On-the-fly Privacy for Location Histograms

George Theodorakopoulos, *Member, IEEE,* Emmanouil Panaousis, *Member, IEEE,*
Kaitai Liang, *Member, IEEE,* and George Loukas, *Member, IEEE*

Abstract—An important motivation for research in location privacy has been to protect against user profiling, i.e., inferring a user's political affiliation, wealth level, sexual preferences, religious beliefs and other sensitive attributes. Existing approaches focus on distorting or suppressing individual locations, but we argue that, for directly protecting against profiling, it is more appropriate to focus on the frequency with which various locations are visited – in other words, the histogram of a user's locations.

We introduce and explore a new privacy notion, namely, on-the-fly privacy for location histograms, in which a mobile user repeatedly submits obfuscated locations to a Location-Based Service aiming for the resulting histogram to resemble a target profile or differ from it. For example, she may want to avoid looking wealthy or to resemble a health conscious person. We describe how to design concrete privacy mechanisms that operate under different assumptions on, e.g., the user's mobility, including provably optimal mechanisms. We use a mobility dataset with 1083 users to illustrate how these mechanisms achieve privacy while minimizing the quality loss caused by the location obfuscation, in the context of two types of Location-Based Services: nearest-PoI, and geofence.

Index Terms—Location privacy, Optimization, Histograms.

1 Introduction

A range of mobile devices, including smartphones, tablets, but also vehicles, can compute the location of the device and submit it to a Location-Based Service (LBS) at the user's request. Such services can be simple and standalone, such as queries for a nearby point of interest ("Find nearest restaurant"), or integrated into a larger service, such as a social network ("Which of my friends are nearby?" or checking into a particular cafeteria, restaurant, cinema) or a recommendation system (Facebook Places, Foursquare, Yelp, or [1]). Alternatively, mobile crowdsensing systems [2] engage users to submit their location together with a variable of interest, e.g., temperature or air quality at their location, thus helping to create a map with useful location-linked information. Finally, even web pages can request the user's location to provide appropriate customization, e.g., of search results.

However, the location information that LBSes need is transmitted out of the device, thus leaving the user's control. This loss of control entails a privacy risk, because location information is sensitive [3]. In certain cases, even a single visit can be detrimental to a person's privacy – consider a female teenager going to an abortion clinic. Even if an individual location visit is not sensitive, a sequence of locations visited in quick succession can be. Consider a construction company representative going to a bank, and then immediately visiting a city planning official. If considered separately, these events do not raise suspicion but jointly they are sensitive.

To address these concerns while not degrading application utility too much, researchers have proposed Location Privacy Preserving Mechanisms (LPPMs) for protecting either individual locations or trajectories [4], [5], [6], [7], [8], [9], [10], [11]. Individual Location Protection (ILP) mechanisms typically either define certain locations as sensitive and do not release them at all to the LBS, or they submit a fake location to the LBS instead of the true sensitive one. The aim is to prevent the LBS from inferring the true location. Trajectory protection mechanisms have the same aim, but the locations they aim to protect are visited by the user in quick succession and are therefore correlated. The task of the trajectory mechanisms cannot be reduced to just repeatedly applying an ILP mechanism. Because of the correlation, e.g., people don't usually move many hundred meters every few seconds, each fake location produced by the trajectory mechanism must be aware of and consistent with the previous ones, otherwise the attacker can detect and filter out the fake locations.

Our focus is on location *histograms* gradually formed by the user submitting locations to the LBS over time. In this context, we protect privacy in the following sense: the user modifies on the fly the locations she submits aiming for them to gradually form a histogram that resembles (or avoids) a certain target. For example, a user may want to avoid forming a location histogram like that of a wealthy tourist, because wealthy tourists may be shown advertisements for expensive products and services. Similarly, a user may want to resemble the histogram of a health conscious person, because that will reduce his health insurance premium or will increase his chances in a dating website. We argue that, for user profiling by advertisers, insurance companies and, generally, by data brokerage companies, histograms of locations matter very much in order to, for example, cluster users into various groups.

For on-the-fly privacy, the frequency of disclosure matters. We address users with *sporadic* location disclosure, not

- *G. Theodorakopoulos is with the School of Computer Science and Informatics, Cardiff University, Cardiff, UK.*
  E-mail: TheodorakopoulosG@cardiff.ac.uk.
- *E. Panaousis and G. Loukas are with the School of Computing and Mathematical Sciences, University of Greenwich, London, UK.*
  E-mail: {e.panaousis, g.loukas}@gre.ac.uk.
- *K. Liang is with the Department of Computer Science, University of Surrey, Guildford, UK.*
  E-mail: k.liang@surrey.ac.uk.

continuous disclosure. So, users submit locations only once in a while and thus the correlation problem in trajectory protection does not apply in our case. ILP mechanisms are more relevant, but in contrast to them we are interested in protecting the *frequencies* of visits to locations, which constitute a separate privacy risk (Figure 1) that is not the focus of ILP mechanisms and thus it is not appropriately addressed by them. The fundamental reason is that ILP can only reduce the frequency of a location, never increase it. Also, they can typically only reduce it to 0, not to any other value. Finally, they cannot make *targeted additions* to certain locations, which may be needed to achieve the new privacy notions that we define in this paper. We provide further high-level examples in Section 2, and we make a more concrete comparison to our algorithms in Section 6.6.

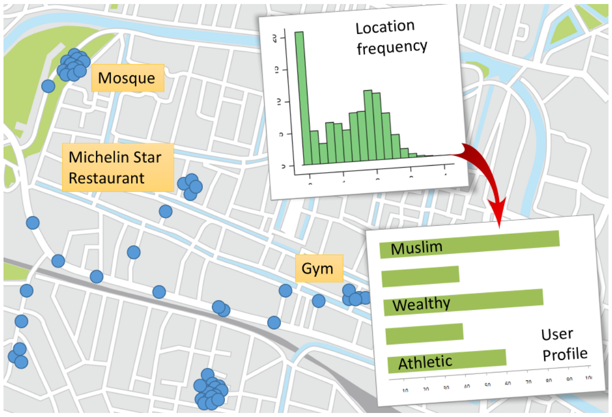


Fig. 1. Illustration of the location frequency profiling privacy threat. A user who frequently visits Mosques will be classified as Muslim; similarly for other frequently visited locations.

We summarize our contributions as follows:

- We introduce a new privacy notion: on-the-fly privacy for location histograms, in which a mobile user repeatedly submits obfuscated locations to a Location-Based Service aiming for the resulting histogram to resemble a target profile or differ from it. By "on-the-fly" we mean that the protection mechanism needs to intervene and obfuscate the location immediately when the user visits that location, instead of waiting to collect all visited locations and then obfuscating them all at once.
- We design a framework for constructing LPPMs that achieves the privacy objectives while minimizing LBS quality loss. The framework is statistical and it is based on likelihood-ratio tests. We show how optimal LPPMs can be constructed.
- We quantify how knowledge of the user's expected mobility helps achieve better privacy with the same or smaller quality degradation. We measure quality of loss incurred by users in a real dataset. Our data show that the quality of loss was low in our dataset. For example, for perfect privacy, the median user submits locations that are about 430m away from her true locations. For slightly less than perfect privacy, this number drops to 150m, which is a reduction of 65%.

2 Why Focus on Histograms Instead of Individual Locations?

As histograms are composed of visits to individual locations, one may think that an existing ILP approach that distorts/suppresses sensitive locations and thus removes them from the eventually formed histogram would be enough to protect histogram privacy as well. In this section, we give examples in which distortion/suppression of individual locations either simply is not powerful/flexible enough to achieve the privacy objective, or it protects privacy but introduces more noise than necessary. In Section 6.6, we make a more direct comparison between ILP and our own algorithms.

The fundamental inflexibility of ILP approaches is twofold: First, ILP only suppresses/distorts locations, therefore it can only lower the visit frequency to a location, never increase it. However, an increase may be necessary to achieve the privacy objective. In this case, an ILP would be completely helpless in achieving the objective. Second, ILP behaves identically every time the user visits the same location, so, in conjunction with the previous observation, we conclude that ILP's only effect is to reduce a location's visit frequency to zero. However, the privacy objective may just need a low frequency value, not necessarily zero, therefore reducing the frequency to zero would unnecessarily affect quality.

Example #1: In a very conservative state that monitors the whereabouts of all its citizens, all are expected to have at least a minimum frequency of visits to religious establishments or places of worship. Similarly, a Health Insurance Company announces that it will raise the premium for all clients whose total visits to gyms, sports centers, and other health-related locations are below a certain minimum frequency. The latter is more than just an example: Unhealthy users can be denied health insurance, as they constitute a larger risk for the insurance company [12].

In both these cases, the user needs to increase her visits to certain types of locations. ILP cannot help, because it can only decrease visits.

