Wisam Eltarjaman, Rinku Dewri*, and Ramakrishna Thurimella

Location Privacy for Rank-based Geo-Query Systems

Abstract: The mobile eco-system is driven by an increasing number of location-aware applications. Consequently, a number of location privacy models have been proposed to prevent the unwanted inference of sensitive information from location traces. A primary focus in these models is to ensure that a privacy mechanism can indeed retrieve results that are geographically the closest. However, geo-query results are, in most cases, ranked using a combination of distance and importance data, thereby producing a result landscape that is periodically flat and not always dictated by distance. A privacy model that does not exploit this structure of geo-query results may enforce weaker levels of location privacy. Towards this end, we explore a formal location privacy principle designed to capture arbitrary similarity between locations, be it distance, or the number of objects common in their result sets. We propose a composite privacy mechanism that performs probabilistic cloaking and exponentially weighted sampling to provide coarse grain location hiding within a tunable area, and finer privacy guarantees under the principle inside this area. We present extensive empirical evidence to supplement claims on the effectiveness of the approach, along with comparative results to assert the stronger privacy guarantees.

Keywords: location privacy; indistinguishability; rank-based query results

DOI 10.1515/popets-2017-0025
Received 2017-02-28; revised 2017-06-01; accepted 2017-06-02.

1 Introduction

Location-based applications form a vast majority of the application eco-system in modern mobile devices. Many conventional applications such as search, driving directions, social messaging, and others, have been redesigned to be location aware. As such, location information forms an important component of interactions with these applications. Much like how the tracking of users across browser sessions became concerning, the ability to access and collect a user's location information over long periods of time is raising concerns as well. The linkability of a user to potentially private information based on gathered location traces have been well-studied [8, 9, 17]. The mobile application arena is also getting flooded with developers that do not always have the security infrastructure to protect the collected data in storage. Despite the risk, location-based applications are undoubtedly popular.

Early proposals on location privacy models centered around the cloaking of the user's location. However, these models failed to provide a formal guarantee on what an adversary can learn by observing the communication between a location privacy preserving mechanism (LPPM) and the application server. More recent proposals such as geo-indistinguishability [1] address this issue by bounding the degree to which an adversary can distinguish between two locations. A persistent assumption in these proposals (more than a decade worth of research) is that location-based queries (or *geo-queries*) produce results that are dictated only by the distance of a result object from the query location. However, clearly verifiable in any popular location-based application, geo-query results are ranked based on multiple criteria, distance being just one of them. Extending location privacy models to bridge this long present gap is therefore important, and forms the motivation for this work.

We begin with a summary of our contributions in this work. Starting from the principle of geo-indistinguishability, we discuss how the principle enforces a notion of quantifiable indistinguishability based on the distance between locations. We demonstrate how, under the use of a weighted sum ranking function, a geo-indistinguishable LPPM allows for distinctions between locations that are equivalent with respect to the query results. The primary contribution of this work is the introduction of the (f, ϵ)-geo-indistinguishable principle. The

Wisam Eltarjaman: University of Denver, E-mail: wisam@cs.du.edu
***Corresponding Author: Rinku Dewri:** University of Denver, E-mail: rdewri@cs.du.edu
Ramakrishna Thurimella: University of Denver, E-mail: ramki@cs.du.edu

principle is designed with an objective similar to geo-indistinguishability, but allows for privacy levels to be controlled as per an arbitrary function of two locations. For example, if a geo-query returns a unique result set for a region, then enforcing the principle with the appropriate f function would guarantee that no two locations in the region are distinguishable from each other. Consequently, as the second contribution, we propose a LPPM that enforces (f, ϵ)-geo-indistinguishability for applications where top-K ranking (using both importance and distance) is performed. We build this LPPM as a two step application that first discloses coarse grain location information in the form of a cloaking region to obtain result ranking metadata, and then executes a probabilistic retrieval mechanism to obtain details on top ranked results with (f, ϵ)-geo-indistinguishability assurances. We provide analytical results that characterize the privacy and the quality of service assurances of the approach. We provide conclusive evidence to support our claims by applying the approach to a nearby points-of-interest (POI) search in a real-world database. In addition, we develop a prototype Android application to demonstrate how third party APIs can be utilized to execute the various steps in the approach, and assess its efficiency and accuracy on a mobile device.

The remainder of the paper is organized as follows. Section 2 presents the request-response (querying) architecture within which a LPPM operates. We move beyond distance-only ranking, and present top-K ranking as the generic sorting operation. Section 3 presents details of our experimental set up. This is provided early on so that we can refer to observed results along with the proposed concepts. Section 4 starts with a discussion of geo-indistinguishability, and later presents (f, ϵ)-geo-indistinguishability, a LPPM based on it, and its characterizations. Section 5 focuses on the accuracy guarantees of the proposed LPPM. Section 6 presents an extensive parametric evaluation of the LPPM, and Section 7 discusses the Android application based on the LPPM. Section 8 provides a discussion on the comparative effectives of the LPPM. Finally, Section 9 presents a brief overview of prior works, and Section 10 concludes the paper. We present proofs of all theorems in the appendix.

2 Top-K Computation

A typical location privacy preserving mechanism (LPPM) for geo-queries may generate an obfuscated location for a query and then retrieve a set of POIs contained within an area centered at the obfuscated location. The retrieved set is then filtered for the user's actual location and presented to the user. We refer to this architecture as a *1-level architecture.* LPPMs can differ in terms of their privacy guarantees depending on how the obfuscated location is generated. They also differ in terms of their communication overhead depending on the size of the area of retrieval. The local filtering of results can be done strictly on the basis of distance (keep only POIs that are within a certain distance of the user), or on a combination of distance and other prominence factors. Note that the former method has received the most attention in the privacy research community, while the latter method is what is deployed in most non-private local search applications [11, 20, 23, 32]. For example, a search for "cafe" in a popular platform such as Google does not always return the nearest cafes, but the top cafes determined by the query location and other *prominence* factors such as user reviews, reference counts, open hours, and business popularity, among others. Therefore, a geo-query search is more accurately a top-K search.

2.1 Querying Architecture

Any LPPM designed along the lines of the above discussion can provide top-K results by modifying the filtering mechanism. However, the architecture levies a high communication overhead owing to the retrieval of all POIs in a large area. The data pertaining to a complete POI typically contains details such as name, address, contact numbers, ratings, photos, and multiple reviews. Since most POIs will be filtered out, the bandwidth consumed while retrieving all such details for the POIs inside the area of retrieval is wasted. We therefore consider a revised querying architecture that significantly reduces this communication overhead.

In the revised architecture, a LPPM retrieves minimal details about the set of POIs within the area of retrieval. It is sufficient to obtain the location and prominence of a POI in this step. Search providers do provide results with such minimal information, for example, a `radarsearch` query in the Google Places API returns the names and locations of POIs within a specified distance of the given location. Prominence values are not yet included in these results, but as discussed next, they can be communicated without revealing the underlying computation function. A filtering process is next applied on the POIs and a relevant subset is determined. All

details for this subset of POIs are then retrieved from the provider and presented to the user. We refer to this revised architecture as the *2-level architecture.* This architecture allows for the ranking of POIs based on an arbitrary function based on location and prominence, and has significantly lower bandwidth overhead.

2.2 Top-K Ranking

Given a set of POIs $\{p_1, p_2, ..., p_n\}$ inside the area of retrieval, with unique identifiers $\{id_1, id_2, .., id_n\}$, locations $\{l_1, l_2, ..., l_n\}$, and prominence $\{\beta_1, \beta_2, ..., \beta_n\}$;$0 \leq \beta_i \leq 1$, we consider a linear ranking function r' given as

$$r'(l_u, p_i) = \alpha d_{norm}(l_u, l_i) + (1 - \alpha)(1 - \beta_i), \quad (1)$$

where l_u is the user's location, d_{norm} is a normalized distance function, and $\alpha > 0$ is the weight assigned to the distance between the POI and the user. Lower rank values are considered better with this function. Nearest neighbor ranking corresponds to the case of $\alpha = 1$. The exact methodology to combine ranking factors is often not available as public knowledge. However, as long as the distance part is not multiplicatively combined with other factors, the assumed ranking function is generic. Also note that the prominence score can be computed from multiple factors. Therefore, the function implicitly captures the combination of more than two factors.

In the event the service provider is unwilling to reveal the α and β_i values, the ranking function can be rewritten as

$$r(l_u, p_i) = \frac{r'(l_u, p_i)}{\alpha} = d_{norm}(l_u, l_i) + \gamma_i \quad (2)$$

where $\gamma_i = \frac{1-\alpha}{\alpha}(1 - \beta_i)$. The ranking of POIs is not affected when using r instead of r'. Therefore, the data retrieved for all POIs inside the area of retrieval can be represented as the set $\Omega = \{\langle id_i, l_i, \gamma_i \rangle | i = 1...n\}$. Locations are typically provided as latitude/longitude pairs; we convert them to cell coordinates as per our experimental setup. Consequently, "locations" imply cell coordinates in the following.

Given Ω and some user location l_u, the top-K result set for l_u can be computed using a brute force search. However, if top-K computations have to be performed repeatedly, more specifically for multiple nearby locations, then a brute force search can result in sluggish performance. Such computations are crucial in our approach; therefore, we need an effective method to compute multiple top-K sets corresponding to a sub-grid of cells. Dewri et al. presented an algorithm for this task where the computation time is significantly reduced by (i) using a kd-tree branch and bound search instead of a linear search, and (ii) using an optimization where top-K search for neighboring cells can be skipped [6]. We employ this algorithm for our needs.

3 Experimental Setup

We begin with our experimental setup and present results alongside the proposed concepts and claims. For most part, the empirical evaluation is performed using a $32 \times 32 km^2$ area centered at downtown Los Angeles, California, USA (34.0522^o N, 118.2428^o W). This area is divided into a grid of cells measuring $100m \times 100m$. While the approach can be applied to any geo-query based application, we focus on the domain of finding nearby points-of-interest. Using the business listings provided in the SimpleGeo database, we are able to query for multiple keywords within this area. Parametric evaluation is shown on three POI keywords, namely *bookstore, gas station,* and *cafe.* There are 155 bookstores, 347 gas stations and 608 cafes inside the evaluation area, thereby giving us three scenarios corresponding to low, medium, and high occupancy POIs. The SimpleGeo database does not include prominence values for POIs; we assigned values to the POIs from $\{0.95, 0.90, ...,0.3, 0.25\}$ using a Zipf distribution [24] with exponent 0.8. Lower prominence scores are more frequent.

