Konstantinos Chatzikokolakis*, Catuscia Palamidessi, and Marco Stronati

Constructing elastic distinguishability metrics for location privacy

Abstract: With the increasing popularity of hand-held devices, location-based applications and services have access to accurate and real-time location information, raising serious privacy concerns for their users. The recently introduced notion of *geo-indistinguishability* tries to address this problem by adapting the well-known concept of differential privacy to the area of location-based systems. Although geo-indistinguishability presents various appealing aspects, it has the problem of treating space in a uniform way, imposing the addition of the same amount of noise everywhere on the map. In this paper we propose a novel elastic distinguishability metric that warps the geometrical distance, capturing the different degrees of density of each area. As a consequence, the obtained mechanism adapts the level of noise while achieving the same degree of privacy everywhere. We also show how such an elastic metric can easily incorporate the concept of a "geographic fence" that is commonly employed to protect the highly recurrent locations of a user, such as his home or work. We perform an extensive evaluation of our technique by building an elastic metric for Paris' wide metropolitan area, using semantic information from the OpenStreetMap database. We compare the resulting mechanism against the Planar Laplace mechanism satisfying standard geo-indistinguishability, using two real-world datasets from the Gowalla and Brightkite location-based social networks. The results show that the elastic mechanism adapts well to the semantics of each area, adjusting the noise as we move outside the city center, hence offering better overall privacy. [1]

Keywords: location privacy, differential privacy, distinguishability metric

DOI 10.1515/popets-2015-0023
Received 2015-02-15; revised 2015-05-13; accepted 2015-05-15.

1 Introduction

The availability of devices capable detecting the geographical position with pretty good accuracy (e.g. wifi-hotspots, GPS, etc.) has led to a proliferation of applications that use the location data to provide a range of services. These applications, called location-based services (LBSs), include points-of-interest retrieval, coupon providers, GPS navigation, location aware social networks, etc. While the value of these services is undeniable, as attested by their vast popularity, at the same time there are growing concerns about the potential privacy breaches that the user is exposed to, due to the constant disclosure of location information.

Among the various approaches proposed in the literature to address the problem of privacy in the use of LBSs, those based on obfuscating the real location by adding random noise [2, 14, 28, 29] have emerged as the most robust to side-information-enhanced attacks. Additionally, the *geo-indistinguishability* framework [2, 14] has the appealing features of providing formal privacy guarantees independent from the user's prior, being robust with respect to combination attacks, and offering a good trade off between privacy and utility. These features are inherited from the framework of differential privacy [13] which prevents an adversary from distinguishing datasets that are "close" based on the Hamming metric. Geo-indistinguishability follows the same principles while using the Euclidean distance between locations.

Formally, a mechanism provides geo-indistinguishability if the user's real location x is *indistinguishable* from other nearby locations x', meaning that the mechanism should report any noisy location z with a probability similar to that of reporting the same z when the real location is x'. The level of similarity depends on the geographical distance between x and x'. More precisely, the ratio between the two probabilities is bound by an expression that grows exponentially with the distance. This means that the mechanism protects the accuracy of the real location but reveals that the user is, say, in Paris instead than London, which is appropriate for the kind of applications (LBSs) we are targeting. Geo-indistinguishability is typically achieved by reporting a location obtained from the user's location by adding noise drawn from a 2-dimensional Laplace distribution.

It is to be noted that the intuition behind most notions of privacy is that of *hiding in a crowd*. The "crowd" may be of

1 This work was partially supported by the European Union 7th FP project MEALS, by the project ANR-12-IS02-001 PACE, and by the INRIA Large Scale Initiative CAPPRIS.

***Corresponding Author: Konstantinos Chatzikokolakis:** CNRS and LIX, École Polytechnique, France
Catuscia Palamidessi: INRIA and LIX, École Polytechnique, France
Marco Stronati: LIX, École Polytechnique, France

different nature: we might want to hide among several people, hide the kind of shop we visit, the activity we perform, etc. Geo-indistinguishability provides this property in the context of the information that can be inferred from the location. By properly setting a parameter of the definition, one can establish the area of the points indistinguishable from the real location, that is, how many elements (resident people, shops, recreational centers etc.) are made indistinguishable from the ones in the real location.

However, one problem with the geo-indistinguishability framework is that, being based on the Euclidean distance, its protection is uniform in space, while the density of the elements that constitute the "crowd" in general is not. This means that, once the privacy parameter is fixed, a mechanism providing geo-indistinguishability will generate the same amount of noise independently of the real location on the map, i.e., the same protection is applied in a dense city and in a sparse countryside. As a consequence, an unfortunate decision needs to be made: one could either tune the mechanism to the amount of noise needed in a dense urban environment, leaving less dense areas unprotected. Or, to ensure the desired level of privacy in low-density areas, we can tune the mechanism to produce a large amount of noise, which will result in an unnecessary degradation of utility in high-density areas.

The idea we pursue in this paper is to adopt a privacy notion that reflects the local characteristics of each area. This can be achieved while maintaining the main principles of geo-indistinguishability, by replacing the Euclidean distance with a constructed *distinguishability metric* $d_{\mathcal{X}}$. The resulting notion, called $d_{\mathcal{X}}$-privacy in [7], ensures that secrets which are close wrt $d_{\mathcal{X}}$ should remain indistinguishable, while secrets that are distant wrt $d_{\mathcal{X}}$ are allowed to be distinguished. We then have the flexibility to adapt the distinguishability metric $d_{\mathcal{X}}$ to our privacy needs.

Going back to the intuition of privacy as being surrounded by a crowd, we can reinterpret it in the light of distinguishability metrics. On one hand people or points of interest can be abstracted as being a *privacy mass* that we can assign differently to every location. On the other hand the concept of being close to a location rich in privacy can be seen as the desire to be similar, or indistinguishable to such a location. Therefore we can express our intuitive privacy with a distinguishability metric that satisfies the following requirement: every location should have in proximity a certain amount of privacy mass. This can be better formalized with a requirement function $\mathrm{req}(l)$ that for every distinguishability level l, assigns a certain amount of privacy mass that must be present within a radius l in the metric $d_{\mathcal{X}}$. Contrary to geo-indistinguishability, that considers space uniform and assigns to every location the same privacy value, we build a metric that is flexible and adapts to a territory where each location has a different privacy importance. In comparison with the Euclidean metric, our metric stretches very private areas and compresses the privacy poor ones in order to satisfy the same requirement everywhere, for this reason we call it an *elastic metric*. By using $d_{\mathcal{X}}$-privacy with a metric that takes into account the semantics of each location, we preserve the strengths of the geo-indistinguishability framework while adding flexibility, borrowing ideas from the line of work of l-diversity [22, 32]. This flexible behavior reflects also on the utility of the resulting mechanism, areas poor in privacy will result in more noisy sanitization.

We then need a way to compute the actual metric $d_{\mathcal{X}}$ satisfying the requirement $\mathrm{req}(l)$. We propose a graph-based algorithm that can efficiently compute an elastic metric for a large number of locations. The algorithm requires a set of locations marked with privacy mass and a privacy requirement to satisfy. Starting from an empty graph, it iteratively adds weighted edges to satisfy the requirement. The resulting distance between two locations is the weight of the shortest path connecting them. Once obtained our metric we show how to use an exponential distribution to obtain automatically a $d_{\mathcal{X}}$-private mechanism that can also be efficiently implemented.

Another problem that arises in the approaches to location privacy based on random noise – and geo-indistinguishability is not immune to it – is that the reiterate use of the mechanism will eventually disclose the frequently visited locations, such as home or office. A solution (commonly employed for various privacy purposes) consist in building a "fence" around sensitive locations so that all points inside are completely indistinguishable from each other. In this way the attacker will be able, after many iterations, to identify the fence but not the exact location inside the fence. We show that such a solution can be elegantly expressed in the distinguishability metric $d_{\mathcal{X}}$, and can be easily incorporated in our algorithm.

Finally, we show the applicability of our technique by evaluating on two real-world datasets. We start by building an elastic metric for Paris' wide metropolitan area, in a grid of $562,500$ locations covering an area of 5600 km^2. Privacy mass is computed from semantic information extracted from the OpenStreetMap database, and the whole computation is performed in under a day with modest computational capability, demonstrating the scalability of the proposed algorithm.

