DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2017 (4):308–328

Konstantinos Chatzikokolakis, Ehab ElSalamouny, and Catuscia Palamidessi

Efficient Utility Improvement for Location Privacy

Abstract: The continuously increasing use of location-based services poses an important threat to the privacy of users. A natural defense is to employ an obfuscation mechanism, such as those providing geo-indistinguishability, a framework for obtaining formal privacy guarantees that has become popular in recent years.

Ideally, one would like to employ an optimal obfuscation mechanism, providing the best utility among those satisfying the required privacy level. In theory optimal mechanisms can be constructed via linear programming. In practice, however, this is only feasible for a radically small number of locations. As a consequence, all known applications of geo-indistinguishability simply use noise drawn from a planar Laplace distribution.

In this work, we study methods for substantially improving the utility of location obfuscation, while maintaining practical applicability as a main goal. We provide such solutions for both infinite (continuous or discrete) as well as large but finite domains of locations, using a Bayesian remapping procedure as a key ingredient. We evaluate our techniques in two real world complete datasets, without any restriction on the evaluation area, and show important utility improvements with respect to the standard planar Laplace approach.

DOI 10.1515/popets-2017-0051

Received 2017-02-28; revised 2017-06-01; accepted 2017-06-02.

1 Introduction

In recent years, the increasing availability of location information about individuals has led to a growing use of systems that record and process location data. Examples include Location Based Services (LBSs), location-data mining algorithms to determine points of interest, and location-based machine learning algorithms to predict traffic patterns.

While these systems have demonstrated to provide enormous benefits to individuals and society, the growing exposure of users' location information raises important privacy issues. On one hand, location information itself may be considered as sensitive. For instance, there have been reported episodes of men tracking women with GPS or specific applications [1, 2], and in California location records have been used in divorce proceedings to prove claims about suspicious movements of spouses [3]. Furthermore, location data can be easily linked to a variety of other information that an individual usually wishes to protect: by collecting and processing such data on a regular basis, it is possible to infer an individual's home or work location, sexual preferences, political and religious inclinations, etc. [4].

It is not surprising, therefore, that a lot of effort has been dedicated to design and implement methods for protecting the user's privacy, while preserving the utility and the dependability of location data for their use in location-based systems. In general, all computational methods for privacy protection are based on degrading the precision of information. In the case of location privacy, one typical way to reduce the precision is by spatial obfuscation, which has the advantage of requiring no trusted third party, since the user himself can run the obfuscation mechanism locally on his device. In this paper we focus on the *randomized* approaches to spatial obfuscation, which in recent years have become increasingly more popular. Much of this success is due to their properties of robustness with respect to attackers that may combine the observation of the user's activity with any *side information* they have about the user or, more generally, about the habits of the population, the characteristics of the territory, etc.

The most influential proposals of this kind have been, to the best of our knowledge, those by Shokri et al. [5] and by Andrès et al. [6]. Both their frameworks are built on rigorous and natural notions of privacy, and both are based on the idea of confusing the adversary by reporting a noisy location, generated from the true location according to some probability distribution.

Specifically, the authors of [5] considered a Bayesian model of adversary. They focused on optimizing the trade-off between privacy and utility and proposed a method to compute the optimal noise-generating mechanism, which consists in formalizing the utility constraints and the privacy target as a linear optimization

Konstantinos Chatzikokolakis: CNRS, Inria and LIX, École Polytechnique, France

Ehab ElSalamouny: Inria, France and Faculty of Computers and Informatics, Suez Canal University, Egypt

Catuscia Palamidessi: Inria and LIX, École Polytechnique, France

problem, and then applying linear programming techniques. The approach of computing an optimal noisy mechanism via linear programming was also adopted in [7] and [8], although they considered the reverse problem of optimizing the utility for a given level of privacy.

On the other hand, the authors of [6] proposed the notion of *geo-indistinguishability*, based on (an extension of) differential privacy [9]. Like differential privacy, geo-indistinguishability is independent from the prior of the adversary, and is robust with respect to composition. Furthermore, it can be implemented in a simple and efficient way using a planar Laplacian as the noise function. Indeed, probably thanks to its efficiency, geo-indistinguishability via the Laplacian mechanism has been adopted as the basis or as a component of several tools and frameworks for location privacy, including: Location Guard [10], LP-Guardian [11], LP-Doctor [12], the system for secure nearby-friends discovery in [13], SpatialVision QGIS plugin [14], and it is one of the possible input methods in STAC [15]. Furthermore, the PIM mechanism [16] can be considered an extension of the planar Laplacian to the case of traces (temporally correlated sequences of points): The authors of [16] attack the problem of the degradation of privacy due to correlation by adding Laplacian noise directly to the convex hull of the trace.

The Laplacian is not the only way to implement geo-indistinguishability efficiently: on a discrete map one can also use a planar variant of the geometric mechanism [17], and if the map is bounded, the exponential mechanism (cfr. [18] for its use in location privacy) and the tight-constraints mechanism [19] are applicable as well. The latter are applicable to any finite domain equipped with an arbitrary privacy metric.

The advantages and disadvantages of these two approaches could be, at a first thought, resumed as follows: Generating the noise via a direct method (Laplacian, geometric, exponential, tight-constraints) is efficient, but there is no guarantee of optimality. Generating the noise via linear programming techniques, on the other hand, is computationally expensive, and not feasible for more than about hundred locations, but it gives the optimal trade-off between privacy and utility.

In this paper we try to improve the situation: on one hand, we explore ways to increase the utility of the direct methods; on the other hand, we study whether it is possible to scale the optimal method to cover large areas, by simply increasing the size of the locations.

It turns out that we can immediately improve the utility of the direct methods by using one of the principles of the optimal mechanism. In general the construction of the latter (for the case in which we fix the privacy constraints and we optimize the utility) is based on a Bayesian technique: given a prior, each reported location is *remapped* into the best possible location, according to the prior and the loss function. Such process can be applied also in the case of the direct methods: Given the prior, one can compute the best remapping with respect to the prior and apply it to the method, thus increasing the utility. The gain in utility is such that, according to our experiments on small areas in which the optimal mechanisms can be constructed, the difference between the expected loss between some of the direct methods and the optimal one is minimal. Furthermore, while in the optimal method finding the best remapping is part of the linear optimization problem and therefore it adds significantly to the complexity, in the case of the direct methods the best remapping is computed separately from the mechanism, and after we know the result of the mechanism. Therefore, it can be computed much more efficiently.

Let us now consider the optimal method in more detail, to see if we can cope with the complexity issue. The variables of the linear program are the probabilities of reporting location z when the true location is x. Hence, their number is the square of the number of locations in the problem. Consequently the linear programming technique can only be applied the number of locations is relatively small: In our experiments, we found that the optimization program would already take several days to produce the result when such number was approaching 100. If we want to cover a large area, one simple solution is to increase the size of the "locations". For instance, if we want to apply the technique in an area of 100 Km^2, we would divide the map in "cells" of 1 Km^2 (for reference, 100 Km^2 corresponds approximately to the San Francisco area).

One problem with this approach is that the scale of typical utility functions is much smaller. For instance, if we intend to use an LBS to find point of interests in walking distance, it seems natural to measure the distance in terms of meters, not kilometers. Now, ignoring distances up to a kilometer in general induces a fictitiously higher utility, because when we discretize a map into cells, the distances are measured as if all the points in a cell were identified with its center. The utility computed using this discretized distance is only an approximation of the real utility, and the coarser is the granularity of the discretization, the less precise is this approximation.

Furthermore, a coarse granularity generates problems not only about utility, but also about privacy, because the privacy constraints usually depend on the distance between points as well. Assume, for instance, the privacy constraints are those of geo-indistinguishability, and consider two points x and x' that are very close but after the discretization belong to different cells. Since they are close they should be highly indistinguishable, but since they are in different cells they will be considered as being located in the centers of their respective cells, and they will therefore become distinguishable at the level allowed by he distance between the two centers.

Contribution

The contributions of this paper are as follows:

- We consider the main methods for geo-indistinguishability (Laplacian, geometric, exponential, tight-constraints) on various kinds of domains: continuous, discrete unbounded, and discrete bounded. We show that their utility increases considerably when we apply the best remapping (wrt a given prior). The comparison is done using the Brightkite and the Gowalla datasets.
- We compare the utility of the above methods and that of the optimal method on small areas where the latter can be computed. We show some of the directed methods, after we apply the remapping, have an expected loss close to that of the optimal one. We also discuss the possibility of applying the optimal method on a larger area, by increasing the size of the location, but we show that the gap between the approximated utility (computed on the coarse grid) and the "real" one would be too large.
- In order to construct the prior, we take a common machine learning approach separating between the training and testing data. From the first part, we construct a global prior used to optimally remap the mechanisms. From the other part of the dataset, we construct a user-specific prior used to measure the utility of the mechanism.

Related work

The most closely-related works have already been discussed in the first part of the introduction. Here we mention some of the other relevant work in the area of location privacy.

As already stated, in general, all computational methods for privacy protection are based on degrading the precision of information. In the particular case of location data, this is obtained essentially in two ways: spatial cloaking and spatial obfuscation.

Spatial cloaking, first proposed in [20], is based on the idea of concealing the user's exact coordinates by reporting a cloaked area, so to meet certain anonymity constraints. Often, the cloaking is not only spatial, but also temporal, so to conceal also the time in which the user was in that position. The anonymity constraints that have been mostly considered in the case of location privacy are: k-anonymity [21–24], l-diversity [25], t-closeness [26], and p-sensitivity [27]. In addition, in order to reduce the linkability between identity and trajectories, [28] proposed the so-called mix-zones. This idea assumes that people will only report their location in certain regions, called "application zones", where a location-based service is offered, e.g. an airport, bank, or coffee shop. In the mix zones, outside the application zones, users will receive new, unused pseudonyms. This helps prevent an attacker from linking pseudonyms, because the new pseudonym could have been assigned to anyone else in the mix zone.

In order to protect *aggregate* location information, randomized methods, and in particular differential privacy, have also been used. For instance, [29] presents a way to statistically simulate the location data from a database while providing privacy guarantees. In [30], a quad tree spatial decomposition technique is used to achieve differential privacy in a database with location patter mining capabilities. On the other hand, Dewri [31] proposes a combination of differential privacy and k-anonymity for the purposes of hiding the location of a single individual.

Spatial obfuscation, on the other hand, works by reducing the precision of the position sent from the user to the server. In contrast to the above approaches, that require a trusted third party, spatial obfuscation mechanisms can be run by user itself [32]. Spatial obfuscation approaches can be classified into two main kinds: deterministic and randomized. We already discussed the randomized ones in the first part of this introduction, but we want to mention here also the approach of [33], that presents various user-centered adaptations of differential privacy to the location context, and exploits the symmetry of circular noise functions to define mechanisms that are very easy to implement, satisfy the required privacy constraints, and under certain conditions provide the same privacy and utility levels as more complex noise functions.

As for the deterministic methods, a typical one is that proposed by Ardagna et al. [34], which consists in blurring the user's location by reporting a larger region (containing the position). Other similar proposals are presented in [35–38].

Unfortunately, deterministic methods are not robust with respect to composition. For instance, if the user reports one zone and then, soon afterwards, an adjacent zone, it can easily be inferred that he must be near the border between these two zones. Furthermore, these methods offer little guarantees against an adversary who already has some side knowledge (aka *prior knowledge*) about the habits of the user or of the population.

Plan of the paper
The next section recalls the necessary preliminary notions. Section 3 introduces the notion of remapping. Section 4 defines the various direct mechanisms and the result of applying the remapping to them. Section 5 compares the various mechanisms (with and without remapping), and the optimal one obtained from linear programming. Section 6 concludes. The proofs of all results can be found in the appendix.