To further validate our claims, we implement an Android application that can use the on-board GPS device, or a simulated GPS that can provide any desired latitude and longitude to the application. Using this application, we perform experiments covering five different cities (Los Angeles, New York, Paris, Vienna and Beijing), and 15 different keywords chosen from the place types list in the Google Places API.

4 Enforcing Indistinguishability

Consider a LPPM that generates some output s and shares it with the service provider as part of the querying process. This output in turn can be used by an adversary to infer potential locations for the user. We assume an attacker model where the adversary

1. knows which LPPM is under use,
2. knows all underlying parameters of the LPPM (except the user's location),

3. can observe all communication between the user and the service provider (can itself be a honest-but-curious adversary), and
4. has a prior knowledge on the user expressed as a probability distribution ϕ over the set of possible locations.

Under this model, we express the inferential capabilities of the adversary in terms of an odds-ratio with respect to any two locations l and l', given as

$$\frac{\Pr(l|s)}{\Pr(l'|s)} = \frac{\Pr(s|l)\phi(l)}{\Pr(s|l')\phi(l')}, \tag{3}$$

If a LPPM generates s independent of the location, i.e. $\frac{\Pr(s|l)}{\Pr(s|l')} = 1$, we can say that $\Pr(l|s) \propto \phi(l)$. In other words, inferences drawn by the adversary after revealing output s can also be drawn with just the prior knowledge, implying that the LPPM did not reveal any information of significance to the adversary. We can then say that any two locations are *indistinguishable* based on the output of the LPPM. Since the prior knowledge of the adversary can vary, and is outside the control of the LPPM, it is often the output probabilities ($\Pr(s|l)$) that are subjected to analysis. The objective is to maintain a degree of indistinguishability between two locations, i.e. the odds-ratio should remain as close as possible to the ratio of the prior probabilities.

4.1 Geo-indistinguishability

The principle of geo-indistinguishability provides a quantifiable degree to which the odds-ratio can deviate from the ratio of the prior probabilities [1].

Definition 4.1 (ϵ-geo-indistinguishability). *A LPPM is ϵ-geo-indistinguishable if, for any output l_z produced by the LPPM, and any two locations $l, l' \in L \subseteq \mathcal{L}$ with $d(l, l^{'}) \leq r$, we can have*

$$\frac{\Pr(l_z|l)}{\Pr(l_z|l')} \leq e^{\epsilon r},$$

where d is a distance function and $\mathcal{L}$ is a set of locations.

For an intuitive understanding, assume that the LPPM is used to query for a user located in Los Angeles downtown. Geo-indistinguishability then ensures that an adversary will be unable to distinguish with high certainty the user's location from other locations in the downtown area; although, the odds-ratio will enable the adversary distinguish between a location in the downtown area versus a location in one of the suburbs (farther away from the user's location). The odds-ratio is always within a factor of $\exp\{\epsilon d(l, l')\}$ of the ratio of prior probabilities. Therefore, the inferential advantage due to the use of the LPPM decreases as the adversary attempts to narrow down the user's location to smaller and smaller areas.

Andrés et al. proposed this principle for LPPMs along with a mechanism that achieves it. Their mechanism generates l_z using a planar Laplace distribution centered at the user's location. Thereafter, all POIs within a distance rad_R from l_z (area of retrieval) are retrieved from the server and then filtered locally. Since, the retrieved results have to be useful to the user, rad_R must be chosen such that user-relevant results are contained in the retrieved content. However, fixing rad_R simply based on the distance of the user from l_z introduces inference risks; the authors therefore provide a procedure to independently decide rad_R such that all POIs within a distance rad_I from the user's actual location (called the area of interest) are retrieved with a high confidence c (as high as 95%).

$$rad_R = rad_I - \frac{1}{\epsilon}\left(LambertW_{-1}\left(\frac{c-1}{e}\right) + 1\right) \tag{4}$$

Note that rad_R is not decided based on the location of the user or the perturbed location l_z. Using Eq. (4), we can choose rad_R and compute the resulting ϵ, or vice versa . In both cases, the mechanism uses an ϵ and rad_R value such that a high percentage c of the probability (of the user at a location) mass falls inside the area of retrieval. With $c = 95\%$, there is at least a 95% chance that the user is within a distance $(rad_R - rad_I)$ from the perturbed location. Therefore, depending on how a mechanism enforces the principle, approximate areas of presence of the user will be revealed with different certainties; the area of retrieval is one such area, but with a high certainty (possibly $\approx 100\%$). We can choose rad_R and rad_I such that $(rad_R - rad_I)$ is a very large quantity; but, doing so will force the mechanism to download a lot of redundant POI data.

4.2 Indistinguishability for Top-K Results

We first introduce the notion of a zone. A zone for a given location tells us what other locations generate similar top-K results. We use top_K to denote a function that returns the top-K POIs for a given location.

Definition 4.2 (Zone). *For a given location l_0 and $0 \leq m \leq K$, a zone Z_m is defined as*

$$Z_m(l_0) = \{l | l \in \mathcal{L}, |top_K(l_0) \cap top_K(l)| = K - m\}.$$

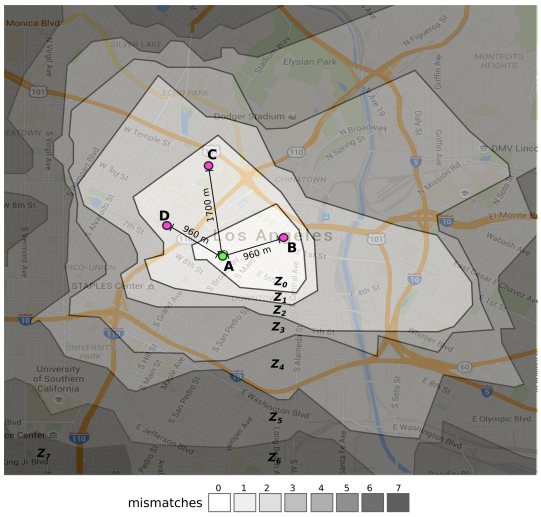

Fig. 1. Variation in the top-10 cafe set in Los Angeles. Ranking performed with $\alpha = 0.8$.

Figure 1 depicts an area of Los Angeles, overlaid with a set of zones. Assume that the user is located at l_0 (say A) in the central zone Z_0. For any other location l in Z_0, the top-10 cafe set is the same as that for l_0 (zero mismatch). When expanding into zone Z_1, the top-10 set undergoes a change in one of the POIs (one mismatch). Therefore, for any location $l \in Z_1$ and $l' \in Z_0$, we have $|top_{10}(l) \cap top_{10}(l')| = 9$. The number of mismatch increases as we move further out from Z_0. By definition, each zone is disjoint from one another, and zone boundaries indicate where the mismatch count relative to $top_K(l_0)$ increases by one (moving outward). In a combinatorial sense, a zone Z_m can encompass up to $\binom{K}{m}$ different top-K sets; but, each such possibility has exactly m mismatches with $top_K(l_0)$. Since the change is minor and happens infrequently, we can retrieve the top-K result of a location in a neighboring zone and still provide high accuracy. The bigger lower zones such as Z_0 and Z_1 are, the more distant the chosen location can be without affecting the accuracy. Note that zones are defined always with respect to a location; in the following, we sometimes do not mention this location when it is clear from the context.

An ϵ-geo-indistinguishable LPPM introduces probabilistic uncertainty based on the distance between two locations. As such, the adversary will gain some inferential advantage when distinguishing between the point pairs (A, B) or (C, D) in Fig. 1. With $\epsilon = \frac{\ln 4}{2000m}$, the prior probability ratio will at most change by a factor of 1.94531 for A and B (or D), and by 3.24901 between A and C. However, from the standpoint of querying from zone Z_0, locations A and B are equivalent since they produce the same result set; similarly C and D are equivalent under the relation of mismatch count.

This brings us to the question of whether geo-indistinguishability can be extended to the context of ranked geo-queries; we would like a LPPM that provides indistinguishability between two locations based on the similarity of query output from the two locations. In this direction, we extend ϵ-geo-indistinguishability as follows.

Definition 4.3 ((f, ϵ)-geo-indistinguishability). *Let $\mathcal{L}$ be a set of locations and $\mathcal{S}$ be the discrete set of outputs produceable by a LPPM M. Given a function f such that $f : \mathcal{L} \times \mathcal{L} \to [0, 1]$ and a privacy parameter $\epsilon \geq 0$, the mechanism M is (f, ϵ)-geo-indistinguishable if $\forall L \subseteq \mathcal{L}$ and $\forall s \in \mathcal{S}$, we have*

$$\frac{Pr(s|l)}{Pr(s|l')} \leq e^{\epsilon\delta},$$

when $f(l \in L, l' \in L) \leq \delta$.

In other words, for any subset of locations where pairwise relations (as measured by the f function) are bounded by some δ, the degree of indistinguishability is also bounded by a function of δ. The principle can also be extended to any subset of $\mathcal{S}$ if M has a continuous range. Since the condition applies to any conceivable subset of locations, we can say that if $f(l, l') = \delta$ for any $l, l' \in L \subseteq \mathcal{L}$, then

$$e^{-\epsilon\delta} \frac{\phi(l)}{\phi(l')} \leq \frac{\Pr(l|s)}{\Pr(l'|s)} \leq e^{\epsilon\delta} \frac{\phi(l)}{\phi(l')}. \tag{5}$$

If f is the normalized Euclidean distance function, the privacy definition states that the relative degree of indistinguishability between two locations resulting from observing an algorithm's output is related to the *distance between the two locations*—the closer the two locations, the harder it should be to distinguish which location was used to generate the output. This is precisely the basis for geo-indistinguishability. However, the f function can also mean other forms of relationships between locations. For example, f can be the fraction of mismatches in the top-K sets of two locations. Under this metric, if one retrieves the same query result from two locations ($\delta = 0$), then it should not be possible to learn which location was used in the query based on the query result. This gives us perfect indistinguishability for locations in Z_0 (Fig. 1), i.e. $\Pr(l|s) \propto \phi(l), \forall l \in Z_0$. Further, if two locations generate "similar" query results, then the degree of indistinguishability should be related to the *degree of similarity*. We can say that the f function is the metric used to relate the similarity of two locations.