Example #2: The Health Insurance Company announces a premium increase for clients whose visits to junk-food restaurants, night-clubs and other locations related to unhealthy lifestyle (poor eating, drinking, not sleeping adequately) are above a certain level.

In this case, an ILP will reduce the frequencies of such visits to zero, but this is unnecessary for privacy and therefore unnecessarily detrimental for quality. Just reducing them to below the acceptable level would suffice.

Note also that, in both Examples #1 and #2, the privacy objective is about increasing or decreasing visit frequencies to *more than one location*. In other words, the relevant objective is a histogram, so the privacy mechanism needs to be aware of both the target histogram to resemble/avoid and of the user's evolving true histogram at the moments where the mechanism needs to make a decision. Neither of these histograms is taken into account by an ILP.

More complicated privacy objectives are also plausible: For example, the Health Insurance Company may increase the premium for users whose visits to junk-food restaurants exceed their visits to gyms. In that case, an ILP can help privacy by reducing the junk-food restaurants visits to zero, but again this may be unnecessary if the user is already visiting gyms more often.

Note that none of the above are defects of ILP mechanisms. They are not designed to achieve histogram privacy, so it is natural that they do not do very well.

3 Related Work

Research on location privacy is conducted both in the security and privacy community and in the database and data mining community. The latter community is concerned with bulk release of data [13]. This is very different from our on-the-fly setting in this paper, which is more common in the security and privacy community: protecting users that move around and send their location repeatedly to an untrusted system. The seminal paper in this line of research is by Beresford and Stajano [14], which introduced the general problem and proposed the concept of mix zones to protect user anonymity. Gruteser and Grunwald [15] were the first to recommended using obfuscation to distort locations before sending them out of the device. Later, Gruteser with Hoh et al. addressed location privacy for paths [16] as opposed to individual locations. Krumm [17] provides an excellent survey of research up until 2008.

Shokri et al. proposed a rigorous quantification of location privacy based on Bayesian estimation [18] and then described a way to protect privacy optimally for sporadic queries [9] and for continuous queries [8]. Andres et al. proposed geo-indistinguishability, an alternative characterization inspired by differential privacy [4] while Eltarjaman et al. introduced the (f, ϵ)-geo-indistinguishable principle [19]. Bringing together location privacy and differential privacy, Fawaz et al. proposed an online private location release mechanism that achieves differential privacy guarantees in indoor environments [20]. Most recently, privacy has been studied in the context of mobile edge computing and energy-aware security in cyber physical systems by Sangaiah et al. [21], [22].

Bindschaedler and Shokri proposed and evaluated the first formal and systematic methodology to generate fake yet semantically real privacy-preserving location traces [23]. They computed the semantic similarity metric between locations. Using this, they found the optimal way to map the visited locations in a pair of traces such that the mapping maximizes the statistical similarity between their mobility models.

None of the above papers aim to directly protect the *accumulated histogram of locations*: they are focused on either individual locations (sporadic disclosure) or trajectories (continuous disclosure). Research on protecting histograms exists only in the setting of interest to the database community. In that research, however, the whole histogram is known to the protection algorithm and it is processed/obfuscated in one go, not on the fly as in our approach. We refer the interested reader to Fanaeepour and Rubinstein [24] for a very recent effort on protecting a histogram in one go.

4 System and Attacker Model

We consider a user who moves within a set of discrete locations R. Each location in R is represented with a semantic label and a pair of geographic coordinates (latitude, longitude), e.g., ⟨`Seafood Restaurant, (40.781558, -73.975792)`⟩. At discrete time instants $t = 1, \ldots, T$, the user wishes to submit her location to an LBS to receive a useful service. The user's location at time instant t is denoted $r_t \in R$. We assume that successive time instants are far enough from each other for the locations to have no significant correlations. This is known as *sporadic* location disclosure in the literature, and it is compatible with LBSes like nearest-Point-Of-Interest or checking into locations, in which the user only discloses her location, e.g., about once a day.

The user wants to protect her privacy in the following sense: She wants her histogram of locations submitted up to each time t to resemble a target histogram H_t^{target}. Alternatively, she may choose to avoid it. A histogram H is a vector whose entries correspond to locations $r \in R$, and the entry for r, denoted $H(r)$, is equal to the number of visits of the user to r. When constructing a location histogram, only the semantic part of the location is taken into account, e.g., all mosques are grouped together.

To protect her privacy, the user employs a privacy mechanism LPPM that can distort locations before submitting them to the LBS. In particular, instead of submitting r_t, the LPPM picks and submits a location $r'_t \in R$ at each time t so as to keep the user's histogram of submitted locations H'_t similar to the target histogram H_t^{target} that she wishes to resemble, or to keep H'_t different from H_t^{target}, if she wishes to avoid the target.

We emphasize that the LPPM distorts locations *on the fly*. As soon as the user visits a location and tries to submit it to the LBS, the LPPM intervenes immediately and distorts it; it cannot wait until the user has visited all locations.

We use $d_p(H'_t, H_t^{\text{target}})$, or equivalently $d_p(t)$, to denote the distance between H'_t and H_t^{target}. This function quantifies privacy. We elaborate on how $d_p(t)$ is defined and computed in Section 4.1. To keep the presentation simple, in the rest of the paper we only focus on Target Resemblance. The mathematics for both Resemblance and Avoidance are very similar – we can derive equivalent algorithms by changing all $\leq$ to $\geq$ below.

Submitting a fake location r'_t instead of the true r_t improves privacy but causes some loss in the quality of service that the user receives from the LBS. So, the LPPM has a dual objective: protect privacy, but also not degrade quality. Note that the location that the LPPM submits will not always differ from the user's true location, but for simplicity we still refer to the submitted locations as "fake." We use $d_q(r_t, r'_t)$ to denote the quality loss when submitting r'_t instead of r_t, and we elaborate further on how it is defined and computed in Section 4.3. All notation and acronyms introduced in this section and in the rest of the paper are listed in Table 1.

4.1 Attacker model - Privacy Objective

We consider the attacker to be the LBS that compiles the fake histogram H'_t from the fake locations that the LPPM submits over time. The attacker's objective is to determine whether H'_t can have plausibly come from the same distribution as the target histogram H_t^{target} or not. Note this is a continuous objective: with each new received location, the attacker

TABLE 1
Notation and Acronyms

R	Set of locations within which the user moves
T	Length of time over which the user moves: $1, \ldots, T$
r_t	User's true location at time $t = 1, \ldots, T$
r'_t	User's submitted location at time $t = 1, \ldots, T$
$d_q(r_t, r'_t)$	Quality loss when submitting r'_t instead of r_t
H'_t	Histogram of locations submitted by the user at times $1, \ldots, t-1$
H_t^{target}	Target histogram to resemble (Target Resemblance problem) or to avoid (Target Avoidance problem)
$d_p(t)$	User privacy at time t
$d_p(H'_t, H_t^{\text{target}})$	Alternative notation for $d_p(t)$ to emphasize that user privacy is the distance between H'_t and H_t^{target}
c	privacy parameter; bound for $d_p(t)$
χ^2	Pearson's Chi-Square statistic
LBS	Location-Based Service
LPPM	Location Privacy Preserving Mechanism
ILP	Individual Location Protection

tries again to determine resemblance using all locations submitted up to that time.

The attacker has the target histogram H_t^{target} in mind, but he does not know that the user is trying to resemble that particular histogram. Actually, he does not even know that the user is trying to resemble any histogram. The attacker only knows the locations that the user is sharing and he aims to detect whether the histogram of these locations is similar to H_t^{target}.

This attacker model is in line with the examples in Section 2, e.g., Example #1 where a government (attacker) is monitoring the locations of all their citizens and they want to detect whether the histogram of locations of each citizen is similar to the "legal" or "orthodox" profile. In such a case, the attacker does not know that a particular citizen is trying to resemble a particular histogram. Also, he does not care about reconstructing the exact histogram of each citizen; he only cares whether it resembles the "legal" profile or not. If the attacker knows that a citizen is modifying the shared histogram to try to resemble the "legal" profile, then the citizen will just be arrested.

To quantify resemblance, the attacker considers that H'_t has been created by sampling t locations independently from an unknown distribution, and H_t^{target} is the expected histogram formed by sampling t locations from a target probability distribution h^{target}. Note that h^{target} does not change with time, whereas H_t^{target} does, so it is simpler to think that h^{target} is what the user chooses to resemble/avoid, and then H_t^{target} is created at each time t as $H_t^{\text{target}}(r) = t \cdot h^{\text{target}}(r)$.

At each time t the attacker chooses between two competing hypotheses, RESEMBLE and DIFFER:

RESEMBLE H'_t has been drawn from h^{target}.

DIFFER H'_t has been drawn from some arbitrary distribution on locations.