We then compare the elastic mechanism to the Planar Laplace mechanism satisfying geo-indistinguishability, on two large areas in the center of Paris as well as a nearby suburb. The evaluation is performed using the datasets of Gowalla and Brightkite[10], two popular location-based social networks, and the widely used Bayesian privacy and utility metrics of Shokri et al. [28]. The results show that the dynamic behavior of the elastic mechanism, in contrast to Planar Laplace, provides adequate privacy in both high and low-density areas.

Contributions

- We propose the use of elastic metrics to solve the flexibility problem of geo-indistinguishability. We formalize a requirement of such metrics in terms of privacy mass, capturing properties such as space, population, points of interest, etc.
- We propose an efficient and scalable graph-based algorithm to compute a metric $d_{\mathcal{X}}$ satisfying this requirement.
- We show that the technique of geo-fences can be elegantly expressed in the metric and incorporated in the algorithm.
- We perform an extensive evaluation of our technique in a large metropolitan area using two real-world datasets, showing the advantages of the elastic metric compared to standard geo-indistinguishability.

1.0.0.1 Plan of the paper

In the next section we recall some preliminary notions about $d_{\mathcal{X}}$-privacy and geo-indistinguishability. In Section 3 we present in detail the elastic metric. First how to extract meaningful privacy resource for each location, then the definition of a privacy requirement and finally the graph-based algorithm that generated the metric. In Section 4 we describe how to model geographical fences with a metric and how to integrate it with the elastic metric algorithm. Finally in Section 5 an elastic mechanism is built and evaluated in comparison with a geo-indistinguishable mechanism.

2 Preliminaries

We briefly recall here some useful notions from the literature.

Probabilistic model

We first introduce a simple model used in the rest of the paper. We start with a set $\mathcal{X}$ of *points of interest* (i.e. a subset of $\mathbb{R}^2$), typically the user's possible locations. Moreover, let $\mathcal{Z}$ be a set of possible *reported locations*, which typically coincides with $\mathcal{X}$, although this is not necessary. For the needs of this paper we consider $\mathcal{X}, \mathcal{Z}$ to be finite.

The selection of a reported value $z \in \mathcal{Z}$ is *probabilistic*; z is typically obtained by adding random noise to the actual location x. The set of probability distributions over $\mathcal{Z}$ is denoted by $\mathcal{P}(\mathcal{Z})$. We define the *multiplicative distance* between two distributions $\mu_1, \mu_2 \in \mathcal{P}(\mathcal{Z})$ as $d_{\mathcal{P}}(\mu_1, \mu_2) = \sup_{Z \subseteq \mathcal{Z}} |\ln \frac{\mu_1(Z)}{\mu_2(Z)}|$, with the convention that $|\ln \frac{\mu_1(Z)}{\mu_2(Z)}| = 0$ if both $\mu_1(Z), \mu_2(Z)$ are zero and ∞ if only one of them is zero.

A *mechanism* is a (probabilistic) function $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ assigning to each location $x \in \mathcal{X}$ a probability distribution on $\mathcal{Z}$, where $K(x)(z)$ is the probability to report $z \subseteq \mathcal{Z}$, when the user's location is x.

Geo-indistinguishability and $d_{\mathcal{X}}$-privacy

The notion of *geo-indistinguishability* was proposed in [2] as an extension of differential privacy in the area of location privacy. The main idea behind this notion is that it forces the output of the mechanism applied on locations x, x', i.e. the distributions $K(x), K(x')$, to be similar when x and x' are geographically close, preventing an adversary from distinguishing them, while it relaxes the constraint when x, x' are far away from each other, allowing a service provider to distinguish points in Paris from those in London. Let $d_{\text{euc}}(\cdot, \cdot)$ denote the Euclidean metric; a mechanism K satisfies ϵ-geo-indistinguishability iff for all x, x':

$$d_{\mathcal{P}}(K(x), K(x')) \leq \epsilon d_{\text{euc}}(x, x')$$

Equivalently, the definition can be formulated as $K(x)(z) \leq e^{\epsilon d_{\text{euc}}(x,x')} K(x')(z)$ for all $x, x' \in \mathcal{X}, z \in \mathcal{Z}$.

The quantity $\epsilon d_{\text{euc}}(x, x')$ can be viewed as the *distinguishability level* between the secrets x and x'. The use of the Euclidean metric d_{euc} is natural for location privacy: the *closer* (geographically) two points are, the *less distinguishable* we would like them to be. Note, however, that any other (pseudo-)metric could be used instead of d_{euc}, such as the Manhattan metric or driving distance, depending on the application. The definition that we obtain by using an arbitrary distinguishability metric $d_{\mathcal{X}}$, i.e. requiring that $d_{\mathcal{P}}(K(x), K(x')) \leq d_{\mathcal{X}}(x, x')$, is referred to as $d_{\mathcal{X}}$-privacy, and is studied on its own right in [7]. In Section 3 it is argued that a properly constructed $d_{\mathcal{X}}$ can provide location privacy while taking into account the semantic properties of each location.

It should be noted that standard differential privacy simply corresponds to $\epsilon d_h(x, x')$-privacy, where d_h is the Hamming distance between databases x, x', i.e. the number of individuals in which they differ. If x, x' are *adjacent*, i.e. they differ in a single individual, then $d_h(x, x') = 1$ and the distinguishability level between such databases is exactly ϵ.

Two characterization results are also given in [2, 7], providing intuitive interpretations of geo-indistinguishability and $d_{\mathcal{X}}$-privacy.

It should be emphasized that geo-indistinguishability aims at protecting the user's *location*, not his *identity* (protecting the link between a person identity and its location is the goal of several techniques in the location privacy literature, see Section 6.0.0.6). In our setting a user might even be authenticated with the LBS, for instance to obtain personalized search results, while wishing to protect his location.

Distinguishability level and ϵ

A point worth emphasizing is the role of the distinguishability level, and its relationship to ϵ in each definition. The distinguishability level between two secrets x, x' is their distance in the *complete* privacy metric employed, that is $\epsilon d_h(x, x')$ for differential privacy, $\epsilon d_{\text{euc}}(x, x')$ for geo-indistinguishability, and $d_{\mathcal{X}}(x, x')$ for $d_{\mathcal{X}}$-privacy. Secrets that are assigned a "small" distinguishability level will remain indistinguishable, providing privacy, while secrets with a large distance are allowed to be distinguished in order to learn something from the system. Typical values that are considered "small" range from 0.01 to $\ln 4$; we denote a small distinguishability level by l^*, and in our evaluation we use $l^* = \ln 2$.

In the case of differential privacy, ϵ is exactly the distinguishability level between adjacent databases (since $d_h(x, x') = 1$ for such databases), hence we directly use our "small" level l^* for ϵ. For geo-indistinguishability, however, ϵ represents the distinguishability level for points such that $d_{\text{euc}}(x, x') = 1$; however, depending on the unit of measurement as well as on the application at hand, we might or might not want to distinguish points at a unit of distance. To choose ϵ in this case, we start by defining a radius r^* of *high protection*, say $r^* = 300$ meters for an LBS application within a big city, and we set $\epsilon = l^*/r^*$. As a consequence, points within r^* from each other will have distinguishability level at most l^*, hence an adversary will be unable to distinguish them, while points will become increasingly distinguishable as the geographic distance between them increases.

Note the difference between $d_{\text{euc}}(x, x')$, the geographic distance between x, x', and $\epsilon d_{\text{euc}}(x, x')$, the distinguishability level between x, x'. In other words, ϵ stretches the Euclidean distance turning it into a distinguishability metric. To avoid confusion, throughout the paper we use r to denote geographical distances and radii, and l to denote distinguishability levels.

In the case of $d_{\mathcal{X}}$-privacy, $d_{\mathcal{X}}(x, x')$ directly gives the distinguishability level between x and x'. In Section 3 we investigate some properties that such a metric should satisfy in order to provide location privacy while taking into account the semantics of each location. Then, in Section 3.4, we propose a graph-based algorithm for constructing such a metric.

Repeated application

Any obfuscation mechanism is bound to cause privacy loss when used repeatedly. In the case of an ϵ-geo-indistinguishable mechanism K, applying it n times will satisfy $n\epsilon$-geo-indistinguishability. This is typical in the area of differential privacy, in which ϵ is thought as a budget which is consumed with every query.