2 Preliminaries

2.1 Geo-indistinguishability

The notion of geo-indistinguishability is a variant of differential privacy for location based systems introduced in [6]. Let $\mathcal{X} \subseteq \mathbb{R}^2$ be the set of *possible locations* of a user, and let $\mathcal{Z}$ be a set of reported values, often assumed to be equal to $\mathcal{X}$. Let $\mathcal{P}(\mathcal{Z})$ denote the set of probability distributions over $\mathcal{Z}$ and let $\mathbf{d}_\otimes$ be the *multiplicative distance* between two distributions $\sigma_1, \sigma_2 \in \mathcal{P}(\mathcal{Z})$, defined as $\mathbf{d}_\otimes(\sigma_1, \sigma_2) = \sup_{Z \subseteq \mathcal{Z}} |\ln \frac{\sigma_1(Z)}{\sigma_2(Z)}|$, with the convention that $|\ln \frac{\sigma_1(Z)}{\sigma_2(Z)}| = 0$ if both $\sigma_1(Z), \sigma_2(Z)$ are zero and ∞ if only one of them is zero.

An *obfuscation mechanism* K probabilistically selects a reported location $z \in \mathcal{Z}$ starting from a real location $x \in \mathcal{X}$, and is modeled by a function $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ mapping each real location to a probability distribution over the reported ones. K satisfies ϵ-geo-indistinguishability iff

$$\mathbf{d}_\otimes(K(x), K(x')) \leq \epsilon\, \mathbf{d}(x, x') \quad \forall x, x' \in \mathcal{X}$$

where $\mathbf{d}$ denotes the Euclidean distance. This definition can be equivalently stated as $K(x)(Z) \leq e^{\epsilon \mathbf{d}(x,x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$.

Note that $\epsilon\, \mathbf{d}$ can be viewed as a *distinguishability metric*: the closest x, x' are wrt this metric, the more similar the distributions $K(x), K(x')$ are required to be. Following [6], we choose ϵ by selecting a radius r in which we want to enjoy strong privacy (eg. $r = 0.1$ km), some small privacy level l for that radius (eg. $l = \ln(1.4)$) and then set $\epsilon = l/r$. This ensures that locations within distance r will have distinguishability level at most l.

Note also that, although the Euclidean distance is a natural choice of distinguishability metric for location privacy, the same definition could be used with a different distance metric capturing a different notion of privacy. Privacy under arbitrary metrics is studied in [39], while [18] proposes a method to construct a distinguishability metric taking into account the semantics of each location. In this paper we mostly assume $\mathbf{d}$ to be Euclidean, but also discuss mechanisms that allow the use of an arbitrary privacy metric.

2.2 Utility

The utility provided by an obfuscation mechanism K depends on the specific application it is used for. Certain applications (eg. weather report) can tolerate a high amount of noise without affecting the provided service, while others (eg. GPS navigation) require finer information. Following several works on location privacy [5, 7], we define utility as the expected *quality loss* of K wrt some prior distribution π:[1]

$$\mathrm{QL}(K, \pi, \mathbf{d}_Q) = \sum_{x \in \mathcal{X}, z \in \mathcal{Z}} \pi(x) K(x)(z) \mathbf{d}_Q(x, z)$$

where $\mathbf{d}_Q$ is a *quality loss metric*, measuring how much the service is degraded when z is reported instead of x. When evaluating a mechanism without a specific application at hand, we commonly use the Euclidean metric for $\mathbf{d}_Q$, since the quality of all LBSs degrades when z moves far away from x.

Another useful generic quality loss metric is the *squared* Euclidean distance $\mathbf{d}^2$. This metric is relevant for applications querying for information in a specified area (eg. POI retrieval), since obfuscation typically requires to increase the area of retrieval which grows with the square of the distance between x and z [6].

Note that all techniques discussed in this paper, with the sole exception of the continuous remap of Section 3.2, work for an arbitrary utility metric $\mathbf{d}_Q$, so they can be easily adapted to the application at hand.

[1] For simplicity, the expression for QL assumes discrete $\mathcal{X}, \mathcal{Z}$.

2.3 The planar Laplace mechanism

The planar Laplace (PL) mechanism is a simple and efficient continuous mechanism satisfying ϵ-geo-indistinguishability. It can be used when $\mathcal{X}$ is any subset of $\mathbb{R}^2$ and the reported set $\mathcal{Z}$ is the full $\mathbb{R}^2$. The mechanism consists at drawing from a 2-dimensional Laplace distribution centered at the real location x, having pdf:

$$D_x(z) = \frac{\epsilon^2}{2\pi} e^{-\epsilon \mathbf{d}(x,z)}$$

Drawing from this distribution can be easily and efficiently performed in polar coordinates, by adding to x a randomly drawn vector expressed as a radius r and angle θ, as follows:

- draw θ uniformly in $[0, 2\pi)$,
- draw p uniformly in $[0, 1)$ and set $r = C^{-1}(p)$,
- report $z = x + \langle r\cos(\theta), r\sin(\theta)\rangle$,

where $C^{-1}(p)$ is the inverse of PL's cumulative distribution function for r, given by $C^{-1}(p) = -\frac{1}{\epsilon}\big(W_{-1}(\frac{p-1}{e}) + 1\big)$ and W_{-1} is the Lambert W function (-1 branch) implemented in several numerical libraries.

Note that, as with any continuous mechanism, using the above algorithm in a machine with floating point arithmetic essential corresponds to using a discretized version of the mechanism, where the obtained location z corresponds to a subset of $\mathbb{R}^2$ around z. In that sense we can think of z as a "single point" drawn from a continuous distribution.

Concerning utility, it can be shown that, due to the symmetry of $\mathbb{R}^2$, the (squared) Euclidean quality loss of PL is independent from π, namely $\forall \pi \in \mathcal{P}(\mathcal{X})$:

$$\mathrm{QL}(\mathrm{PL}, \pi, \mathbf{d}) = 2/\epsilon \quad \text{and} \quad \mathrm{QL}(\mathrm{PL}, \pi, \mathbf{d}^2) = 6/\epsilon^2 \qquad (1)$$

2.4 Optimal mechanisms

Although the planar Laplace mechanism is simple and efficient, its utility is not guaranteed to be optimal among all mechanisms satisfying ϵ-geo-indistinguishability. If $\mathcal{X}$ is finite and small in size, given π and $\mathbf{d}_Q$ an optimal mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ can be constructed by solving a linear optimization problem [7]. In the finite case the mechanism can be represented by a $|\mathcal{X}| \times |\mathcal{X}|$ stochastic matrix K, where K_{xz} is the probability to report z from x. Having elements of K as variables, the optimal mechanism is given as a solution to:

$$\begin{aligned} &\text{minimize } \mathrm{QL}(K, \pi, \mathbf{d}_Q) \\ &\text{subject to } K_{xz} \le e^{\epsilon \mathbf{d}(x,x')} K_{x'z} \quad x, x', z \in \mathcal{X} \\ &\qquad\quad \text{and } \textstyle\sum_z K_{xz} = 1 \quad x \in \mathcal{X} \end{aligned}$$

The constructed mechanism is guaranteed to satisfy ϵ-geo-indistinguishability, and have no worse utility than any other mechanism K' also satisfying the same privacy definition. However, solving the above problem is feasible only when $|\mathcal{X}|$ is very small. In fact, the number of variables and the number of constraints of the linear optimization problem are respectively *quadratic* and *cubic* on the number of locations. In [7] an approximation technique is proposed using spanners, but even in this case the technique is necessarily limited to at most a few hundred locations.

Note that this technique can be employed to an arbitrary privacy metric $\mathbf{d}$, not necessarily the Euclidean one. Moreover, [8] proposes similar constructions taking also into account inference error constraints.

3 Remapping locations

Remapping the output of a mechanism is a simple yet effective technique for improving its utility while satisfying the same level of privacy. In this section we discuss the technique in general, the optimal way of performing it and an efficient way to remap points on a continuous plane. In Section 4 we further discuss remapping for various types of mechanisms, and in Section 5 we show that remapping can be effective in practice for substantially improving the mechanism's utility.

Given a mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$, and starting from a location x, we can first use K to draw a noisy location z, then remap z to a new location $z^* = R(z)$ using a remapping function[2] $R : \mathcal{Z} \to \mathcal{Z}$, and finally report z^*. Let KR denote the composed mechanism (if K, R are written as stochastic matrices, the composition is their product KR). The following result is a straightforward adaptation of the well-known fact that remap reserves differential privacy ([40] Prop. 2.1).

Proposition 3.1 (Remap preserves privacy). *Let $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ be a mechanism and $R : \mathcal{Z} \to \mathcal{Z}$ be a remap. If K satisfies ϵ-geo-indistinguishability then so does KR.*

On the other hand, remapping can improve the mechanism's utility. Geo-indistinguishability requires nearby locations x, x' to produce z with the same probability, but z does not have to be far way from x. Con-

[2] In general R could also be probabilistic, although in this paper we only consider deterministic ones, since an optimal deterministic remap always exists.

sider, for instance, two locations close to the sea, say in San Francisco (Fig 6). A mechanism such as the planar Laplace adds noise symmetrically in all directions, leading to a high chance of z falling into the sea. Knowing that the user is unlikely to be in the sea, we can remap it back into the edge of the city, bringing it closer to the true location. Of course, such a remap will decrease the utility if a user is indeed in the sea, but the chances of this happening is small; on average utility is improved.

Clearly, one wishes to employ the remap providing optimal utility, captured by the following property.

Definition 3.2 (Optimal remap). *A remap R is optimal for K wrt $\pi, \mathbf{d}_Q$ iff $\mathrm{QL}(KR, \pi, \mathbf{d}_Q) \leq \mathrm{QL}(KR', \pi, \mathbf{d}_Q)$ for all remaps R'.*

Optimality can be achieved by a *Bayesian* remap, choosing the location z^* that minimizes the expected loss wrt the *posterior* distribution $\sigma \in \mathcal{P}(\mathcal{X})$ obtained from π, K after observing z by applying Bayes' rule. Let $\sigma = \mathbf{Post}(\pi, K, z)$ denote this distribution, given by

$$\sigma(x) = \frac{\pi(x)K(x)(z)}{\sum_{x'} \pi(x')K(x')(z)} \qquad x \in \mathcal{X}$$

Proposition 3.3 (Bayesian remap). *Given π, K, $\mathbf{d}_Q$, the remap defined by:*

$$R(z) = \operatorname*{arg\,min}_{z^* \in \mathcal{Z}} \textstyle\sum_{x\in\mathcal{X}} \sigma(x)\mathbf{d}_Q(x, z^*) \qquad \textit{where}$$
$$\sigma = \mathbf{Post}(\pi, K, z)$$

is optimal for K wrt $\pi, \mathbf{d}_Q$.

3.1 Efficiency of the Bayesian remap

The Bayesian remap consists of computing the posterior $\mathbf{Post}(\pi, K, z)$ and then finding the point $z^* \in \mathcal{Z}$ that minimizes the expected loss. If $\mathcal{X}, \mathcal{Z}$ are finite, the remap can be computed directly using its definition (Prop 3.3). Note that computing the remap for an existing mechanism K is far more efficient that constructing an optimal mechanism (Section 2.4); the latter is feasible only for sizes of the order of 10^2, while a direct computation of R can be performed for sizes two to three orders of magnitude larger.

Moreover, even when iterating over all $z^* \in \mathcal{Z}$ is infeasible, the remap can be approximated by considering only locations within a certain $\mathbf{d}_Q$-ball $B_t(z)$ centered at z, replacing $\arg\min_{z^*\in\mathcal{Z}}$ by $\arg\min_{z^*\in B_t(z)}$. The choice of the threshold t could be done in various ways, for instance we can choose the smallest t such that $K(x)(B_t(x)) \geq 0.99$ for all $x \in \mathcal{X}$. Intuitively, under this threshold we expect the posterior σ to be very small outside the examined area, while $\mathbf{d}_Q$ is large, making locations in that area unlikely candidates for the remap. Note that any remap function preserves geo-indistinguishability, so from the point of view of privacy we are safe to use any heuristic might make it faster.

The complexity in this case depends only on the threshold t, not on $|\mathcal{X}|, |\mathcal{Z}|$. In fact, this technique makes the remap applicable even if the space is *infinite* (but discrete); in our experiments with an infinite grid (Section 5) each remap took only a few milliseconds without any loss in utility. Note also that this technique works with any choice of $\mathbf{d}_Q$, making the remap a generic and efficient method for practical applications.