Intuitively, if the attacker chooses the RESEMBLE hypothesis, this means he believes that the submitted locations H'_t resemble the target histogram. If the attacker chooses the DIFFER option, this means he believes the opposite: the submitted locations H'_t do not resemble the target histogram. Because this is the Target Resemblance problem, the user aims to make the attacker choose RESEMBLE.

The attacker's task is a *hypothesis test* and to conduct such tests it is standard to use likelihood ratios[1]. The alternatives, such as Bayesian inference, would be applicable if the attacker had some prior probability about each hypothesis [25].

The attacker conducts the likelihood ratio test as follows: First, compute the probability of the observed data H'_t under the RESEMBLE hypothesis, $\Pr[H'_t; h^{\text{target}}]$. This is the likelihood of the RESEMBLE hypothesis. Then, compute the highest probability of H'_t under any distribution on locations, which gives the likelihood of the DIFFER hypothesis. The ratio of these two likelihoods is called *likelihood ratio statistic*, and it is equal to

$$\text{lrs}(t) = \sum_r H'_t(r) \log \frac{H'_t(r)}{H_t^{\text{target}}(r)}, \tag{1}$$

where r ranges over all locations in H_t^{target}, and $H'_t(r)$ and $H_t^{\text{target}}(r)$ are the frequencies of visits to location r in the respective histograms. In the literature this is called *simple goodness-of-fit test* [25, Sec. 4.2.3].

Another popular statistic is Pearson's chi-squared statistic:

$$\chi^2(t) = \sum_r \frac{\left(H'_t(r) - H_t^{\text{target}}(r)\right)^2}{H_t^{\text{target}}(r)}. \tag{2}$$

Either statistic would produce similar results. To keep the description general, we use the term *privacy distance* to refer to the preferred statistic, and we denote it by $d_p(t)$. At certain points where it aids presentation, we use $d_p(H'_t, H_t^{\text{target}})$ as an equivalent form for $d_p(t)$. In the evaluation we set $d_p(t)$ to $\chi^2(t)$.

The fundamental observation is that, if the RESEMBLE hypothesis is true, then $H'_t(r)$ and $H_t^{\text{target}}(r)$ will be close to each other for all locations r, and so $d_p(t)$ will take a small value. In contrast, if the DIFFER hypothesis is true, then $d_p(t)$ will be large. Therefore, the attacker will choose RESEMBLE if $d_p(t)$ is small and DIFFER if $d_p(t)$ is large.

The user selects a privacy parameter c and wishes to keep $d_p(t)$ below c, because in Target Resemblance the user aims to make the attacker accept RESEMBLE. The intuitive meaning of c is the following: If $d_p(t) = c$, then the user's submitted histogram is e^c times more likely to have come from an arbitrary distribution than from h^{target}. This is similar to the meaning of ϵ in differential privacy, in which the two hypotheses relate to two datasets that differ by one element.

The LPPM may aim to enforce the bound $d_p(t) \leq c$ at all times

$$\max_{t=1,\ldots,T} d_p(t) \leq c, \tag{3}$$

[1] For example, differential privacy is defined with the help of a likelihood ratio.

which means that the user wishes to make the attacker accept RESEMBLE at all times.

Alternatively, the user may care about resembling the target not at all times but only at a subset of times, for example just at the final time $t = T$,

$$d_p(T) \leq c, \tag{4}$$

which essentially means that the user only cares about resembling the target when she has submitted all her locations, but not before.

We refer to this bound as the *privacy constraint*. In the evaluation we explore both variants of the privacy constraint: strictly enforce at all times, and enforce at the end.

4.2 Mobility Knowledge

When choosing a fake location to submit, we expect it would help if the LPPM knew how the user moves within the set of locations R, but of course this knowledge may or may not be available. We distinguish two variants, both of which we explore in the evaluation:

4.2.1 No knowledge

The LPPM does not know how the user moves within R. In this case, when taking a decision at time t, the LPPM only knows the submitted histogram H'_t and the true histogram H_t.

4.2.2 Known mobility profile π

The LPPM knows that the user visits location $r \in R$ with probability $\pi(r)$, $\sum_r \pi(r) = 1$. This helps because the true histogram (normalized, i.e., divided by t) will eventually converge to π, so the LPPM can anticipate this convergence and choose fake locations in a better way.

Note that a new possibility opens up for the user: She can choose to aim for Target Resemblance in expectation, where the expectation is taken over the user's mobility profile π:

$$\mathbb{E}_\pi[d_p(t)] \leq c. \tag{5}$$

As above, the bound could be enforced at all times $t = 1, \ldots, T$, or just at $t = T$.

This is a less strict privacy objective than the ones listed above in (3) and (4), because it is possible that $d_p(t)$ will exceed c, but we expect that the quality loss would be smaller. In the long run, i.e., as t gets larger, the user's true (normalized) histogram will converge to π, so the probability of exceeding c would be negligible.

4.3 Quality Loss

The quality loss $d_q(r_t, r'_t)$ incurred when submitting r'_t instead of r_t to the LBS obviously depends on the kind of service provided by the LBS. For example, for an LBS that returns the nearest restaurant to the submitted location, d_q should increase proportionately with the geographic distance between r_t and r'_t. The problem with sending a fake location is that the LBS response will not be the nearest one to the user's true location, so the user will have to travel a longer distance.

In contrast, for a geofence-based LBS, the quality loss is not smooth. The only thing that matters is whether the submitted location and the true location are on the same side of the geofence or not, so the quality loss is binary: either zero or a maximum value.

In both these cases, though, the quality loss function depends on the geographic distance between r_t and r'_t. We use the term *geographic* d_q for those functions that depend on the geographic coordinates of r_t and r'_t, but not directly on their semantic labels.

Different yet are the considerations that apply for LBSes that provide location recommendations: d_q should be determined not by geographic distance, but by the semantic difference between the submitted location and the true location. The recommendation from the LBS may change when submitting e.g., a Bar instead of a Restaurant, but submitting one Bar instead of another makes no difference. We use the term *semantic* d_q for these cases.

Our approach applies for both geographic d_q and semantic d_q. In fact, we can accommodate any function as d_q, as long as $d_q(r', r) \geq d_q(r, r) = 0, \forall r', r \in R$. To be usable in our approach, d_q just needs to reflect that there is no quality loss when submitting the true location, and that submitting a fake location never achieves better quality than submitting the true location. Even though we use the term "distance," note that d_q does not need to satisfy all the distance axioms; in particular, it does not need to be either symmetric, $d(x, y) = d(y, x)$, nor to satisfy the triangle inequality.

The user wishes to *minimize* the total quality loss,

$$\min \sum_{t=1}^{T} d_q(r_t, r'_t), \tag{6}$$

or, when the user's mobility profile π is known, the user may wish to minimize the expected total quality loss:

$$\min \mathbb{E}_\pi \left[\sum_{t=1}^{T} d_q(r_t, r'_t) \right]. \tag{7}$$

To keep the presentation simple, in the rest of the paper we focus on geographic distances only. In the evaluation we use a distance-proportional d_q and a geofence-based d_q, with both the total and the expected variants. We use the Haversine geographic distance, and for the geofence-based d_q we select the geofence radius to be 200m, which aligns with Android's recommendation of at least 100-150 meters, although our approach can obviously be used with any other value[2].

4.4 Optimal Design of the Histogram Privacy Mechanism

Having defined all the components of our approach, we can now state the optimization problem that we solve to obtain the optimal LPPM. Intuitively, the problem is to find a rule for choosing fake locations r'_t that minimizes the quality loss, subject to the privacy constraint.

Problem (Optimal Privacy for Histograms)**.** *Given a target histogram over the locations R, a d_q quality distance function, a d_p privacy distance function and an associated privacy parameter*

[2] https://developer.android.com/training/location/geofencing.html#choose-the-optimal-radius-for-your-geofence

c, and possibly knowledge about the user's mobility, produce an algorithm that minimizes the quality loss, subject to the privacy constraint.

