This is an author created version of the article. The original manuscript is available from http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.34.

Exploiting Service Similarity for Privacy in Location Based Search Queries

Rinku Dewri, *Member, IEEE,* and Ramakrisha Thurimella

Abstract—Location-based applications utilize the positioning capabilities of a mobile device to determine the current location of a user, and customize query results to include neighboring points of interests. However, location knowledge is often perceived as personal information. One of the immediate issues hindering the wide acceptance of location-based applications is the lack of appropriate methodologies that offer fine grain privacy controls to a user without vastly affecting the usability of the service. While a number of privacy-preserving models and algorithms have taken shape in the past few years, there is an almost universal need to specify one's privacy requirement without understanding its implications on the service quality. In this paper, we propose a user-centric location-based service architecture where a user can observe the impact of location inaccuracy on the service accuracy before deciding the geo-coordinates to use in a query. We construct a local search application based on this architecture and demonstrate how meaningful information can be exchanged between the user and the service provider to allow the inference of contours depicting the change in query results across a geographic area. Results indicate the possibility of large default privacy regions (areas of no change in result set) in such applications.

Index Terms—Privacy-supportive LBS, location privacy, service quality.

1 Introduction

The consumer market for location-based services (LBS) is estimated to grow from 2.9 billion dollars in 2010 to 10.4 billion dollars in 2015 [1]. While navigation applications are currently generating the most significant revenues, location-based advertising and local search will be driving the revenues going forward. The legal landscape, unfortunately, is unclear about what happens to a subscriber's location data. The non-existence of regulatory controls have led to a growing concern about potential privacy violations arising out of the usage of a location-based application. While new regulations to plug the loopholes are being sought, the privacy-conscious user currently feels reluctant to adopt one of the most functional business models of the decade.

Privacy and usability are two equally important requirements for successful realization of a location-based application. Privacy (location) is loosely defined as a "personally" assessed restriction on when and where someone's position is deemed appropriate for disclosure. To begin with, this is a very dynamic concept. Usability has a two fold meaning—a) privacy controls should be intuitive yet flexible, and b) the intended purpose of an application is reasonably maintained. Towards this end, prior research have led to the development of a number of privacy criteria, and algorithms for their optimal achievement. However, there is no known attempt to bring into view the mutual interactions between the accuracy of a location coordinate and the service quality from an application using those coordinates. Therefore, the question of what minimal location accuracy is required for a LBS application to function, remains open. The common man's question is: "how important is my position to get me to the nearest coffee shop?"—which unfortunately remains unanswered in the scientific community.

It is worth mentioning that a separate line of research in analyzing anonymous location traces have revealed that user locations are heavily correlated, and knowing a few frequently visited locations can easily identify the user behind a certain trace [2], [3]. The privacy breach in these cases occurs because the location to identity mapping results in a violation of user anonymity. The proposal in this work attempts to prevent the reverse mapping—from user identity to user location—albeit in a user-controllable manner.

1.1 Related Work

Location obfuscation has been extensively investigated in the context of privacy. Obfuscation has been earlier achieved either through the use of dummy queries or cloaking regions. In the dummy query method, a user hides her actual query (with the true location) amongst a set of additional queries with incorrect locations [4], [5]. The user's actual location is one amongst the locations in the query set. The additional processing overhead at the LBS, resulting from the dummy queries, must be addressed while using this method. Cheng et al. propose a data model to augment uncertainty to location data using circular regions around all objects [6]. They use imprecise queries that hide the location of the query issuer and yield probabilistic results. The results are modeled as the amount of overlap between the query range and

• *R. Dewri and R. Thurimella are with the Department of Computer Science, University of Denver, CO 80208, USA. Email:{rdewri,ramki}@cs.du.edu.*

the circular region around the queried objects. Yiu et al. propose an incremental nearest neighbor processing algorithm to retrieve query results [7]. The process starts with an anchor, a location different from that of the user, and it proceeds until an accurate query result can be reported. The work focuses on reducing the communication cost of the repeated querying mechanism.

Trusted third party based approaches rely on an anonymizer that creates spatial regions to hide the true location of users. The use of spatial and temporal cloaking to obfuscate user locations was first proposed by Gruteser and Grunwald [8]. Continuing on, Gedik and Liu develop a location privacy architecture where each user can specify maximum temporal and spatial tolerances for the cloaking regions [9]. Drawing inspiration from the concept of k-anonymity in database privacy [10], Gedik and Liu enforce a location k-anonymity requirement while creating the cloaking regions. This requirement ensures that the user will not be uniquely located inside the region in a given period of time. Ghinita et al. propose a decentralized architecture to construct an anonymous spatial region, and eliminate the need for the centralized anonymizer [11]. In their approach, mobile nodes utilize a distributed protocol to self-organize into a fault-tolerant overlay network, from which a k-anonymous cloaking set of users can be determined. Kalnis et al. propose that all obfuscation methods should satisfy the reciprocity property [12]. This prevents inversion attacks where knowledge of the underlying anonymizing algorithm can be used to identify the actual object [13]. Parameter specification remains the biggest hindrance to real world application of these techniques. Even when a user has advanced knowledge to comprehend the implications of a parameter setting on location privacy, the impact on service is unknown in these approaches. Refer to Section 1 of the supplementary file for additional literature review.

1.2 Contributions

Our contributions in this work are two-fold. First, we propose a novel architecture for LBS applications that is directed towards revealing privacy/utility trade-offs to a user before an actual geo-tagged query is made. Unlike a typical competitive architecture where the LBS provider does not actively participate in making privacy decisions, we envision a *privacy-supportive LBS* as a provider willing to provide supplemental information for making "informed" privacy decisions. An informed decision implies that the LBS user operates under reasonable knowledge about the service level implications of revealing her location with a given degree of inaccuracy. Under this platform, a user first obtains an overview of the impact of using inaccurate locations in a certain query. Thereafter, the actual query made to the service provider is geo-tagged with a location that the user has carefully chosen to balance result accuracy and location privacy. We describe in Section 2 the underlying rationale, setting, expectations and components that go into such an architecture. Refer to Section 2 of the supplementary file for a separate study, which demonstrates that users have the flexibility of adding significant noise to their locations and still obtain accurate search results.

As our second contribution, we present in Section 3, a proof of concept design for a privacy-supportive local search LBS. Given a search term (e.g. generic ones such as "cafes", and targeted ones such as "starbucks coffee") and a highly generalized user location (e.g. the metropolitan city), the privacy-supportive LBS generates a concise representation of the variation in the 10-nearest neighbor result set as a hypothetical user moves across the large metropolitan area. Once the representation is communicated to the user, she can infer the geographic variability that can be introduced in her location coordinates to retrieve all or a subset of the result set. Our results, using a publicly available local business database, indicate that the proposed approach can precisely reveal the area boundaries within which the result set is fully preserved (a default privacy level). Further, we observe a high degree of precision in estimating the area boundaries when user requirements on result set accuracy are relaxed (i.e. location sensitivity is hardened). Section 4 presents the empirical results to support these claims.

2 Privacy-supportive LBS

Future LBS architectures must make room for a service provider to cooperate with the user in making sound privacy decisions. There is a growing skepticism on how a LBS provider handles (or might handle) location data. If strong market adoption is an agenda item for these businesses, then it becomes their responsibility to present evidence that the sought location accuracy is indeed a characteristic requirement of the application. Further, regulatory enforcements on location data procurement, and subsequent liability in the event of improper handling, can make the collection of unnecessarily precise geo-locations an unattractive choice. From a computational perspective, only the service provider maintains the database of queried objects in real time. Therefore, it is reasonable that differences (or similarities) in the output of a query can be efficiently computed at the server side. A user cannot make informed privacy decisions without this computation. In light of these arguments, a privacy-supportive LBS seems both appropriate and important. Note that a simple opt-in LBS is not privacy-supportive, since the implications of not using ones geo-location is not available to the user.