On the other hand, if $\mathcal{Z}$ is continuous the problem becomes harder, since, even in a bounded area $B_t(z)$, we would still need to consider uncountably many points. This case is discussed in the next section.

3.2 Efficient continuous remap

For many applications, we would like to have the flexibility to position a user anywhere on a plane, as well as to report arbitrary locations, hence it is convenient to take $\mathcal{X}, \mathcal{Z}$ to be the full $\mathbb{R}^2$.

The first problem in this case is computing the posterior. Although users could be potentially located anywhere in the plane, in practice the prior is constructed from some finite dataset of past service uses, POIs, etc. Hence, we can assume that a posterior $\sigma \in \mathcal{P}(\mathbb{R}^2)$ of *finite support*, can be constructed for the remap (see Section 4.1.1 for a possible way of constructing σ). Now, assuming a posterior σ of finite support, denoted by $\lceil\sigma\rceil$, the problem is that we still have an uncountable set of possible candidates for z^*. The reason is that, although $\lceil\sigma\rceil$ is finite, the z^* minimizing the expected loss is not necessarily an element of the support.

Thankfully, for the common case of the Euclidean and squared Euclidean metrics, constructing the optimal z^* can be done efficiently. Intuitively, z^* should be "in between" the points in the support of σ, that is in the convex hull of $\lceil\sigma\rceil$. A natural choice is the centroid of the points in $\lceil\sigma\rceil$ weighted by their probabilities:

$$\textsc{Centroid}(\sigma) = \textstyle\sum_{x\in\lceil\sigma\rceil} \sigma(x)\, x$$

The centroid can be shown to minimize the expected *squared* distance $\mathbf{d}^2$.

Proposition 3.4. *For all $\sigma \in \mathcal{P}(\mathbb{R}^2)$ with finite $\lceil\sigma\rceil$:*

$$\textsc{Centroid}(\sigma) = \operatorname*{arg\,min}_{z^* \in \mathbb{R}^2} \textstyle\sum_{x\in\lceil\sigma\rceil} \sigma(x)\, \mathbf{d}^2(x, z^*)$$

Consider, for instance, the posterior $\sigma(\mathbf{0}) = 0.2$, $\sigma(\mathbf{1}) = 0.8$. The centroid $c = 0.2 \cdot \mathbf{0} + 0.8 \cdot \mathbf{1}$ gives the minimum expected squared distance 0.104.

On the other hand, somewhat unexpectedly, the centroid *does not* minimize the expected standard Euclidean distance $\mathbf{d}$. In the example above, the centroid gives expected distance 0.28, while $\mathbf{1}$ itself gives expected distance 0.16. The problem of finding the point that minimizes the weighted distance from a given set is known as the Weber problem, and is solved by the Weiszfeld algorithm, first discovered in 1937 and later rediscovered with various improvements [41]. The algorithm starts from $y_0 = \textsc{Centroid}(\sigma)$ and iterative modifies it as follows:

$$y_{i+1} = \sum_{x\in\lceil\sigma\rceil} \frac{\sigma(x)\, x}{\mathbf{d}(x, y_i)} \bigg/ \sum_{x\in\lceil\sigma\rceil} \frac{\sigma(x)}{\mathbf{d}(x, y_i)}$$

We denote by $\textsc{Weiszfeld}(\sigma)$ the result of this iteration with some stopping condition (eg, when the improvement in the expected distance drops below some ϵ).

Note that both $\textsc{Centroid}(\sigma)$ are $\textsc{Weiszfeld}(\sigma)$ are practical even for large sizes. Section 4.1.1 discusses this technique in the context of the planar Laplace mechanism, and the evaluation of Section 5 shows its practical applicability. Finally, note that for an arbitrary $\mathbf{d}_Q$, the optimal solution can be approximated by discretizing the space and applying the approximation technique of Section 3.1.

3.3 The choice of prior

A crucial element of the Bayesian remap is the prior distribution π from which the posterior σ is constructed. In theory, the prior should describe the behaviour of the user, and a good quality prior, constructed from a large dataset of past behaviour of that user, is assumed to be available. In practice, however, there are two major problems with this assumption.

First, general-purpose obfuscation mechanisms should be able to work with *new users* for which *no information* whatsoever is available. For instance, a user installing the Location Guard browser plugin [10] is expecting to use it immediately, but the plugin has no information about this user.

Second, even after an extended training period and assuming that all user's movements are recorded, the prior constructed from this data is not necessarily of good quality. For instance, such a prior would assign zero probability to all locations never visited in the past, but clearly the user might visit some new locations in the future. Moreover, such a prior would mostly contain locations that the user visits on a daily basis, such as its home or work. Remapping using such a prior is similar to revealing the prior to the provider: we might often assume that an adversary already has this prior knowledge, but if he doesn't we shouldn't reveal it to him ourselves.

The problem of prior knowledge is particularly crucial for the *proper evaluation* of a mechanism. Constructing a prior from a dataset and then evaluating it on the *same* dataset (a common practice in the literature of location privacy) might be misleading, especially for datasets containing only a few locations per user. In the extreme case, a prior constructed from a single visited location will lead to a remap that is perfect for *that* location, but likely very bad for *any other* location the user might visit in the future. Such a practice is similar to training and testing a learning algorithm on the same dataset.

To deal with the above issues, we assume the remap is performed using a *global prior* π that describes the behaviour of an *average user*. Possible ways for constructing such a prior are discussed in the following section. In the evaluation of Section 5 we construct the prior from a training dataset containing *no data whatsoever* of the users of the testing set. This simulates a remapping applied to a new user seen for the first time, based on generic data about past uses of the specific service.

On the other hand, a drawback of using a global prior is that the remapping is not guaranteed to improve the expected loss of users who behaviour is drastically different that that prior. However, in our evaluation we see that the percentage of users who are actually “hurt” by the remap is small. One way to mitigate this problem is to measure how different the global prior is from the user's behaviour and avoid the remap if it becomes too large. We leave such a study as future work.

Obtaining a differentially private global prior

There are several approaches for constructing a good quality global prior for remapping. A user-independent approach is to use semantic information about each location, such as the density and variety of POIs, which can be obtained from sources such as OpenStreetMap. On the other hand, a user-dependent approach, using information from users of the service itself, can provide a prior of improved quality. For instance, a prior could be constructed from aggregate information about past uses of the service, made available from the service provider. However, since publishing such aggregate information could compromise the users' privacy, the provider should employ some privacy protection mechanism, such as differential privacy, for publishing this information.

Obtaining a global prior under differential privacy can be achieved using standard techniques. For each user we construct a prior π_i (a vector of probabilities for each location) and then compute the average $\frac{1}{n}\sum_i \pi_i$. The average query is sensitive to changes in a single user if the number of users n is small. Hence, a standard technique is to require that n is above a certain *threshold* N in order to release the result. N should be selected in advance (independently from the database) and if $n < N$ we should add dummy users in the computation or refuse to provide an answer. By modifying a single prior π to π' the ℓ_1 sensitivity is $\|\frac{1}{n}(\pi-\pi)'\|_1 \leq \frac{1}{N}(\|\pi\|_1 + \|\pi'\|_1) = \frac{2}{N}$. Hence, we can achieve ϵ-differential privacy by adding noise from $Lap(\frac{2}{N\epsilon})$ to each element of the average prior [9]. The result is not necessarily a probability vector, but can be converted to one by post-processing, setting negative values to 0 and normalizing.

Note that the choice of N is important: large values allow to release the average with limited noise for database of size at least N, while sacrificing utility for smaller databases. An evaluation of this technique is provided in Section 5.

Note also that the stardard setting of differential privacy assumes a trusted curator that has access to all data, and hence he can compute the real answer and then add noise to it. However, under the standard assumption of a polynomial time adversary, there exist secure multiparty computation techniques (e.g. [42]) that allow to compute noisy aggregates in a *distributed* setting. In such a setting users maintain their own data and participate in a protocol in order to directly compute the noisy answer under differental privacy.

3.4 Remap and adversarial error

We have already discussed that remapping locations is safe wrt geo-indistinguishability; we now show that it is also safe wrt *adversarial error*, a well-known model for quantifying location privacy proposed in [43]. In this model, an adversary performs an inference attack, producing a guessed location x^* starting from the reported location z. Such an attack can be modeled by a stochastic matrix H, where H_{zx^*} is the probability to guess x^* when seeing z. Then, privacy is measured by the expected error of an optimal attack, that is:

$$\textsc{AdvError}(K, \pi, \mathbf{d}_A) = \min_H \mathrm{QL}(KH, \pi, \mathbf{d}_A)$$

Here, the metric $\mathbf{d}_A(x, x^*)$ captures the adversary's loss when he guesses x^* while the real location is x. A crucial observation is that any remap R can only *increase* the adversary's error, since a remap that reduces the error could be performed by the adversary himself.

Proposition 3.5. *For all π, K, R, $\mathbf{d}_A$:*

$$\textsc{AdvError}(K, \pi, \mathbf{d}_A) \leq \textsc{AdvError}(KR, \pi, \mathbf{d}_A)$$

4 Practical Mechanisms

In this section we discuss several mechanisms satisfying geo-indistinguishability that can be applied to a generic LBS. Our emphasis is on practical applicability: the mechanisms should be efficient and applicable to realistic domain sizes, without any unreasonable assumptions.

We classify the mechanisms based on the restrictions they assume on the structure of the domains $\mathcal{X}, \mathcal{Z}$. In the more flexible case, the domains can be the full continuous plane. However, in some applications it might be reasonable to restrict to a discrete, either infinite or a finite but realistically large domain. Mechanisms for each category can be clearly applied to the more restrictive cases, but the extra restrictions allow for the construction of additional mechanisms with certain advantages.

4.1 Mechanisms for the continuous plane

In the general unrestricted case of the full continuous plane, the natural choice is to apply the planar Laplace mechanism described in Section 2.3. The PL mechanism is simple and efficient, and due to its generality and lack of restrictions, it is used in all known applications of geo-indistinguishability. However, even on a continuous plane, the remapping technique can be practically employed to improve the mechanism's utility, as discussed in the next section.

4.1.1 Remapping the planar Laplace mechanism

The challenge for remapping the PL mechanism lies in the continuity of its domain. Using a continuous prior on $\mathbb{R}^2$ is not realistic, it's hard to construct such a prior, let alone to use it. Instead, we assume that prior information about the service is given in the form of a dataset of previous uses of the service, or even in the form of generic information about points of interest (POIs). Let $Q \subset \mathbb{R}^2$ be a possibly large but finite set of locations, each $q \in Q$ associated with a weight $w(q) \geq 0$. For instance Q might contain the locations from which the service has been accessed in the past, with $w(q)$ being the number of users accessing the service from that location. Or Q might be a list of POI relevant for the service, with $w(q)$ capturing the popularity of each POI.

Given a noisy location z drawn from PL, the goal is to first construct a reasonable posterior $\sigma \in \mathcal{P}(\mathbb{R}^2)$ of finite support. To do so, we need to restrict Q (which can be very large) to a limited area around z. Let $t = C^{-1}(0.99)$, where C^{-1} is the inverse of PL's cumulative distribution function for the radius (see Section 2.3), and let $Q_t = Q \cap B_t(z)$. Intuitively, PL reports a point within distance t from x with probability 99%, hence it is reasonable to remap z to a point z^* no farther that t away from z.

Note that Q_t can be computed efficiently using a spatial data-structure such as kd-trees. The 0.99 threshold can be used to trade efficiency for accuracy, smaller values will lead to a smaller Q_t but might cause the optimal point to be outside the area. If Q is dense, a pre-processing phase could be used to reduce the size of Q_t, merging points into small clusters, and setting $w(q)$ to the weight of the cluster.