What is the advantage of having an arbitrary f function in the privacy definition? The notion of establishing distinguishability based on arbitrary metrics is not new. Chatzikokolakis et al. also discuss a generalized differential privacy principle called $d_{\mathcal{X}}$-privacy where inputs similar to each other with respect to the $d_{\mathcal{X}}$ function should produce outcomes with similar probabilities [4]. A primary takeaway from the work is that different metrics can be useful in different applications. Geo-indistinguishability instantiates this principle using an Euclidean distance metric on locations. Reed and Pierce had earlier explored it in a context where similarities between expressions in functional programs are captured based on the values they produce [25]. (f, ϵ)-geo-indistinguishability is a restatement of such a general principle along with an explicit scaling factor ϵ. When limiting geo-indistinguishability to be strictly based on how close, or far apart, two locations are, we are unable to exploit any implicit similarities that two locations can have in the particular application. For example, if a query's result is not highly sensitive to the location (e.g. searching for top-10 hospitals), then even two distant locations ought to be fairly indistinguishable from one another; or, perhaps we want two locations to be indistinguishable based on how (dis)similar they are in terms of the population demographics. A distance based definition does not allow such possibilities; similarity distance is not always same as physical distance. One may ask for guarantees that such similar locations (as per f) exists in the real world. They may or may not depending on how f is defined and the type of the application. When considering top-K POI search, their existence is probable since businesses can be sparsely placed, and service providers are motivated towards finding the "best matches" instead of simply the "nearest matches." This work demonstrates how instantiating f as the mismatch function can be advantageous in applications performing top-K search based on distance and prominence.

If we consider $\Pr(l)/\Pr(l')$ to reflect the inferential capability of the adversary (closer to 1 means uncertainty), then Eq. (5) tells us that

$$\ln \frac{\text{capability after mechanism's output}}{\text{capability before mechanism's output}} \leq \epsilon\delta.$$

Then, ϵ corresponds to how much privacy leakage from a mechanism is allowed in the worst case ($\delta = 1$). We may set $\epsilon = 0$ to force no privacy leakage in any circumstance, but doing so can make the mechanism unproductive. However, if the f function mostly produces a δ value close to zero for the location pairs under consideration, then the privacy leakage is still close to nil. In that case, the value of ϵ can be very high as well, albeit the privacy leakage is contained (the worst case never happens). We revisit the assignment of ϵ in Section 5.3 and determine how it can be chosen to induce a certain level of utility.

The next question is whether a (f, ϵ)-geo-indistinguishable mechanism exists. Indeed, the general differential privacy mechanism suggested by McSherry and Talwar holds the evidence that a (f, ϵ)-geo-indistinguishable mechanism is possible [18]. This mechanism is driven by a quality function that can associate a real valued score to any $(s \in \mathcal{S}, l \in \mathcal{L})$ pair, with higher scores being more desirable.

Theorem 4.1. *Let q be a quality function, $q : \mathcal{S} \times \mathcal{L} \to \mathbb{R}^+$. Given $l \in \mathcal{L}$ and $\epsilon \geq 0$, the general mechanism M_g chooses output s with probability* $\Pr(s|l) \propto \exp\left\{\frac{\epsilon}{2} q(s, l)\right\}$. *The general mechanism M_g is $(f, \epsilon C)$-geo-indistinguishable, if $\forall L \subseteq \mathcal{L}$, we have*

$$\max_{s' \in \mathcal{S}; l, l' \in L} \big(q(s', l) - q(s', l')\big) = C\delta,$$

where $\delta = \max_{l, l' \in L} f(l, l')$ and C is a constant.

The general mechanism requires that, in all subsets of locations, the sensitivity of the quality function (maximum difference in scores) is always within a constant factor of the maximum value of the f function in the subset. Therefore, as more and more locations are considered, the sensitivity of the quality function must grow at a rate proportional to the change effectuated in the maximum f value. The proportionality constant C dictates the inferential advantage controlled by the mechanism.

4.3 An Application for 2-level Querying

The application we consider for a 2-level querying architecture first retrieves location and prominence data on a set of POIs, computes the top-K sets of specific locations, and then retrieves details on K POIs. The specific steps are as follows.

Step 1 (Probabilistic cloak) Choose a location l_q uniformly at random from within a radius rad_I from the user's location l_u. Send l_q to the server.

Step 2 (Minimal download) Download the location and prominence of all POIs matching the search keyword that are within a distance rad_R of l_q.

Step 3 (Local computation) Locally compute the top-K set of every location (cell) within a radius of rad_C from l_q.

Step 4 (Indistinguishable retrieval) Choose one of these sets using a (f, ϵ)-geo-indistinguishable mechanism and retrieve details for POIs in the set from the server.

Note that, always retrieving details for the top-K set corresponding to the user's true location should be avoided, since the set may be unique to the user's location, or a small area around it.

4.3.1 rad_I, rad_R and rad_C

We refer to the areas created by using the three radii as A_I, A_R and A_C respectively. Figure 2 illustrates the relationships we establish between the three values. A_I signifies an area of interest for the user, i.e. all relevant POIs contained in A_I should be used in determining the top-K sets. A typical `nearbysearch` using the Google Places API requires specification of a similar radius. Consequently, A_R should fully encompass A_I. This can be achieved by setting $rad_R \geq 2rad_I$. A typical `radarsearch` using the Google Places API allows for this value to be up to $50km$. Top-K sets are computed for every location (cell) in A_C, and a choice is made from within these sets. We want the top-K set of the user to be a part of this sampling; since l_q is always within a distance of rad_I from l_u, inclusion of the user's top-K set can be guaranteed by setting $rad_C \geq rad_I$. l_q, A_R and A_C are known publicly, but A_I is not known since the user location is a secret. Therefore, the choice of rad_C reveals a first level approximation of the area of presence of the user. We choose rad_I to control this approximation, and subsequently set $rad_C = rad_I$. Service providers can limit the number of POIs returned from within the area of retrieval (`radarsearch` puts a limit of 200); therefore, we choose rad_R to the minimum value necessary, i.e. $2rad_I$. This approach leaves us with making one parametric choice, rad_I, with the other two decided as $rad_R = 2rad_I$ and $rad_C = rad_I$.

4.3.2 Probabilistic cloak

The four steps in the 2-level querying architecture is composed of two disclosure mechanisms—probabilistic cloak and indistinguishable retrieval. Probabilistic cloak reveals a coarse approximation of the user's location, given in terms of the area A_C centered at l_q. Since l_q is always within a distance rad_I from the user location l_u, this disclosure can be controlled by setting larger values

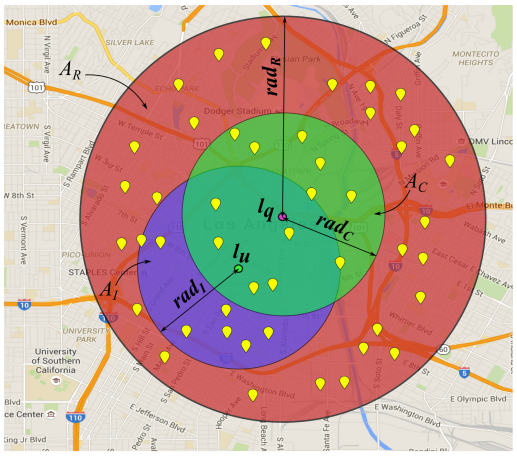

Fig. 2. Areas induced by the three radii values. l_u is the true user location and l_q is a randomly generated location within distance rad_I of l_u. Location and prominence data is downloaded for all POIs inside A_R. $rad_I : rad_R : rad_C = 1 : 2 : 1$.

for rad_I. l_q is chosen uniformly at random, thereby limiting the leakage on where the user is within the area of retrieval.

Proposition 4.1. *Given* l_q, rad_I, rad_R *and* rad_C, *where* $rad_R = 2rad_I$ *and* $rad_C = rad_I$, *we have*
(i) $\Pr(l|l_q, rad_I, rad_R, rad_C) = 0$ *if* $l \notin A_C$, *and*
(ii) $\forall l \in A_C$, $\Pr(l|l_q, rad_I, rad_R, rad_C) \propto \phi(l)$.

The above proposition tells us that, as a result of the probabilistic cloak, the adversary gets to know that the user is not outside of A_C. However, locations inside A_C are still indistinguishable (discrimination is possible only to the extent possible by already existing prior knowledge ϕ). Therefore, the odds-ratio expressed in Eq. (3) remains unchanged for locations inside A_C. In the next section, we introduce our mechanism for indistinguishable retrieval that limits the leakage of the user position when POI details are retrieved. As such, the subsequent analysis is focused on the privacy guarantees within A_C.

4.3.3 Indistinguishable retrieval

For step 3, we use the computation process discussed in Section 2.2 to determine the top-K sets of all locations in A_C. Let $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ represent the collection of these top-K sets corresponding to the locations $l_1, l_2, ..., l_m \in A_C$, i.e. $top_K(l_i) = t_i$. For step 4, we consider the following instantiation of the general mechanism M_g.

Definition 4.4 (Mechanism M_{fgi}). *Let the quality function $q : \mathcal{T} \times A_C \to \mathbb{R}^+$ be the fraction of matches between a set $t \in \mathcal{T}$ and the top-K set of location $l \in A_C$, i.e.*

$$q(t, l) = \frac{|t \cap top_K(l)|}{K}.$$

Given the user location l_u, mechanism M_{fgi} outputs a set $t \in \mathcal{T}$ with probability $Pr(t|l_u) \propto e^{\frac{\epsilon}{2} q(t, l_u)}$.

The quality function in mechanism M_{fgi} is a measure of the overlap between two top-K sets. Consequently, the probability with which a set is chosen by the mechanism decays exponentially as it becomes more and more different from the top-K set corresponding to the user's location. Details are subsequently retrieved for the POIs included in the output produced by M_{fgi}, which now adds to the knowledge of the adversary.

Theorem 4.2. *Mechanism M_{fgi} is (f, ϵ)-geo-indistinguishable for locations in $\mathcal{L} = A_C$ when*

$$f(l, l') = 1 - \frac{|top_K(l) \cap top_K(l')|}{K}.$$

The f function here is the fraction of mismatches in the top-K sets of two cells l and l'. (f, ϵ)-geo-indistinguishability implies that, for all pairs of locations in A_C whose top-K sets have at most a fraction of δ mismatch, the probabilities of producing a certain output from either location in the pair will differ at most by a factor of $e^{\epsilon\delta}$ and at least by a factor of $e^{-\epsilon\delta}$ of each other. For location pairs where there are no mismatches ($\delta = 0$), the probabilities will be equal. For location pairs with complete mismatch ($\delta = 1$), the probability ratio is between e^{ϵ} and $e^{-\epsilon}$. This captures the guarantee that any location l (including the user location l_u) will be indistinguishable in zone Z_0 (defined corresponding to $top_K(l)$), and difficult to distinguish from locations in nearby zones. The best case happens when the entire A_C is covered in a single zone, a possibility that can emerge when POIs are sparse, and their ranking involves both distance and prominence.