2.1 Setting

The communication setting we assume includes one or more users equipped with GPS-enabled devices, and a LBS provider possessing a database of points-of-interest (POI). These points-of-interest may be static, as in local business listings, or dynamic, as in a friend-finder service where users frequently check-in/out of the underlying

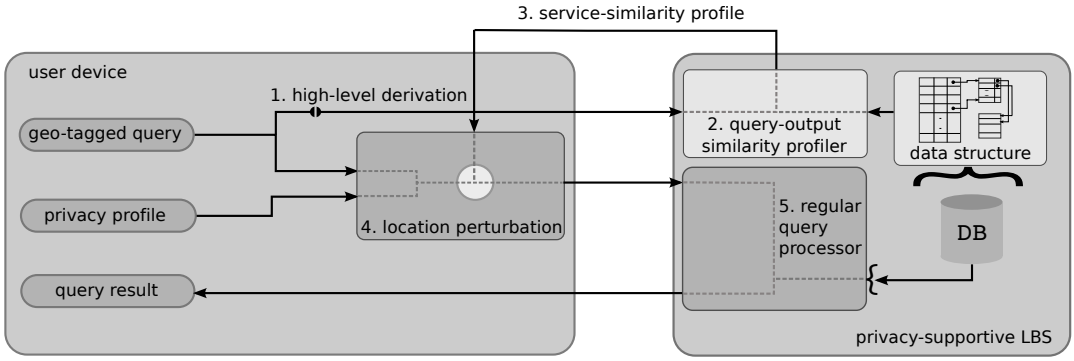

Figure 1. Communication order for a location-based query in the presence of a privacy-supportive LBS.

social-networking platform. Similar to in almost all operating LBS applications, user access to the service is augmented by a geographic tag identifying the position of the user. Authentication may or may not be required to use the service, although many applications claim to be able to provide a better result set in the latter case. The service itself may require other parameters to be specified, such as search keywords or profile descriptions. The geographic tag in the query is typically the GPS-coordinates of the user device, but can also be a carefully crafted location as explained in the next subsection.

2.2 Architecture

The location disclosure mechanism in a privacy-supportive LBS architecture employs an intermediate communication with the LBS. A high-level schematic of the communication pattern is depicted in Fig. 1. The user device forwards the query to the LBS, albeit uses a high-level generalization of the user's geographic location in it. This generalization may be derived as per user-specification (say at the level of the city), or obtained automatically from the location approximation that a provider can infer using a cell-towers and wifi-access points database[1]. In response to this first query phase, the user obtains a *service-similarity profile*. This profile is a representation of the similarities in the query output at different geographic locations. The exact form taken by this profile, as well as the data structures employed in computing this profile, may vary from application to application. A location perturbation engine on the user side then determines a noisy location to use based on the user's *privacy profile* and the retrieved service-similarity profile. The LBS processes the query with respect to the noisy location.

A user can manually interact with the service-similarity profile to assess which locations have the highest (or acceptable) level of result set similarity, within the constraints of the location noise she wants to infuse into the query. In this case, a good visualization of the similarity profile is required. Although this is the most flexible method of putting the trade-off information to use, such high degree of interaction will affect the usability of the application, specially when queries are made frequently. Hence, we assume that action axioms have been provided by the user to make the process automatic. The privacy profile then states how a location is to be selected for different categories of applications, their importance, and the relative location sensitivity. Policy specifications such as these, and their integration into the decision making process, warrant an extensive exploration. We will avoid this frontier in this work. A naive approach is to allow the user to select a location sensitivity level (much like choosing the ringer-state in a mobile phone), assess query result accuracy at the corresponding location granularity (using the similarity profile), and notify the user if the accuracy drops below a threshold. Note that the policy executes within a user's device and reveals little or no information on how locations get chosen.

2.3 Privacy expectations and threat model

We interpret location privacy as the *accuracy* with which an adversary can determine the position of a user. This interpretation resembles the intuitive perception that a location estimated closer to our true position is more encroaching on our privacy than a relatively distant estimation. However, the privacy-supportive architecture does not make any assumption on what is "distant" and what is "close enough." This is a significant departure from statistical measures of privacy, where a statement on "what is private" must be made pro-actively before issuing the query. A privacy-supportive LBS does not require this decision until the user determines the usability of the information that would be revealed as a result of the location disclosure, if at all. In light of this difference, the architecture, its underlying algorithms, or the service provider itself, cannot make any claims on the enforced level of privacy. It only facilitates the process to enforce personally desirable levels of location privacy after careful consideration of its impact. On similar grounds, we assume a threat model where the

1. Creating and updating cell-towers and wi-fi access point maps is a costly affair. The businesses that do so (Skyhook, Google, Apple, Navizon, etc.) often consider it proprietary. The legal standard for accessing these databases is currently being litigated in a number of cases (http://epic.org/privacy/location_privacy).

provider is semi-honest (follows protocol but may be curious). Note that, on one hand, even the weakest of the adversaries may learn the precise locations of a privacy-indifferent user (one who always reveals the true location), while on the other, even the strongest of the adversaries may learn nothing additional from a privacy-paranoid user. A privacy-aware user would use the system to her advantage, perhaps frequently revealing accurate (not necessarily precise) positions, and occasionally the heavily perturbed ones. An adversary who can classify these locations as real or dummy, infers some knowledge about the user's whereabouts—however, this is information that the user has opted to reveal in the first place.

3 A LOCAL SEARCH APPLICATION

Mobile local search is demonstrating an upward market trend, the gap with the desktop counterpart diminishing in the next three years, and then rising further[2]. Given the penetration of web-enabled handheld devices in the consumer market, it has become exceedingly common for a user to instantly look up the information she seeks to find. These search queries are estimated to produce 27.8 billion more queries than desktop-search by the year 2016. A vast majority of the users performing mobile search seek access to information pertinent in the locality of the query. Multiple LBS applications—e.g. Where, AroundMe, MeetMoi, Skout and Loopt—have spawned in the past few years to address this market segment. In general, a local search application provides information on local businesses, events, and/or friends, weighted by the location of the query issuer. Location and service accuracy trade-offs are clearly present in a local search LBS. A privacy-supportive variant is therefore well-suited for this application class. Local search results tend to cycle through periods of plateaus and minor changes as one moves away from a specified location. The plateaus provide avenues for relaxation in the location accuracy without affecting service accuracy, while the minor changes allow one to assess accuracy in a continuous manner.

3.1 Problem statement

In the traditional usage of a local search application, the user would communicate a search keyword to the provider, and retrieve a ranked list of records matching the search term. Let us denote the items that match the search term in the points-of-interest database by $\mathcal{P} = \{P_1, P_2, ..., P_N\}$. A ranking function $\mathcal{R}$ is applied to this set and a top-k subset of the ranked results is returned to the user. Since neighboring results are considered more useful, the ranking function would utilize the geo-location of the user. We use $\mathcal{R}_k(\mathcal{P}, pos)$ to collectively denote this result set when retrieved with respect to the position pos.

2. Source: BIA/Kesley Press Releases, April 2012

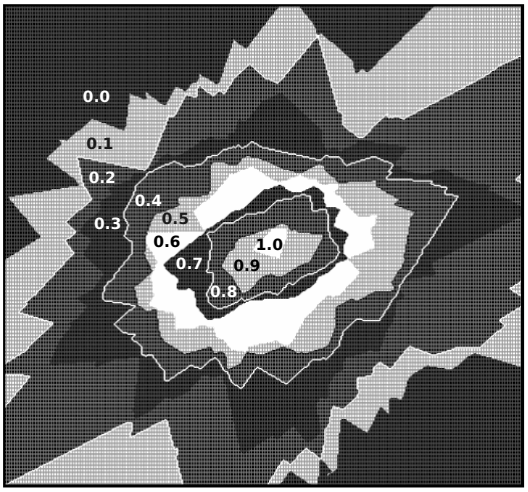

Figure 2. Hypothetical query result set similarity with the user at the center of the area.