The situation is similar in the case of $d_{\mathcal{X}}$-privacy: applied n times, a mechanism will satisfy $nd_{\mathcal{X}}$-privacy. This means that the distinguishability level between x, x' after n applications is $nd_{\mathcal{X}}(x, x')$; if $d_{\mathcal{X}}(x, x') > 0$ then as n grows x and x' are bound to become completely distinguishable. However, if we use a pseudo-metric such that $d_{\mathcal{X}}(x, x') = 0$, then x, x' are completely indistinguishable, and will remain so under any number of repetitions n. This property will be exploited by the "fence" technique of Section 4.

Planar Laplace and Exponential mechanisms

The typical approach for achieving differential privacy is adding noise from some sort of Laplace-like distribution. In the case of the Euclidean distance, the *Planar Laplace (PL)* mechanism [2] can be employed. When applied on location x, this mechanism draws a location z from the continuous plane with probability density function:

$$\frac{\epsilon}{2\pi} e^{-\epsilon d_{\text{euc}}(x,z)}$$

In [2] a method to efficiently draw from this distribution is given, which uses polar coordinates and involves drawing an angle and a radius from a uniform and a gamma distribution respectively. The mechanism can be further discretized and truncated, and can be shown to satisfy ϵ-geo-indistinguishability.

Furthermore, in the case of an arbitrary distinguishability metric $d_{\mathcal{X}}$, a variant of the Exponential mechanism [23] can be employed. When applied at location x, this mechanism reports z with probability:

$$c_x e^{-\frac{1}{2} d_{\mathcal{X}}(x,z)} \quad \text{with} \quad c_x = \left(\textstyle\sum_{z'} e^{-\frac{1}{2} d_{\mathcal{X}}(x,z')}\right)^{-1}$$

where c_x is a normalization factor. This mechanism can be shown to satisfy $d_{\mathcal{X}}$-privacy. Note the difference in the exponent between the two mechanisms: the Exponential mechanism has a factor $\frac{1}{2}$ missing from the Planar Laplace; in the proof of $d_{\mathcal{X}}$-privacy, this factor compensates for the fact that the normalization factor c_x is different for every x, in contrast to the Planar Laplace while the normalization factor $\frac{\epsilon}{2\pi}$ is independent from x. The advantage of this technique is the possibility of obtaining a privacy mechanism independently of the metric used, allowing us to focus solely on the metric design.

Utility

The goal of a privacy mechanism is not to hide completely the secret but to disclose enough information to be useful for some service while hiding the rest to protect the user's privacy. Typically these two requirements go in opposite directions: a stronger privacy level requires more noise which results in a lower utility.

Utility is a notion very dependent on the application we target. An example of noise insensitive applications are weather forecast services, where utility remains unchanged even if the reported location is kilometers away from the real one. On the contrary an POI research application can tolerate lower noise addition in order to report meaningful results.

However, to evaluate and compare *general-purpose* location obfuscation mechanisms, the general principle is to report locations as close as possible to the original ones. A natural and widely used choice [5, 28, 29] is to define utility as the *expected geographical distance* between the actual and the reported locations. Hence, we define the average error of mechanism K on location x as:

$$E_K(x) = \sum_z K(x)(z)\ d_{\mathrm{euc}}(x, z)$$

In the case of the Planar Laplace mechanism, the error is independent from x (due to the symmetry of the continuous plane) and is given by $E_{\mathrm{PL}}(x) = 2/\epsilon$.

3 An elastic distinguishability metric

Rarely we are interested in hiding our geographical location per se, more commonly we consider our location sensitive because of the many personal details it can indirectly reveal about us. For this reason reducing the *accuracy*, through perturbation or cloaking, is considered an effective technique for reporting a location that is at the same time meaningful and decoupled from its sensitive semantic value. In the preliminaries we explained how in geo-indistinguishability the privacy level is configured for a specific radius r^*, that is perceived as private. Using $r^* = 300$ m for a large urban environment is based on the fact that a large number of shops, services and people can be found within that radius, limiting the power of inference of the attacker. This is an intuitive notion of location privacy that we call *hiding in a crowd*, where the crowd represents the richness and variety that a location provides to the user's privacy.

As explained in the introduction, the use of $\epsilon d_{\mathrm{euc}}$ as the distinguishability metric has a major drawback. The simple use of geographical distance to define privacy ignores the nature of the area in which distances are measured. In a big city, ϵ can be tuned so that strong privacy is provided within 300 meters from each location but in a rural environment, they are not perceived as sufficient privacy. And even inside a city, such a protection is not always adequate: within a big hospital an accuracy of 300 meters might be enough to infer that a user is visiting the hospital.

In this section we address this issue using a custom distinguishability metric that is adapted to the properties of each area, an *elastic metric*. More specifically we discuss properties that such a metric should satisfy, and in the next section we present an algorithm for efficiently computing such a metric.

Once obtained the elastic metric, we can plug it in the $d_{\mathcal{X}}$-privacy definition and obtain an *elastic privacy definition* for location privacy, much like was done for geo-indistinguishability. Furthermore we can use the Exponential mechanism presented in Section 2 to obtain a *sanitization mechanism* that satisfies our elastic privacy definition.

3.1 Privacy mass

The main idea to overcome the rigidity of geo-indistinguishability is to construct a distinguishability metric $d_{\mathcal{X}}$ that adapts depending on the properties of each area. However in order to distinguish a city from its countryside, or on a finer scale, a crowded market place from a hospital, we first need to assign to each location how much it contributes to the privacy of the user. In other words we consider *privacy as a resource* scattered on the geographical space and each locations is characterized by a certain amount of privacy.

More precisely the privacy of *location* x depends on the *points that are indistinguishable from* x. Let $\mathrm{cover}(x)$ denote the set of points that are "highly" indistinguishable from x. For the moment we keep $\mathrm{cover}(x)$ informal, it is properly defined in the next section. Intuitively, the privacy of x depends:

1. on the *number* of points in $\mathrm{cover}(x)$: an empty set clearly means that x can be inferred, while a set $\mathrm{cover}(x)$ containing a whole city provides high privacy. This corresponds to the idea that hiding within a large area provides privacy.
2. on the semantic *quality* of points in $\mathrm{cover}(x)$: it is preferable for $\mathrm{cover}(x)$ to contain a variety of POIs and highly populated locations, than points in a desert or points all belonging to a hospital. This corresponds to the idea that hiding within a populated area with a variety of POIs provides privacy.

To capture this intuition in a flexible way we introduce the concept of *privacy mass*. The privacy mass of a location x, denoted by $m(x)$, is a number between 0 and 1, capturing the location's value at providing privacy. We also denote by $m(A) = \sum_{x \in A} m(x)$ the total mass of a set A. The function $m(\cdot)$ should be defined in a way such that a set of points containing a *unit of mass* provides sufficient cover for the user.

Hence, the metric we construct needs to satisfy that

$$m(\text{cover}(x)) \geq 1 \qquad \forall x \in \mathcal{X}$$

Following the idea that privacy comes by hiding within either a "large" or "rich" area, we define $m(x)$ as

$$m(x) = a + q(x)b \tag{1}$$

where a is a quantity assigned to each location simply for "occupying space", $q(x)$ is the "quality" of x and b is a normalization factor. The quality $q(x)$ can be measured in many ways; we measure it by querying the OpenStreetMap database for a variety of POIs around x, as explained in Section 3.3. Assuming $q(x)$ to be given, we can compute a and b as follows: we start with the intuition that even in empty space, a user feels private if he is indistinguishable within some large radius r_{large}, for instance 3000 m (r_{large} can be provided by the user himself). Let $B_r(x) = \{x' \mid d_{\text{euc}}(x, x') \leq r\}$ denote the Euclidean ball of radius r centered at x. Letting x be a location in empty space, i.e. with $q(x) = 0$, intuitively we want that $\text{cover}(x) = B_{r_{\text{large}}}(x)$, and $m(\text{cover}(x)) = 1$, hence

$$a = \frac{1}{|B_{r_{\text{large}}}(x)|}$$

Similarly, in an "average" location in a more private place, like a city, a user feels private if he is indistinguishable within some smaller radius r_{small}, for instance 300 m (r_{small} can be also provided by the user himself). Let

$$avg_q = E_x q(B_{r_{\text{small}}}(x))$$

be the average quality of a r_{small} ball (where expectation is taken over all location in the city). On average we establish that such a ball contains one unit of privacy mass, thus we get:

$$1 = a \cdot |B_{r_{\text{small}}}(x)| + b \cdot avg_q \qquad \text{hence}$$
$$b = \frac{1}{avg_q}\left(1 - \frac{|B_{r_{\text{small}}}(x)|}{|B_{r_{\text{large}}}(x)|}\right)$$

Note that the intuitive requirement of being indistinguishable from a set of entities with some semantic characteristics is widely used in the privacy literature. Most notably, k-anonymity [4, 12, 19, 25] requires to be indistinguishable from group of at least k individuals, while l-diversity [22, 32] adds semantic diversity requirements: hiding among k hospitals is not acceptable since it still reveals that we are in a hospital. It should be emphasized, however, that although we follow this general intuition, we do so inside the geo-indistinguishability framework, leading to a privacy definition and an obfuscation mechanism vastly different from those of the aforementioned works, as explained in more details in Section 6.0.0.6.