The weights $w(q), q \in Q_t$ can be though of (after normalization) as a finite prior on $\mathbb{R}^2$. Applying Bayes' law to this prior, using the pdf of PL, we can construct the following posterior σ with finite support $\lceil \sigma \rceil = Q_t$:

$$\sigma(x) = \frac{w(x)e^{-\epsilon \mathbf{d}(x,z)}}{\sum_{q \in Q_t} w(q)e^{-\epsilon \mathbf{d}(q,z)}} \qquad x \in Q_t$$

Finally, the remapped point z^* is $\text{W}\textsc{eiszfeld}(\sigma)$ (if $\mathbf{d}_Q = \mathbf{d}$) or $\text{C}\textsc{entroid}(\sigma)$ (if $\mathbf{d}_Q = \mathbf{d}^2$).

Note that, in practice, the dataset Q might not be detailed enough to provide sufficient information for every location. A new user might access the service from a location in which no or few other users have been in the past. A remap using low quality data is likely to decrease the user's privacy, hence we use a simple heuristic for assessing the data's quality: if the size (or alternatively the total weight) of Q_t is below a certain threshold $q_{\min}$, then we skip the remap and report z directly. The complete algorithm is displayed in Fig 1.

Algorithm 1: Planar Laplace with remap

```
Data: $x \in \mathbb{R}^2$, $\epsilon > 0$, $q_{\min} \geq 0$, $Q \subset \mathbb{R}^2$,
      $w : Q \to \mathbb{R}^+$
Result: An obfuscated location $z^*$
draw $z \sim PL_\epsilon(x)$ ;
set $t = C^{-1}(0.99)$ ;
compute $Q_t = Q \cap B_t(z)$ ;
if $|Q_t| < q_{min}$ then
    return $z$
else
    compute
    $\sigma(x) = \frac{w(x)e^{-\epsilon \mathbf{d}(x,z)}}{\sum_{q \in Q_t} w(q)e^{-\epsilon \mathbf{d}(q,z)}} \quad x \in Q_t$ ;
    compute $z^* = \text{W}\textsc{eiszfeld}(\sigma)$;
    // or $z^* = \text{C}\textsc{entroid}(\sigma)$;
    return $z^*$
```

4.2 Mechanisms for the discrete plane

For some applications, it is reasonable to assume that users are located on a discrete grid, and reported points should always be on that grid. However, we might still want to cover very large areas (or the whole world), hence it is convenient to think of this grid as being infinite. The PL mechanism can of course be employed in this case, with or without remap, by further projecting its output on the grid. However, the discrete nature of the domain allows us to use the Planar Geometric mechanism, a discretized version of PL.

4.2.1 The planar geometric mechanism

The planar geometric mechanism can be seen as a discretized version of the planar Laplace mechanism described in Section 2.3 (similarly to the relationship between the one-dimensional geometric and Laplace mechanisms). The two mechanisms share the idea of producing the output by adding random noise to the user's location x such that the added noise is independent of the real location of the user.

Precisely, let the Euclidean space $\mathbb{R}^2$ be partitioned into a grid of square cells such that the side length of every cell is $s > 0$. The centers of these cells form therefore an infinite grid of points $\mathcal{G}$, in which the horizontal (or vertical) distance between any successive two points is exactly s. We call this distance the 'spacing' of $\mathcal{G}$. Then we define the planar geometric mechanism PG as a probabilistic function from $\mathcal{G}$ to itself as follows:

$$\mathrm{PG}(x)(z) = \lambda\, e^{-\epsilon\, \mathbf{d}(x,z)} \qquad x, z \in \mathcal{G} \tag{2}$$
$$\text{where } \lambda = 1/ \sum_{(i,j)\in\mathbb{Z}^2} e^{-\epsilon\, s\, \sqrt{i^2+j^2}}$$

In general, every output location z of a mechanism can be seen as the sum of the input location x and an Euclidean 'noise' vector u having the magnitude $\mathbf{d}(x,z)$, and describing the added obfuscation noise. From this perspective, it is easy to see from (2) that the probability assigned by PG to every noise vector depends only on its magnitude regardless of the input of the mechanism. The normalization constant λ ensures that such probabilities sum up to 1, forming a valid distribution on the vectors drawn from $\mathcal{G}$. It is also easy to see that the planar geometric mechanism satisfies ϵ-geo-indistinguishability on $\mathcal{G}$.

Sampling from a discrete mechanism, such as PG, can be performed efficiently since we have an explicit construction of its probability mass function. First, λ can be approximated to any precision, which only needs to be done once for every choice of ϵ, s. Then, to draw a random location z, we uniformly select $p \in [0,1]$ and we iterate on $\mathcal{G}$, starting from x, by visiting locations in increasing order of distance from x. Once the cumulative probability of all visited points reaches p, we report the last visited location.

Finally, remapping can be applied to PG in two ways: one approach is to employ the continuous remap described in the previous section, and project the result to the grid. An alternative approach, is to directly use the definition of the Bayesian remap, but restrict to a radius around z, as discussed in Section 3.1. The second approach has the advantage of being applicable for any choice of $\mathbf{d}_Q$, but is less efficient, especially if the spacing s is small.

4.3 Mechanisms for large but finite grids

The third case that we consider is that of an application within a limited region, for which users are assumed to lie on a finite grid. Still, we would like to cover a large area, and more importantly to use a fine cell size, hence the grid size should be realistically large.

Clearly, the PL and PG mechanisms, with or without remap, can still be employed in this case, by projecting (truncating) their output to the finite grid. Note that the truncation step is an instance of post-processing, since it is independent from the user's real location x, hence it preserves geo-indistinguishability.

On the other hand, the finite nature of the domain allows us to construct alternative mechanisms, that we discuss in the following sections.

4.3.1 Tight-Constraints mechanisms

The tight-constraints mechanism was introduced by [19] in the general setting of $\mathbf{d}$-privacy for any finite space $\mathcal{X}$ of secrets equipped with a privacy metric $\mathbf{d}$. In our context of ϵ-geo-indistinguishability, this mechanism, denoted by TC is defined as follows.

Definition 4.1 (Tight-Constraints mechanism [19]). *Given a finite set of locations $\mathcal{X}$ and $\epsilon > 0$, TC : $\mathcal{X} \to \mathcal{P}(\mathcal{X})$ is a mechanism satisfying:*

$$\mathrm{TC}(x)(z) = e^{-\epsilon\, \mathbf{d}(x,z)}\, \mathrm{TC}(z)(z) \quad \forall x, z \in \mathcal{X}.$$

The above definition relates the probability of reporting an output z, when the real location is z itself, with the same probability from some other location x. This relation means precisely that the ϵ-geo-indistinguishability constraint for the inputs x, z and the output z is satisfied with equality. With regard to this characteristic, the mechanism is named after those 'tight' constraints. It is shown in [19] that TC satisfies ϵ-geo-indistinguishability; furthermore, with respect to the *binary loss* function – defined as $\mathbf{d}_Q(x,z) = 0$ iff $x = z$ and 1 otherwise – this mechanism is optimal for a set of priors called 'regular' priors. While this optimality is guaranteed only for the binary loss function, we experimentally show in Section 5 that TC provides also a substantial improvement in comparison to other mechanisms when the loss is measured as the Euclidean distance between the input and output locations. In particular, TC enjoys substantially better utility than the Exponential mechanism, the only other (efficient) mechanism applicable to an arbitrary privacy metric $\mathbf{d}$.

Existence and construction

It is important to remark that the mechanism TC may not exist in some cases, more precisely when there are no collection of conditional probability distributions (one for every input) that satisfy the tight constraints of Def 4.1. It is shown by the authors of [19] that the necessary and sufficient condition of the existence of this mechanism is related to the privacy-constraints matrix Φ with entries indexed by the elements of $\mathcal{X}$ as follows.

$$\phi_{xz} = e^{-\epsilon\, \mathbf{d}(x,z)} \quad \forall x, z \in \mathcal{X}.$$

Then the mechanism TC exists if and only if there is a vector $\boldsymbol{\mu}$ indexed by $\mathcal{X}$ such that

$$\Phi\, \boldsymbol{\mu} = \mathbf{1} \quad \text{and} \quad \mu_z \geq 0 \quad \forall z \in \mathcal{X}. \tag{3}$$

Every entry μ_z of the vector $\boldsymbol{\mu}$ is precisely the probability of reporting the location z when the user is at z, i.e. $\mu_z = \mathrm{TC}(z)(z)$ for every $z \in \mathcal{X}$. Therefore using the entries of this vector and Definition 4.1, all the conditional probabilities of the mechanism TC are easily obtained by the following simple equation.

$$\mathrm{TC}(x)(z) = \phi_{xz}\mu_z \quad \forall x, z \in \mathcal{X}. \tag{4}$$

When the domain of locations of the users is regarded as a discrete set of points, e.g. a grid, we show in this paper that TC provides significantly less expected loss in comparison to other well-known mechanisms, e.g. the planar Laplace mechanism. However, this advantage comes at the computational cost required to obtain $\boldsymbol{\mu}$ by solving the system of $|\mathcal{X}|$ linear equations in (3). In the following we exploit the graph symmetries between the points of $\mathcal{X}$ to reduce this cost of constructing TC.

15	14	13	12	11	11	12	13	14	15
14	10	9	8	7	7	8	9	10	14
13	9	6	5	4	4	5	6	9	13
12	8	5	3	2	2	3	5	8	12
11	7	4	2	1	1	2	4	7	11
11	7	4	2	1	1	2	4	7	11
12	8	5	3	2	2	3	5	8	12
13	9	6	5	4	4	5	6	9	13
14	10	9	8	7	7	8	9	10	14
15	14	13	12	11	11	12	13	14	15

Fig. 1. Symmetric cells in a grid of size 10×10. The cells having the same label are symmetric to each other, i.e. in the same equivalence class. The highlighted region contains one cell of every equivalence class, and shows that the grid has 15 class.

Reducing the construction cost using symmetries

The domain of locations $\mathcal{X}$ together with the distances between its individual points can be regarded as a weighted graph in which the vertices are the points of $\mathcal{X}$, and the weight for every two vertices x, z is exactly the Euclidean distance $\mathbf{d}(x, z)$. A bijective mapping $\rho : \mathcal{X} \to \mathcal{X}$ is called an automorphism on $\mathcal{X}$ if this mapping preserves the distance between every $x, z \in \mathcal{X}$ in the sense $\mathbf{d}(\rho(x), \rho(z)) = \mathbf{d}(x, z)$. Using the notion of automorphisms, we define symmetric points as follows.

Definition 4.2 (symmetric points)**.** *Consider a domain of points $\mathcal{X}$. For any two points $x, x' \in \mathcal{X}$, we say that x is symmetric to x', written as $x \approx x'$ if there is an automorphism $\rho : \mathcal{X} \to \mathcal{X}$ such that $\rho(x) = x'$.*

It is clear that $\approx$ is an equivalence relation, and therefore it partitions the domain $\mathcal{X}$ into a set of equivalence classes written as $\mathcal{X}/_{\approx}$. Figure 1 illustrates a grid $\mathcal{X}$ of 10×10 cells (locations), in which symmetric cells are annotated by the same label. In other words, the label on every cell refers to the equivalence class that the cell belongs to. With respect to the equivalence classes of $\mathcal{X}$, we define a matrix $\Phi^{\approx}$ of which the rows and columns are indexed by elements (classes) of $\mathcal{X}/_{\approx}$ as follows. For every $c, c' \in \mathcal{X}/_{\approx}$,

$$\phi^{\approx}_{cc'} = \sum_{z \in c'} \phi_{xz} \quad \text{where } x \text{ is any member of } c. \tag{5}$$

Then the following theorem shows that instead of constructing TC by solving the original equations (3) which involves the large matrix Φ, this mechanism can be constructed more efficiently using the smaller matrix $\Phi^{\approx}$.