3.1.1 An ideal scenario

Let us next consider a hypothetical scenario where the user has access to a matrix that shows the percentage similarity of the result set with respect to the user's current location. In order to formalize this map, let us superimpose a grid of $r \times c$ cells on a geographic area $\mathcal{G}$. In local search, it is sufficient to restrict focus to this geographic area while determining the set $\mathcal{P}$. The position of the user in the grid is given as $p = \langle x_0, y_0 \rangle$. Let Sim be a similarity function, defined in this application as follows.

$$Sim(\langle x, y \rangle, \langle x', y' \rangle) = \frac{|\mathcal{R}_k(\mathcal{P}, \langle x, y \rangle) \cap \mathcal{R}_k(\mathcal{P}, \langle x', y' \rangle)|}{k}.$$

For brevity, we will also use $\mathcal{R}_k(\mathcal{P}, \langle x, y \rangle)$ and $\mathcal{R}_k(\mathcal{P}, \langle x', y' \rangle)$ as arguments to the Sim function. Let $\mathcal{S}_{x_0,y_0}$ be a matrix of r rows and c columns, with

$$\mathcal{S}_{x_0,y_0}[i,j] \quad = \quad Sim(\langle x_0, y_0 \rangle, \langle i, j \rangle)$$

Hence, $\mathcal{S}_{x_0,y_0}$ is a cell-by-cell measure of the similarity of the result set retrieved for the user's position relative to that retrieved for any other position in the grid. As depicted in Fig. 2, this matrix allows the user to identify cell boundaries where the result set similarity gradually decreases from 100% to 0%. We can call them the *service-contour* of the issued query. The innermost region in the figure, $\mathcal{S}_{x_0,y_0} = 1.0$, is the *default privacy* region—the user can claim to be anywhere in that region and yet retrieve the same result set as she would do by using her precise coordinates. The size of this default region is a characteristic feature of the distribution of the points in the set $\mathcal{P}$ across the grid.

The service-contour of a query reveals the regions where a certain percentage of the top-k results is retained. Given a certain requirement on the fraction of results that must be retained (i.e. the utility that must be maintained), the area of the corresponding region is a measure of the privacy achievable by the user,

since a query originating from any point in the region will return a result set with the desired utility. The user can calculate these regions for any level of utility requirement, which in other words imply that an overall picture of the privacy/utility trade-offs is available to the user for decision making. Trading between service accuracy and location inaccuracy is then a question of choosing a point in one of the demarcated regions.

Unfortunately, the user device cannot compute $\mathcal{S}_{x_0,y_0}$ without access to $\mathcal{P}$, which resides at the LBS provider. The LBS cannot compute $\mathcal{S}_{x_0,y_0}$ since it requires access to the exact position $\langle x_0, y_0 \rangle$. The question we investigate is: what form of information can the LBS provide to the user to help infer the service-contour?

3.1.2 Service-contour inferencing

There exists a trivial solution to the raised question—push the set $\mathcal{P}$ and the ranking function $\mathcal{R}$ to the user, and perform the top-k ranking locally on the user device. As one can see, this solution clearly ignores underlying communication overheads and policies on sharing business intelligence. Note that the set $\mathcal{P}$ is not simply a collection of positions, but includes additional attributes about the businesses located at those positions. This could range from names, addresses, categories, subcategories, to specifics such as value, feedback scores, and entire profiles of individuals with personal information. The ranking function $\mathcal{R}$ is often a well-guarded business secret on how these attributes are combined. Another approach is to send a set of similarity matrices to the user, one each corresponding to a specific coordinate in the grid. The approach requires the computation and transfer of an inordinate amount of information ($O(r^2c^2)$). Given a geographic area, our objective is to restrict the transfer of information to a bounded size, or $O(1)$. The service-contour inferencing problem is then defined as follows.

Service-contour inferencing: *Give a set of points $\mathcal{P}$ on a geographic area (represented as a $r \times c$ grid), a ranking function $\mathcal{R}$, and a similarity function Sim, find functions* $\mathbf{Enc}$ *and* $\mathbf{Dec}$ *such that*

1) *output $\mathcal{T} = \mathbf{Enc}(\mathcal{P}, \mathcal{R}, Sim)$ is $O(1)$ in size, and*
2) *assuming $\mathcal{S}'_{x,y} = \mathbf{Dec}(\mathcal{T}, \langle x, y \rangle)$, with $\langle x, y \rangle$ being any point on the grid, we have $\mathcal{S}'_{x,y} = \mathcal{S}_{x,y}$.*

3.1.3 Approximate inferencing

Without the bounded size constraint, the service-contour inferencing problem can be solved by computing the top-k results for each point in the grid, and then conveying an identification vector with respect to each point. An identification vector uniquely identifies the k results corresponding to a point. The service-contour can then be exactly generated. This is an attractive choice provided the communication overhead is not exceedingly high. Note that the top-k results induce a set of order k Voronoi regions [14], [15], [16], each region sharing a certain result set. Therefore, the information to be conveyed may be highly compressible. We shall use the communication overhead of this method as a benchmark in the experimental analysis.

$\mathcal{V}$:

V_1	{a,b,c,d,e}
V_2	{a,b,c,f,g}
V_3	{f,g,h,i,j}

$\mathcal{V}_{Sim}$:

	V_1	V_2	V_3
V_1	1	0.6	0
V_2	0.6	1	0.4
V_3	0	0.4	1

$\mathcal{I}$:

1	1	1	2	2
1	1	1	2	2
1	1	2	2	2
1	1	2	3	3
3	3	3	3	3

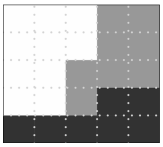

Figure 3. Set $\mathcal{V}$ shows hypothetical top-5 result sets on a 5×5 grid. $\mathcal{I}$ depicts which result set is applicable at a point. $\mathcal{V}_{Sim}$ shows pairwise similarity of the 3 unique result sets for the grid. The image is a compact representation of $\mathcal{I}$ and $\mathcal{V}_{Sim}$ —grey color codes used are: *1-white=1.0*, *2-grey=0.6* and *3-black=0.0*.

Consider a hypothetical scenario where the top-k results corresponding to a point can be represented by one of V symbols. Further, a maximum entropy condition is achieved under arbitrary distribution of the points in $\mathcal{P}$ across the grid. Therefore, each symbol is equi-probable ($1/V$). Under this setting, no lossless compression of the symbol sequence describing the top-k results across the grid can achieve a compression level better than $\log_2 V$ bits per point, i.e. $rc\log_2 V$ bits for $\mathcal{T}$. Assuming a 320×320 grid on a $32 \times 32km^2$ area (a point then resembles a $100m \times 100m$ area), and $V = 1000$ unique top-k result sets generated for the points in this area, this number is around $124.5KB$. While this is not a large data transfer in itself, repeated querying will result in an accumulated overhead that is a significant fraction of typical bandwidth limitations. We seek algorithms that can avoid such a communication overhead (even in the worst case); however, provide a good approximation of $\mathcal{S}_{x,y}$. Note that this observation assumes a worst case scenario and only pertains to the ability to correctly determine if two points have different (or the same) result sets. Computing the similarity would involve encoding additional identifier data corresponding to every set.

3.2 Privacy-supported local search

The crucial piece of information to infer the service-contour is the similarity measure Sim that tells the percentage overlap in the result sets from two points. Given that the top-k result sets (the output of $\mathcal{R}$) do not always change as one moves from one point to the next, the same calculation is performed (operates on same data) by Sim for most pairs of points. Let us denote by $\mathcal{V}$ the set of distinct outputs of $\mathcal{R}$ for the points of the grid, i.e. $\mathcal{V} = \{\mathcal{R}_k(\mathcal{P}, \langle x, y \rangle) | 1 \leq x \leq c, 1 \leq y \leq r\}$. Note that the size of $\mathcal{V}$ is going to be comparatively smaller than the size of the grid. Let $\mathcal{V}_{Sim}$ be a matrix that denotes

the Sim values on pairs of elements of $\mathcal{V}$, i.e.

$$\mathcal{V}_{Sim}[i,j] = Sim(V_i, V_j),\ V_i, V_j \in \mathcal{V}.$$

Next, we define a $r \times c$ index matrix $\mathcal{I}$ such that $\mathcal{I}[i,j] = t$ implies $\mathcal{R}_k(\mathcal{P}, \langle i,j \rangle) = V_t$, where V_t is a member of $\mathcal{V}$. Fig. 3 captures the relationship between $\mathcal{V}, \mathcal{V}_{Sim}$ and $\mathcal{I}$. In the same figure, we also see another representation of the three sets in the form of a 5×5 pixel image. The color of each pixel is indicative of points having the same value in $\mathcal{I}$. In addition, the similarity measure, as computed in $\mathcal{V}_{Sim}$, can be inferred from the shades of the colors.

$$Sim(\langle x,y \rangle, \langle x',y' \rangle) = 1 - |color(x,y) - color(x',y')|$$

For example, the result set similarity between the points $\langle 3,3 \rangle$ and $\langle 5,5 \rangle$ is $\mathcal{V}_{Sim}[2,3] = 0.4$, which can also be derived as $1 - |0.6 - 0.0|$. The advantage here is that the similarity information is conveyed without the need to communicate $\mathcal{V}$. The representation is rather straightforward in this example, but need not be so for arbitrary $\mathcal{V}, \mathcal{V}_{Sim}$ and $\mathcal{I}$.