3.2 Requirement

Having fixed the function $m(x)$, we turn our attention to the requirement that our distinguishability metric $d_{\mathcal{X}}$ should satisfy in order to provide adequate privacy for all locations.

Let $B_l(x)$ denote the $d_{\mathcal{X}}$-ball of distinguishability level l. The $d_{\mathcal{X}}$-privacy property ensures that, the smaller l is, the harder it will be to distinguish x from any point in $B_l(x)$. Our requirement is that $B_l(x)$ should collect an appropriate amount of privacy mass:

$$m(B_l(x)) \geq \text{req}(l) \qquad \forall l \geq 0, x \in \mathcal{X} \tag{2}$$

where $\text{req}(l)$ is a function expressing the required privacy mass at each level. The algorithm of Section 3.4 ensures that the above property is satisfied by $d_{\mathcal{X}}$.

It remains to define the $\text{req}(l)$ function. Let l^* denote a "small" distinguishability level (see Section 2 for a discussion on distinguishability levels and what small means. In this paper we use $l^* = \ln 2$). Points in $B_{l^*}(x)$ will be "highly" indistinguishable from x, hence $B_{l^*}(x)$ plays the role of $\text{cover}(x)$ used informally in the previous Section. Privacy mass was defined so that $m(\text{cover}(x)) \geq 1$, hence we want $\text{req}(l^*) = 1$.

Moreover, as the $d_{\mathcal{X}}$-distance l from x increases, we should collect even more mass, with the amount increasing quadratically with l (since the number of points increases quadratically). Hence we define $\text{req}(l)$ as a quadratic function with $\text{req}(0) = 0$ and $\text{req}(l^*) = 1$, that is:

$$\text{req}(l) = \left(\frac{l}{l^*}\right)^2$$

Defining the requirement in terms of privacy mass is a flexible way to adapt it to the properties we are interested in. Indeed we can re-obtain geo-indistinguishability as a special case of our new framework if all locations are considered just for their contribute in space, not quality, i.e. $q(x) = 0$. The requirement is then to be indistinguishable in certain area and in an Euclidean metric it is simply the area of the circle with radius l, a function that is indeed quadratic in l.

3.3 Extracting location quality

Our definition of privacy mass depends on the semantic quality $q(x)$ of a point x. To compute the quality in a meaningful way, we used the OpenStreetMap database [2] to perform geo-localized queries. The open license ODbL of the database allows to download regional extracts that can then be loaded in a GIS database (Postgresql+PostGIS in our case) and queried

2 http://www.openstreetmap.org

for a variety of geo-located features. The data in many urban areas is extremely fine grained, to the level of buildings and trees. Furthermore there is a great variety of mapped objects produced by more that 2 millions users.

In order to extract the quality of a cell $q(x)$, we perform several queries reflecting different privacy properties and we combine them in one aggregate number using different weights. In our experiments we query for a variety of Points Of Interest in the tag class `amenity`, such as restaurants and shops, and for the number of buildings in a cell. The buildings are an indication of the population density, in fact despite the database provides a `population` tag, it is for large census areas and with a scarce global coverage, while the `building` tag can be found everywhere and with fine resolution. Considering the simple nature of the queries performed we believe the resulting grid captures very well the concept of hiding in the crowd that we wanted to achieve (a sample can be viewed in Figure 2a). We leave more complex query schemes as future work as the main focus of this paper is on the metric costruction, described in detail in Section 3.4.

Among possible improvements three directions seem promising.

First, one strength of $d_{\mathcal{X}}$-privacy is that it is independent of prior knowledge that an attacker might have about the user, making the definition suitable for a variety of users. However in some cases we might want to tailor our mechanism to a specific group of users, to increase the performance in terms of both privacy and utility. In this case, given a prior probability distribution over the grid of locations, we can use it to influence the privacy mass of each cell. For instance, if we know that our users never cross some locations or certain kind of POIs, we can reduce their privacy mass.

Second, we are interested in queries that reward variety other that richness e.g. a location with 50 restaurants should be considered less private than one with 25 restaurant and 25 shops. Similar ideas have been explored in l-diversity, for a detailed discussion of the differences in our approaches refer to Section 6.0.0.6.

Finally, different grids could be computed for certain periods of the day or of the year. For instance, our user could use the map described above during the day, feeling private in a road with shops, but in the evening only a subset of the tags should be used as many activities are closed, making a road with many restaurants a much better choice.

Once we have enriched every location x with a quality $q(x)$, we can compute the resource function $m(x)$ as described previously. In the next part we describe how to exploit this rich and customizable information to automatically build an elastic metric satisfying a requirement req.

3.4 An efficient algorithm to build elastic metrics

In this section we develop an efficient algorithm to compute a distinguishability metric d_x that satisfies the quadratic requirement defined before. The metric we produce is induced by an undirected graph $\mathcal{G} = (\mathcal{X}, E)$, that is the main structure manipulated by the algorithm, where vertices are locations and edges (x, d, x') are labeled with the distance between locations. The distance between two locations is the shortest path between them and thanks to this property instead of computing $|\mathcal{X}|^2$ edges, we can actually keep just a subset and derive all other distances as shortest paths.

We start with a fully disconnected graph where all distances are infinite (thus each location is completely distinguishable) and we start adding edges guided by the requirement function. We work in *iterations* over the grid, where at each iteration we add only one edge per vertex, stopping when req is satisfied for all vertices. The reason to work in iterations is that even if at iteration i a vertex can only reach a certain number of cells, because of the other edges added during the same iteration, at $i + 1$ it will find itself connected to many more vertices. This approach distributes edges uniformly which provides two main advantages. First, it increases the locality of connections which in turn reduces the average error (or increases the utility) of the resulting mechanism. Second, it leads to a smaller number of edges, thus decreasing the size of the graph.

The requirement function $\mathrm{req}(l)$ is used as a guideline to define the edges. Let $\mathrm{req}^{-1}(m) = l^*\sqrt{m}$ be the inverse of req.[3] This function tells us at what distinguishability level $l = \mathrm{req}^{-1}(m)$ we should find m amount of privacy mass in order to satisfy the requirement. For each location x we keep a temporary level l_x that is updated at each iteration using req^{-1} and stops at a predefined maximum value $l^\top$. At the beginning l_x is set using only the privacy mass provided by x alone but adding edges will take into account also the ball of points reachable within l_x. In other words the temporary level of each location indicates up to what level of distinguishability the requirement is satisfied.

We then start the iterations and for each vertex x that hasn't already reached $l^\top$ we recompute an updated l_x. The update is necessary to take into account other connections that may have been added for other vertices and that could increase the ball of x. In order to add a new edge we need a strategy

3 Note that our algorithm is not tied to the specific quadratic requirement function, it can work with an arbitrary function req. Even if $\mathrm{req}(l)$ is not invertible (e.g. for a step-like requirement), we could use $\mathrm{req}^*(m) = \inf\{l \mid \mathrm{req}(l) \geq m\}$ in place of req^{-1}.

to find a candidate vertex x' to connect to. The strategy we employ is `next-by-geodistance`, that returns the cell x' geographically closest to x, but still not visited. In the resulting metric locations are more indistinguishable to nearby locations, reducing the average error of reported points. Once we have a candidate location x' there are two possible situations. If the distance between x and x' is greater than l_x, we need to lower it to satisfy the requirement, so we add an edge (x, l_x, x'). Otherwise if the distance is shorter or equal than l_x, this means that x' is already in the l_x ball of x, so we ask `next-by-geodistance` for another candidate. For each vertex not completed an edge is added to the graph and the process is repeated in iterations until all locations reach $l^\top$.