Theorem 4.3 (Tight-Constraints mechanism)**.** *A TC mechanism exists for a domain of locations $\mathcal{X}$ if and only if there is vector $\boldsymbol{\mu}^{\approx}$ such that*

$$\Phi^{\approx}\, \boldsymbol{\mu}^{\approx} = \mathbf{1} \quad \textit{and} \quad \mu^{\approx}_c \geq 0 \quad \forall c \in \mathcal{X}/_{\approx}. \tag{6}$$

Furthermore, TC is obtained by setting for every $c \in \mathcal{X}/_{\approx}$,

$$\mathrm{TC}(x)(z) = e^{-\epsilon\, \mathbf{d}(x,y)}\, \mu^{\approx}_c \quad \forall x \in \mathcal{X}, \forall z \in c. \tag{7}$$

Using the above theorem, the cost of constructing TC for a domain of locations $\mathcal{X}$ is dramatically reduced by exploiting the symmetries between its locations. For example, in Figure 1, the grid of 100 locations contains only 15 equivalence classes. In general for any grid of size $n \times n$, the number of classes is $n^2/8 + n/4$ if n is even and $(n+1)^2/8 + (n+1)/4$ if n is odd. Clearly this makes solving Equations (6) significantly cheaper than solving the original equations (3).

4.3.2 The exponential mechanism

The exponential mechanism Exp is a generic mechanism applicable to any finite domain $\mathcal{X}$ with an arbitrary privacy metric $\mathbf{d}$. It is given by:

$$\mathrm{Exp}(x)(z) = \lambda_x e^{-\frac{1}{2}\epsilon \mathbf{d}(x,z)} \text{ where } \lambda_x = 1/\textstyle\sum_z e^{-\frac{1}{2}\epsilon \mathbf{d}(x,z)}$$

Compared to the other mechanisms, the $\frac{1}{2}$ factor in the exponent compensates for the fact that the normalization factor λ_x depends on the secret x. The smaller exponent leads to a greater variance of the noise, hence the utility of this mechanism is the worse among those discussed in this section, with the advantage, on the other hand, of being very simple and at the same time applicable to any metric $\mathbf{d}$. The exponential mechanism is used in [18] to achieve privacy wrt a constructed "elastic" metric, adapted to the semantics of each location.

4.3.3 Optimal mechanism built on a coarse grid

Since our domain is finite, we could apply the construction of 2.4 to obtain an optimal mechanism for a certain prior π. However, if $\mathcal{X}$ contains more than a few hundred locations, as it is common in realistic scenarios, this construction is not feasibly applicable.

A natural solution would be to construct the mechanism on a *coarser* grid $\mathcal{X}'$ containing larger cells obtained by merging together several cells of $\mathcal{X}$. Once K is constructed from $\mathcal{X}'$, a user located at $x \in \mathcal{X}$ would draw a noisy location from $K(x')$, where x' is the cell of the coarse grid in which x is situated. This technique is essentially *implied* by the use of very coarse grids in the evaluation of the mechanisms in [5, 7, 8], however it has various shortcomings, as discussed in Section 5.3.

5 Evaluation

We have evaluated all mechanisms discussed above on two real-world datasets from the Gowalla and Brightkite social networks. The results for Gowalla are presented in this section; those for Brightkite, can be found in the appendix. For the infinite continuous and discrete case, we perform an evaluation on the *complete datasets*, showing that an efficient remapping can be performed without any restriction on the area of interest. For the finite case, we consider a large geographical region covering most of the San Francisco peninsula.

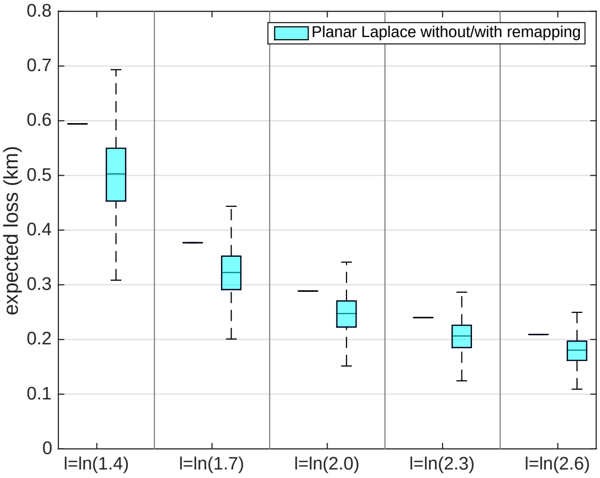


Fig. 2. Results for the continuous case, Euclidean loss

To ensure the practical applicability of the proposed mechanisms, we take a common machine learning approach separating between the training and testing (evaluation) data. More precisely, we split the entire location dataset into two non-overlapping parts. The first part, containing the location data of approximately 80% of the users, is seen as the training set. From this part, we construct a global prior computed as the average of individual priors of all users visiting the region, which is then used to optimally remap the proposed mechanisms. Note that this remap is optimized for the global prior rather than being overfitted to a specific user.

The other part of the dataset, which contains the data of the remaining 20% of the users is seen as the testing set for evaluating the constructed mechanisms. More precisely, we construct a user-specific prior for every user having at least 20 check-ins (in total for the infinite case, or in the specified region for the bounded case), and measure the expected loss of the mechanism for the user using her own prior. Although the mechanisms are trained only on the users of the training part of the dataset, we find that they offer a low level of quality loss also for the users in the testing set.

We should emphasize that the split is performed on users (not on checkins): *no data whatsoever* for the testing users is available in the training dataset. This very conservative approach aims at simulating the case of a new user for which we have no information, other than what is generally available about the whole service. Improving the utility in this case is a strong result; clearly, if we trained on part of the user's own data, the results could be further substantially improved.

All evaluated mechanisms are constructed to satisfy ϵ-geo-indistinguishability, using $\epsilon = l/r$ where $r =$

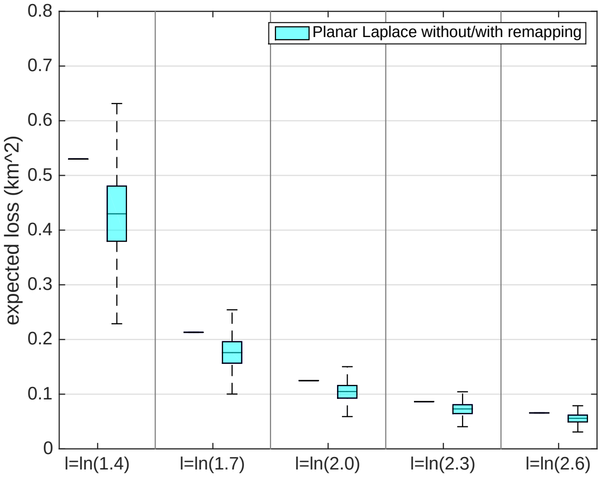


Fig. 3. Results for the continuous case, squared Euclidean loss

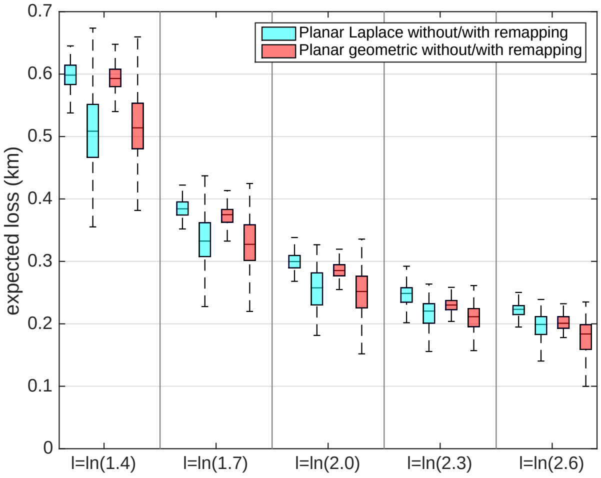


Fig. 4. Results for the discrete infinite case

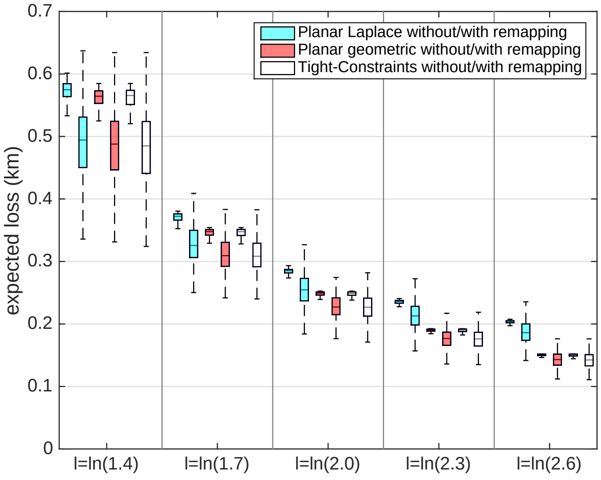


Fig. 5. Results for a large but finite grid (SF area)

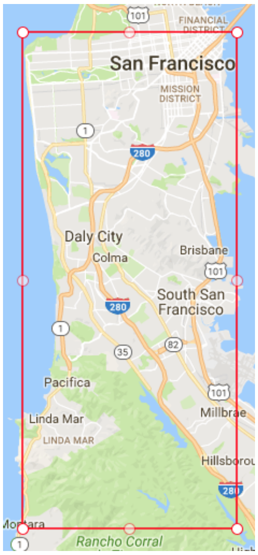


Fig. 6. A rectangular region of size 12 km × 28 km covering most of the San Francisco peninsula

0.1 km and l ranging from $\ln(1.4)$ to $\ln(2.6)$. Utility is measured wrt $\mathbf{d}_Q = \mathbf{d}$, unless stated otherwise.

For the continuous case, we evaluate the planar Laplace mechanism, by itself, or coupled with the continuous remapping technique of Section 3.2. The dataset Q is the whole training set consisting of 48K users and more than 5M checkins. From the list of checkins we create a kd-tree, allowing to quickly construct Q_t. To avoid a single user greatly affecting the posterior, we take $w(x) = 1/u$ where u is the number of checkins of that user in Q_t, and also set $q_{\min} = 20$. In our evaluation, each remap took only a few milliseconds, despite the large dataset. The mechanisms were evaluated in the complete testing dataset of 12K users.

The results are shown in Fig 2 (for $\mathbf{d}_Q = \mathbf{d}$) and Fig 3 (for $\mathbf{d}_Q = \mathbf{d}^2$), where boxplots of the average loss for each user are given for various values of l. Note that the average loss for PL without remap is always the same – given by (1) – hence the boxplots reduce to a single line. On the other hand, the average loss when remapping is employed depends on the user: since remapping is performed using a global prior, it might not always give an improvement. Still, although no user data is used for training, the evaluation shows a substantial improvement in terms of expected loss for most users. The results for $\ln(1.4)$ show that the remapped PL has a mean expected loss of 499 m, while the corresponding mean of the unremapped PL is 594.4 meters, i.e. 19% higher. Of course, not all users benefit from the remap, however only 8.17% of the users have any increase in expected loss, while only 1.27% have an increase of 10% (59.4 m) or more.

For the infinite discrete case, we perform the same evaluation, mapping all points in the Gowalla dataset to a infinite grid of cell size 0.1 km (hence, despite the discretization, locations are given with a relatively high precision). The PL and PG mechanisms are evaluated, with and without remapping. For PG, although we are interested in the Euclidean loss, we use the generic approximation technique of Section 3.1, demonstrating its efficiency for large datasets. The results are shown in Figure 4; similarly to the continuous case, remapping is shown to be effective, despite the use of a generic prior. Moreover, PG slightly outperforms PL in most cases, which is expected since it is tailored to the discrete grid.

Finally, for the evaluation on a large but finite domain, we use the region delimited by the red line shown in Figure 6, covering most of the San Francisco area. This region, 12km × 28km, is bounded from south and north by the latitudes 37.5395, 37.7910, and from west and east by longitudes -122.5153, -122.3789. In the case of the Gowalla dataset, our training set has 5216 users having 103052 check-ins in the considered region. On the other hand, the testing part consists of 273 users, where each one has at least 20 check-ins in the region. To allow acceptable precision of this discretization, we set the side length of every cell to be only 200 meters, hence splitting our region in a 60×140 grid of 8400 cells. Note that, although still finite, this size is two orders of magnitude larger that the one typically considered for constructing optimal mechanisms [5, 7], typically performed for domains of 50 to 100 cells.

