varepsilon}{\Delta A}x\right)$$

where $\Delta A$, called the sensitivity of $A$, is the maximum difference in the output of $A$ when run on all possible pairs of neighboring databases.

As an example, consider the case where an analyst needs to determine the number of cars towed from a specific area of the city. The sensitivity of the query is 1 since adding a new car in the dataset can only change the output of the above query by 1. Hence, a differentially private answer to the query would be the real answer plus a noise drawn from distribution $Lap(1/\varepsilon)$.

So far, we have shown that in order to protect the privacy of the individuals, it is required to add noise to the dataset. It is, therefore, a well suited question to ask whether this noise won’t impact the accuracy of the data processing algorithms too much. It turns out that the error, both for the Laplace mechanism and for other DP building blocks, is always proportional to $1/\varepsilon$. Thus, the privacy budget serves a double role: on one hand it specifies the level of privacy protection, but on the other hand it controls the error incurred in the output of the algorithm.

To achieve good accuracy at reasonable levels of privacy, it is therefore important to build a private algorithm from several building blocks, by making use of the composition properties of DP. The sequential composition property states that running two $\varepsilon$-DP algorithms in sequence on the same dataset produces a $2\varepsilon$ DP algorithm, that is the privacy budget of the entire algorithm is the sum of the budgets of each step. In our example with the towed cars, if the analyst would then ask how many of the towed cars are registered to owners from outside the city, then this is a case of sequential composition.

On the other hand, the parallel composition property states that for algorithms consisting on steps run on disjoint partitions of the dataset, the privacy budget is only the maximum budget required by each of the steps. For example, running a counting query with privacy budget $\varepsilon$ on disjoint subsets of a dataset results in a $\varepsilon$-DP algorithm. For the towed cars example, this is equivalent with an analyst asking how many cars where towed in two disjoint regions of the city: a newly towed car can increase only one of the answers by 1, so the total privacy budget is just $\varepsilon$.

These composition properties allow us to handle the cost of using DP: for some applications, the injected noise can be prohibitively large in the case of na"ively adapting a non-private algorithm. Thus, the development of new algorithms to boost accuracy by combining individual steps becomes a necessity. Next, we will illustrate how a good-accuracy DP-algorithm can be devised and how the resulting output can be processed to obtain accurate results in a real-world location-based application.
Use Case: Differentially-Private Crowdsourced Data Collection

Due to the widespread availability of mobile devices with a broad array of sensing features and the ability to interface with external sensors, environmental sensing using crowdsourcing has become increasingly popular. This growing trend is beneficial for a wide range of applications, such as pollution levels monitoring or emergency response. In such a setting, authorities can quickly and inexpensively acquire data about forest fires, environmental accidents or dangerous weather events.

One particular task that is relevant in this scenario is that of detecting anomalous phenomena. Typically, this requires to determine a heatmap capturing the distribution of a certain sensed parameter (e.g., temperature, CO2 level) over a geospatial domain. When the parameter value in a certain region reaches a predefined threshold, then an alarm should be triggered, signaling the occurrence of an anomaly. Furthermore, the alarm should identify with good accuracy the region where the dangerous event occurred, so that countering measures can be deployed to that region.

However, there are important privacy concerns related to crowdsourced sensing. Contributed data may reveal sensitive private details about an individual’s health, lifestyle choices, and may even impact the physical safety of a person. To protect against such disclosure we present a solution using DP, based on the research from [6].

Consider the example of a forest fire, where mobile users report air temperature in various regions. To model the fire spread, one needs to plot the temperature distribution, which depends on the values reported by individual users, and the users’ reported locations.

We consider a two-dimensional geographical region and a phenomenon characterized by a scalar value (e.g., temperature, or CO2 concentration) within domain $[0, M]$. A number of $N$ mobile users measure and report phenomenon values recorded at their location. If a regular grid is super-imposed on top of the data domain, then the histogram obtained by averaging the values reported within each grid cell provides a heatmap of the observed phenomenon. Since our focus is on detecting anomalous phenomena, the actual value in each grid cell is not important; instead, what we are concerned with is whether a cell value is above or below a given threshold $T$, $0 < T < M$.

To protect the privacy of the mobile users we propose the architecture in Figure 4 where each user reports her sensed values to a trusted data collector who, in turn, sanitizes the set of reported values according to differential privacy with parameter $\varepsilon$, producing a data structure representing a noisy index of the data domain (i.e., a private spatial decomposition, PSD). This data structure is then released to data recipients (i.e., general public) for processing. They are able to answer queries with arbitrary granularity, as is suitable for their specific data uses. Furthermore, this model allows each data recipient to choose a different threshold value $T$ in their analysis. In practice, the trusted collector role can be fulfilled by cell phone companies, which already know the locations of mobile users, and may be bound by contractual obligations to protect users’ location privacy. The collector may charge a small fee to run the sanitization process, or can perform this service free of charge, and benefit from a tax break, e.g., for supporting environmental causes.