We make the following observations:

- This is a general problem that encompasses several variants, as discussed above. Solving the problem creates a *custom* LPPM for each input. For instance, different users have different mobility profiles and privacy objectives, and different LBSes lead to different d_q. Hence, our contribution is not a single LPPM, but rather a framework to construct LPPMs given the above mentioned inputs (Figure 2).
- This problem, and the analysis and the evaluation in the following sections, are about the *construction* of the LPPM, which only needs to be done *once*. Of course, in the case of no mobility knowledge, the construction is a greedy algorithm anyway, so it can be done as the user moves. As we will see, it is very computationally efficient. In the case of known mobility, the construction involves solving a constrained non-linear optimization problem, which would be done offline. After the LPPM is constructed, *using* it is a simple table lookup as we will see, so it can be used very fast and with trivial computational power. Of course, the table needs to be stored on the user's device, but the number of entries in the table are in the hundreds or low thousands, each of which is a real number, so the overall storage requirements are low.
- We have chosen to minimize quality loss subject to a privacy constraint. We could have instead chosen to achieve a guaranteed quality level, while minimizing the privacy loss. Mathematically, that would require a trivial modification to the problem statement and to our algorithm, i.e., only swap the objective function with the privacy constraint. Philosophically, both approaches make sense. If a user needs a privacy guarantee, then the approach we follow is more appropriate. If a user needs a quality guarantee, because without it the application could become hard to use, then the alternative is better. We favor having a privacy guarantee based on a privacy parameter as is typical in security and privacy mechanisms – e.g., ϵ in differential privacy or geoindistinguishability.
- We have left the choice of the target profile h^{target} to the user, because the user is best placed to know what inference they want to protect against. However, it may be difficult for a user to know what the location profile is e.g., for a typical wealthy user. This is an important question, but we do not address it in this paper. We expect that experts on city planning, demographics research, or location-based marketing could easily create a typical profile with locations in a city that wealthy users or other types of users visit. Specifically for wealthy individuals, anyone living long enough in a city could probably compile a list of places where the wealth hang out. Alternatively, a user may want to resemble a generic person's profile, which would be the average popularity of each location.

5 LPPM Variants

5.1 Baseline: No Protection

As a baseline to compare all other LPPMs against, we use an algorithm that merely submits the user's true location at all times to the LBS. Obviously, there is no quality loss in this case, but privacy is worse than other algorithms. For this algorithm, we measure and report privacy as $d_p(T) = \chi^2(T)$, i.e., the privacy distance function after the user has submitted all her locations to the LBS at time $t = T$. We call this algorithm No-Protection.

5.2 No Knowledge and Deterministic Enforcement

The next variant that we evaluate is one with no mobility knowledge and in which the privacy constraint is enforced at all times. The quality loss is the total loss, i.e., $\sum_{t=1}^{T} d_q(r_t, r'_t)$. We call this algorithm No-Knowledge.

Since No-Knowledge does not know anything about the user's mobility pattern, the selection at time t can only be done in a greedy manner, using the submitted histogram up to and including time t H'_t, the quality metric d_q, and the privacy constraint,

$$d_p(t) = \chi^2(t) = \sum_{r'} \frac{\left(H'_t(r') - H_t^{\text{target}}(r')\right)^2}{H_t^{\text{target}}(r')} \leq c. \quad (8)$$

No-Knowledge works as follows: Assume the user is at location r at time t. Among all possibilities for r' that do not violate the privacy constraint, choose the one with minimum $d_q(r, r')$. If all r' violate the privacy constraint, choose one with least violation. This last possibility is unavoidable in general, but the reason is more technical than substantial. If the privacy constraint is very strict, e.g., $c = 0$, then H'_t must be very close or identical to H_t^{target} for all t. But H_t^{target} may not have integer entries, as it may have been produced as $t \cdot h^{\text{target}}$, whereas H'_t will always have integer entries.

The problem with aiming for strict enforcement at all times is that it can result in avoidable quality loss by replacing r with r' and then, at a later time, replacing r' with r. For example, consider a target in which locations r_1 and r_2 must have equal frequencies $h^{\text{target}}(r_1) = h^{\text{target}}(r_2)$. Assume that, at first, the user visits r_1 frequently and r_2 rarely, and then later the opposite happens so the two frequencies would eventually balance each other out. This means the target would be eventually achieved without any intervention from the LPPM. However, the LPPM has no mobility knowledge, so it cannot know how the two frequencies will evolve, and also it aims to enforce the privacy constraint at all times. Therefore, it will at first distort r_1 to r_2, and then distort r_2 back to r_1, each time causing quality loss. This motivates why a quality-sensitive user may want to aim either for final-time enforcement at time $t = T$, as opposed to at all times, or for enforcement in expectation, as opposed to deterministic enforcement.

5.3 Known Mobility π and Enforcement in Expectation at time T

In this variant, which we call Known-Mobility, the LPPM knows that the user visits location $r \in R$ with probability $\pi(r)$, $\sum_r \pi(r) = 1$, and aims to make the final histogram H'_t

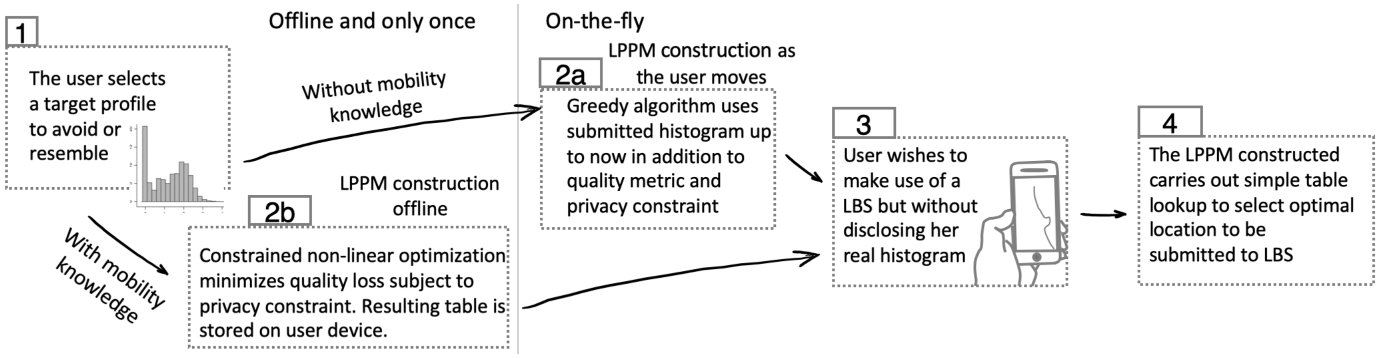

Offline and only once
On-the-fly
1
The user selects a target profile to avoid or resemble
Without mobility knowledge
With mobility knowledge
2a
LPPM construction as the user moves
Greedy algorithm uses submitted histogram up to now in addition to quality metric and privacy constraint
2b
LPPM construction offline
Constrained non-linear optimization minimizes quality loss subject to privacy constraint. Resulting table is stored on user device.
3
User wishes to make use of a LBS but without disclosing her real histogram
4
The LPPM constructed carries out simple table lookup to select optimal location to be submitted to LBS

Fig. 2. Illustration of the use of the framework for constructing and using LPPMs.

resemble the target in expectation. To achieve this, the LPPM will report a fraction $f_{rr'}$ of r-visits as r'-visits. So, for each location r', the reported frequency will be $T\sum_r \pi(r) f_{rr'}$ in expectation, and we want to compute $f_{rr'}$ values that make these reported frequencies such that the privacy objective is satisfied. The LPPM will then pick r' with probability $f_{rr'}$, independently each time the user visits r.

The expected number of times when the $r \to r'$ distortion will happen is $T\pi(r)f_{rr'}$, and this will cause expected total quality loss equal to $T\pi(r)f_{rr'}d_q(r,r')$. The overall optimization problem is then to choose $f_{r,r'}$ values that minimize the expected total quality loss, subject to the privacy constraint:

$$\text{Minimize} \sum_{r,r'} T\pi(r) f_{rr'} d_q(r,r') \tag{9}$$

$$\text{subject to} \sum_{r'} \frac{\left(T\sum_r \pi(r) f_{rr'} - H_T^{\text{target}}(r')\right)^2}{H_T^{\text{target}}(r')} \leq c \tag{10}$$

$$\sum_{r'} f_{rr'} = 1, \forall r \tag{11}$$

$$f_{rr'} \geq 0, \forall r, r'. \tag{12}$$

Even before solving this problem, we can prove the intuitive fact that there should not be distortion "flows" in both directions between two separate locations, i.e., both $r \to s$ and $s \to r$. Assume the opposite were true, $f_{rs} > 0$ and $f_{sr} > 0$ and also assume without loss of generality that $\pi(r)f_{rs} < \pi(s)f_{sr}$, i.e., that the $r \to s$ flow is larger than the $s \to r$. Then, the following transformation would reduce quality loss and it would still satisfy the constraints: First, reduce f_{rs} to zero and increase f_{rr} to $f_{rr} + f_{rs}$. This just means that location r will now keep to itself the flow that it was previously sending to s, which obviously is good for quality. Then, decrease f_{sr} to $f_{sr} - \frac{\pi(r)}{\pi(s)} f_{rs}$ and increase f_{ss} by the same amount. Again, this is good for quality, because s keeps to itself some of the flow it was sending to r. So, the overall quality is improved.