4.3.4 Radius choice and privacy

A mechanism can enforce (f, ϵ)-geo-indistinguishability over any subset of locations. Any choice will result in some trade-off between privacy and cost. Our approach balances it by saying that the area inferable is always A_C, and within that area, distinguishability is a function of the similarity of query results. While in a typical execution of the geo-indistinguishability based mechanism in Section 4.1, there is a negligible chance that the area of retrieval does not contain the user, here there is no such uncertainty about A_C. It is therefore important to stress that M_{fgi} can enforce the principle only for locations inside $\mathcal{L} = A_C$ (Thm. 4.2). The credibility of a mechanism then depends on the size of the area it reveals (with high certainty) and how is privacy enforced inside that area. As before, no radii value is decided based on the location of the user; they are directly dependent on the system parameter rad_I. Both approaches benefit from the choice of larger radii values since the revealed area is then of a larger size. However, an approach designed for a 1-level architecture will require the download of large amounts of data, while an approach for a 2-level architecture will only download location and prominence data. In this regard, a 2-level architecture scales better and can accommodate the requirement of larger retrieval areas (arising from the choice of larger rad_I values) without much affect on the communication cost.

4.3.5 Characterization

Mechanism M_{fgi} makes locations in zone Z_0 indistinguishable from each other; however, the degree of indistinguishability reduces with respect to locations in other zones. An an example, consider $\epsilon = 30$ in (f, ϵ)-geo-indistinguishability, with the user being at l_0. Then for a location l that has the same top-K result set as that of l_0 ($l \in Z_0(l_0)$; $\delta = 0$), the posterior probability ratio (Def. 4.3) is one, meaning the locations are indistinguishable. When $\delta = 0.1$ (1 mismatch), the ratio is at most 20.09. On the other hand, enforcing ϵ-geo-indistinguishability with $\epsilon = 0.00474$ (area of interest of radius $1km$ will be inside area of retrieval of radius $2km$ with 95% confidence), a location l that is at most 500 meters away from l_0 will have the ratio to be at most $e^{500\epsilon} = 10.68$. Therefore, if it is the case that all locations within 500 meters of l_0 are in $Z_0(l_0)$, then clearly (f, ϵ)-geo-indistinguishability provides a stronger privacy guarantee than ϵ-geo-indistinguishability. In the following theorem, we generalize this intuition, and characterize the size that zones need to be if (f, ϵ)-geo-indistinguishability is to be a preferable choice.

Theorem 4.3. *Let s be the output of a ϵ_{gi}-geo-indistinguishable mechanism and $\tilde{s}$ be the output of a (f, ϵ_{fgi})-geo-indistinguishable mechanism when the input (user) location is l_0. Let $L_m = \{l | l \in \mathcal{L}, d(l_0, l) \geq {}^{\epsilon_{fgi} m}/_{\epsilon_{gi} K}\}$ with d as the Euclidean distance function. If $Z_m(l_0) \subseteq$*

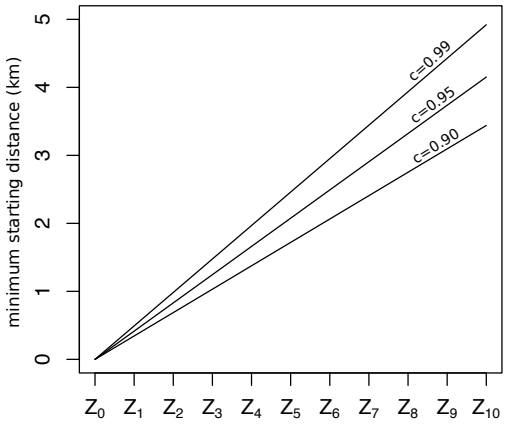

Fig. 3. Desired minimum starting distance of a zone relative to a location.

L_m for all $m \in \{0, 1, ..., K\}$, then $\forall l \in \mathcal{L}$

$$\frac{Pr(l|\widetilde{s})}{Pr(l_0|\widetilde{s})} \leq \frac{Pr(l|s)}{Pr(l_0|s)}.$$

The above theorem characterizes when a (f, ϵ)-geo-indistinguishable mechanism is not worse than a conventional geo-indistinguishable mechanism in terms of the discriminatory advantage (the odds-ratio) introduced by the mechanisms. In conventional geo-indistinguishability, relative to any fixed location l_0, indistinguishability as measured by the output probability ratio diminishes continuously with increasing distance from l_0; whereas the changes generate a monotonic step function in M_{fgi}. Theorem 4.3 implies that the step function $\frac{\epsilon_{fgi}}{\epsilon_{gi}} f(l_0, \cdot)$ should preferably grow slower than the distance function $d(l_0, \cdot)$. Therefore, any zone Z_m should start at a distance of $\frac{\epsilon_{fgi} m}{\epsilon_{gi} K}$ or more from l_0. Figure 3 shows this minimum distance in three different scenarios. Each scenario captures the case when the ϵ values are chosen such that a given level of confidence (c) is always present in the service quality. The parameter ϵ in geo-indistinguishability is chosen so that the area of retrieval ($rad_R = 2km$) contains the area of interest ($rad_I = 1km$) with confidence c [1]—the three corresponding ϵ_{gi} values are $\epsilon_{0.99} = 0.00664, \epsilon_{0.95} = 0.00474$, and $\epsilon_{0.90} = 0.00389$ (the subscript indicates the confidence level). Correspondingly, the parameter in (f, ϵ)-geo-indistinguishability is obtained such that at most 2 mismatches can happen with probability c—the three corresponding ϵ_{fgi} values are $\epsilon_{0.99} = 32.67, \epsilon_{0.95} = 19.68$, and $\epsilon_{0.90} = 13.38$. We discuss the methodology for this in Section 5.3. At a confidence level of 95%, zones are required to have a span of at least $415.19m$, which changes to $492.02m$ at 99%. When a zone is wider than this minimum necessary size, it allows subsequent zones to be narrower by an equal amount. Since not all locations in Z_0 are always the required distance away from the closest border of Z_0, it is clear that the inequality does not hold for all query locations l_0. Nonetheless, the inequality may still hold farther out if subsequent zones are wider than necessary. On 1000 random queries in Los Angeles with $\alpha = 0.8$, we observed that the average radius of Z_0 and Z_1 (relative to the centroid of Z_0) is $908m$ and $2.086km$ respectively for a dense POI such as cafe, while it is $1.212km$ and $2.473km$ for a sparse POI such as bookstore.

4.3.6 Composite privacy

The 2-level application is composed of two forms of location disclosure: one direct (in the form of a perturbed location), and one indirect (in the form of a top-K set). As such, the overall privacy enforced in such an application is dependent on what inferences can be drawn by an adversary using a composition of all outputs produced by all mechanisms in the application. In our particular instantiation, the probabilistic cloaking mechanism protects the privacy of the user at a coarse (tunable) level. Nonetheless, the mechanism's dependence on rad_I helps an adversary immediately disqualify locations that are more than rad_I distance away from the center of the area of retrieval. However, the uniform random sampling performed in the mechanism does not allow finer grain localization of the user than what is already possible using the prior knowledge of the adversary (Prop. 4.1). Subsequently, the privacy inside the area of retrieval is dictated by the indistinguishable retrieval mechanism (M_{fgi}). The composite privacy guarantee in our application can therefore be stated as satisfying (f, ϵ)-geo-indistinguishability inside the area of local computation (A_C), and non-existent on areas outside. We demonstrate in Section 8 that such a composite mechanism can induce higher estimation errors than a single step mechanism for a Bayesian adversary.

We can replace the probabilistic cloaking mechanism by a standard geo-indistinguishable mechanism in order that locations outside A_C also carry a non-zero probability for the user to be present. In most practical usages, such a mechanism will still be tuned such that a significant portion of the probability mass (95% or 99%) is still inside the area the retrieval, implying that the user is "very likely" to be inside the retrieval area. For composite privacy, the posterior knowledge resulting from the use of geo-indistinguishability will serve as the prior knowledge in the analysis of the subsequent

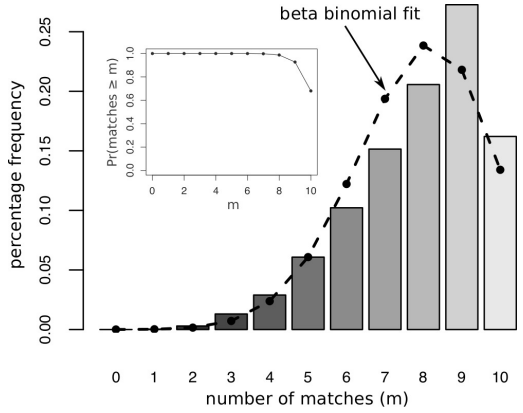

Fig. 4. Observed and fitted base match distribution ($rad = 2km, K = 10$) for cafes in Los Angeles, CA, USA. Inset figure shows probability of obtaining matches in the set chosen by mechanism M_{fgi}.

(f, ϵ)-geo-indistinguishable retrieval mechanism. When a high confidence value c is used, this posterior knowledge is more revealing than when using the uniform probabilistic cloaking mechanism. This highlights the importance of exploring disclosure mechanisms in the first step that are more (privacy) protective than that in the subsequent retrieval step. We have not studied here how coarse grain and fine grain disclosures in the two steps can be best composed to obtain better levels of composite privacy than when using a single 1-level application.

5 Retrieval Accuracy

The retrieval accuracy in the 2-level application described above is determined by the number of matches in the top-K set chosen by mechanism M_{fgi} and the top-K set corresponding to the user location. This in turn is influenced by the density of relevant POIs in the neighborhood of the user. Therefore, our approach includes some observations derived from real world POI categories and their densities.

5.1 Base Match Distribution

A top-K ranking function emphasizes both distance and prominence of a POI. As a result, the top-K set corresponding to a location does not undergo abrupt changes in neighboring locations. It can therefore be expected that, irrespective of the use of any privacy mechanism, the top-K set relative to the user's location will have matches with the top-K set of nearby locations. The base match distribution attempts to capture this similarity as a probability mass function.