![Figure 4: System Model](image-url)
To construct the index, one could partition the dataspace according to a regular grid and split the available privacy budget between two aggregate query types, one counting user locations in each grid cell, and the other summing reported values. These values can then be used to compute the heatmap by dividing the two quantities to obtain an average of the values in each cell.

However, to increase the accuracy of the future processing steps, we generalize the above procedure for several levels. At each level, recursively for each cell, we compute a noisy estimate of the number of users and the sum of their reported values and decide based on these whether to stop the recursion or split the cell into a number of subcells to which the same algorithm would be applied. The goal is to get an estimate of the phenomenon distribution that is as accurate as possible within the limits of differential privacy. The main issue is that, whereas the computations between two cells of the same level of the hierarchy are independent and can thus use the parallel composition property, the computations of noisy values for cells from two different levels are sequential, thus requiring a splitting of the privacy budget between levels. This in turn imposes a limit on the height of the resulting PSD or increases the inaccuracies within it.

To solve this issue, as illustrated in Figure 4, we introduce a data processing algorithm to be run by each data recipient after receiving the finalized PSD from the trusted collector. We assume that the recipient is interested in building a heatmap according to a recipient resolution grid (rrg). Recall that our solution is designed to be flexible with respect to recipient requirements, and each recipient may have its own rrg of arbitrary granularity. The objective of heatmap construction is to determine for each rrg cell a binary outcome: positive if the value derived for the cell is above \( T \), and negative otherwise.

Figure 5 shows an example of rrg superimposed on the PSD index. The PSD has three levels, the root being split into four cells. The bottom layer in the diagram represents the rrg. The shaded cell in the rrg layer represents the cell for which the algorithm is currently determining the outcome.

Since the recipient has no other information other than the PSD, we assume that the count and sum values inside a PSD cell are uniformly distributed over the cell’s extent. Hence, for each rrg cell we compute \( \overline{n} \) (i.e., count of users located within cell) and \( \overline{s} \) (i.e., the sum of all sensed values within that cell) in proportion to the overlap between the rrg and PSD cells, normalized by the PSD cell area. If one rrg cell overlaps two or more PSD cells, the values for \( n \) and \( s \) are determined as the weighted sum of the values corresponding to each PSD cell, where the weight is represented by the overlap amount.

Note that, although the leaf nodes of the tree have a resolution that is closer to that of the rrg grid, the privacy budget allocated to these nodes is small so it is likely that the injected noise will overcome the actual values in the cell. On the other hand, while the values in cells closer to the root are closer to their non-private counterpart, the cells that are closer to the root are covering a large part of the domain, hence they smooth the phenomenon too much and using only these cells results in missing areas where the phenomenon is above the threshold.

In our solution, we account for these factors. Instead of naively dividing estimates for \( n \) and \( s \) in each rrg grid cell (which may have low accuracy), we evaluate individually the outcome based on information at each PSD level, and then combine the outcomes through a voting process in order to determine the outcome for each individual rrg cell.

Returning to the example in Figure 5, assume that threshold \( T = 80 \). We determine the outcome of the gray cell at the rrg layer by using the outcomes for all the marked PSD cells on the three levels shown (cells are marked using a small black square). Specifically, the Level 1 PSD cell containing the shaded grid cell has \( \overline{n} = 30 \) and \( \overline{s} = 1050 \), resulting in a phenomenon value \( \overline{\rho} = \frac{\overline{s}}{\overline{n}} = 35 \), below the threshold \( T = 80 \). Hence, the root cell’s vote would be negative, meaning that with the information from that layer, the grayed grid cell does not present an anomalous reading.

However, at Level 2 of the PSD, we have \( \overline{n} = 20 \) and \( \overline{s} = 1700 \), resulting in a value of 85, greater than the threshold. Hence, this layer will contribute a positive vote. Similarly, at Level 3, \( \overline{n} = 8 \) and \( \overline{s} = 800 \) which also results in a positive vote.

The resulting outcome for any rrg cell depends on the distribution of the votes it has received. We could use
the difference between positive and negative votes, but this will report a biased result for grid cells overlapping multiple PSD cells at the same level. A better solution is to use the ratio of positive votes to the total votes. In our example, the grayed cell got two positive votes and a single negative one, hence it would be marked as anomalous.

4 Conclusions

Location-based services are becoming a ubiquitous part of the mobile computing revolution. However, despite their positive impact, they also introduce serious privacy concerns. Existing techniques for location protection invariably inject a certain amount of uncertainty in the reported location data, whether they make use of spatial cloaking, where exact locations are replaced with regions, or whether random noise is injected in the answer to statistical queries, as in the case of differential privacy. There is an inherent connection between privacy-preserving transformations that reduce data accuracy and techniques for uncertain data processing. In this article, we studied the case of differentially-private location data protection, which is currently the de-facto standard for preserving location privacy. We illustrated how one can analyze and bound the inaccuracy introduced by differentially-private noise, and discussed potential approaches, such as voting, that deal with uncertainty.

References