3.2.1 Multi-dimensional scaling

The example above involves determining three greyscale color codes (values in $[0,1]$) such that the Euclidean distance between two values is proportional to the similarity measurements given by $\mathcal{V}_{Sim}$. The objective is not different when $\mathcal{V}_{Sim}$ has a significantly more number of entries. We adopt the classical method of multi-dimensional scaling at this step. The multi-dimensional scaling problem is stated as follows for the problem at hand.

Multi-dimensional scaling: *Given a set of top-k result sets $\mathcal{V} = \{V_1, V_2, ..., V_n\}$ and a similarity matrix $\mathcal{V}_{Sim}$, obtain a set of n m-dimensional vectors $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n$ that minimizes*

$$\sum_{i<j} \left(\text{Euc}\left(\mathbf{c}_i, \mathbf{c}_j\right) - \left(1 - \mathcal{V}_{Sim}[i,j]\right)\right)^2.$$

Euc is the Euclidean distance function. The scaling happens from a k-dimensional space to a m-dimensional space. For the case when a minimum value of zero exists (and is found), the Euclidean distance between any two vectors $\mathbf{c}_i$ and $\mathbf{c}_j$ is equal to the dissimilarity between two result sets V_i and V_j. Such distance preserving embedding of high dimensional data is readily useful for data visualization. Numerical solvers for a multi-dimensional scaling problem are included in most statistical packages. We use the implementation provided in the cmdscale function of the R statistical package. The implementation follows the analysis of Mardia [17]. We use a value of $m = 3$ since it allows one to graphically visualize the similarity trend in the form of a RGB color image. Higher values of m allow for the possibility of better distance preservation, but results in a larger encoded size.

The **Enc** function based on 3-dimensional scaling then operates as follows: each component of the $\mathbf{c}_i$ vectors are normalized to the $[0,1]$ interval, and a $r \times c$ pixel image is created with the RGB color of pixel (i,j) set to $\mathbf{c}_{\mathcal{I}[i,j]}$. This image is the output $\mathcal{T}$ produced by the **Enc** function and communicated to the user. Although a vector $\mathbf{c}_i$ can take infinite values in $[0,1]^3$, the number of possibilities reduce to 16.7 million due to the color mapping. Fig. 1 in Appendix A (see supplementary file) illustrates an example image created by **Enc** for 10-nearest Starbucks coffee shop locations in the city of Los Angeles, CA (1024 square kilometers area centered around Los Angeles City Hall).

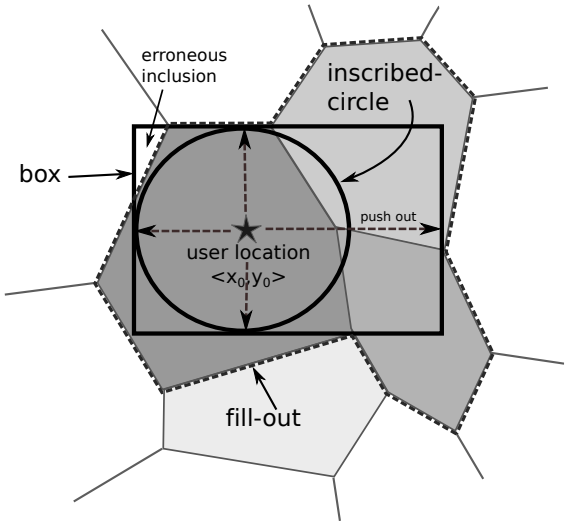

Figure 4. Heuristics for service-contour inferencing. Shaded regions depict true areas with a given service similarity. Output of fill-out is shown as a dashed-line around the determined area.

3.2.2 Inferring the service-contour

In order to retrieve the service-contour from $\mathcal{T}$, the **Dec** function uses the location of the user $\langle x_0, y_0 \rangle$ as a point of reference for similarity comparison. Let $\mathcal{T}_{x,y}$ be the RGB color vector at the (x,y) pixel in $\mathcal{T}$. The Euclidean distance between $\mathcal{T}_{x_0,y_0}$ and the color vector $\mathcal{T}_{i,j}$ of any other pixel (i,j) (a point in the grid) attempts to closely estimate the dissimilarity measure—the similarity estimate then being $\mathcal{S}'_{x_0,y_0}[i,j] = 1 - \text{Euc}(\mathcal{T}_{x_0,y_0}, \mathcal{T}_{i,j})$. The **Dec** function then simply computes this estimate for all possible points $\langle i,j \rangle$ in the grid. Computation of the service-contour can also be parameterized by a threshold δ such that points in the grid with a similarity estimate higher or equal to δ are the only ones identified. To do so, one can begin at point $\langle x_0, y_0 \rangle$ and continue to explore neighboring points as long as the similarity estimate satisfies the threshold. We explore three fast heuristics in order to avoid a point by point generation of the service-contour. Fig. 4 illustrates the difference between them.

Box: Starting from the user location $\langle x_0, y_0 \rangle$, a box is grown by pushing the four edges outward (in clockwise order), one point-step at a time. Edge pushing along a direction is stopped whenever doing so will result in the

inclusion of a point with similarity estimate less than δ.

Inscribed-circle: Box-expansion tends to cover inaccurate points (those outside the threshold) in the corner areas, specially when similarity estimates are not exact. A circular region inscribed in the box, centered at $\langle x_0, y_0 \rangle$, eliminates such errors on the corners of the box.

Fill-out: While an inscribed-circle is good at reducing the error in some cases, it cannot cover irregular shaped regions within the δ threshold. The fill-out method expands the circular region by including neighboring points that has the same color vectors as points within the inscribed-circle.

An interactive process of inference would involve determining the service-contour for a given value of δ (say 90%), and then progressively growing it depending on the area of the region inferred at a certain threshold. We refrain from using methods based on computational geometry due to their higher processing requirements.

Note that we have excluded the possibility of a malicious server model in this scheme. A malicious server can manipulate the similarity data to create the impression that no two neighboring cells have the same result set. However, it would not be correct to state that such manipulations will force the user to reveal her precise location. The decision on whether a default privacy region is sufficiently large enough is user-driven. A distorted picture of the similarity profile may in fact drive the user to believe that no reasonable privacy can be achieved in the application, and thereby discontinue using it. In another case, a privacy-aware user may still pick a location from a larger area, i.e. trade accuracy (although based on distorted information) for privacy. Hence, even after a malicious server manipulates the similarity matrix intelligently, it is not guaranteed that the location communicated by the user is true, or a consequence of the privacy/accuracy trade-off process. In addition, the server must also keep the user motivated to use the service. This in itself is much more difficult once the user observes discrepancies in the final query answers and the physical realities. A formal evaluation substantiating these arguments would be useful; otherwise distributed methods to share trust scores on service providers can be sought to identify malicious servers.

4 Empirical evaluation

The empirical evaluation is performed using the SimpleGeo Places dataset that contains information on more that 20 million places around the world, and distributed under the Creative Commons open license. The US part of the dataset has 12,993,248 entries, with data corresponding to multiple business categories and sub-categories. Entries are maintained in the GeoJSON format, and includes attributes such as name, latitude/longitude, address, phone numbers, classifiers (category, type, subcategory) and tags. In our study, a place is considered a match for the search keyword if it includes the keyword in any of these attributes, and the city matches the city attribute. The evaluation is performed for the four largest cities in USA—Los Angeles, Houston, Chicago and New York. One of the factors influencing the top-k results is the number of objects returned by a query, and their distribution around the query point. The existence of a large number of objects implies that the top-k results are likely to change for small changes in location. For objects that are low in density, large variations in the location are possible without changing the result set. This behavior can be reasonably assumed irrespective of the density of users in the city. Therefore, we choose large cities where we can obtain different densities of objects, specially ones with high densities. Objects that are high in density in large cities may not be so in a smaller city. Hence, we believe that a comprehensive evaluation can be performed by considering these large cities.