Definition 5.1 (Base match distribution). *The base match distribution $\omega_{rad,K}$ is the probability distribution corresponding to the discrete random variable $R_{rad,K} : \mathcal{L} \times \mathcal{L} \rightarrow \{\Theta, 0, 1, ..., K\}$ where*

$$R_{rad,K}(l, l') = \begin{cases} \Theta & , \textit{if } d(l, l') > rad \\ |top_K(l) \cap top_K(l')| & , \textit{otherwise} \end{cases}.$$

The base match distribution $\omega_{rad,K}(m)$ provides the probability that any two locations within a distance rad of each other will have m matches in their top-K sets. For example, Fig. 4 shows a histogram of the number of matches (top-10 "cafe" sets) seen in a sample of 10^6 location pairs in Los Angeles, with locations in a pair being at a distance of at most 20 cells ($2km$) from each other. We obtain an estimate of the base match distribution $\hat{\omega}_{20,10}$ by fitting a beta-binomial distribution to this data. This estimate is useful in obtaining an insight into the approximate scale of ϵ that needs to be chosen in M_{fgi} for the mechanism to generate useful results. It is impractical to estimate a base match distribution for every possible search keyword; therefore, we also validate the comparative effectiveness of using a simple binomial distribution, or even a uniform distribution.

5.2 Match Probability

When the base match distribution is skewed towards higher matches, a uniform sampling from the different top-K sets can itself lead to a majority of high matches. For example, $\hat{\omega}_{20,10}$ implies a match of 8 or more in approximately 60% of the cases. Mechanism M_{fgi} further scales these probabilities to make the drawing of high match sets significantly more likely.

Proposition 5.1. *Mechanism M_{fgi} produces an output t for the user location l_u such that*

$$\Pr(|t \cap top_K(l_u)| \geq m) = \frac{\sum_{i=m}^{K} \omega_{rad,K}(i) \exp\left\{\frac{\epsilon}{2K} i\right\}}{\sum_{j=0}^{K} \omega_{rad,K}(j) \exp\left\{\frac{\epsilon}{2K} j\right\}},$$

where $\omega_{rad,K}$ is the base match distribution for the top-K search.

We can cluster the candidate top-K sets into equivalence classes based on the number of matches they have with $top_K(l_u)$. The base distribution then provides an estimate of the percentage of sets with a given

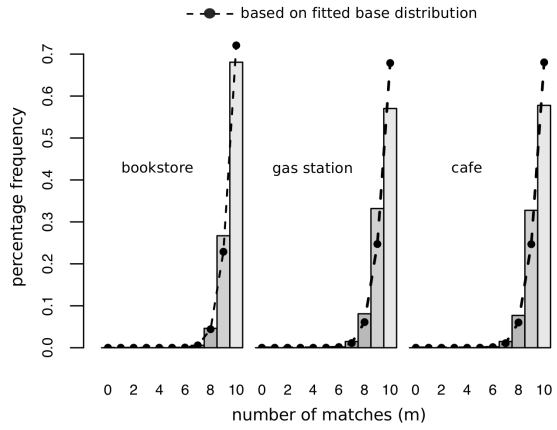

Fig. 5. Observed match frequencies in three different POI categories. Top-10 sets computed with $\alpha = 0.8$ and $rad_C = 2km$. M_{fgi} samples using $\epsilon = 30$.

quality score. Mechanism M_{fgi} exponentially scales the probability of choosing an output with higher quality score. The inset plot in Fig. 4 shows the impact of this scaling when picking a top-10 cafe set in the example. The scaling increased the probability of obtaining 8 or more matches to 98% with $\epsilon = 30$. Figure 5 depicts the match frequencies in three different POI categories, having low, medium and high occupancy across the query area. Mechanism M_{fgi} is used here with $rad_C = 2km$ and $\epsilon = 30$, and $\alpha = 0.8$ for top-10 ranking. For each category, the data points are generated by performing queries from 1000 randomly chosen locations within the experiment area, with 100 executions of M_{fgi} at each location. The match probabilities computed from using a fitted base distribution reasonably captures the observed match frequencies. As expected, sparse POIs (bookstore in this case) induce a higher retrieval accuracy.

5.3 Choosing ϵ

The choice of ϵ directly influences the output probabilities of the sets, and in turn impacts the retrieval accuracy. We can ensure that M_{fgi} provides a minimum of m matches with confidence c by solving for ϵ in the following equation derived from Prop. 5.1.

$$c \sum_{i=0}^{m-1} \omega_{rad,K}(i) \exp\left\{\frac{\epsilon}{2K} i\right\} - (1-c) \sum_{i=m}^{K} \omega_{rad,K}(i) \exp\left\{\frac{\epsilon}{2K} i\right\} = 0 \quad (6)$$

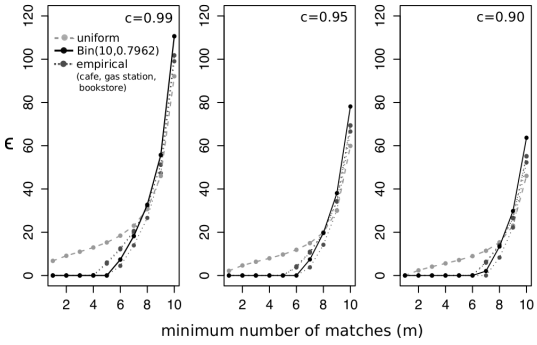

Fig. 6. Computed ϵ using various base distributions and confidence levels $c = 0.99, 0.95$ and 0.90.

We use a Newton-Raphson iterative solver in R to solve for ϵ. Figure 6 illustrates the minimum ϵ value necessary to guarantee at least m matches (x-axis) in the chosen set with a confidence of $90\%, 95\%$ and 99%. While the use of the base match distribution is preferable in determining ϵ, it is not necessarily practical. The figure also presents the ϵ values obtained by using two other distributions in lieu of the base match distribution—a uniform distribution signifying no knowledge of the base distribution, and a binomial distribution with parameters $n = 10$ and $p = 0.7962$. The parameters of the binomial distribution are chosen such that approximately $\frac{2}{3}$ of the probability mass is concentrated in values greater than 7. This choice is made after analyzing the empirical base distribution of 15 different POI categories, where the total probability mass in 8, 9 and 10 matches is observed to be between 60-75%. The binomial distribution approximates the trends of the three low, medium, and high occupancy POI categories better than the uniform distribution. It overestimates ϵ when higher match counts are desired. Based on the binomial base distribution, a value of $\epsilon = 32.67$ gives us a 99% probability of obtaining 8 or more matches.

6 Parametric Evaluation

The performance of the proposed 2-level application is determined by a combination of three parameters, namely α : the weight given to distance in the ranking function, rad_I : the area of interest, and ϵ : the privacy parameter in M_{fgi}. We provide comparative results of their impact on the retrieval accuracy for the three example POI categories. The default values are $rad_I = 2km$ and $\epsilon = 30$, with top-10 ranking performed using $\alpha = 0.8$.

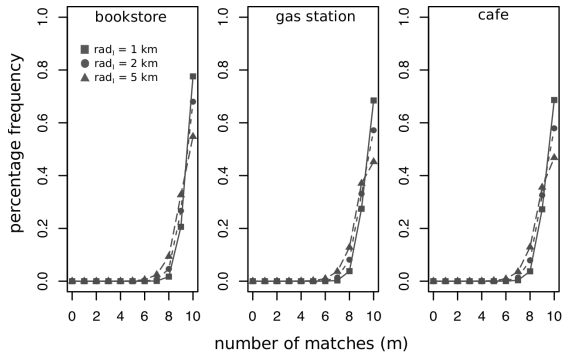

Fig. 7. Impact of the area of interest (rad_I) on retrieval accuracy. $rad_C = rad_I$.

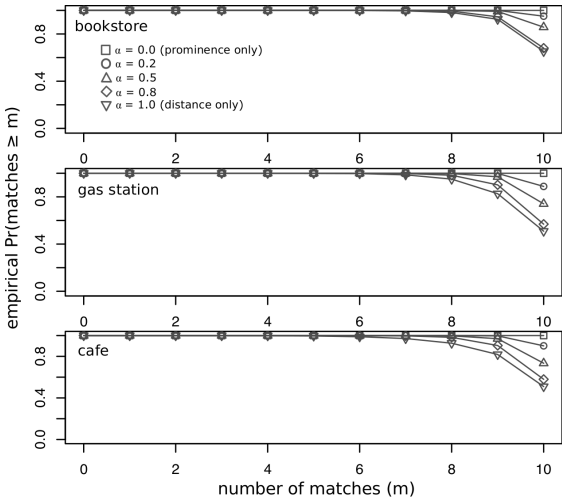

Fig. 8. Impact of α on retrieval accuracy.

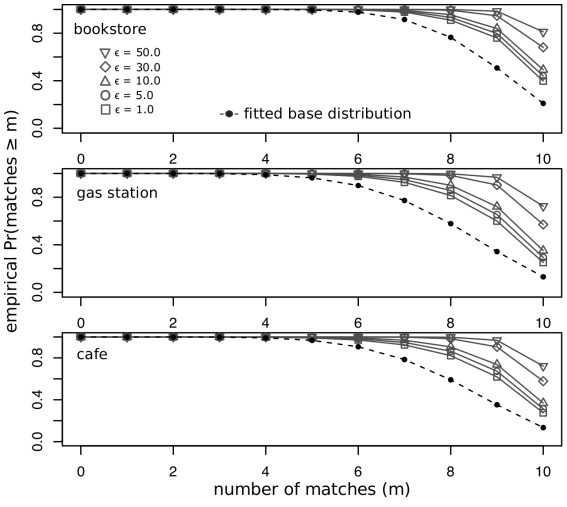

Fig. 9. Impact of privacy parameter (ϵ) on retrieval accuracy.

rad_I **Impact.** Figure 7 depicts the percentage frequency when exactly m matches are obtained between the actual top-K result set and that generated by M_{fgi}. The chances of retrieving the exact set drops as the area of interest becomes larger, while that of retrieving a set with one or two mismatches increases. The behavior is not surprising since larger rad_I values, correspondingly a larger rad_C, imply that the potential set of outputs contains a comparatively smaller fraction of samples with $m = 10$. As such, the base distribution has a lower mass at that point. However, increasing rad_I also creates higher chances of covering zones Z_1 and Z_2 . The number of potential sets in Z_1 and Z_2 are combinatorially higher than in Z_0 (single top-K set); increasing rad_I creates avenues for inclusion of more of these sets. As long as rad_I is not set so large that other low match sets get included in majority, we can expect to retain the high retrieval accuracy. At $rad_I = 5km$, we still obtain 8 or more matches with probability greater than 90%, higher in some POI categories.