All constraints are still satisfied:

- The fractions f are still nonnegative and sum to 1 for both r and s: $\sum_{r'} f_{rr'} = 1$ and also $\sum_{r'} f_{sr'} = 1$.
- For r, the incoming flow $\sum_\rho \pi(\rho) f_{\rho r}$ was affected in two of its terms, $\rho = r$ and $\rho = s$: The first term was $\pi(r)f_{rr}$ and it became $\pi(r)(f_{rr} + f_{rs})$ when increasing f_{rr} to $f_{rr} + f_{rs}$, thus it increased by $\pi(r)f_{rs}$. The second term was $\pi(s)f_{sr}$, and it became $\pi(s)(f_{sr} - \frac{\pi(r)}{\pi(s)} f_{rs})$, thus it decreased by $\pi(r)f_{rs}$. So, the overall sum $\sum_\rho \pi(\rho) f_{\rho r}$ does not change.
- Similarly, for s, the sum $\sum_\rho \pi(\rho) f_{\rho s}$ does not change.

5.4 Perfect Privacy

In the setting when π is known, the case $c = 0$ implies that the user wishes to perfectly resemble the target profile in expectation. This is expected to cause a larger quality loss compared to $c > 0$, but it is a useful baseline in the opposite direction from the no-protection baseline. It shows how large the quality loss is for perfect privacy, so any algorithm or heuristic that causes more quality loss than this should not be considered.

Mathematically, this is a very interesting special case, because the privacy constraint is simplified and it becomes a linear function of $f_{rr'}$. In fact, as we now show, the quality loss is exactly equal to the Earth Mover's Distance (EMD) [26] between π and H_T^{target} and the $f_{rr'}$ values become equivalent to the flow coefficients that arise when computing the EMD.

Setting $c = 0$ in the optimization problem (9), the inequality constraint (10) becomes equivalent to:

$$T\sum_r \pi(r) f_{rr'} = H_T^{\text{target}}(r'), \forall r' \tag{13}$$

because all the squared terms in (10) must be equal to zero.

We now divide the first constraint by T and recall that $\frac{H_T^{\text{target}}(r')}{T} = h^{\text{target}}(r')$. We multiply the other two constraints by $\pi(r)$, and we set $g_{rr'} = \pi(r)f_{rr'}$. The resulting problem is exactly the EMD between π and h^{target}, with the "ground costs" being $Td_q(r,r')$.

$$\text{Minimize } T\sum_{r,r'} g_{rr'} d_q(r,r') \tag{14}$$

$$\text{subject to} \sum_r g_{rr'} = h^{\text{target}}(r'), \forall r' \tag{15}$$

$$\sum_{r'} g_{rr'} = \pi(r), \forall r \tag{16}$$

$$g_{rr'} \geq 0, \forall r, r'. \tag{17}$$

Note that this new problem is almost completely independent of T, as T only appears in the function to minimize. For any other value of $t = 1, \ldots, T-1$, the coefficients $f_{rr'}$

would be the same as for $t = T$, and the optimal value (the quality loss) would just be $t \sum_{r,r'} g_{rr'} d_q(r, r')$. So, we conclude that this problem computes the expected quality loss at each time instant, and we just need to multiply it by t to find the expected quality loss after t time instants. We call this algorithm Perfect Resemblance.

5.5 Run-time analysis

The time it takes to compute Known-Mobility depends on the solver that is used. There are standard solvers for non-linear constraint optimization problems that scale well for the size of problem that we consider in this paper. The number of variables $f_{rr'}$ to be computed is equal to the product of two numbers: The number of different geographic locations that a user visits, and the number of different semantic locations needed to specify the target. In practice, users do not visit many different locations. As we show in the Evaluation with our analysis of the dataset, 90% of the users visit less than 150 locations. More importantly, we do not expect the target histogram/profile to need many locations for its specification. For example, the "wealthy person" profile could be "Spends 50% of their time in Airports, Hotels, and Spas, equally spread, and the remaining time on any other location." So, in total, the problem may need up to 1000 variables, which takes a few minutes even on low-end laptops.

The time to compute Perfect Resemblance is at most that of solving a linear program, which is known to be polynomial. The EMD calculation is a special case of a linear program, a transportation problem, whose complexity is not resolved as far as we know, but it can be solved in $O(n^3 \log n)$ time as a minimum cost network flow problem, where n is the length of the histograms.

6 Experimental Evaluation

In this section we evaluate each of the described algorithms to establish its privacy-quality tradeoff. Note that the Perfect Resemblance algorithm is by construction optimal, and so is the Known-Mobility algorithm, given its constraints. So, the evaluation we present for these two algorithms should more appropriately be considered an illustration of the best possible privacy (for Perfect Resemblance) and the best possible quality for a given privacy constraint (for Known-Mobility). There can be no other approach or heuristic that performs better.

6.1 Experiment setup

We select a single semantic target histogram and, for a range of privacy parameters c, we compute the quality loss that each algorithm achieves when aiming to resemble that target with $d_p \leq c$. Because we choose Target Resemblance as the privacy notion to achieve, *lower c values are better*, because they imply closer resemblance to the target.

We consider two different types of LBS: a nearest-PoI LBS whose quality loss is equal to the geographical distance between the true location and the submitted location, and a geofence LBS whose quality loss is 0 if the geographical distance is less than a given threshold (200m) and 1 otherwise. So, we compute two d_q functions, one for each LBS, from the dataset that we describe below. Finally, for the algorithms that take as input the user's mobility profile π, this is also computed from the dataset.

6.2 Dataset

We use a dataset with 1083 users, each contributing a sequence of Foursquare check-ins in New York City [27]. Each check-in is a quadruplet `(user-id, label, latitude, longitude)` in which the label is a semantic label from Foursquare's categories [28], for instance `(985, Office, 40.74441206, -73.98341417)`. We treat the successive location visits by a user as being far enough in time from each other to be independent of each other. In other words, knowing where the user currently is, does not give any information about where the user will go next. For convenience, we still refer to each sequence of locations as a trajectory.

The dataset contains 227428 visits to 43207 unique locations. This is an average of 5.3 visits to each location, but of course there is large variance across locations. Location (40.75079479, -73.99357639), Pennsylvania Station, is visited 1147 times, making it the most visited location. These visits may have been by the same users over and over, so to measure location popularity we exclude repeated visits. Excluding repeated visits, the most popular location is the parking lot at JFK Airport Terminal 4, (40.64508935, -73.78452301), with 274 out of 1083 distinct users visiting it at least once.

Turning to semantic labels, there are 251 unique labels in the dataset. The most visited one is `Bar` with 15978 visits, and it is also the most popular after excluding repeated visits: 960 out of 1083 distinct users visited a Bar at least once. As with locations, label popularity varies a lot across labels.

When it comes to users, we see that the trajectories of different users vary widely both in the number of distinct locations visited and in the total number of location visits (length of each user's trajectory). More than 90% of users make up to 370 total visits each and the median trajectory length is 147 total visits (Figure 3). The total quality loss is a sum over all locations that a user visits, hence some users may incur a higher quality loss just because their trajectory is longer. We explore this issue further in Section 6.5.

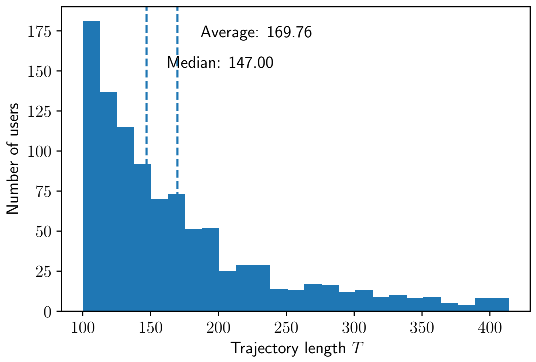


Fig. 3. Histogram of user trajectory lengths.

6.3 Computation of algorithm inputs d_q and π from dataset

Because geographic locations are given as (latitude, longitude) pairs, we use the Haversine distance function to compute d_q. For the nearest-PoI LBS, d_q is exactly equal to this distance (in km), while for the geofence LBS d_q is 0 if the Haversine distance is less than a threshold radius of 200m and it is 1 otherwise.

The user's mobility profile π is simply computed as the normalized histogram of locations that the user visits in the dataset, i.e., $\pi(r)$ is the user's relative frequency of visiting location r.

6.4 Privacy-quality tradeoff

We now illustrate the tradeoff between privacy and total quality loss in each of our algorithms. We emphasize that the actual values shown cannot be considered either good or bad. They are just the privacy and quality that users get if they want to resemble a certain target. For example, if a user deems, the quality loss for a certain value of the privacy parameter to be too high, then this just means that it is not possible for that user to achieve the desired privacy for an acceptable quality loss. The user must relax either the privacy goal or accept worse quality.

The target histogram we use as an example has four semantic labels, each with probability 0.125: Jewelry Store, Ski Area, Stadium, Pool. This is supposed to portray a rich athlete who spends 50% of their time spread equally across locations with these four labels, and the remaining 50% on any other locations. The user aims to resemble this target to within a d_p distance of c; the privacy parameters we use are $c \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, but we plot only $c \in \{5, 25, 50\}$ in most plots.