For this grid, we construct the truncated planar Laplace mechanism, the truncated planar geometric mechanism, and the tight-constraints mechanism on the 8400 cells covering the required region, with and without remap. The results are displayed in Figure 5, showing that the tight-constraints and the geometric mechanisms clearly outperform the planar Laplace. It is important here to remark that each one of these two mechanisms has an advantage over the other one. The geometric mechanism enjoys the feature that it exists for all levels of privacy due to the symmetry of the infinite grid on which it is constructed. The tight-constraints mechanism, on the other hand, might not exist for very strong levels of privacy; however, it has the advantage of being applicable to any privacy metric $\mathbf{d}$, not necessarily the Euclidean one.

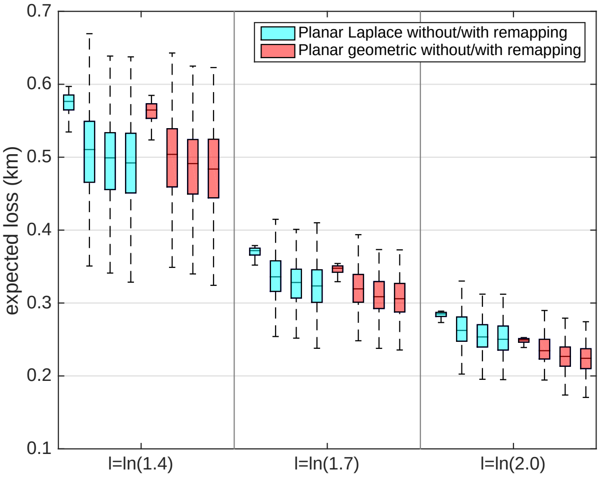


Fig. 7. Remaps using a differentially private prior

Finally, we compare the tight-constraints and the exponential mechanism, the only two (efficient) ones that are applicable to an arbitrary privacy metric $\mathbf{d}$. Apart from the Euclidean one, we evaluate the two mechanisms using the well known maximum[3] (or Chebyshev) metric, defined as $\mathbf{d}_\infty(x,x') = \epsilon \max_i(|x_i - x'_i|)$. This metric corresponds to a privacy property requiring the same level of privacy within a square area, instead of a circle. Note that in both cases $\epsilon = l/r$ scales the corresponding metric, deciding the radius (or the length of the square's sides) in which a privacy level l is required.

It turns out that, for the Euclidean metric, the tight-constraints mechanism significantly outperforms the exponential mechanism. Concerning the maximum metric, the tight-constraints mechanism does not exist for the strong levels of privacy, but when it exists, it outperforms the exponential one. On the other hand, the latter is simpler to compute and always applicable. The details of this comparison are shown in Appendix B.

5.1 Using a differentially private prior

In this section, we evaluate the remapping technique using a differentially private prior, as discussed in Section 3.3. We perform the same evaluation in the SF area, the only difference being that Lap($\frac{2}{N \ln(3)}$) noise is applied to the global prior. Unfortunately, the Gowalla dataset is not ideal for such an evaluation since it contains only 5216 users in the training set for the SF area. As a consequence, we perform two evaluations, one with $N = 5000$, which is relatively small for averag-

[3] Note that the maximum metric is a natural choice for location *traces* [6, 44], although we here use it for single locations.

ing queries, and one with $N = 50000$. In the latter case, we duplicate the users so that we reach the required number; although the results in the latter case are not realistic, we get an idea of what we can expect if we had that many users.

The results are shown in Figure 7. Four versions of PL and PG are evaluated: in the leftmost one no remap is applied while in the rightmost one a remap using the exact global prior is applied. For the two middle ones, a remap using a differentially private global prior is applied, using $N = 5000$ and $N = 50000$ respectively. As expected, the quality of the remap is reduced by the noise applied in the prior. However, even in the case $N = 5000$ the loss is limited, while for $N = 50000$ the noisy prior is almost as good as the exact one.

5.2 Comparison with the optimal method

We have shown that the mechanisms PL, PG, and TC are scalable with respect to the size of the location domain, in contrast to the optimal mechanism OPT, which involves a linear optimization feasible only for a small number of locations. In order to compare these mechanisms with OPT, therefore, we have to consider a relatively small grid: according to our experiments, solving the optimization problem on hundred locations would already require several days of running time. We choose a grid of 10 × 10 square cells, each of size 0.2km × 0.2km, thus covering an area of 2km × 2km. Note that a spacing of 0.2km is still an acceptable granularity, and in line with previous evaluations. Of course, we could choose a coarser grid and cover a larger area (cfr. Section 4.3.3), but there are various problems with that idea, that will be discussed in the next section.

We overlay this grid on a selected dense region in San Francisco, selected in such a way that it captures a large number of check-ins based on Gowalla dataset. More precisely this region is bounded from south and north by the latitudes 37.7737, 37.7917 and from west and east by longitudes -122.4135, -122.3908, and contains 38963 check-ins. Using an average prior computed from these checkins, we construct OPT, and also the remapped versions of the PL, PG, and TC, for various privacy levels. Figure 8 displays the boxplot for the utilities of those mechanisms for 110 testing users.

We can see that, for relatively strong levels of privacy (e.g. $l = \ln(1.4)$), the mechanisms PL, PG, and TC are significantly improved by remapping. This improvement becomes less significant in the case of relaxed privacy (e.g. $l = \ln(2.6)$). In fact when the privacy is strong, a large amount of noise is added by non-remapped mechanisms, giving a considerable margin for reducing the noise according to the available prior.

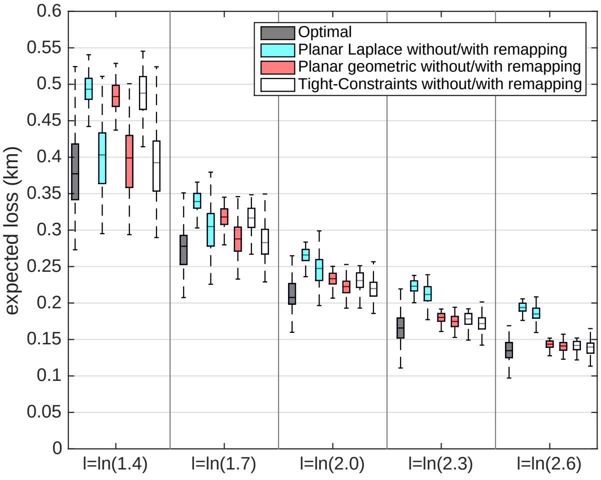


Fig. 8. A Comparison between the optimal mechanism, PL, PG, and TC, constructed on a grid of 100 cells covering a small region in San Francisco

In comparison with OPT, we observe that, for high privacy levels (e.g. $l = \ln(1.4)$), the remapped versions of PL, PG, and TC are only slightly worse than OPT. However, for more relaxed privacy levels, only PG and TC are close to OPT, while PL is not. This is due to the fact that PL adds enough noise to guarantee privacy for all the points of the continuous region, in contrast to PG, TC and OPT, which are tailored to protect only the centers of the cells overlaid on that region (and which are the only points considered for utility).

5.3 Criticisms to the use of a coarse grid

As discussed in Section 4.3.3, one can in general cover a large area with a small number of locations by using a large-granularity grid. However, this idea has a number of drawbacks.

First of all, the granularity of the grid affects privacy in the geo-indistinguishability model. The geo-indistinguishability constraints guarantees that for two locations y, y' in the coarse grid, $K(z), K(y')$ are as similar as the distance $\mathbf{d}(y, y')$ between them. If, however, the mechanism is used by users located at points x, x' of the fine grid (by projecting to the coarse one), the privacy they enjoy is as large as the *distance between the projected points*. Two points x, x' could be projected to the same coarse cell, in which case their effective distance is 0; however, two close points at a border of a coarse cell could be projected to different cells, in which case their effective distance will be much larger that $\mathbf{d}(x, x')$.

A second, related, problem is the following: For a fixed level of privacy l, the more we enlarge the distances between cells the more we relax the geo-indistinguishability constraints, so that, in order to maximize utility, cells will tend to "report themselves" with probability closer and closer to 1, unless they have prior 0. Thus a very coarse grid will generate a quasi-deterministic mechanism, and the resulting effect will be similar to the spacial cloacking (except for the cells with prior 0): points within the same coarse cell will be almost indistinguishable, while points of different coarse cells will be almost completely distinguishable. Appendix C illustrates this problem on real data.

Finally, also the computation of the utility depends on the granularity of the cells, because the distances are computed with respects to the centers of the cells, and the probabilities of points in the same cell are lumped together (or, in other words, users are seen as located at the centers of the cells). Hence, an optimal mechanism constructed on a coarse grid, is by no means guaranteed to be optimal also on a finer one. Appendix D shows that indeed the optimal mechanism constructed on a coarse grid does not compare favorably to the other mechanisms when its utility is (re)computed with respect to the points of the finer grid.

6 Conclusion

In this work, we studied mechanisms for location privacy with emphasis on being *practical*, for realistic domain sizes and without unreasonable assumptions about the prior information available about the user. We discussed such solutions under various constraints for the domain of locations: the complete unconstrained continuous plain, the discrete plain or large but finite domains, using a Bayesian remap as a key ingredient. An extended evaluation was provided on two real-world datasets, showing considerable improvements wrt the standard planar Laplace mechanism.

References

[1] K. Orland, "Stalker Victims Should Check For GPS." The Associated Press, 2003. http://www.cbsnews.com/news/stalker-victims-should-check-for-gps/.

[2] J. Brownlee, "This Creepy App Isn't Just Stalking Women Without Their Knowledge, It's A Wake-Up Call About Facebook Privacy (Update)," 2012. http://www.cultofmac.com/157641/.

[3] J. Simerman, "FasTrak to courthouse." East Bay Times, 2007. http://www.eastbaytimes.com/2007/06/05/fastrak-to-courthouse/.

[4] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[5] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. L. Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proc. of CCS*, pp. 617–627, ACM, 2012.

[6] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proc. of CCS*, pp. 901–914, ACM, 2013.

[7] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. of CCS*, 2014.

[8] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 299–315, 2015.

[9] C. Dwork, "Differential privacy," in *Proc. of ICALP*, vol. 4052 of *LNCS*, pp. 1–12, Springer, 2006.

[10] "Location guard." https://github.com/chatziko/location-guard.

[11] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proc. of CCS*, pp. 239–250, ACM Press, 2014.

[12] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection of mobile apps' location privacy threats," in *Proc. of USENIX Security 2015*, pp. 753–768, USENIX Association, 2015.

[13] C. Ma and C. W. Chen, "Nearby friend discovery with geo-indistinguishability to stalkers," *Procedia Computer Science*, vol. 34, pp. 352 – 359, 2014.

[14] "Qgis processing provider plugin." https://github.com/SpatialVision/differential_privacy.

[15] L. Pournajaf, L. Xiong, V. Sunderam, and X. Xu, "Stac: Spatial task assignment for crowd sensing with cloaked participant locations," in *Proceedings of the 23rd SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, GIS '15, pp. 90:1–90:4, ACM, 2015.

[16] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. of CCS*, pp. 1298–1309, ACM, 2015.

[17] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," in *Proc. of STOC*, pp. 351–360, ACM, 2009.

[18] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Constructing elastic distinguishability metrics for location privacy," *PoPETs*, vol. 2015, no. 2, pp. 156–170, 2015.

[19] E. ElSalamouny, K. Chatzikokolakis, and C. Palamidessi, "Generalized differential privacy: Regions of priors that admit robust optimal mechanisms," in *Horizons of the Mind*, vol. 8464 of *LNCS*, pp. 292–318, Springer Int. Publishing, 2014.

[20] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of MobiSys*, USENIX, 2003.

[21] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information (abstract)," in *Proc. of PODS*, pp. 188–188, ACM Press, 1998.

[22] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[23] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[24] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Trans. Knowl. Data Eng*, vol. 13, no. 6, pp. 1010–1027, 2001.

[25] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Trans. on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[26] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity.," in *ICDE*, vol. 7, pp. 106–115, 2007.

[27] A. Solanas, F. Sebé, and J. Domingo-Ferrer, "Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond," in *Proc. of PAIS 2008*, ACM Int. Conf. Proceeding Series, pp. 61–69, ACM, 2008.