α **Impact.** Figure 8 depicts the impact of α on the retrieval accuracy. $\alpha = 0$ signifies ranking based only on prominence, and hence there is a single top-K set corresponding to all locations. $\alpha = 1$ signifies a K-nearest-neighbor ranking; this case presents the least favorable condition for mechanism M_{fgi}. The case of $\alpha = 1$ also demonstrates what happens when a uniform distribution is assumed for the POI importance; since ranking uses relative scores, using $\alpha = 1$ or using the same γ_i value for all POIs produces the same ranks. Any deviation from a uniform distribution favorably impacts the expected accuracy since we can increase the chances for two locations to have the same top-K results. Between these two extreme conditions, differences in the retrieval accuracy is mostly observed for $m = 9$ and $m = 10$. The differences are less prominent in the sparsely distributed POI as changes in the top-K set are unlikely for small changes in the user location. With $\alpha = 0.2$ or $\alpha = 0.5$ (half the weight on the distance value), we obtain a significantly high probability ($> 95\%$) of obtaining 9 or more matches. Therefore, accurate results can be retrieved irrespective of how distance and prominence are weighed in the ranking by the service provider. The high accuracy in retrieving true results can be attributed to the exponential scaling performed by the mechanism to the output probabilities of high scoring sets.

ϵ **Impact.** Figure 9 depicts the impact of ϵ on the retrieval accuracy. Lower values of ϵ reduce the influence of the exponential weights on the base match distribution. At $\epsilon = 0$, the mechanism samples proportional to the base distribution. High matches can be made more likely by increasing its value. Observe that the differences in match probability is more prominent in cases

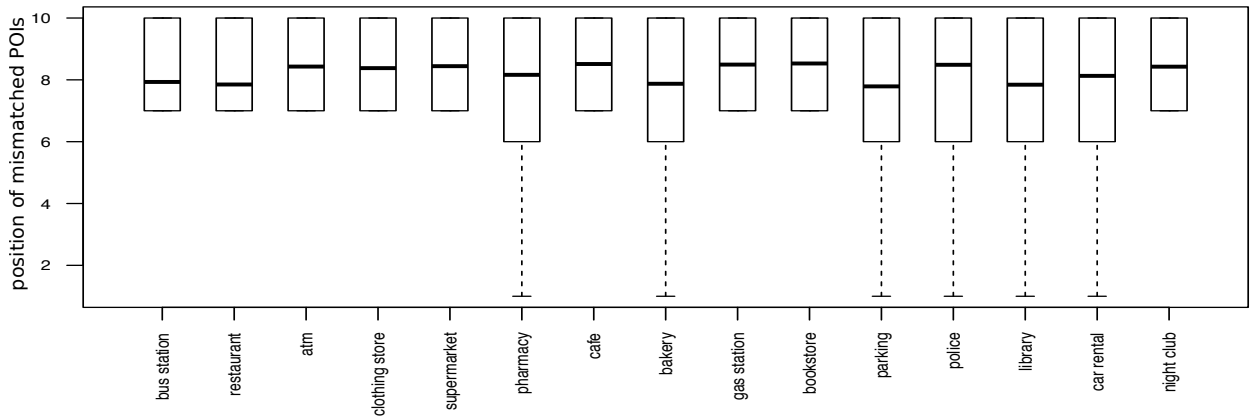

Fig. 10. Position of POIs in the true top-K set when missed by the application.

such as $m = 9$ and $m = 10$. Even with a small value such as $\epsilon = 1$, we observe probabilities as high as 80% for 7 or 8 matches. We discussed in Section 5.3 how the parameter can be appropriately chosen when a given level of certainty is desired in the number of matches. High values for the parameter can still be chosen (for better accuracy) when most location pairs under consideration are likely to produce similar results (e.g. sparse POIs). Since ϵ reflects the worst case inferential capability (when mismatch is 100%), the resulting impact on privacy can still be relatively low as the mismatch function's value acts as a scale down multiplier to ϵ.

7 An Android Implementation

We implemented the example 2-level POI search application in Android using the Google Places API to perform the queries. The application allows the user to input a search keyword and reads the device GPS (or a simulated GPS) to obtain the user location. It then retrieves POI locations for the search category using a `radarsearch` query. A `radarsearch` query returns locations and unique identifiers for POIs within a specific radius (rad_R) of the query point. The area of interest (rad_I) is a configurable parameter which we set to $2km$ in the following; correspondingly $rad_R = 2rad_I = 4km$. 10-nearest-neighbor ranking is performed ($\alpha = 1$), partly because prominence data is not yet available using the Places API, and partly because K-nearest-neighbor search produces the worst case behavior as per the parametric evaluation. Details are then retrieved (using the `details` endpoint and identifiers of the POIs) for 10 POIs decided by the (f, ϵ)-geo-indistinguishable mechanism M_{fgi} with $\epsilon = 30$. The application is run on a Nexus 5X smartphone over a 4GLTE connection. All networking tasks are performed using a thread-pool with 4 threads, and HTTP persistent connections (to reduce latency).

We use a desktop application to perform 1000 `radarsearch` queries from random locations for each of the 15 chosen search keywords and in each of the five chosen cities (Los Angeles, New York, Paris, Vienna and Beijing), giving a total of 75000 queries. We also run mechanism M_{fgi} to pick a set for details retrieval. This process allows us to compute retrieval accuracy and analyze the ranks of missed POIs. For a subset of 100 queries (per city per keyword) chosen uniformly at random, the Android application is executed on the smartphone and performance results such as timing and bandwidth usage are gathered. We restrict the experiments on the smartphone to a smaller subset since running all 75000 queries from the phone would incur a large cumulative 4G bandwidth ($\approx 7.6GB$).

Rank of missed POIs. Figure 10 shows whisker plots of the position (1 = highest rank, to 10 = lowest rank) of POIs that appear in the actual top-10 set but are missed by the application. Results from the five cities are summarized across different categories. The median position is approximately 8, with POIs in the top 6 positions being retrieved at least 75% of the times. This highlights that changes appearing in the top-10 sets are incremental and often starts in the lower ranked POIs.

Retrieval accuracy. Figure 11 summarizes the percentage number of times (empirical probability) when at least a given number of matches are found. The key point we highlight here is that the observations are very similar across the different cities (Fig. 11a) and across different keywords (Fig. 11b). The observations

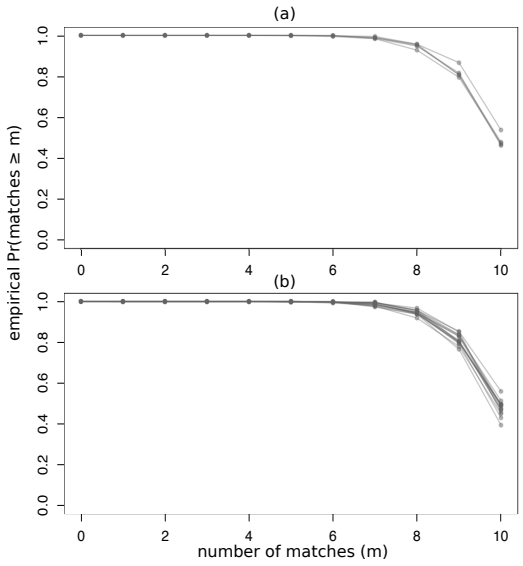

Fig. 11. Retrieval accuracy (a) per city (all keywords) and (b) per keyword (all cities).

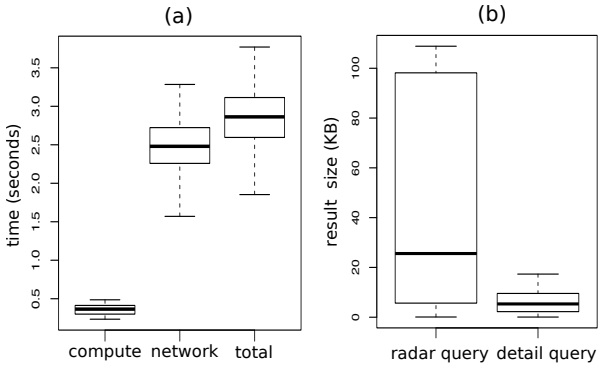

Fig. 12. (a) End to end time of one query in the Android application. (b) Size (KB) of JSON file retrieved in a `radarsearch` and POI detail query. $rad_R = 4km$.

are in accordance with the results seen in the evaluation performed within the Los Angeles area alone.

Runtime performance. Figure 12a shows the quartiles of the end-to-end time to execute one complete query in the Android application. The end-to-end time consists of compute and network time. Compute time includes the parsing of network data, computing top-K sets, computing the probability mass function, sampling using the mechanism, and updating the user interface with details of retrieved results. Network time includes connection time to Google servers, issuing requests, and then buffering of responses. As a result of the fast top-K computation algorithm, the compute time is under half a second in all cases. The network communication takes the most time, contributing a median of 2.5 seconds. Note that the total time to execute a query in a typical search application (e.g. Google Maps search for Android) averages around 2.5 seconds. Therefore, the overhead introduced in the 2-level application is negligible.

Communication cost. Figure 12b shows the size of the responses (as JSON files) received from performing a `radarsearch` and a POI detail query. A `radarsearch` query returns an average size of 43.3KB of data, a median of 26.1KB, and sizes are between 100KB to 110KB in 25% of the queries. Each query to retrieve details about one POI returns an average of 6.3KB, and a median of 5.4KB. The 2-level application performs one `radarsearch` query and retrieves details on K (= 10) POIs, therefore incurring a median cost of 78.8KB and an average of 107KB per query. A 1-level application will have to retrieve details on all POIs inside the area of retrieval, which amounts to an average size of 1.2MB per query for 200 POIs found inside the area of retrieval. Therefore, the total bandwidth cost in the 2-level architecture can be around 10 times (107KB vs 1.2MB) lower than in the 1-level architecture.

8 Comparative Performance

In Section 4, we provided a characterization for query locations when the proposed mechanism can provide higher indistinguishability than a ϵ-geo-indistinguishable mechanism. Next, we present a comparative assessment in terms of a separate privacy metric, namely the expected estimation error of the adversary. For a given prior distribution ϕ on locations, the expected estimation error of the adversary measures the average distance between the true location of the user and the location estimated by the adversary [27]. Therefore, this metric computes the privacy level taking into consideration the likelihood of the user being in locations favorable under Thm. 4.3, as well as those that are not. Consider the geo-indistinguishable mechanism in Section 4.1, where the output produced by the mechanism is a perturbed location l_z. The expected estimation error of a Bayesian adversary is then computed as

$$experr_M = \sum_{l,l_z,l' \in \mathcal{L}} \phi(l) \Pr(l_z|l) \Pr(l'|l_z) d(l,l'). \quad (7)$$

Using Eq. (4), we compute the minimum required value of ϵ such that the area of interest with $rad_I = 1km$, is contained within an area of retrieval (center l_z and $rad_R = 2rad_I = 2km$) with confidence $c = 0.90$, giving us $\epsilon_{0.90} = 0.00389$.