We first show the tradeoff for each algorithm separately, using the nearest-PoI LBS, so quality loss is measured in km. Then we compare all algorithms in one plot, which we also do for the geofence LBS.

No-Protection: This algorithm just submits the user's true location to the LBS without any modification. The quality loss is obviously 0 for both types of LBS. The privacy values that we show in Figure 4 indicate how close each user's true histogram is to the target. The reason for the observed variability is that the true histograms of some users already happen to be close to the target histogram, so even submitting their true histogram provides them a certain degree of privacy in the sense of Target Resemblance. Having said that, this algorithm provides the least possible privacy across all algorithms.

Perfect Resemblance: The privacy achieved in this algorithm (Figure 5) is the best possible across all algorithms, i.e., it corresponds to the case $c = 0$, but the quality loss is also the largest. In this sense, it is the polar opposite of the No-Protection algorithm. Note that, by construction, for Perfect Resemblance there is no tradeoff. Privacy is perfect (d_p is 0), so we report the total quality loss it produces for each user.

We observe a long tail in the quality loss values, evident also in that the average quality loss is more than 70% higher than the median. The median quality loss is 72.2km, which means that, over the course of a whole trajectory, the median user will submit locations that are in total 72km away from the true locations.

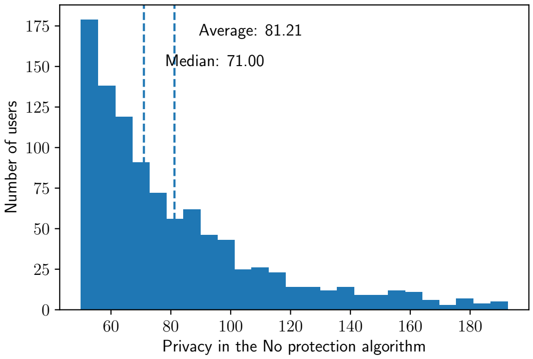


Fig. 4. Histogram of privacy values achieved by No-Protection. Because this mechanism just submits the user's true location, the figure essentially shows how far each user's true histogram is from the target histogram.

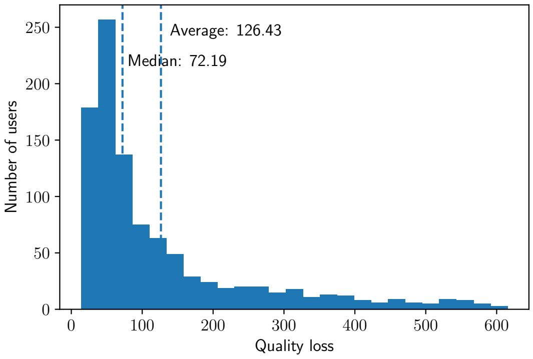
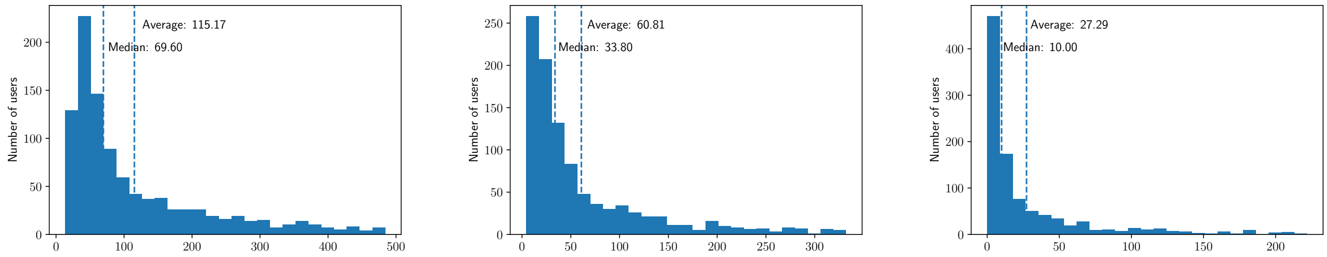


Fig. 5. Histogram of quality loss values achieved by Perfect Resemblance.

No-Knowledge privacy: For this algorithm, we show side-by-side the total quality loss for $c = 5, 25, 50$ in Figure 6. As in Perfect Resemblance, we observe that in each plot there is wide variability across users: In all cases, there are users with a quality loss four or more times higher than the average. When comparing across the three values of c, the median quality loss starts at 69.6km ($c = 5$), which is very similar to the 72.2km in Perfect Resemblance and indeed slightly better, even though Perfect Resemblance is a mobility-aware algorithm and No-Knowledge is not. This is the effect of $c = 0$ for Perfect Resemblance versus $c = 5$ for No-Knowledge: Perfection in privacy comes at a significant cost in quality. The relaxation in c from 0 to 5 is more than enough to counterbalance the lack of mobility knowledge.

Known-Mobility privacy: As with No-Knowledge privacy, we plot Known-Mobility privacy for $c = 5, 25, 50$ in Figure 7. This algorithm is expected to achieve the best quality loss, and indeed this is the case. Across all values of c, it produces values significantly lower than any of the other algorithms (except No-Protection of course). Note the horizontal axis range is different in each subplot (and different from the subplot ranges in Figure 6), so the quality



(a) No-Knowledge $c = 5$

(b) No-Knowledge $c = 25$

(c) No-Knowledge $c = 50$

Fig. 6. Quality loss for No-Knowledge, Nearest-PoI LBS.

values of Known-Mobility are even better than what a direct visual comparison suggests.

6.4.1 Comparative performance across algorithms

We now compare the privacy-quality tradeoff across algorithms for all ten values of c from 5 to 50 and for both the nearest-PoI LBS and the geofence LBS. For each algorithm, we show the quartiles of the quality loss (median, 0.25 and 0.75 quantiles), in Figure 8 for the nearest-PoI LBS and in Figure 9 for the geofence LBS with a radius of 200m.

For comparison, we also show the median privacy of the No-Protection algorithm, and the median and 0.25-0.75 quantiles of the Perfect Resemblance algorithm.

We observe that the quality loss for the Known-Mobility algorithm is consistently better than No-Knowledge, as expected, across both LBSes. The difference between the two quantifies the value of knowing the user's mobility.

In absolute terms, the Known-Mobility algorithm does not seem to have terrible quality loss, even for low values of c. Its median for $c = 5$ is significantly lower than even the 0.25 quantile of Perfect Resemblance, which again shows how much difference a relaxation in c makes. The same effect exists for both LBSes, and it is even more pronounced in the geofence LBS.

It is also interesting and somewhat surprising that there is non-negligible overlap between the quality loss values of No-Knowledge and Known-Mobility, even though the former is mobility-unaware. This is explained by the wide variability across users: Some users' true histograms are so dissimilar from their resemblance target that even Known-Mobility does not reduce their quality loss a lot. Conversely, some histograms are so similar to the target that even No-Knowledge, *not knowing* their mobility, does not hurt quality very much. On the whole, however, mobility clearly helps.

6.5 Impact of trajectory length

In the previous sections, we always show the total quality loss over the trajectory of a user. However, as we show when analyzing the dataset (Figure 3), users have different trajectory lengths, so users with long trajectories may have a systematically higher quality loss than users with short trajectories. We indeed observe this effect across all algorithms, and we show it in indicative plots (Figure 10) for Perfect Resemblance, No-Knowledge ($c = 5$), and Known-Mobility ($c = 5$).

To account for this effect, we now plot quality loss divided by trajectory length, i.e., we plot the quality loss per location visit. The general observation is that all loss-per-visit plots are less skewed than the corresponding total-loss plots, as evidenced by the averages being much closer to the medians.

In Perfect Resemblance we see that the median quality loss per location visit is 0.43 km (Figure 11). Given that this algorithm gives the best privacy and worst quality, 430m seems acceptable.

In No-Knowledge the median quality loss drops from 420m ($c = 5$) to 200m ($c = 25$) to 70m ($c = 50$) – see Figure 12. We can see how the distribution of users noticeably shifts from being concentrated around 250m-500m ($c = 5$) to around 0-150m ($c = 50$). The concentration is more pronounced for the loss per location visit than for the total loss, and the difference between the Average and the Median is smaller than for the total loss.

Known-Mobility provides, as expected, the lowest quality loss per visit, with medians of 150m ($c = 5$), 60m ($c = 25$), and 20m ($c = 50$) in Figure 13. Comparing $c = 5$ with Perfect Resemblance, which has a 430m loss, we see that relaxing the requirement for perfect privacy can reduce the quality loss by about 65%.