For each city, a $1024km^2$ area is used as the high-level generalization $\mathcal{G}$ to generate the similarity profile. A 320×320 cells grid is superimposed on this area. Each cell then reflects a $100m \times 100m$ area. This approach implicitly assumes that positioning a user in a cell is equivalent to exactly locating her. For Los Angeles and Houston, the city center is at the center of this grid ($\langle 160, 160 \rangle$). For Chicago and New York, the city centers are at $\langle 288, 160 \rangle$ and $\langle 32, 160 \rangle$ respectively. The geographic co-ordinates are provided in Appendix A. Euclidean distance based nearest neighbor is used as the ranking function, with $k = 10$. We employ the cover tree algorithm by Beygelzimer et al. [18] to determine the 10 nearest query matches with respect to a point on the grid.

Instead of experimenting with a large corpus of search keywords, we generalize the notion of query points into low, medium and high density objects. Low density objects result from targeted queries, with frequencies ranging from 10 to 50 within the grid. Queries resulting in 50 to 200 objects are considered medium density, while frequencies higher than that are considered high density. We were able to generate low density objects by using search terms such as "bowling", "electronics store" and local grocery store names in the cities. Medium density objects are generated from search terms such as "starbucks coffee" and "police". High density objects are generated by heavily generic terms such as "atm" and "gas station." For the high density case, frequencies were often observed to be in the range of 400 to 900. The search keyword itself does not hold much importance for this study, but is used to retrieve query point distributions that reflect the real world. The results below combine performance measures irrespective of what search term produced them, the only distinction being made is with respect to the density.

4.1 Evaluation process

Performance of the $\mathbf{Enc}$ and $\mathbf{Dec}$ functions are measured using *precision* and *recall* metrics. Given a threshold δ, we arrive at a set of points Z on the grid that the user can

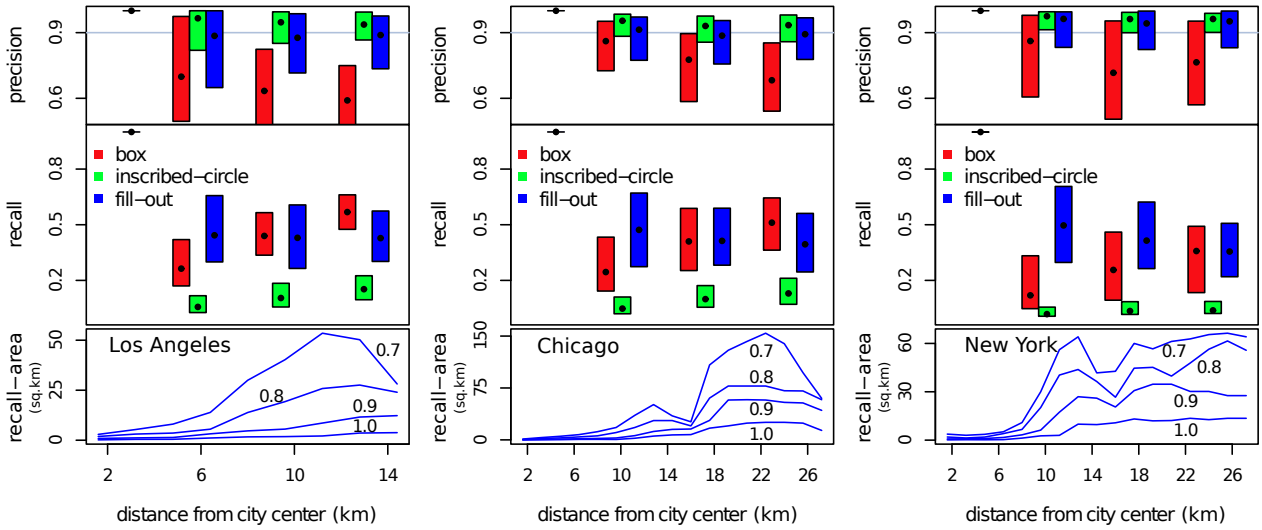

Figure 5. Precision and recall when searching for "starbucks coffee" in a given city. Each plot shows performance of fill-out for $\delta = 1.0$ (leftmost) and then three sets of rectangles, one each for $\delta = 0.9, 0.8$ and 0.7 (from left to right). Lower edge of a rectangle represents 10^{th} percentile, upper edge represents the median (50^{th} percentile), and the dot represents 25^{th} percentile. Also shown is the area recalled (in km^2) by the fill-out heuristic as a user moves away (distance in km) from the city center. Trend lines are marked with the corresponding δ value.

use to perturb her location. Depending on the accuracy of maintaing similarities, and the subsequent estimation by the three heuristics, this set of points may be over or underestimated. If Z_{true} is the true set of points satisfying the threshold, then the precision is given as the fraction of points in Z that are also in Z_{true}. Recall is the percentage of points in Z_{true} that are also in Z.

$$Precision = \frac{|Z \cap Z_{true}|}{|Z|}; \qquad Recall = \frac{|Z \cap Z_{true}|}{|Z_{true}|}$$

Precision can be viewed as the probability that the service similarity guarantee (within the threshold) is not violated. Recall measures the ability to identify the areas where a certain level of service similarity is guaranteed. While precision can be viewed as a measure of the quality of service, the absolute recalled area ($|Z \cap Z_{true}|$) is the size of the geographic region where the user can hide herself, and yet retrieve true query results (within the threshold). In other words, the recall-area may be viewed as a measure of the privacy level obtained by the user.

Experiments are performed for four service similarity thresholds: $\delta = 1.0, 0.9, 0.8$ and 0.7. For each value, precision and recall are calculated for the three heuristics using a sample of points as the user location $\langle x_0, y_0 \rangle$ on the grid. The sample consists of 1521 points uniformly distributed on the grid—a sample point every $800m$ ($0.5mi$) along the horizontal and vertical directions. For $\delta = 1.0$, results are only reported for the fill-out heuristic.

4.2 The case of "starbucks coffee"

The case of locally searching a coffee shop—e.g. "starbucks coffee"—often comes up in location privacy discussions. We present the detailed comparative results with respect to a privacy-aware user trying to find the nearest Starbucks coffee shop location. Fig. 5 and Fig. 6 show the comparative efficiency of the three heuristics in the four cities. For each city, the precision and recall plots show the performance of fill-out for $\delta = 1.0$ (leftmost) and then three sets of rectangles, one each for $\delta = 0.9, 0.8$ and 0.7 (from left to right). A precision and recall of 1.0 for fill-out at $\delta = 1.0$ implies that a privacy-indifferent user does not lose any accuracy in the result set as a result of the process. In addition, the heuristic exactly reveals the default privacy region with respect to the issued query. For the other δ values, each rectangle shows the 10^{th} percentile (lower edge), 25^{th} percentile (center dot) and 50^{th} percentile (upper edge) of the computed precision and recall values. Recall that the p^{th} percentile is the value below which p percentage of the observations lie. The inscribed-circle and fill-out heuristics guarantee 90% or more precision for 75% (25^{th} percentile) of the points sampled on the grid (possible user locations), across the four cities. This is observed irrespective of the service similarity requirement imposed by a user. Precision for the box heuristic is comparatively worse because of its tendency towards erroneous inclusion of points. As expected, inscribed-circle clearly improves upon this, but results in an extensive pruning of the identified regions (poor recall). It is not difficult to create a heuristic with high precision; however, the desirable one has high recall as well.