[28] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[29] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *Proc. of ICDE*, pp. 277–286, IEEE, 2008.

[30] S.-S. Ho and S. Ruan, "Differential privacy for location pattern mining," in *Proc. of SPRINGL*, pp. 17–24, ACM, 2011.

[31] R. Dewri, "Local differential perturbations: Location privacy under approximate knowledge attackers," *IEEE Trans. on Mobile Computing*, vol. 99, no. PrePrints, p. 1, 2012.

[32] F. Durr, P. Skvortsov, and K. Rothermel, "Position sharing for location privacy in non-trusted systems," in *Proc. of PerCom 2011*, pp. 189–196, IEEE, 2011.

[33] E. ElSalamouny and S. Gambs, "Differential privacy models for location-based services," *Trans. on Data Privacy*, vol. 9, no. 1, pp. 15–48, 2016.

[34] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. of DAS*, vol. 4602 of *LNCS*, pp. 47–60, Springer, 2007.

[35] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *Proc. of WWW*, pp. 237–246, ACM, 2008.

[36] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. of PERVASIVE*, vol. 3468 of *LNCS*, pp. 152–170, Springer, 2005.

[37] M. Xue, P. Kalnis, and H. Pung, "Location diversity: Enhanced privacy protection in location based services," in *Proc. of LoCA*, vol. 5561 of *LNCS*, pp. 70–87, Springer, 2009.

[38] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. of ICDCS*, pp. 620–629, IEEE, 2005.

[39] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of Differential Privacy using metrics," in *Proc. of PETS*, vol. 7981 of *LNCS*, pp. 82–102, Springer, 2013.

[40] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theor. Comp. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.

[41] L. Cooper and I. Katz, "The weber problem revisited," *Computers & Mathematics with Applications*, vol. 7, no. 3, pp. 225 – 234, 1981.

[42] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. of EUROCRYPT*, vol. 4004 of *LNCS*, pp. 486–503, Springer, 2006.

[43] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proc. of S&P*, pp. 247–262, IEEE, 2011.

[44] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "A predictive differentially-private mechanism for mobility traces," in *Proc. of PETS*, vol. 8555 of *LNCS*, pp. 21–41, Springer, 2014.

A The Brightkite dataset

In this section we evaluate the direct mechanisms (Laplacian, geometric, and tight-constraints) on the Brightkite dataset. The setup, including all configuration options, is identical to the one used for Gowalla (Section 5). The dataset is split into a training set of 40K users and a testing set of 10K users.

The results for the continuous case are shown in Fig 9. Similarly to the case of Gowalla, the remap provides a substantial utility improvement in most cases (although slightly less than that of Gowalla). For ln(1.4) the remapped PL has a mean expected loss of 503 m

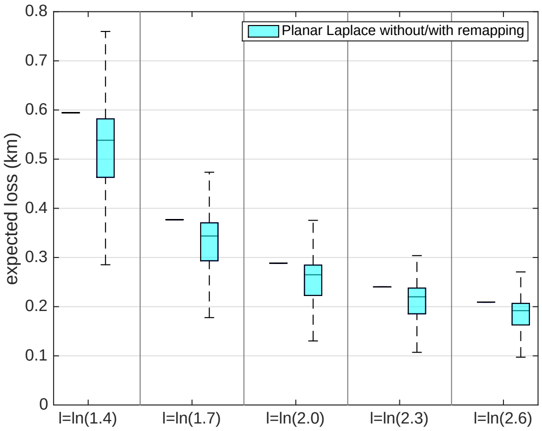


Fig. 9. Brightkite: results for the continuous case, Euclidean loss

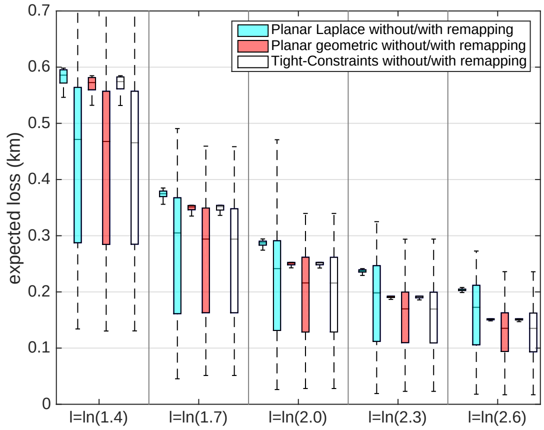
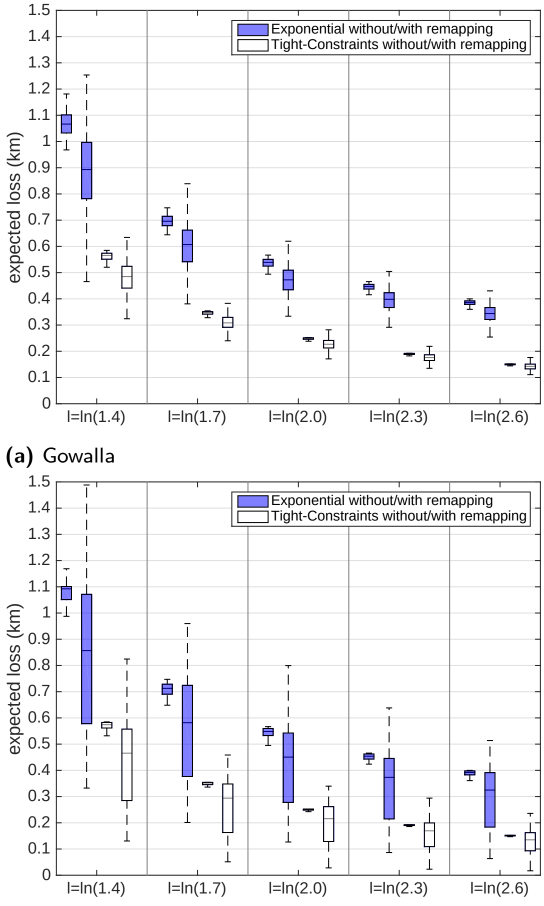


Fig. 10. Brightkite: results for a large but finite grid (SF area)



(a) Gowalla

(b) Brightkite

Fig. 11. The expected loss of the exponential and tight-constraints mechanisms that satisfy geo-indistinguishability on the centers of the large grid (60 × 140 cells) covering San Francisco.

(and median expected loss of 538 m) while the corresponding mean of the unremapped PL is 594.4 meters, i.e. 18% higher. On the other hand, the variance of the error for Brightkite is higher than the one for Gowalla. For this dataset 18.06% of the users have any increase in expected loss, while only 2.66% have an increase of 10% (59.4 m) or more.

The results for a large but finite grid are shown in Fig 10. Again, the utility improvement in most cases is substantial; in fact the median expected error is lower that the one for Gowalla. On the other hand, the variance of the error is visibly much higher. This difference in variance can be explained by the lower quality of data in the Brightkite dataset: many users have checkins in a single location, repeated thousands of times.

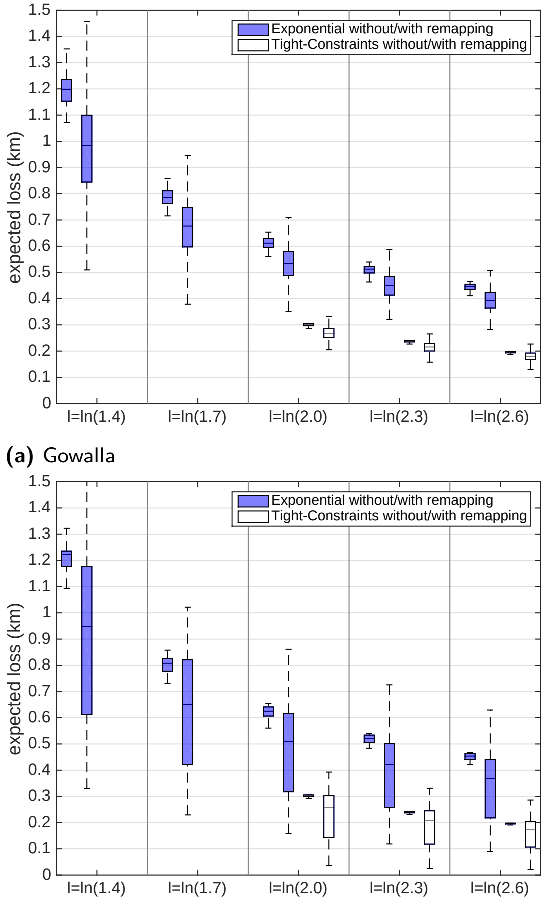
B The tight-constraints vs the exponential mechanism

In this section we compare the tight-constraints and the exponential mechanism, the only two (efficient) ones that are applicable to arbitrary privacy metrics.

Figure 11 illustrates the case of the Euclidean metric, showing that the utility of the tight-constraints mechanism significantly outperforms that of the exponential one. For the maximum metric, defined as $\mathbf{d}_\infty(x, x') = \epsilon \max_i(|x_i - x'_i|)$, the results are shown in Figure 12. We observe in this case that the tight-constraints mechanism does not exist for the stronger levels of privacy ln(1.4) and ln(1.7), while it exists for other levels, in which cases it outperforms the exponential mechanism. Hence we can conclude that whenever the tight-constraints mechanism exists, it outperforms the exponential one, while the latter is simpler to compute and always applicable.

C Diagonal probabilities of the optimal mechanism on a coarse grid

Figure 13 shows the diagonal elements for the optimal mechanism that is constructed on a coarse grid of 14×6 cells covering San Francisco. These elements are organized in the same *grid* format and the value reported in each cell represents the probability of reporting that cell when the user is located in the same cell. Note that all values are either 0 or almost 1. The 0 is due to the fact that the prior is null for those cells.



(a) Gowalla

(b) Brightkite

Fig. 12. The expected loss of the exponential and tight-constraints mechanisms for the maximum metric $\mathbf{d}_\infty$

D Comparative evaluation of the optimal mechanism constructed on a coarse grid

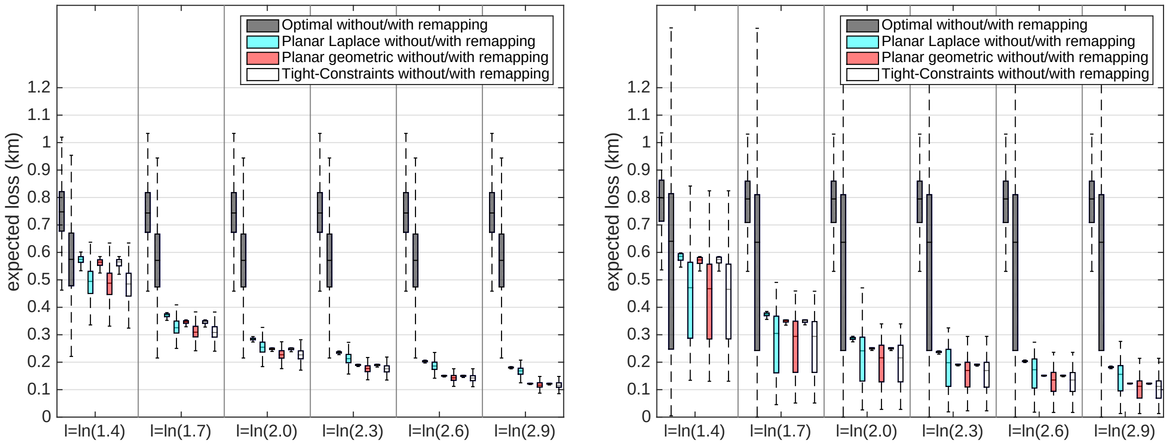
We consider here the optimal ϵ-geo-indistinguishable mechanism constructed on a coarse grid obtained by partitioning the grid of 8400 cells covering the San Francisco area (cfr. Section 5). into 6×14 cells spacing 2.0 km. This mechanism is optimal (by definition) for the centers of the coarse grid, but its utility degrades sensibly computed on the original finer grid. More precisely, we consider the expected loss computed by remapping the outputs of this mechanism to the cells of the original fine grid, using the global prior constructed from the training data. Figure 14 compares such expected loss with that of the other mechanisms, evaluated for the individual users in the testing data, using their own priors, and for various levels of privacy.