Our experiments showed that completion of the last few tens of vertices can take extremely long and they are localized mostly on the border of the grid. This is due to the fact that points close to the border have fewer neighbors, making it harder for them to find a candidate to connect to. As a consequence they need to reach much further away, taking more iterations and resulting in a higher average error because of the long connections created. For this reason we use a stopping condition that checks, at the end of every iteration, if all the nodes remaining to complete are closer to the border than a certain *frame* constant. If they are, the algorithm stops without completing their requirement. All locations inside this frame of the grid can be reported as sanitized locations but cannot be used as secret locations. The *frame* value is a compromise between the algorithm running time and usable grid size, in our experiments we used 3% of the grid size.

It can be shown that the metric $d_\mathcal{X}$ constructed by this algorithm does satisfy the requirement req for all $l \leq l^\top$. The stopping level $l^\top$ can be set arbitrarily high, but in practice setting it to any value larger than 10 has no effect on the resulting metric. As shown in the evaluation of Section 5, the algorithm can scale to an area of half a million locations with modest computing resources. This is several orders of magnitude better that techniques computing optimal obfuscation mechanisms [5, 27, 29] which can only handle a few hundred points within reasonable time constraints. Of course, our method gives no optimality guarantees, it only constructs one possible metric among those satisfying the requirement.

We believe further improvements in performance are possible in three directions, that we leave as future work. When working in privacy poor areas, like in the country, the algorithm spends a considerable time compressing large areas, as expected. We believe that this work could be avoided by grouping together several locations already when laying down the grid. We would have a coarser resolution in the country, which is acceptable, and a large speed up in the metric construction. A second obvious improvement would come from running the algorithm in parallel on portions of the map and merging the results. The problem arises on the borders of the submaps, where on adjacent locations we have connections with sharply different shapes. We believe however that computing several submaps leaving a frame of uncompleted points and them running again the algorithm in the entire map could provide a reasonable result. Finally several strategies can be applied in the choice of the next candidate and in the way we perform the iterations that could have an impact on speed and utility of the mechanism by completing faster the requirement.

```
foreach x ∈ 𝒳 do
  l_x := req^{-1}(m({x}))
while ∃x. l_x ≠ l^⊤ do
  foreach x ∈ 𝒳 do
    l := req^{-1}(m(B_{l_x}(x)))
    do
      x' := next-by-geodistance(x)
    while d_𝒳(x, x') ≤ l_x
    E := E ∪ {(x, l_x, x')}
```

3.5 Practical considerations

We believe that the techniques presented are practical enough to deploy a location privacy mechanism in a real setting. The resources required both in terms of hardware and time are very limited, consider that the mechanism evaluated in the next section was built in a day on a medium Amazon E2C instance. As already mentioned querying the database is easily parallelizable and the same grid can be reused to build several metrics. We could imagine having a choice of pre-computed grids, for different flavors of location quality and times (as explained in Sec 3.3), on top of which the user could tune the requirement.

The computation of the metric is the most demanding part of the process but proved reasonably fast for an area that can easily contain all the movements of the user and many optimization are still possible in the algorithm. We imagine these two computationally intensive activities to be performed by a remote server, with seldom updates from the OpenStreetMap database.

The user's phone needs to take care only of downloading an extract of the metric to use in the Exponential mechanism. For every request, the mechanism computes the exponential distribution of sanitized locations and draws from it, which amounts to a trivial computation, both in memory and time. In principle we could also avoid contacting a third party by saving the entire metric locally on the phone, as a reference our metric for Ilê de France is only 58MB.

From a user's perspective, the amount of configuration required to run the mechanism varies according to her needs. It can go from no configuration, in the case she downloads a precomputed map, to scaling the privacy mass requirement, to full customization in the case the users wants to tailor the queries to her specific needs.

4 Incorporating fences in the metric

As discussed in the introduction another issue of geo-indistinguishability is that, repetitive use of a mechanism from the same location is bound to reveal that location as the number of reports increases. This is crucial for locations that the user frequently visits, such as his home or work location. Such locations cannot be expected to remain indistinguishable in the long run; repetitive use of the mechanism is bound to reveal them with arbitrary accuracy.

Despite the fact that all privacy mechanism are susceptible to this privacy erosion over time, the compositionality property of ϵ-geo-indistinguishability quantifies exactly this privacy degradation: repetitive use of a mechanism causes a linear accumulation of ϵ, lowering the privacy protection guaranteed. There are techniques to alleviate this effect, such as the predictive mechanism of [8], but they are not enough in this highly recurrent cases.

This problem, especially for the home-work locations, has been already studied in the literature [16], although in the context of anonymity i.e. how to match a user identity to a pair of home-work locations. In fact our interest is focused on reducing the accuracy of the reported location, because of the sensitive data the attacker can infer from it. Even if the user is authenticated with a LBS, for instance to notify her friends that she is home, it is still valuable to not disclose the precise address (that friends know anyway).

4.1 Fences

For highly recurrent locations we propose the use of geo *fences*, areas on the map where the user's movements are completely hidden and that are considered known to the attacker. This technique is not novel and indeed has been widely used by a large number of LBS. For example, personal rental services (e.g. Airbnb) allow the user to indicate an area where the good to be rented is located, so that other users can evaluate if it is at a convenient distance without compromising the privacy of the owner. Despite the vast use of fences in practice, to the best of our knowledge, there is a lack of works in the literature about their implementations or evaluating their effectiveness.

Our contribution consist in a simple formalization of fences in the framework of distinguishability metrics. This construction allows to hide completely sensitive locations within a fence, while permitting the use of any other $d_{\mathcal{X}}$-private mechanism outside. Given a privacy metric $d_{\mathcal{X}}$, we define a new fenced metric d_F as:

$$d_F(x, x') = \begin{cases} 0 & x, x' \in F \\ d_{\mathcal{X}}(x, x') & x, x' \notin F \\ \infty & \text{otherwise} \end{cases}$$

Outside the fence the original metric $d_{\mathcal{X}}$ is in place while inside the fence all points are completely indistinguishable. The advantage is that being zero the distance *inside* the fence, any repeated use of the mechanism from the sensitive location comes for *free*, effectively stopping the linear growth of the budget. In practice we keep reporting uniformly points inside the fence, thus leaking no information to the attacker, other than the public fact that we are inside the fence. Indeed it should be noted that any movement *across* the fence is completely distinguishable, the attacker knows when we are in or out.

Regarding utility, in this case it simply depends on the size of the fence, in direct contrast with privacy.

Automatic Configuration of position and size

In order to configure the position and size of the fences, the user input would be the best option (as shown in [6]), however they could also be inferred and suggested automatically. In [15] the authors developed an attack to identify POI of a specific user, from a set of mobility traces. A similar technique could be employed on the user's phone, over a training period, to collect and analyze her movements for a few days. The mechanism would then automatically detect recurrent locations and suggest the user to fence them, possibly detecting more than just home/work locations.

With the use of geolocated queries, such as those used to extract privacy points in Section 3.3, we could determine the size of the fence so to include a reasonable amount of buildings for home and other POIs for work.

Compatibility with the elastic metric algorithm

It should be noted that fences are applicable to any distinguishability metric, including but not limited to the elastic metric presented in Sec 3. Not only we can incorporate fences in our elastic definition but also in our graph based algorithm, in a simple and efficient way. It amounts to connecting the locations inside the fence with zero labeled edges and to

leave them disconnected from the nodes outside. When running the algorithm we should also take care to maintain this disconnection of the fence. In order to avoid adding edges from the inside we simply set the temporary radius r_x of all nodes inside the fence to $d^\top$, so that the algorithm considers them completed and skips them. On the other side, to avoid adding edges from the outside, we need to modify the function `next-by-geodistance` so to avoid considering a candidate any location inside a fence. Both alterations to the algorithm are trivial to implement and have no effect on performances. The only drawback of the presented method is that the fences need to be set before building the metric, which is inconvenient as for each user we are obliged to recompute, for the most part, the same metric. Despite this our experiments show that in under a day is possible to generate the metric so this remains an effective technique for practical purposes, especially considering the very static nature of the fences.