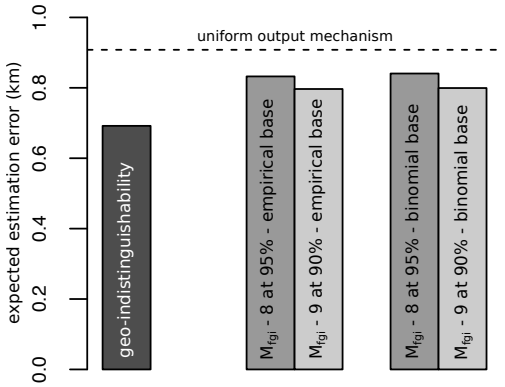

Fig. 13. Expected estimation error of adversary.

For our 2-level approach, a perturbed location l_q is produced in the probabilistic cloaking step, and then a set $s \in \mathcal{T}$ is produced in the retrieval step. The expected error is therefore computed as

$$experr_M = \sum_{l,l_q,l' \in \mathcal{L}; s \in \mathcal{T}} \phi(l) \Pr(l_q|l) \Pr(s|l_q,l) \Pr(l'|s,l_q) d(l,l'). \quad (8)$$

$\Pr(s|l_q,l)$ is as per mechanism M_{fgi}, and $\Pr(l'|s,l_q)$ is computed as

$$\frac{\Pr(s|l',l_q)\Pr(l_q|l')\phi(l')}{\sum_{l''}\Pr(s|l'',l_q)\Pr(l_q|l'')\phi(l'')}. \quad (9)$$

Recall that the output is generated by first selecting a location (l_q) for retrieval and then choosing an output from one of the top-K sets. Specifically, $\Pr(l_q|l) = 0$ if $d(l,l_q) > rad_I$ and is $\frac{1}{\pi rad_I^2}$ otherwise (l_q is uniformly chosen for POI location retrieval). Similarly, $\Pr(s|l_q,l)$ is zero if s is not the top-K set of some location within a distance of rad_I ($rad_C = rad_I$) from l_q. Note that the expected estimation error computation accounts for each step in both mechanisms. As such, the metric's value reflects an overall privacy level enforced by the mechanisms. For the prior distribution, we consider a uniform distribution inside an area with $1km$ radius centered at Los Angeles downtown (34.0522^o N, 118.2428^o W). We consider the ϵ parameter in M_{fgi} under two accuracy requirements: 8 or more matches with 95% confidence, and 9 or more matches with 90% confidence. We compute the parameter by solving Eq. (6) using the empirical base match distribution corresponding to the search keyword and a $\texttt{Bin}(10, 0.7962)$ distribution as the base match distributions.

For a mechanism that results in uniform probabilities for the terms in Eq. (7), the expected error in the given scenario is $908m$. Using uniform probabilities also signify an "ignorant" adversary lacking any prior knowledge about the user, or the mechanism in use. Such a mechanism only reveals the area of retrieval, and that the user is most likely somewhere inside it. Figure 13 shows the expected error for a top-10 search with the keyword "cafe." The ϵ-geo-indistinguishable mechanism provides an expected error of $691m$; comparatively, the use of mechanism M_{fgi} results in an expected error of $840m$ when using the binomial base distribution (8 matches at 95% confidence level). The difference between the two approaches also appears in the resulting bandwidth usage. There is an average of 117 cafes inside an area of retrieval of radius $2km$. Using the average response sizes reported in Section 7, a query using a geo-indistinguishable mechanism would result in the usage of 737.1KB, compared to approximately 85KB with the (f,ϵ)-geo-indistinguishable mechanism.

9 Related Work

Anonymity sets. Location privacy has earlier been achieved through the use of obfuscation and dummy queries. A user can hide her actual query in a set of dummy queries and achieve location privacy [15]. Gruteser and Grunwald [12] proposed the use of spatial and temporal cloaking to obfuscate user locations. The cloaking is performed at a trusted third party site. Individual preferences in terms of temporal and spatial tolerances can also be incorporated during such cloaking [10]. Enforcing properties such as k-anonymity ensures that users will not be uniquely located inside a region in a given period of time. Multiple other suggestions are available on how the cloaking region should be formed [2, 7, 16, 19]. Kalnis et al. proposed that all obfuscation methods should satisfy the reciprocity property [13] in order to prevent inversion attacks where knowledge of the underlying anonymizing algorithm can be used to identify the actual user.

Beyond anonymity sets. Moving beyond anonymity sets, Khoshgozaran et al. proposed a protocol where K-nearest neighbor queries are reduced to a set of private block retrieval operations on a database [14]. These retrievals can be performed using a tamper-resistant processor located at the server so that the content provider is oblivious of the retrieved blocks.

Xu and Cai argued that privacy should be treated as a feeling-based property, and proposed using the popularity of a public region as the privacy level [31]. Soriano et al. showed that the privacy assurances of this

model do not hold when the adversary possesses footprint knowledge on the spatial regions over time [29]. Niu et al. recently revisited the use of dummy queries with the objective of addressing side information that the adversary may have on query probabilities from different locations [21]. In a subsequent work, the authors demonstrated that caching of query results can help improve the privacy in dummy query models [22]. These works are driven by an entropy-based privacy metric.

Shokri et al. argued that location privacy should be quantified based on the expected estimation error of an adversary [26]. They provided a method to arrive at different types of inferences regarding a user's location based on a known mobility profile of the user. Using methods of likelihood estimations, the authors showed that above measures such as the anonymity set size or entropy do not correctly quantify the privacy enforced by the method [28].

Differential privacy. Dewri introduced the idea of merging a well-known form of privacy in databases, namely differential privacy, and k-anonymity [5]. Under this model, an anonymity set of size k is first formed and then an obfuscated location is generated such that the probabilities of reporting this location from any of the k locations are close to each other. Andrés et al. improved this approach by proposing geo-indistinguishability [1]. Xiao and Xiong further proposed differential privacy based extensions to prevent inferences in continuous query systems [30]. These integrations of location privacy and differential privacy remain the state-of-the-art in privacy models for location privacy protection. The primary drawback of these models is that the choice of the obfuscated location is driven only by privacy requirements, and no attempt is made to accommodate or exploit its impact on the query results.

Privacy-accuracy trade-off. Examination of the privacy/accuracy trade-off in location-based applications is rare. Shokri et al. explored an optimal location obfuscation method that can hinder privacy attacks and provide the best service quality, essentially targeting an equilibrium solution in a Stackelberg Bayesian game [27]. They compute quality loss as the average dissimilarity in service quality between the user's true location and a pseudo-location. Privacy is computed as the expected error of the adversary in an inference attack. Along similar lines, Bordenabe et al. provided a mechanism to minimize the service quality loss for a given degree of geo-indistinguishability [3]. Similar to most of the earlier works, both of these works assume that service quality in an application is directly proportional to the distance between the pseudo-location and the true location.

To the best of our knowledge, Dewri et al.'s prior work is the only known attempt to consider arbitrary ranking functions for local search results [6]. They presented a fast top-K computation algorithm suitable for use in a mobile device, and provided evidence that top-K sets do not change significantly for nearby locations. They also provided empirical results demonstrating that reasonable privacy can be achieved for certain prior distributions. However, the work falls short of providing a formal privacy guarantee, especially when assumptions on prior distributions cannot be made.

10 Conclusions

In this work, we presented a LPPM for points-of-interest retrieval where a query point is cloaked within an area of retrieval, and all locations inside the area whose top-K result sets have the same number of mismatches relative to the top-K set of the query point become equally indistinguishable. We theoretically characterized when the approach provides stronger levels of privacy than a geo-indistinguishable mechanism, and provided the framework necessary to tune the mechanisms to guarantee a required level of accuracy. The empirical evaluation drives us to the conclusion that our LPPM can retain high similarity with the sought top-K set, irrespective of the how much contribution distance makes to the ranking, the density of the POIs in the search area, or variations in the ϵ parameter. The mechanism can execute on a mobile device without generating any noticeable delays or incurring excessive bandwidth cost. It also induces expected errors (for an adversary) that are closer to that produced by a uniform output mechanism.

As with geo-indistinguishability, the privacy assurances in (f, ϵ)-geo-indistinguishability also degrade if used to protect multiple locations. For the scenario where n queries are made in sequence, the effective ϵ value in both mechanisms is $n\epsilon$ at the end of the queries. Clearly, an inherent trade-off can be achieved in the accuracy of query results and the corresponding privacy guarantees. It may also be useful to determine the largest area A_C that can be realized by the proposed mechanism for a given POI distribution and a given level of communication cost in the 2-level architecture. Doing so will enable clearer comparisons of privacy levels induced by the mechanism, all else being equal. We leave the in-depth exploration of these aspects for future work.

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential Privacy for Location-Based Systems. In *Proceedings of the 20th ACM Conference on Computer and Communications Security*, pages 901–914, 2013.

[2] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting Anonymous Location Queries in Mobile Environments with Privacy Grid. In *Proceedings of the 17th International World Wide Web Conference*, pages 237–246, 2008.

[3] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Optimal Geo-Indistinguishable Mechanisms for Location Privacy. In *Proceedings of the 21th ACM Conference on Computer and Communications Security*, pages 251–262, 2014.

[4] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the Scope of Differential Privacy Using Metrics. In *Proceedings of the 2013 International Symposium on Privacy Enhancing Technologies*, pages 82–102, 2013.

[5] R. Dewri. Local Differential Perturbations: Location Privacy Under Approximate Knowledge Attackers. *IEEE Transactions on Mobile Computing*, 12(12):2360–2372, 2013.

[6] R. Dewri, W. Eltarjaman, P. Annadata, and R. Thurimella. Beyond the Thin Client Model for Location Privacy. In *Proceedings of the 2013 International Conference on Privacy and Security in Mobile Systems*, pages 1–8, 2013.

[7] R. Dewri, I. Ray, I. Ray, and D. Whitley. Query m-Invariance: Preventing Query Disclosures in Continuous Location-Based Services. In *Proceedings of the 11th International Conference on Mobile Data Management*, pages 95–104, 2010.

[8] J. Freudiger, R. Shokri, and J.-P. Hubaux. Evaluating the Privacy Risk of Location-Based Services. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, pages 31–46, 2011.

[9] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. De-anonymization Attack on Geolocated Data. *Journal of Computer and System Sciences*, 80(8):1597–1614, 2014.

[10] B. Gedik and L. Liu. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[11] Google. Google Places API. https://developers.google.com/places/web-service/search#PlaceSearchRequests, 2017. [Online; accessed 1-March-2017].

[12] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services*, pages 31–42, 2003.