0.998964	0.000000	0.000000	0.000000	0.000000	0.997930
0.997929	0.000000	0.000000	0.000000	0.000000	0.997929
0.996894	0.997928	0.000000	0.000000	0.997928	0.996894
0.996896	0.996893	0.000000	0.000000	0.000000	0.997928
0.996896	0.995861	0.996896	0.996896	0.996894	0.996894
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.996896	0.995862	0.995862	0.995862	0.995862	0.996896
0.997930	0.996896	0.996896	0.996896	0.996896	0.997930

Fig. 13. The diagonal entries of the optimal mechanism that satisfies ϵ-geo-indistinguishability for $\epsilon = \ln(1.4)/0.1$ ($l = \ln(1.4)$) on a coarse grid of size 14×6 cells covering San Francisco.

One clear implication from Figure 14 is that the optimal mechanism constructed on a coarse grid (6×14 cells) of the region is no longer optimal for finer and more practical discretization scheme of the region into a grid of 60×140 cells. Although this situation is improved by constructing a remapped version of such mechanism on the fine-grained grid (using a global prior), the resulting mechanism still incurs higher quality loss, i.e. worse utility, relative to the other mechanisms (the Laplace, the geometric, and the tight-constraints ones). Another important observation is that this optimal mechanism (on the coarse grid) and its remapped version maintain the same expected loss as the privacy level l grows. This is due to the fact that the optimal mechanism constructed on such coarse grid becomes almost a deterministic 'cloaking' mechanism (cfr. Section 5.3), regardless of the privacy level. As a consequence, the remapped version of this fixed mechanism to the cells of a finer grid depends only on the global prior on this fine grid, with no regard to the privacy level.

In contrast to the uniform behavior of the optimal mechanism and its remapped version, we observe that the expected losses of other mechanisms are significantly influenced by the imposed privacy level and provide better trade-off between the privacy requirements and the utility.



(a) Gowalla

(b) Brightkite

Fig. 14. The expected loss of privacy mechanisms using Gowalla (a) and Brightkite (b) datasets on the geographical region of Figure 6. The optimal mechanism is constructed on a coarse grid (6×14 cells) covering the region, and its remapped version is constructed on the large fine grained grid (60×140 cells). The other mechanisms are constructed directly on the fine grained grid. The parameter l describes the level of privacy within 0.1km.

E Proofs

We present here all proofs omitted from the paper due to space constraints.

The following is a straightforward adaptation of a similar result for standard differential privacy ([40] Prop. 2.1).

Proposition 3.1 (Remap preserves privacy)**.** *Let $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ be a mechanism and $R : \mathcal{Z} \to \mathcal{Z}$ be a remap. If K satisfies ϵ-geo-indistinguishability then so does KR.*

Proof. The proof lies on the fact that the probability for KR to report an observation in $Z \subseteq \mathcal{Z}$, is the same as the probability of K to report an observation in $R^{-1}(Z) = \{z \in \mathcal{Z} : R(z) \in Z\}$:

$$KR(x)(Z) = K(x)(R^{-1}(Z)) \qquad \forall x \in \mathcal{X}, Z \subseteq \mathcal{Z} \tag{8}$$

Hence, for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$ we have that:

$$\begin{aligned} KR(x)(Z) &= K(x)(R^{-1}(Z)) && (8)\\ &\leq e^{\epsilon\mathbf{d}(x,x')}K(x')(R^{-1}(Z)) && \epsilon\text{-geo-ind. of } K\\ &= e^{\epsilon\mathbf{d}(x,x')}KR(x')(Z) && (8)\end{aligned}$$

□

Proposition 3.3 (Bayesian remap)**.** *Given π, K, $\mathbf{d}_Q$, the remap defined by:*

$$R(z) = \operatorname*{arg\,min}_{z^* \in \mathcal{Z}} \textstyle\sum_{x\in\mathcal{X}} \sigma(x)\mathbf{d}_Q(x, z^*) \qquad \textit{where}$$
$$\sigma = \mathbf{Post}(\pi, K, z)$$

is optimal for K wrt $\pi, \mathbf{d}_Q$.

Proof. From the definition of $R(z)$ we have that, for all $z, z^* \in \mathcal{Z}$,

$$\sum_x \pi(x)K(x)(z)\mathbf{d}_Q(x, z^*) \geq \sum_x \pi(x)K(x)(z)\mathbf{d}_Q(x, R(z)) \tag{9}$$

Let $R' : \mathcal{Z} \to \mathcal{Z}$ be an abritrary remapping function. We have that

$$\begin{aligned} \mathrm{QL}(KR', \pi, \mathbf{d}_Q) &= \sum_{\substack{x\in\mathcal{X}\\ z\in\mathcal{Z}}} \pi(x)K(x)(z)\mathbf{d}_Q(x, R'(z))\\ &\geq \sum_{\substack{x\in\mathcal{X}\\ z\in\mathcal{Z}}} \pi(x)K(x)(z)\mathbf{d}_Q(x, R(z))\\ &\qquad \text{(using (9) with } z^* = R'(z))\\ &= \mathrm{QL}(KR, \pi, \mathbf{d}_Q)\end{aligned}$$

□

Proposition 3.4. *For all $\sigma \in \mathcal{P}(\mathbb{R}^2)$ with finite $\lceil\sigma\rceil$:*

$$\textsc{Centroid}(\sigma) = \operatorname*{arg\,min}_{z^*\in\mathbb{R}^2} \textstyle\sum_{x\in\lceil\sigma\rceil} \sigma(x)\,\mathbf{d}^2(x, z^*)$$

Proof. Let $x \in \mathbb{R}^2$, we denote by $\|x\|$ the Euclidean norm. Recall that $\|x\|^2 = x \cdot x$ where $\cdot$ denotes the dot product. Let $c = \textsc{Centroid}(\sigma), z^* \in \mathbb{R}^2$, we have that:

$$\begin{aligned} &\sum_{x\in\lceil\sigma\rceil} \sigma(x)\|x - z^*\|^2\\ &= \sum_{x\in\lceil\sigma\rceil} \sigma(x)\|x - c + c - z^*\|^2\end{aligned}$$

$$
\begin{aligned}
&= \sum_{x\in\lceil\sigma\rceil} \sigma(x)(x-c+c-z^*)\cdot(x-c+c-z^*)\\
&= \sum_{x\in\lceil\sigma\rceil} \sigma(x)\left(\|x-c\|^2+2(x-c)\cdot(c-z^*)+\|c-z^*\|^2\right)\\
&= \sum_{x\in\lceil\sigma\rceil} \sigma(x)\|x-c\|^2+2(c-z^*)\cdot\sum_{x\in\lceil\sigma\rceil}\sigma(x)(x-c)+\\
&\quad \sum_{x\in\lceil\sigma\rceil}\sigma(x)\|c-z^*\|^2
\end{aligned}
$$

Finally, since $\sum_{x\in\lceil\sigma\rceil}\sigma(x)=1$, $\sum_{x\in\lceil\sigma\rceil}\sigma(x)(x-c)=\mathbf{0}$ and $\|c-z^*\|^2\geq 0$ we get that

$$
\begin{aligned}
\sum_{x\in\lceil\sigma\rceil}\sigma(x)\|x-z^*\|^2 &= \sum_{x\in\lceil\sigma\rceil}\sigma(x)\|x-c\|^2+\|c-z^*\|^2\\
&\geq \sum_{x\in\lceil\sigma\rceil}\sigma(x)\|x-c\|^2
\end{aligned}
$$

□

Theorem 4.3 (Tight-Constraints mechanism)**.** *A TC mechanism exists for a domain of locations $\mathcal{X}$ if and only if there is vector $\boldsymbol{\mu}^{\approx}$ such that*

$$\Phi^{\approx}\boldsymbol{\mu}^{\approx}=\mathbf{1} \quad \textit{and} \quad \mu^{\approx}_c\geq 0 \quad \forall c\in\mathcal{X}/_{\approx}. \qquad (6)$$

Furthermore, TC is obtained by setting for every $c\in\mathcal{X}/_{\approx}$,

$$\mathrm{TC}(x)(z)=e^{-\epsilon\,\mathbf{d}(x,y)}\,\mu^{\approx}_c \quad \forall x\in\mathcal{X},\forall z\in c. \qquad (7)$$

Proof. In the following we will denote the equivalence class of every $x\in\mathcal{X}$ by c_x. Suppose that there is a vector $\boldsymbol{\mu}^{\approx}$ that satisfies (6). Then it holds for every $x\in\mathcal{X}$ that $\sum_{c\in\mathcal{X}/_{\approx}}\phi^{\approx}_{c_x c}\,\mu^{\approx}_c=1$. The latter equation can be expanded using the definition (5) of $\phi^{\approx}_{c_x c}$ as

$$\sum_{c\in\mathcal{X}/_{\approx}}\sum_{z\in c}\phi_{xz}\,\mu^{\approx}_c=1.$$

From the above equation, it is clear that (3) is satisfied with the vector $\boldsymbol{\mu}$ defined as $\mu_z=\mu^{\approx}_c$ for all $c\in\mathcal{X}/_{\approx}$ and $z\in c$. Since also $\mu_z\geq 0$, the tight-constraints mechanism exists. In this case Eq. (4), which evaluates the probabilities of TC, yields Eq. (7) since $\phi_{xz}=e^{-\epsilon\,\mathbf{d}(x,y)}$ and $\mu_z=\mu^{\approx}_c$.

Conversely, suppose that the tight-constraints mechanism exists, i.e. there is a vector $\boldsymbol{\mu}$ that satisfies the system of equations (3). We show in the following that there is vector $\boldsymbol{\mu}^{\approx}$ that satisfies the reduced system (6). Let $\{\rho_i : i=1,2,\ldots,n\}$ be the set of all different automorphisms on $\mathcal{X}$. Now consider any class $c\in\mathcal{X}/_{\approx}$, and any member $x\in c$. Then by (3), it holds

$$\sum_{z\in\mathcal{X}}\phi_{\rho_i(x)\rho_i(z)}\,\mu_{\rho_i(z)}=1, \quad 1\leq i\leq n.$$

The above equations are exactly the instances of (3) that correspond to the locations $\rho_i(x)$ for $1\leq i\leq n$, i.e. the members of c. Note also in the above summation, $\rho_i(z)$ considers every element of $\mathcal{X}$ because y iterates on all these elements and ρ_i is bijective. Using the fact that $\phi_{\rho_i(x)\rho_i(z)}=\phi_{xz}$, we combine the above equations into one as follows.

$$\sum_{c'\in\mathcal{X}/_{\approx}}\sum_{z\in c'}\phi_{xz}\left(\sum_{i=1}^{n}\mu_{\rho_i(z)}\right)/n=1.$$

In the above equation, note that the quantity $(\sum_{i=1}^{n}\mu_{\rho_i(z)})/n$ is non-negative and depends only the class c' of y. Denoting this quantity by $\mu^{\approx}_{c'}$, and recalling that $\sum_{z\in c'}\phi_{xz}=\phi^{\approx}_{cc'}$, the above equation can be rewritten as follows.

$$\sum_{c'\in\mathcal{X}/_{\approx}}\phi^{\approx}_{cc'}\,\mu^{\approx}_{c'}=1.$$

Since c was chosen arbitrarily, the above equation holds for every member of $\mathcal{X}/_{\approx}$. Since also $\mu^{\approx}_{c'}\geq 0$ for all c', the system of equations in (6) is satisfied. □

Proposition 3.5. *For all π, K, R, $\mathbf{d}_A$:*

$$\textsc{AdvError}(K,\pi,\mathbf{d}_A)\leq\textsc{AdvError}(KR,\pi,\mathbf{d}_A)$$

Proof. The result is a direct consequence of the fact that for any inference attack H on KR, there exists an attack H' on K such that

$$\mathrm{QL}(KRH,\pi,\mathbf{d}_A)=\mathrm{QL}(KH',\pi,\mathbf{d}_A)$$

This comes from the associativity of matrix multiplication by taking $H'=RH$. □