As future work we are investigating the possibility to carve the fences from an already built metric. The challenge in this case is re-enforcing the requirement for all affected points and it is not clear for now how spread out is the impact on other nodes and what is the effect on utility.

5 Evaluation

In this section we perform an extensive evaluation of our technique in Paris' wide metropolitan area, using two real-world datasets. We start with a description of the metric-construction procedure, and we discuss the features of the resulting metric as well as the obfuscation mechanism obtained from it. Then, we compare the elastic mechanism to the Planar Laplace mechanism satisfying geo-indistinguishability, using data from the Gowalla and Brightkite social networks. The comparison is done using the privacy and utility metrics of [28]. It should be emphasized that the metric construction was completely independent from the two datasets, which were used only for the evaluation. All the code used to run the evaluation is publicly available [1].

5.1 Metric construction for Paris' metropolitan area

We build an elastic metric $d_{\mathcal{X}}$ for a 75 km $\times$ 75 km grid centered in Paris, roughly covering its extended metropolitan area. Each cell is of size 100 m $\times$ 100 m, and the set of locations $\mathcal{X}$ contains the center of each cell, giving a total number of $562,500$ locations. The area covered is shown in Fig 1; note that the constructed metric covers the larger shown area, the two smaller ones are only used for the evaluation of the mechanism in the next section.

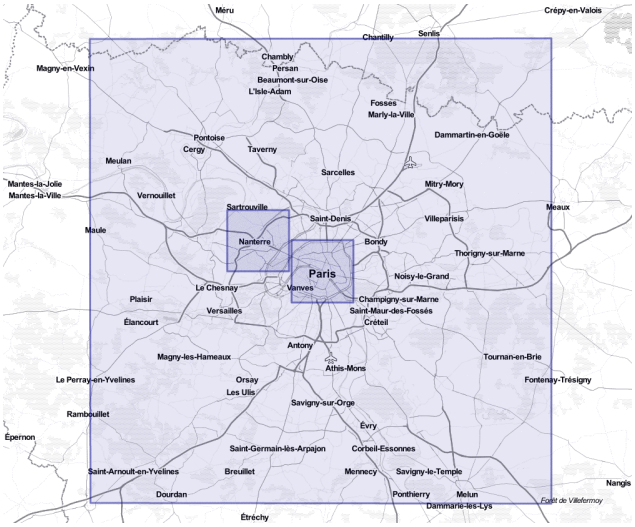
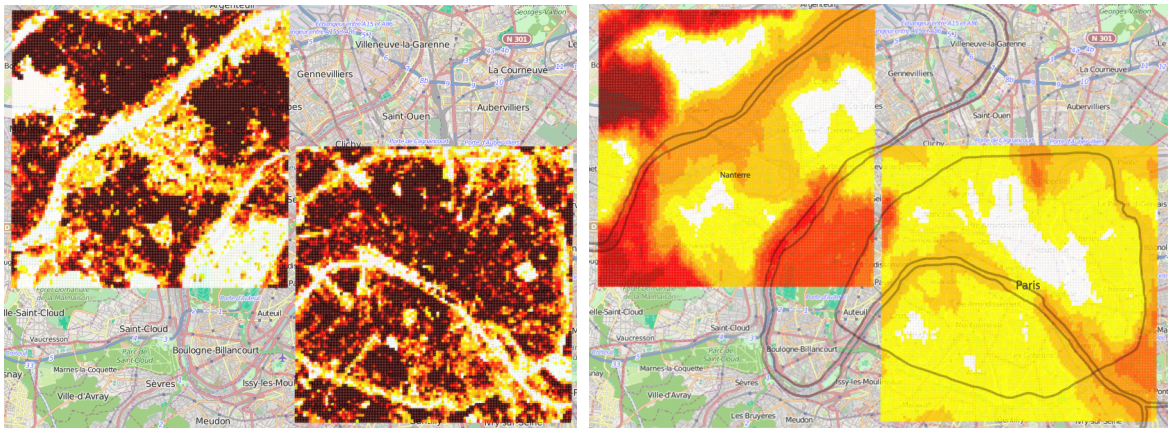


Fig. 1. Coverage of EM with two subregions: Nanterre suburb on the left and Paris city on the right

The semantic quality $q(x)$ of each location was extracted from OpenStreetMap as explained in Section 3.3, and the privacy mass $m(x)$ was computed from (1) using $r_{\text{small}} = 300$ m and $r_{\text{large}} = 3$ km. The resulting mass of each location is shown in Figure 2a, where white color indicates a small mass while yellow, red and black indicate increasingly greater mass. The figure is just a small extract of the whole grid depicting the two smaller areas used in the evaluation: central Paris and the nearby suburb of Nanterre. Note that the colors alone depict a fairly clear picture of the city: in white we can see the river traversing horizontally, the main ring-road and several spots mark parks and gardens. In yellow colors we find low density areas as well as roads and railways while red colors are present in residential areas. Finally dark colors indicate densely populated areas with presence of POIs.

For this grid, we use the algorithm presented in Section 3.4 to compute an elastic metric $d_{\mathcal{X}}$ with the quadratic requirement of (2), configured with $l^* = \ln 2$. The whole computation took less than a day on an entry-level Amazon EC2 instance. This performance of the algorithm is already sufficient for real-world use: the metric only need to be computed once, while the computation can be done by a server and the result can be then transmitted to the user's device. Note that the algorithm can deal with sizes several orders of magnitude bigger that techniques computing optimal obfuscation mechanisms [5, 27, 29], which makes it applicable to more realistic scenarios.

We then construct an Exponential mechanism (described in Section 2) using $d_{\mathcal{X}}$ as the underlying metric. We refer to the resulting obfuscation mechanism as the Elastic Mecha-



(a) Privacy mass $m(x)$ of each location

(b) Expected error $E_{\mathrm{EM}}(x)$ at each location

Fig. 2. Paris' center (right) and the nearby suburb of Nanterre (left)

nism (EM). The mechanism is highly adaptive to the properties of each location: high-density areas require less noise than low-density ones to achieve the same privacy requirement. Figure 2b shows our utility metric per location, computed as the expected distance $E_{\mathrm{EM}}(x)$ between the real and the reported location. Compared to Figure 2a it is clear that areas with higher privacy mass result to less noise. Populated areas present a good and uniform error that starts to increase on the river and ring-road. On the other hand, the large low-density areas, especially in the Nanterre suburb, have a higher error because they need to report over larger areas to reach the needed amount of privacy mass.

Finally, Figure 3 shows a boxplot of the expected error for each location in the two areas. It is clear that the amount of noise varies considerably, ranging from a few hundred meters to several kilometers. It is also clear that locations in central Paris need considerably less noise that those in the suburban area. For comparison, the Planar Laplace mechanism (compared against EM in the next section) has a constant expected error for all locations.

Note that the expected error will always be higher than the r_{small} used in the normalization. For example in a location that satisfies its requirement in 300 m it would be 870 m. This is expected and it is due to the nature of the exponential noise added.

5.2 Evaluation using the Gowalla and Brightkite datasets

In this section we compare the Elastic Mechanism (EM) constructed in the previous section with the Planar Laplace mechanism [2] satisfying standard geo-indistinguishability. For the evaluation we use two real-world datasets from location-based social networks.

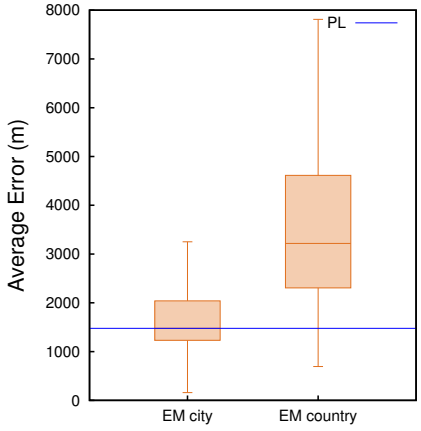


Fig. 3. Expected error $E_{\mathrm{EM}}(x)$ per location

5.2.0.2 The Gowalla and Brightkite datasets

Gowalla was a location-based social network launched in 2007 and closed in 2012, after being acquired by Facebook. Users were able to "check-in" at locations in their vicinity, and their friends in the network could see their check-ins. The Gowalla dataset [10] contains $6,442,890$ public check-ins from $196,591$ users in the period from February 2009 to October 2010. Of those, $9,635$ check-ins were made in Paris' center area and 429 in the Nanterre area (displayed in Fig 1).