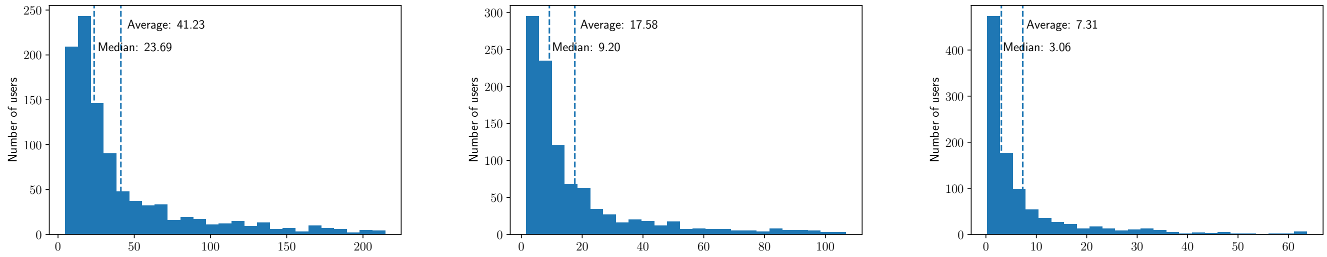
Comparing all algorithms for both types of LBSes (Figures 14 and 15) the picture is similar to the total quality loss, except the overlap between No-Knowledge and Known-Mobility is not that high any more. This shows more clearly than before the benefit of knowing the user's mobility.

6.6 Target Avoidance: Comparison to ILP mechanisms

Having described our proposed algorithms, it is now easier to make a comparison to ILP mechanisms, all of which are geared towards on Target Avoidance, to the best of our knowledge.

We point out two problems that would result if one uses an ILP to achieve the Target Avoidance objective. The cause of these problems is that ILP algorithms essentially remove all visits to sensitive locations.

First of all, in Target Avoidance, a target histogram to avoid is given, but it is not obvious how to even define the sensitive locations given that histogram. If we set the sensitive locations to be the locations in the histogram, the existing algorithms will submit a histogram with all zeros in those locations. However, if the target histogram also has



(a) Known-Mobility $c = 5$

(b) Known-Mobility $c = 25$

(c) Known-Mobility $c = 50$

Fig. 7. Quality loss for Known-Mobility, Nearest-PoI LBS.

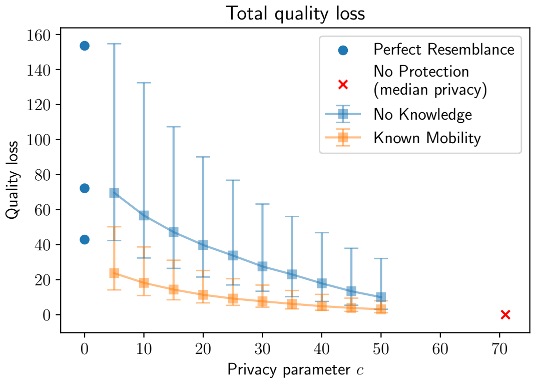


Fig. 8. Nearest-PoI LBS: Tradeoff between privacy and quality loss.

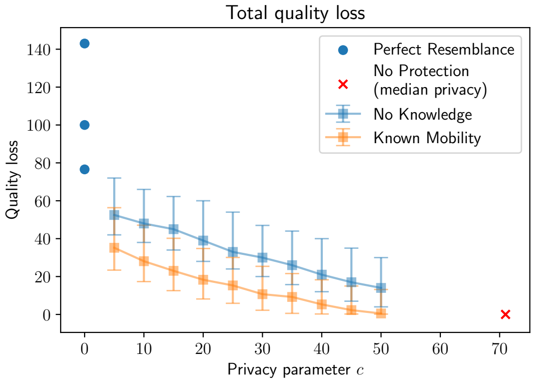


Fig. 9. Geofence LBS (radius 200m): Tradeoff between privacy and quality loss.

all zeros in its locations, then the submitted histogram is identical to the target, even though the user wishes to avoid it.

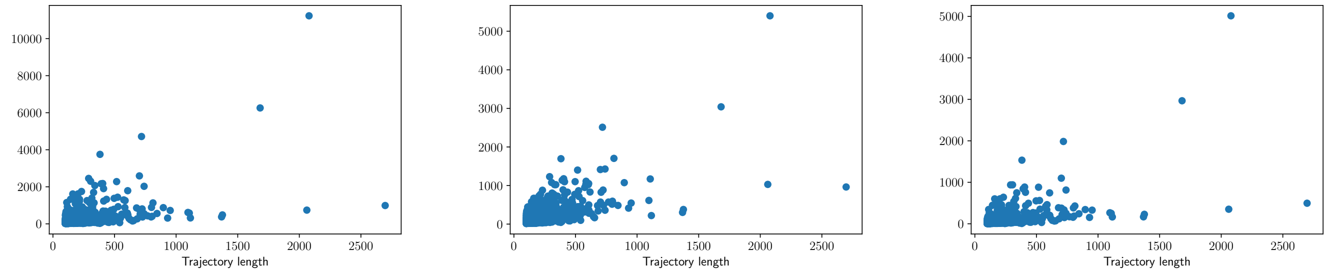
Second, ILP algorithms incur unnecessary quality loss by removing all visits. To see this, let's assume that a user visits a sensitive location, and the algorithm chooses to submit the nearest location to the user instead of the sensitive one. This is the most favorable choice for quality, and it is exactly what our No-Knowledge privacy algorithm does as well. However, the existing algorithm will *always* choose a different location from the true location, whereas our No-Knowledge algorithm will choose a different location *only if* it is necessary for satisfying the privacy constraint. In that sense, our No-Knowledge algorithm is always at least as good, never worse, so we can consider the performance plots of No-Knowledge as an upper bound for the performance of existing algorithms.

Even if we calibrate an ILP to only remove an appropriate fraction of visits to a location, instead of removing all visits, it can still not make targeted *additions* to locations whose frequency needs to increase. If we modify the ILP to also make targeted additions, then we have more or less re-invented No-Knowledge.

7 Conclusion and Future Work

We address the problem of on-the-fly protection of location histograms. A user submits locations dynamically to a Location-Based Service and wishes to make on-the-fly modifications to continuously resemble (or, alternatively, avoid) a target histogram. We describe an optimization framework for optimally designing a protection mechanism that simultaneously achieves a user-chosen privacy parameter and minimizes quality loss.

We believe this line of research can be fruitfully extended to other areas, e.g., histograms of web page visits, web search queries, and watching videos, all of which can be used to profile users. In the context of location privacy, on-the-fly protection can be applied to protect other patterns beyond histograms: whether or not the user has visited a certain combination of locations, e.g., a Casino and a Bank; a Gym visit at least 7 times within a week, or a Cinema visit followed by a Restaurant visit within one day. Finally, we could consider more than one users, and in that case the time and location dimension can be combined with a social dimension: e.g., the adversary wants to detect whether user A visited the same location as user B more than 5 times in the past week at the same time (implying that user A often meets user B), or whether user A consistently visits a location the day after user B visits it (implying that user B might be leaving a message/package for user A to collect). We are not aware of work that addresses these objectives in the on-the-fly setting, although in the static setting (privacy-preserving data mining) it is common to protect sensitive patterns.



(a) Quality loss for Perfect Resemblance.

(b) Quality loss for No-Knowledge $c = 5$.

(c) Quality loss for Known-Mobility $c = 5$.

Fig. 10. Quality loss as a function of trajectory length, nearest-PoI LBS.

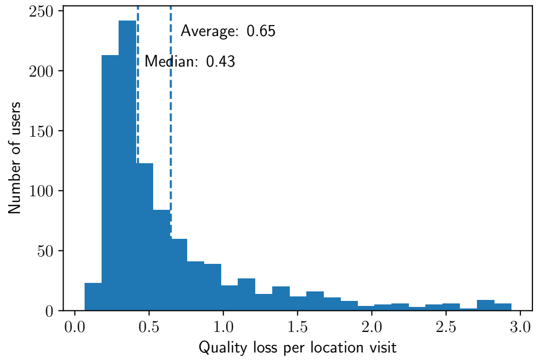


Fig. 11. Quality loss per location visit for Perfect Resemblance.

References

[1] D. Quercia, N. Lathia, F. Calabrese, G. D. Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 971–976.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, November 2011.

[3] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show me how you move and I will tell you who you are," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. ACM, 2010, pp. 34–41.

[4] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2013, pp. 901–914.

[5] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "A predictive differentially-private mechanism for mobility traces," in *Privacy Enhancing Technologies*. Springer, 2014, pp. 21–41.

[6] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, "Preventing velocity-based linkage attacks in location-aware applications," in *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems (ACM-GIS 2009)*. ACM, 2009, pp. 246–255.

[7] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 1959–1972.

[8] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy and Security (TOPS)*, vol. 19, no. 4, December 2016.

[9] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. L. Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *ACM Conference on Computer and Communications Security (CCS)*, October 2012.