[13] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[14] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr. Location Privacy: Going beyond k-Anonymity, Cloaking and Anonymizers. *Journal of Knowledge and Information Systems*, 26(3):435–465, 2011.

[15] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique Using Dummies for Location-Based Services. In *Proceedings of the IEEE International Conference on Pervasive Services*, pages 88–97, 2005.

[16] F. Liu, K. A. Hua, and Y. Cai. Query l-Diversity in Location-Based Services. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 436–442, 2009.

[17] C. Y. T. Ma, D. K. Y. Ma, N. K. Yip, and N. S. V. Rao. Privacy Vulnerability of Published Anonymous Mobility Traces. *IEEE/ACM Transactions on Networking*, 21(3):720–733, 2013.

[18] F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

[19] M. F. Mokbel, C. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services Without Compromising Privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774, 2006.

[20] Moz, Inc. The 2015 Local Search Ranking Factors. https://moz.com/local-search-ranking-factors, 2015. [Online; accessed 1-March-2017].

[21] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li. Achieving K-anonymity in Privacy-aware Location-based Services. In *Proceedings of the 33rd Annual IEEE International Conference on Computer Communications*, pages 754–762, 2014.

[22] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li. Enhancing Privacy through Caching in Location-based Services. In *Proceedings of the 34th Annual IEEE International Conference on Computer Communications*, pages 1017–1025, 2015.

[23] B. O'Clair, D. Egnor, and L. E. Greenfield. Scoring local search results based on location prominence, 2011. US Patent 8,046,371.

[24] D. M. W. Powers. Applications and Explanations of Zipf's Law. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pages 151–160, 1998.

[25] J. Reed and B. C. Pierce. Distance Makes the Types Grow Stronger: A Calculus for Differential Privacy. In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, pages 157–168, 2010.

[26] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying Location Privacy. In *Proceedings of the 32nd IEEE Symposium on Security and Privacy*, pages 247–262, 2011.

[27] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. L. Boudec. Protecting Location Privacy: Optimal Strategy Against Localization Attacks. In *Proceedings of the 19th ACM Conference on Computer and Communications Security*, pages 617–627, 2012.

[28] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an Old Cloak: k-Anonymity for Location Privacy. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, pages 115–118, 2010.

[29] M. Soriano, S. Qing, and J. Lopez. Time Warp: How Time Affects Privacy in LBSs. In *Proceedings of the 12th International Conference on Information and Communications Security*, pages 325–339, 2010.

[30] Y. Xiao and L. Xiong. Protecting Locations with Differential Privacy under Temporal Correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1298–1309, 2015.

[31] T. Xu and Y. Cai. Feeling-Based Location Privacy Protection for Location-Based Services. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pages 348–357, 2009.

[32] Yelp Inc. Yelp API v3. https://www.yelp.com/developers/documentation/v3/business_search, 2017. [Online; accessed 1-March-2017].

APPENDIX

Proof of Theorem 4.1

For some output $s \in \mathcal{S}$, and any $l, l' \in L \subseteq \mathcal{L}$ such that $\max_{l,l' \in L} f(l,l') = \delta$ and $\max_{s' \in \mathcal{S}; l,l' \in L} (q(s',l) - q(s',l')) = C\delta$, we have

$$\frac{\Pr(s|l)}{\Pr(s|l')} = \exp\left[\frac{\epsilon}{2}(q(s,l) - q(s,l'))\right] \frac{\sum_{s' \in \mathcal{S}} \exp\left[\frac{\epsilon}{2} q(s',l')\right]}{\sum_{s' \in \mathcal{S}} \exp\left[\frac{\epsilon}{2} q(s',l)\right]}$$
$$\leq \exp\left[\frac{\epsilon}{2} C\delta\right] \exp\left[\frac{\epsilon}{2} C\delta\right] = e^{\epsilon C \delta}. \blacksquare$$

Proof of Proposition 4.1

Let the user be at location l_u and ϕ be the prior distribution that the adversary has on the user's location. l_q is selected uniformly at random from A_I (disc centered at l_u with radius rad_I; see Fig. 2). Then

$$\Pr(l_q|l, rad_I) = \begin{cases} p & , d(l_q,l) \leq rad_I \\ 0 & , otherwise \end{cases},$$

where p is the constant probability value under the definition of a uniform distribution.

(i) A_C is centered at l_q and has radius $rad_C = rad_I$. For any point $l \notin A_C$, we have $d(l_q,l) > rad_I$. Therefore, l_q could not have been selected if the user was at $l \notin A_C$.

(ii) For $l, l' \in A_C$, we have $d(l_q,l) \leq rad_C$ and $d(l_q,l') \leq rad_C$. Then,

$$\begin{aligned} & \frac{\Pr(l|l_q, rad_I, rad_R, rad_C)}{\Pr(l'|l_q, rad_I, rad_R, rad_C)} \\ = & \frac{\Pr(l_q, rad_I|l)\phi(l)}{\Pr(l_q, rad_I|l')\phi(l')} \\ = & \frac{\Pr(l_q|l, rad_I) Pr(rad_I|l)\phi(l)}{\Pr(l_q|l', rad_I) Pr(rad_I|l')\phi(l')} \\ = & \frac{p\phi(l)}{p\phi(l')} \quad [rad_C = rad_I \text{ is chosen statically}] \\ = & \frac{\phi(l)}{\phi(l')}. \end{aligned}$$

Therefore, $\forall l \in A_C$, $\Pr(l|l_q, rad_I, rad_R, rad_C) \propto \phi(l)$.$\blacksquare$

Proof of Theorem 4.2

f is the fraction of mismatches in the top-K sets of two locations $l, l' \in \mathcal{L} = A_C$. If $\max_{l,l' \in L \subseteq \mathcal{L}} f(l,l') = \delta$, i.e. $f(l,l') \leq \delta$ then $|top_K(l) \cap top_K(l')| \geq K(1-\delta)$.

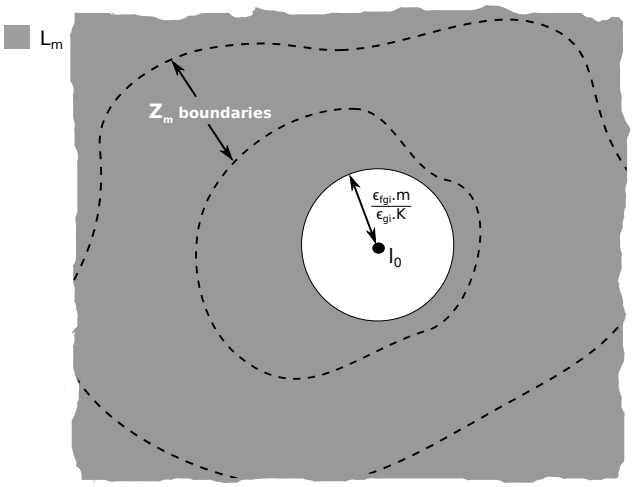

Fig. 14. $Z_m(l_0)$ and L_m with $Z_m(l_0) \subset L_m$.

Consider $|top_K(l) \cap top_K(l')| = K(1-\delta)$. Therefore, the two top-K sets differ in $K\delta$ elements. The maximum difference in quality is provided by the set t which has the least overlap with one of the sets, and the most overlap with the other while satisfying the condition. If t overlaps in $K\delta + b$ elements in one set, then at least b elements of t will also appear in the other set. Therefore, the maximum difference in quality scores in this case will be $\frac{K\delta+b}{K} - \frac{b}{K} = \delta$.

For cases where $|top_K(l) \cap top_K(l)| > K(1-\delta)$, the sets will differ in less than $K\delta$ elements; so the maximum difference in quality scores will be less than δ.

Combining both cases, when $|top_K(l) \cap top_K(l')| \geq K(1-\delta)$, the maximum difference in quality scores will be δ.

$$\max_{s' \in \mathcal{S}; l,l' \in L \subseteq \mathcal{L} \text{ such that } f(l,l') \leq \delta} \big(q(s,l) - q(s,l')\big) = \delta.$$

Here the constant $C = 1$. Therefore, by Thm. 4.1, the mechanism is (f, ϵ)-geo-indistinguishable.$\blacksquare$

Proof of Theorem 4.3

Figure 14 illustrates the relationship between $Z_m(l_0)$ and L_m. For all $l \in Z \subseteq Z_m(l_0) \subseteq L_m$, we have $f(l,l_0) = \frac{m}{K}$. Therefore,

$$\begin{aligned} \max_{l \in Z} \frac{\Pr(\widetilde{s}|l)}{\Pr(\widetilde{s}|l_0)} &= e^{\epsilon_{fgi} m/K} \\ &\leq e^{\epsilon_{gi} d(l,l_0)}, \forall l \in Z \qquad [\text{since } l \in L_m] \\ &\leq \max_{l \in Z} \frac{\Pr(s|l)}{\Pr(s|l_0)}. \end{aligned}$$

Considering Z as singleton sets ($Z = \{l\}$), we obtain, $\forall l \in Z_m(l_0)$

$$\frac{\Pr(\widetilde{s}|l)}{\Pr(\widetilde{s}|l_0)} \leq \frac{\Pr(s|l)}{\Pr(s|l_0)}.$$

Using the fact that $\mathcal{L} = \bigcup_{m=0}^{K} Z_m(l_0)$, we have from Eq. (3), $\forall l \in \mathcal{L}$

$$\frac{\Pr(l|\widetilde{s})}{\Pr(l_0|\widetilde{s})} \leq \frac{\Pr(l|s)}{\Pr(l_0|s)}.\blacksquare$$

Proof of Proposition 5.1

For any output choice t that has i matches with $top_K(l_u)$, we have $q(t, l_u) = i/K$. Since the expected number of top-K sets with i matches in a radius of rad is $\omega_{rad,K}(i)$, we have

$$\Pr(|t \cap top_K(l_u)| = i) = \omega_{rad,K}(i) \frac{\exp\left\{\frac{\epsilon}{2}\frac{i}{K}\right\}}{\sum_{j=0}^{K} \omega_{rad,K}(j) \exp\left\{\frac{\epsilon}{2}\frac{j}{K}\right\}}.$$

Therefore,

$$\begin{aligned}\Pr(|t \cap top_K(l_u)| \quad &\geq \quad m) = \sum_{i=m}^{K} \Pr(|t \cap top_K(l_u)| = i) \\ &= \quad \frac{\sum_{i=m}^{K} \omega_{rad,K}(i) \exp\left\{\frac{\epsilon}{2K} i\right\}}{\sum_{j=0}^{K} \omega_{rad,K}(j) \exp\left\{\frac{\epsilon}{2K} j\right\}}.\blacksquare\end{aligned}$$