Brightkite was another location-based social network created in 2007 and discontinued in 2012. Similarly to Gowalla users could check-in in nearby locations and query who is nearby and who has been in that location before. The Brightkite dataset [10] contains $4,491,143$ check-ins from

$58,228$ users. Of those, $4,014$ check-ins were made in Paris' center and 386 in Nanterre.

These datasets are particularly appealing for our evaluation since a check-in denotes a location of particular interest to the user, and in which the user decided to interact with an actual LBS. This is in sharp contrast to datasets containing simply mobility traces, which just contain user movements without any information about the actual use of an LBS.

5.2.0.3 Privacy metrics

Since the EM and PL mechanisms satisfy different privacy definition, to perform a fair comparison we employ the widely used Bayesian privacy metric of [28]. This metric considers a Bayesian adversary having a prior knowledge π of the possible location of the user and observing the output of the mechanism K. After seeing a reported location z the attacker applies a strategy $h : \mathcal{Z} \to \mathcal{X}$ to remap z to the real secret location where he believes the user could be, e.g. if z is in a river, it is likely that the user is actually in a nearby location x on the banks.

The AdvError metric measures the expected loss of an attacker trying to infer the user's location. It is defined as:

$$\textsc{AdvError}(K, \pi, h, d_A) = \sum_{x,z} \pi(x)\ K(x)(z)\ d_A(x, h(z))$$

where d_A is a metric modeling the adversary's loss in case he fails to identify the user's real location. Notice how the loss function is not applied directly to the reported location z, but to the remapped location $h(z)$. A rational adversary will use the strategy h^* minimizing his error, hence [28] proposes to use $\textsc{AdvError}(K, \pi, h^*, d_A)$ as a privacy metric, where

$$h^* = \arg\min_h\ \textsc{AdvError}(K, \pi, h, d_A)$$

Note that h^* can be computed efficiently using the techniques described in [29].

In our evaluation the secrets are POIs in each dataset. We use two commonly used loss functions modelling different types of adversaries: the first is the *binary* loss function d_{bin}, modelling an adversary interested in the semantics of the user's POI, hence trying to guess exactly in which one he is located.

$$d_{\text{bin}}(x, z) = \begin{cases} 0 & x = z \\ 1 & x \neq z \end{cases}$$

Then $\textsc{AdvError}(K, \pi, h, d_{\text{bin}})$ expresses the adversary's *probability of error* in guessing the user's POI. Second, we use the Euclidean loss function d_{euc}, modelling an adversary who is interested in guessing a POI *close* to the user's, even if that POI is unrelated to his activity. In this case, $\textsc{AdvError}(K, \pi, h, d_{\text{euc}})$ gives the adversary's *expected error* in meters in guessing the user's POI.

We should emphasize an important difference between the two adversaries: d_{bin} tries to extract semantic information from the actual POI, and is less effective in dense areas where the number of POIs is high. On the other hand, d_{euc} is less sensitive to the number of POIs: if many POIs are close to each other, guessing any of them is equally good. This difference is clearly visible in the evaluation results.

We use each dataset to obtain a prior knowledge π^* of an "average" user of each social network, by considering all check-ins within the areas of interest. Note that the datasets do not have enough check-ins *per-user* to construct *individual* user profiles. Indeed, most users have checked-in in each location at most once, hence any profile built by $n-1$ check-ins would be completely inadequate for inferring the remaining one. As a consequence, we assume that the adversary will compute his best strategy h^* by using the global profile π^*.

Finally, we use

$$\textsc{AdvError}(K, \pi_u, h^*, d_A) \qquad d_A \in \{d_{\text{bin}}, d_{\text{euc}}\}$$

as our privacy metric, where π_u is the user's individual prior (computed only from the user's check-ins), and

$$h^* = \arg\min_h\ \textsc{AdvError}(K, \pi^*, h, d_A)$$

is the adversary's strategy computed from the global profile. Hence, the individual priors π_u are only used for averaging and not for constructing the strategy.

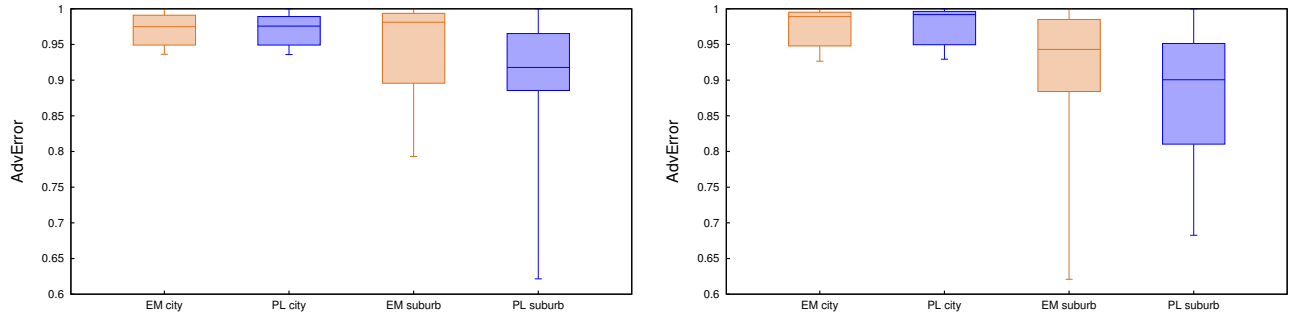
5.2.0.4 Utility metrics

The utility of an obfuscation mechanism is in general closely tied to the application at hand. In our evaluation we want to avoid restricting to a particular one; as a consequence we use two generic utility metrics that are reasonable for a variety of use cases. We measure utility as the *expected distance* between the real and the reported locations, using two distance functions. In some applications, service quality degrades linearly as the reported location moves away from the real one; in such cases the Euclidean distance d_{euc} provides a reasonable way of measuring utility. Other applications tolerate a noise up to a certain threshold r with almost no effect on the service, but the quality drops sharply after this threshold. In such cases, we can measure utility using the follwoing distance metric:

$$d_r(x, z) = \begin{cases} 0 & d_{\text{euc}}(x, z) < r \\ 1 & ow. \end{cases}$$

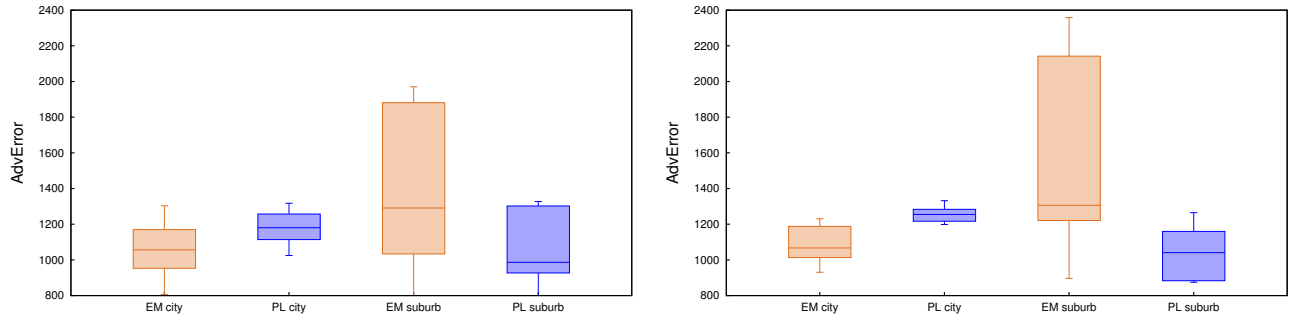
5.2.0.5 Results

We carry out our evaluation in two different areas of the Paris metropolitan area, with very different privacy profiles. The



(a) Gowalla **(b)** Brightkite

Fig. 4. Per-user binary AdvError of the EM and PL mechanisms for each area



(a) Gowalla **(b)** Brightkite

Fig. 5. Per-user Euclidean AdvError of the EM and PL mechanisms for each area

first area is the city of Paris intra-muros, very private on average, while the second is the adjacent suburb of Nanterre, where already the concentration of privacy mass is much lower.