Fill-out improves upon the recall of inscribed-circle without heavily degrading the precision. However, the recall values themselves are all below 50%. The bottom of each plot shows trend lines depicting how the area recalled ($|Z \cap Z_{true}|$ in km^2) by the fill-out heuristic changes as a user moves away from the city center. The

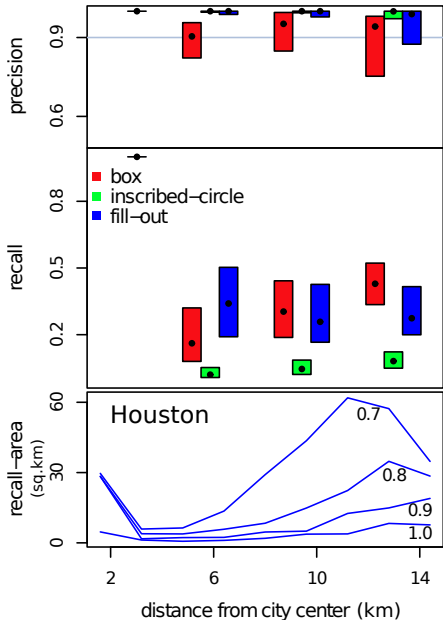

Figure 6. Precision and recall when searching for "starbucks coffee" in the city of Houston, Texas. See Fig. 5 caption for details.

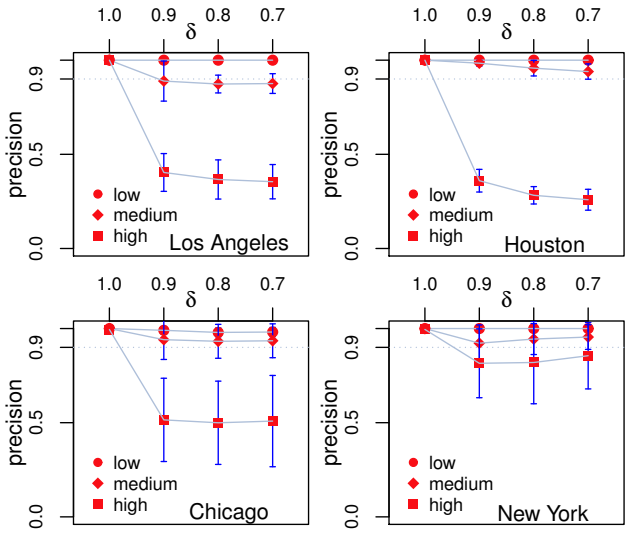

Figure 7. Precision of fill-out heuristic for different service similarity thresholds ($\delta = 0.7, 0.8, 0.9, 1.0$) and object densities (*low,medium,high*). Vertical bar shows one-standard-deviation.

query object ("starbucks coffee") has a relatively higher concentration near the city center areas. The trend line for $\delta = 1.0$ (for which fill-out has 100% recall) indicates that the default privacy region may not be significantly large when query objects are concentrated. However, areas as large as 20-40 km^2 become available within $8km$ ($\sim 5mi$) of the city center, provided one or two incorrect results are acceptable. This is despite the poor recall of the heuristic. These areas will presumably be large enough for a privacy-conscious user, given that the observations hold more strongly for regions that see lesser crowd. Note that changing the service accuracy requirement further down can expand the determined area. Object locations in this case, although not the nearest ones, will not be unrealistically far away.

4.3 Precision/recall trends

The precision and recall trends we observe for the case of "starbucks coffee" are repeated for the other medium density experiment (derived using the keyword "police"). For the fill-out heuristic, Fig. 7 shows the mean (across the search keywords) of the 25^{th} percentiles of the precision scores for different object densities. Full precision for low density objects is almost guaranteed, irrespective of the service accuracy threshold. However, the approach has difficulty maintaining those same values for high density objects. High density objects are often located close to each other, thereby creating a scenario where moving small distances significantly changes the result set. It also means that finding such objects is not difficult in the real world. Note that the density designation is not based on what is being queried—a "gas station" could be a high density object in parts of a city, and low/medium in others. In the latter case, when finding one could become difficult by simply looking around, local search is possible in a privacy-supportive manner. The ranking function is also a crucial component in deciding the density of objects. For instance, a ranking function that accounts for local reviews of restaurants while making suggestions, will result in a low density categorization for the keyword "restaurants", meaning the top-k result set does not change significantly even for a high concentration of restaurants in the area.

The recalled area is also significantly large for low density objects, occasionally dropping when clusters of such objects are found. Fig. 8 depicts this drop for the cities of Chicago and New York. The observation reinstates the fact that object densities can be locally high. The conclusions made in the "starbucks coffee" case remains applicable in general to the recalled area for medium density objects. Refer to Section 3 in the supplementary file for results on the communication overhead associated with the proposed methodology.

4.4 Conclusions

Based on the observations from the empirical study, we make the following conclusions on the efficacy of a privacy-supportive local search application.

Precise geo-locations are necessary for result set accuracy when the queried objects exist as a dense cluster in the search area. It seems unlikely that both location privacy and result exactness can be maintained in this case. A privacy-supportive application would allow the user to aggressively trade-off the service similarity requirement to determine a sufficiently large area for

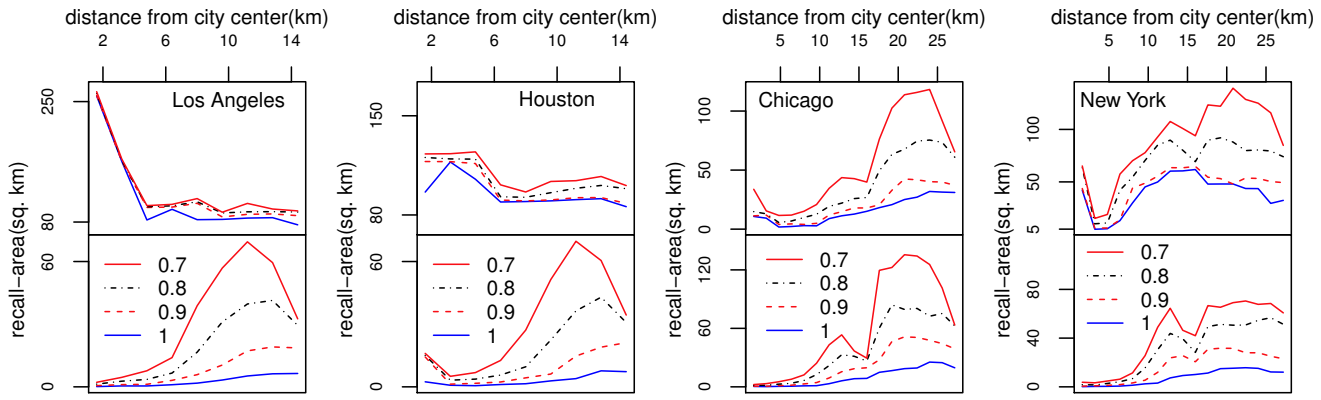

Figure 8. Area (km^2) recalled by the fill-out heuristic for different service similarity thresholds ($\delta = 0.7, 0.8, 0.9, 1.0$), as user moves away (distance in km) from city center. Top plots are for low density objects and bottom plots for medium density objects.

location perturbation. Given the high density of objects, resulting objects can still be expected to be in the near vicinity.

When object density is not dense, location accuracy has a minor role to play in retrieving relevant results. A privacy-supportive application would help identify the large default-privacy regions resulting in such situations.

Next generation telecommunication systems could very well make it possible to quickly (and cost-effectively) transfer all information required to infer the service-contour exactly. Until then, approximate inferencing algorithms can be used to reduce the communication overhead.

5 Summary

In this paper, we proposed a novel architecture to help identify privacy and utility trade-offs in a location-based service. The architecture has a user-centric design that delays the sharing of a location coordinate until the user has evaluated the impact of its accuracy on the service quality. Using the prototypical example of a local search application, we showed the form of information that can be exchanged between the user and the provider to enable a privacy-supportive LBS. Section 4 of the supplementary file suggests some future directions of research for this work.

References

[1] J. Sythoff and J. Morrison, *Location-Based Services: Market Forecast, 2011-2015*. Pyramid Research, 2011.

[2] P. Golle and K. Partridge, "On the Anonymity of Home/Work Location Pairs," in *Proceedings of the 7th International Conference on Pervasive Computing*, 2009, pp. 390–397.

[3] H. Zang and J. Bolot, "Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, 2011, pp. 145–156.