[10] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2015, pp. 1298–1309.

[11] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 239–250.

[12] M. Crossley, "Discrimination against the unhealthy in health insurance," *U. Kan. L. Rev.*, vol. 54, p. 73, 2005.

[13] G. Ghinita, "Privacy for location-based services," *Synthesis Lectures on Information Security, Privacy, & Trust*, vol. 4, no. 1, pp. 1–85, 2013.

[14] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive computing*, no. 1, pp. 46–55, 2003.

[15] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 31–42.

[16] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in GPS traces via uncertainty-aware path cloaking," in *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2007, pp. 161–171.

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

[18] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *IEEE Symposium on Security and Privacy (SP)*, May 2011.

[19] W. Eltarjaman, R. Dewri, and R. Thurimella, "Location privacy for rank-based geo-query systems," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 77–96, 2017.

[20] K. Fawaz, K.-H. Kim, and K. G. Shin, "Privacy vs. reward in indoor location-based services," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 102–122, 2016.

[21] A. K. Sangaiah, D. V. Medhane, G. Bian, A. Ghoneim, M. Alrashoud, and M. S. Hossain, "Energy-aware green adversary model for cyber physical security in industrial system," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[22] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, July 2019.

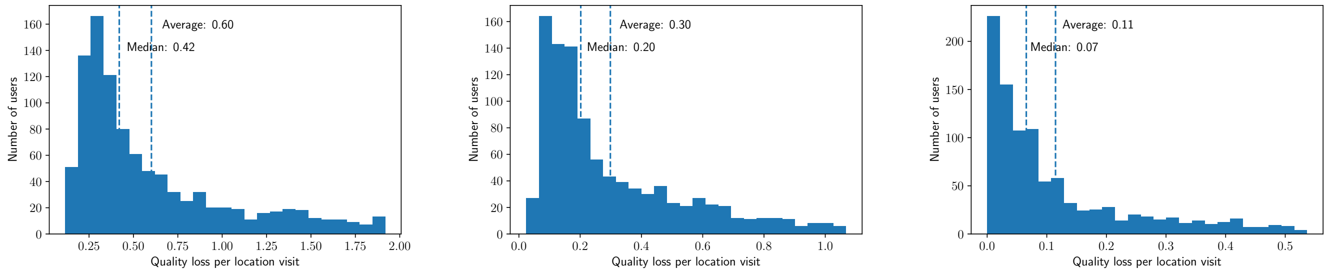
[23] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 546–563.

[24] M. Fanaeepour and B. I. Rubinstein, "Differentially private counting of users' spatial regions," *Knowledge and Information Systems*, vol. 54, no. 1, pp. 5–32, 2018.

[25] J.-Y. Le Boudec, *Performance Evaluation of Computer and Communication Systems*. EPFL Press, Lausanne, Switzerland, 2010.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 59–66.

[27] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

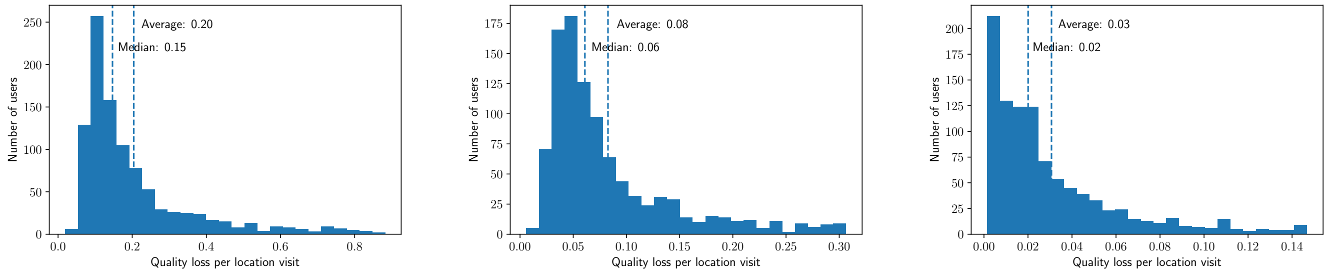


(a) No-Knowledge $c = 5$

(b) No-Knowledge $c = 25$

(c) No-Knowledge $c = 50$

Fig. 12. Quality loss per location visit for No-Knowledge, Nearest-PoI LBS.

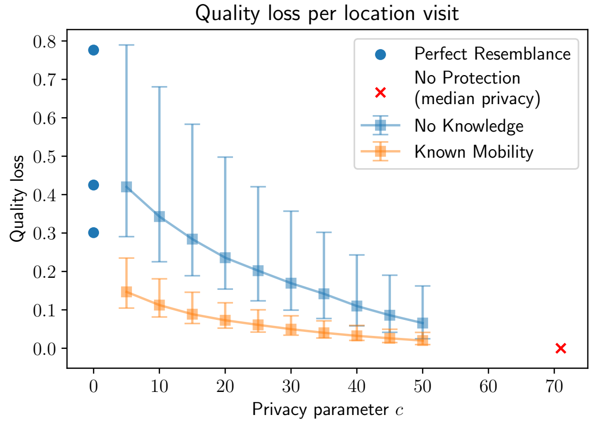


(a) Known-Mobility $c = 5$

(b) Known-Mobility $c = 25$

(c) Known-Mobility $c = 50$

Fig. 13. Quality loss per location visit for Known-Mobility, Nearest-PoI LBS.



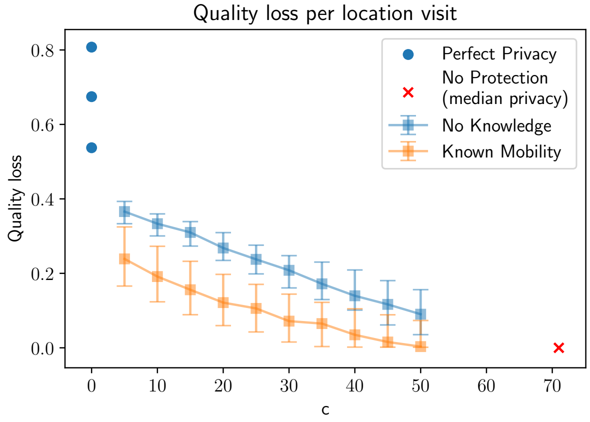


Fig. 14. Nearest-PoI LBS: Tradeoff between privacy and quality loss per location visit.

Fig. 15. Geofence LBS (radius 200m): Tradeoff between privacy and quality loss per location visit.

[28] Foursquare. (2018) Venue categories. [Online]. Available: https://developer.foursquare.com/docs/resources/categories

George Theodorakopoulos received the Diploma degree from the National Technical University of Athens, Greece, in 2002, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 2004 and 2007, all in electrical and computer engineering. He is a Senior Lecturer at the School of Computer Science & Informatics, Cardiff University, since 2012. From 2007 to 2011, he was a Senior Researcher at the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. He is a coauthor (with John Baras) of the book Path Problems in Networks (Morgan & Claypool, 2010).

Emmanouil Panaousis received the Diploma degree in Informatics and Telecommunications from the University of Athens, Greece, in 2006, and the M.Sc. degree in Computer science from the Athens University of Economics and Business, Greece, in 2008, and the Ph.D. degree in Mobile Communications Security from Kingston University London, U.K., in 2012. He is an Associate Professor at the School of Computing and Mathematical Sciences, University of Greenwich, since 2019. Prior to this, he held appointments at the University of Surrey, University of Brighton, Imperial College London, Queen Mary University London and Ubitech Technologies Ltd. His most recent research roles include: Principal Investigator (PI) of H2020 CUREX project on cyber risk management for health data exchange; PI of H2020 SECONDO project on security economics with use cases in AI-enabled smart environments; PI of H2020 SPEAR project on the intrusion detection for smart grid infrastructures. His core expertise is on designing and implementing robust defences against adversarial behaviour.

Kaitai Liang received the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, in 2014. He is currently an Assistant Professor with the Department of Computer Science, University of Surrey, U.K. His research interests are applied cryptography and information security, in particular, encryption, network security, big data security, privacy enhancing technology, blockchain, lattice-based crypto and security in cloud computing.

George Loukas received the Ph.D. degree in network security from Imperial College London (2006). He is currently an Associate Professor in Cyber Security and Head of the Internet of Things and Security research group at the University of Greenwich, as well as the Project Coordinator of H2020 EUNOMIA tackling disinformation online and Principal Investigator of several other inter- national research projects related to the security of smart homes, the Internet of Things, autonomous vehicles, and human-as-a-sensor systems. He has authored or co-authored more than 70 journals and conference publications. His book on cyber-physical attacks was included in ACM's top ten list in the computing milieux category of 2015. He is on the Editorial Board of the BCS Computer Journal and Elsevier's Simulation Modelling Practice and Theory.