To obtain a fair comparison, the Planar Laplace PL mechanism is configured to have the same utility as the EM mechanism in Paris' center. We computed the utility of EM using the global profile π^* from both datasets, using four distance functions, namely d_{euc} and d_r with $r = 1200, 1500$ and 1800 meters. In each case, we computed the parameter ϵ of PL that gives the same utility; we found that in all 8 cases the ϵ we obtained was almost the same, ranging from 0.001319 to 0.001379. Since the difference between the values is small, we used a single configuration of PL with the average of these values, namely $\epsilon = 0.001353$. With this configuration, we then compare the two mechanisms' privacy in both areas.

Note that, although we configured the *expected* utility of both mechanisms to be the same in the city, the EM's behaviour is highly dynamic: in the most private areas of Paris EM uses much less noise: in 10% of the city's locations PL adds 50% or more noise than EM.

The results for the binary adversarial error are shown in Figure 4. We can see that in the center of Paris both mechanisms provide similar privacy guarantees. The dynamic nature of PL does not affect its privacy: locations in which the noise is lower have more POIs in proximity, hence even with less noise it is still hard to guess the actual one. On the other hand, in the suburb there is a sharp degradation for PL. The reason is that the number of POIs in both datasets is much smaller in Nanterre than in Paris, the distance between them is greater, and the resulting priors are more "informed". Hence it is considerably easier for the adversary to distinguish such POIs, leading to a probability of error as low as 0.62. On the other hand, EM maintains a higher AdvError in the suburb, by introducing a higher amount of noise.

To match EM's privacy guarantees in the suburb, PL should be configured with a higher amount of noise. However, since Planar Laplace treats space in the same way everywhere, this would lead to a high degradation of utility in Paris' center, which is unfortunate since (i) the extra noise is unnecessary to provide good privacy in the center, and (ii) the extra noise could render the mechanism useless in a dense urban environment where accurate information is crucial. In short, the flexibility of the elastic mechanism allow it to add more noise when needed, while offering better utility in high-density areas.

The results for the Euclidean adversarial error are shown in Figure 5. Here, we see a sharp difference wrt the binary adversary: the effectiveness of guessing a POI *close* to the real one is not affected much by the number of POIs (guessing any of them is equally good). As a consequence, EM, which adds less noise in dense areas with a great number of POIs, scores a lower adversarial error in the city (although the difference is moderate). The median error for EM in Gowalla is 1056 meters while for PL it is 1180 meters. The motivation behind PL was that, in a dense urban area, it is harder for the adversary to extract semantic information from the reported location even if less noise is used. But the Euclidean adversary is not interested in semantic information, so he scores better with less noise.

In the suburb, on the contrary, the picture is reversed. Due to the fact that priors in Nanterre are more "informed" making remapping easier, the adversarial error for PL decreases compared to the city. On the other hand, EM adapts its noise to the less dense environment, leading to much higher adversarial error.

Note that the Nanterre area is still quite close to the city center and highly populated itself. The fact that we can already see a difference between the two mechanisms so close to the center is remarkable; clearly, the difference will be much more striking as we move away from the city center (unfortunately, the datasets do not contain enough data in these areas for a meaningful quantitative evaluation).

Finally, we should emphasize that the mechanism's construction and evaluation were completely independent: no information about Gowalla's or Brightkite's list of check-ins was used to construct the metric. The only information used to compute d_x was the semantic information extracted from OpenStreetMap.

6 Related Work and Conclusion

6.0.0.6 Related work

Concerning location privacy, there are excellent works and surveys [20, 26, 31] that present the threats, methods, and guarantees.

A large body of works developed from k-anonymity, originally a notion of privacy for databases [4, 12, 19, 22, 24, 25, 32]. Many of the shortcomings of k-anonymity, outlined in [30], are addressed in the current main trend based on the expectation of distance error [11, 18, 28, 29]. Both are dependents on the adversary's side information, contrary to our approach, as are some other works [9] and [3].

Extraction of privacy points uses ideas similar to k-anonymity or l-diversity but the mechanisms are very different in nature. The privacy of our resulting mechanism is due to the obfuscation technique, it is independent of the attacker side knowledge and doesn't require any third party to operate. Furthermore we protect privacy as the location accuracy and not its anonymity.

Notions that abstract from the attacker's knowledge based on differential privacy can be found in [21] and [17] although only for *aggregate* information.

In this work we extend and generalize geo-indistinguishability [2] and in order to so we go back to the notion it is based on, d_x-privacy [7] that provides a metric extension of differential privacy. This family of definitions abstracts from the attacker's prior knowledge, and is therefore suitable for scenarios where the prior is unknown, or the same mechanism must be used for multiple users.

Regarding the construction of finite mechanisms, [27] proposes a linear programming technique to construct an optimal obfuscation mechanism with respect to either the expectation of distance error or geo-indistinguishability. In [5] the authors propose again a linear programming technique to compute a geo-indistinguishable mechanism with optimal utility. Their approach uses a spanner graph to approximate the metric in a controlled way. Our algorithm does not provide optimality with respect to privacy nor utility, it guarantees the respect of a privacy requirement while achieving good utility. Moreover the state of the art in optimal mechanism construction is limited to few tens of locations while the purpose of our technique is to scale to several thousands of points.

6.0.0.7 Conclusion

In this paper, we have developed a novel elastic privacy metric that allows to adapt the privacy requirement, as hence the amount of applied noise, to the properties of each location. We have formalized a requirement for such metrics based on the concept of privacy mass, and using semantic information extracted from OpenStreetMap. We have developed a graph-based algorithm to efficiently construct a metric satisfying this requirement for large geographical areas. We have discussed how the geo fencing technique can be elegantly expressed in the metric and incorporated in its construction. Finally, we have performed an extensive evaluation of our technique in Paris' wide metropolitan area, using two real-world datasets from the Gowalla and Brightkite social networks. The results show that the adaptive behavior of the elastic mechanism can offer better privacy in low-density areas by adjusting the amount of applied noise.

References

[1] https://github.com/paracetamolo/elastic-mechanism.

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

[3] 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*, volume 4602 of *LNCS*, pages 47–60. Springer, 2007.

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

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

[6] A. J. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Proc. of UbiComp 2010*. ACM, 2010.

[7] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of Differential Privacy using metrics. In *Proc. of PETS*, volume 7981 of *LNCS*, pages 82–102. Springer, 2013.

[8] K. Chatzikokolakis, C. Palamidessi, and M. Stronati. A predictive differentially-private mechanism for mobility traces. In *Proc. of PETS*, volume 8555 of *LNCS*, pages 21–41. Springer, 2014.

[9] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. of PET*, volume 4258 of *LNCS*, pages 393–412. Springer, 2006.

[10] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. ACM, 2011.

[11] R. Dewri. Local differential perturbations: Location privacy under approximate knowledge attackers. *IEEE Trans. on Mobile Computing*, 99(PrePrints):1, 2012.

[12] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of PERVASIVE*, volume 3468 of *LNCS*, pages 152–170. Springer, 2005.

[13] C. Dwork. Differential privacy. In *Proc. of ICALP*, volume 4052 of *LNCS*, pages 1–12. Springer, 2006.

[14] K. Fawaz and K. G. Shin. Location privacy protection for smartphone users. In *Proc. of CCS*, pages 239–250. ACM Press, 2014.

[15] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. on Data Privacy*, 4(2):103–126, 2011.

[16] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Proc. of PerCom*. IEEE, 2009.

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

[18] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of SecureComm*, pages 194–205. IEEE, 2005.

[19] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services. In *Proc. of ICDE Workshops*, page 1248, 2005.

[20] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

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

[22] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Trans. on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[23] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proc. of FOCS*, pages 94–103. IEEE, 2007.

[24] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng*, 13(6):1010–1027, 2001.

[25] P. Shankar, V. Ganapathy, and L. Iftode. Privately querying location-based services with SybilQuery. In *Proc. of UbiComp*, pages 31–40. ACM, 2009.

[26] K. G. Shin, X. Ju, Z. Chen, and X. Hu. Privacy protection for users of location-based services. *IEEE Wireless Commun*, 19(2):30–39, 2012.

[27] R. Shokri. Optimal user-centric data obfuscation. Technical report, ETH Zurich, 2014. http://arxiv.org/abs/1402.3426.

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

[29] 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*, pages 617–627. ACM, 2012.

[30] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proc. of WPES 2010*, pages 115–118 115–118 115–118, 2010.

[31] M. Terrovitis. Privacy preservation in the dissemination of location data. *SIGKDD Explorations*, 13(1):6–18, 2011.

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