[4] M. Duckham and L. Kulik, "A Formal Model of Obfuscation and Negotiation for Location Privacy," in *Proceedings of the 3rd International Conference on Pervasive Computing*, 2005, pp. 152–170.

[5] 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*, 2005, pp. 88–97.

[6] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving User Location Privacy in Mobile Data Management Infrastructures," in *Proceedings of the 6th Workshop on Privacy Enhancing Technologies*, 2006, pp. 393–412.

[7] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services," in *Proceedings of the 24th International Conference on Data Engineering*, 2008, pp. 366–375.

[8] 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*, 2003, pp. 31–42.

[9] B. Gedik and L. Liu, "Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[10] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[11] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems," in *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 371–380.

[12] 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*, vol. 19, no. 12, pp. 1719–1733, 2007.

[13] G. Ghinita, K. Zhao, D. Papadias, and P. Kalnis, "A Reciprocal Framework for Spatial k-Anonymity," *Journal of Information Systems*, vol. 35, no. 3, pp. 299–314, 2010.

[14] P. K. Agarwal, M. de Berg, J. Matousek, and O. Schwarzkopf, "Constructing Levels in Arrangements and Higher Order Voronoi Diagrams," in *Proceedings of the 10th Annual Symposium on Computational Geometry*, 1994, pp. 67–75.

[15] F. Aurenhammer and O. Schwarzkopf, "A Simple On-line Randomized Incremental Algorithm for Computing Higher Order Voronoi Diagrams," in *Proceedings of the 7th Annual Symposium on Computational Geometry*, 1991, pp. 142–151.

[16] D.-T. Lee, "On k-Nearest Neighbor Voronoi Diagrams in the Plane," *IEEE Transactions on Computers*, vol. C-31, no. 6, pp. 478–487, 1982.

[17] K. V. Mardia, "Some Properties of Classical Multidimensional Scaling," *Communications on Statistics – Theory and Methods*, vol. A, no. 7, pp. 1233–1241, 1978.

[18] A. Beygelzimer, S. Kakade, and J. Langford, "Cover Trees for Nearest Neighbor," in *Proceedings of the Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 97–104.

Supplement: Exploiting Service Similarity for Privacy in Location Based Search Queries

Rinku Dewri, *Member, IEEE*, and Ramakrisha Thurimella

1 Additional related work

Location privacy preservation has received significant interests over the past decade, both across policy makers and academic researchers. Legislative enforcements to preserve location privacy dates back to the United State's Communication Act of 1934, wherein "Section 222 requires telecommunications carriers to provide confidentiality for customer information as proprietary information of another common carrier." Disclosure is only allowed during emergency situations, or with permissions of the customer. Efforts are ongoing to enforce more specific laws related to geolocation information tracking and sharing (e.g. Location Privacy Act of 2011, currently in the first step of the legislative process). However, laws are often regional—while policies in the European Union may require every user to consent to location sharing, a policy in the United States may require users to *opt-out* of a default sharing. Nonetheless, the important question that still remains open is whether a user can derive any reasonable utility out of the location-based service and still protect her location information?

Multiple suggestions are available on how a cloaking region should be formed. Bamba et al. enforce a location l-diversity requirement in addition to k-anonymity, where the number of still-object counts must also be above a user-specified threshold [1]. Liu et al. propose that a minimum level of entropy should also be maintained in the queries originating from the cloaking region [2]. Dewri et al. have extended these concepts to the case of continuous services [3], [4]. Shin et al. introduce profile anonymization in cloaking regions, wherein at least $k-1$ other users with the same profile (denoted by a vector) as the request issuer is present [5]. Riboni et al. make a similar argument, but in the context of service parameters. Inferences that can be drawn based on these parameters are avoided by smoothing the differences among the distribution of the parameters in requests from different cloaking regions [6].

A mix zone model is presented for location privacy by Beresford and Stajano [7]. The objective of mix zones is to prevent tracking of long-term user movements, while short-term revelation of location data is permissible. A trusted middleware usually mixes the identities of users in specific zones, thereby preventing continuous tracking. Extensions of this technique are proposed for the scenario where user movements are constrained to road networks [8].

Mokbel et al. explore query processing of different types on spatial regions – private queries over public data, public queries over private data, and private queries over private data [9]. Their effort is directed towards facilitating different query formats using cloaking regions. Lee et al. explore privacy concerns in path queries where source and destination inputs may reveal personal information about users [10]. They propose the notion of obfuscated path queries where multiple sources and destinations are specified to hide the true inputs. Although we do not focus on continuous location-based services in this work, it is worth noting that certain locations (home or work places) reveal more information about a user. Hence, the privacy expectations are also bound to be different when users are at such locations. Historical location data is used by Xu and Cai in a variant of location k-anonymity, where the cloaking region is required to have at least k different footprints [11]. In a later work, the authors argue that the impact of a privacy parameter, such as k, on the level of privacy is often difficult to perceive. Hence, they treat privacy as a feeling-based property and propose using the popularity of a public region as the privacy level [12]. Each user specifies a spatial region as her privacy index, and the cloaking region for the user must at least have the same popularity as that of the specified region. An entropy based computation is used to define the popularity of a spatial region. Soriano et al. show that the privacy assurances of this model do not hold when the adversary possesses footprint knowledge on the spatial regions over time [13]. Shokri et al. propose a framework to quantify location privacy based on the expected estimation error of an adversary [14]. This work provides 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 show that measures such as the anonymity set size or entropy, do not correctly quantify the privacy enforced by the method [15].

• *R. Dewri and R. Thurimella are with the Department of Computer Science, University of Denver, CO 80208, USA. Email:{rdewri,ramki}@cs.du.edu.*

Table 1
Minimum area (km^2) in which local search results (10 nearest neighbors) are same for a given percentage of the continental United States landmass. Value in parenthesis shows minimum area that shares 9 out of the 10 results.

keyword	90%	75%	50%	25%	10%
atm	1 (2.4)	1 (6.4)	2 (20.8)	6 (73)	21.4 (273.2)
bus station	1 (6)	1 (15.8)	4 (60.9)	16.4 (229.4)	61.4 (721.12)
cafe	1 (2)	1 (6)	2 (20.3)	6 (85.3)	24 (267.1)
car rental	1 (3)	1 (8)	2 (28.7)	8 (117.95)	34.88 (450.66)
gas station	1 (2)	1 (5)	2 (17)	5 (59.8)	18.3 (208.5)
hospital	1 (2)	1 (6)	2 (18.4)	5 (69.2)	20.4 (297.32)
library	1 (5.9)	1 (14)	3.9 (54.2)	11.9 (152.88)	38.5 (409.8)
lodging	1 (8)	1 (22.2)	4.6 (83.2)	18.6 (301.65)	74 (887)
night club	1 (4)	1 (12.5)	3.5 (62)	15 (257.25)	65.5 (891.4)
parking	1 (5)	1 (15.2)	3.6 (50.8)	13.15 (206.7)	60.2 (632.6)
pharmacy	1 (2)	1 (6)	2 (18.2)	6 (73.5)	23.1 (288)
police	1 (6.9)	1 (17)	3.9 (55)	12.8 (167.6)	44.3 (438.08)

Data transformation is another method to prevent the inference of locations. Agrawal et al. propose an encryption technique called OPES (Order Preserving Encryption Scheme) that allows comparison operations to be directly applied on encrypted data [16]. Operand decryption is however required for computing SUM and AVG. Wong et al. overcome this drawback by developing an asymmetric scalar-product preserving encryption [17]. This allows the preservation of relative distances between database points. Khoshgozaran et al. employ Hilbert curves to transform the data points and then answer queries in the transformed space [18]. The parameters of the transformation, called the Space Decryption Key, is assumed to be not known to an adversary. A new paradigm in location privacy is based on private information retrieval (PIR) techniques. Khoshgozaran et al. propose K nearest neighbor queries that can be reduced to a set of PIR block retrievals [19]. 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. Papadopoulos et al. further warrant the need to retrieve the same number of blocks across queries [20]. While the use of PIR techniques in providing location privacy is an interesting direction to explore, computational inefficiency or the dependence on additional hardware makes these approaches currently unsuitable for mainstream adoption.

2 A MOTIVATING STUDY

The literature reviewed in this work highlights the efforts of the academic community to prevent the sharing of "pure" location information. An universal assumption in most of these methods is that the user, by default, is unwilling to share her location, irrespective of the service-level impacts. One can argue that a user willing to do so will simply avoid using the privacy-preserving transformation. It is our opinion that individuals do not view privacy as an immutable property, but rather as a personal yet adaptable element. For instance, while a mobile user may keep her GPS device turned off most of the time, she may occasionally turn it on to achieve Shopkick™ (www.shopkick.com) rewards when visiting a departmental store. This user's perspective on location privacy is guided by prospective gains from revealing her location. As another example, a user may precisely reveal her location (irrespective of its sensitivity) while looking for nearby emergency care centers; the same user may not be willing to do so while getting a listing of nearby local businesses. This user's perspective on location privacy is requirement driven, depending on the assessed (personally) importance of location sensitivity and service usefulness. We performed an empirical study to determine if a location-based search application can generate any utility to an extreme user (always paranoid about revealing current location) in this latter category.

Consider a grid of cells, each 1000x1000m^2, across the continental United States landmass. An individual located at any of these cells issues a local search query that retrieves the 10 nearest businesses matching the search term. Table 1 lists, for a given percentage of cells in the grid where the individual could be located, the number of other cells that would receive the same query answer as received by this individual. The values in parenthesis indicate the number of other cells that would retrieve at least 9 out of the 10 businesses retrieved by the individual. In the context of the paranoid user, this data highlights that, for most places that the user could be located (say 75% of the landmass), she has the freedom to use a location coordinate anywhere in an area of size *at least* 6.4 km^2 and still retrieve 9 out of the 10 nearest ATMs. The statistics can be different depending on the actual search term issued by the user. In addition, an area of 6.4 km^2 may still not be comforting enough for the user. A possibility then is to consider an area that guarantees a 8 out of 10 match. This process presents an adaptive mechanism for the user who can choose to trade-off location accuracy at the expense of service accuracy, or vice versa. The challenge however lies in the fact that the user does not necessarily have the requisite resources (both in terms of computation and data) to compute these areas. On the other hand, the LBS provider that performs the local search has the

computational and data resources to compute the area boundaries, provided it can accurately (and quickly) convey the information without requiring the user to reveal her location.

3 Communication Overhead

The communication overhead is measured by the amount of information that is to be transferred to the user to infer the service-contour. The baseline for our comparison is the size of the sets $\mathcal{I}$ and $\mathcal{V}$, after compression using the DEFLATE algorithm. The object identifiers for elements in $\mathcal{V}$ are obtained from the unique identifiers assigned by SimpleGeo in its database. The data transferred when using the **Enc** function includes the compressed version of the set $\mathcal{T}$.

For low density objects, the transferred data has a 35% reduction in size from that of the baseline data. Although the absolute size of the baseline data is in the range of 5 to 10 KB, the impact of the improvement is seen when aggregated over a number of queries. The reduction factor (transferred data size over baseline data size) is rather varying for medium density objects—values ranging from 0.8 to 0.15 in some cases. Absolute values for the baseline data are observed to be in the range of 25 to 150 KB. The critical factor contributing to the difference in size is the set $\mathcal{V}$, which in turn depends on the number of distinct result sets that can be obtained within a geographic area.

4 Future Directions

One of the assumptions we made in the empirical study is that the rank order of the top-k results is not important. Without this assumption, the (dis)similarity measurement will have to be redefined to include disagreements in the result ordering. Higher utility will be maintained if the result objects that are the closest to the user are indeed retrieved by the mechanism.

For a continuous query LBS model, the policy that determines the final choice of the location must also induce realistic correlations between subsequent locations. This would involve analyzing the current service-contour from multiple reference points, in an effort to generate a reasonable trajectory of future locations. The difficulty appears because of the possibility of dynamic updates to the objects data base. Additional directions include reducing the communication overhead, efficiently solving the service-contour inferencing problem for a moving objects data base, augmenting the inference process with clear privacy policy descriptions, and integrating application sensitivity into the decision making process.

Dynamic updates in our application environment can occur by addition/deletion of objects. These updates can happen in the background, and the query processor can have access to the updated database as soon as the update operations are complete. We note that, as a result of this process, the query performance will not degrade, although stale results may be generated for a brief period. If the update time is not significant, a locking mechanism can be enforced to guarantee result validity. It is also important to note that frequent updates to POI databases are not likely. In this work, we did not consider the possibility of mobile POI points (for example, as in a friend finder service where the searched objects are also mobile). We believe that the case of mobile POIs needs an extensive and a formal study in its own, since the locations of the moving objects may be sensitive information. In such a case, obtaining the service-contour is not as straightforward as in the case of a local search.

Appendix A

Table 2
City center co-ordinates used in the empirical study.

City	Latitude	Longitude
Los Angeles	34.053691^0N	118.243126^0W
Houston	29.760177^0N	95.3692910^0W
Chicago	41.87045^0N	87.629905^0W
New York	40.713256^0N	74.005905^0W

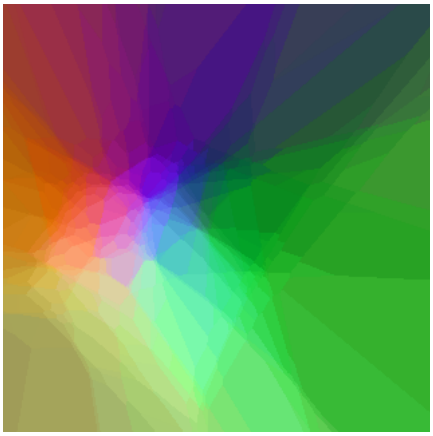

Figure 1. Output $\mathcal{T}$ of **Enc** function based on multi-dimensional scaling for a query involving "starbucks coffee" as the search term in the city of Los Angeles, CA. The ranking function is 10-nearest-neighbors. Note: color variations are lost in greyscale viewing.

References

[1] 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*, 2008, pp. 237–246.

[2] 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*, 2009, pp. 436–442.

[3] R. Dewri, I. Ray, I. Ray, and D. Whitley, "On the Formation of Historically k-Anonymous Anonymity Sets in a Continuous LBS," in *6th International ICST Conference on Security and Privacy in Communication Networks*, 2010, pp. 71–88.

[4] ——, "Query m-Invariance: Preventing Query Disclosures in Continuous Location-Based Services," in *Proceedings of the 11th International Conference on Mobile Data Management*, 2010, pp. 95–104.

[5] H. Shin, J. Vaidya, and V. Atluri, "A Profile Anonymization Model for Location Based Services," *Journal of Computer Security*, vol. 19, no. 5, pp. 795–833, 2011.

[6] C. B. D. Riboni, L. Pareschi and S. Jajodia, "Preserving Anonymity of Recurrent Location-Based Queries," in *Proceedings of the 16th International Symposium on Temporal Representation and Reasoning*, 2009.

[7] A. R. Beresford and F. Stajano, "Mix Zones: User Privacy in Location-Aware Services," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp. 127–131.

[8] B. Palanisamy and L. Liu, "MobiMix: Protecting Location Privacy with Mix-Zones Over Road Networks," in *Proceedings of the 27th International Conference on Data Engineering*, 2011, pp. 494–505.

[9] 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*, 2006, pp. 763–774.

[10] K. C. K. Lee, W.-C. Lee, H. V. Leong, and B. Zheng, "OPAQUE: Protecting Path Privacy in Directions Search," in *Proceedings of the 25th International Conference on Data Engineering*, 2009, pp. 1271–1274.

[11] T. Xu and Y. Cai, "Exploring Historical Location Data for Anonymity Preservation in Location-Based Services," in *IEEE INFOCOM 2008*, 2008, pp. 1220–1228.

[12] ——, "Feeling-Based Location Privacy Protection for Location-Based Services," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009, pp. 348–357.

[13] 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*, 2010, pp. 325–339.

[14] 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*, 2011, pp. 247–262.

[15] 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*, 2010, pp. 115–118.

[16] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order Preserving Encryption for Numeric Data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2004, pp. 563–574.

[17] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamouslis, "Secure kNN Computation on Encrypted Databases," in *Proceedings of the 35th SIGMOD International Conference on Management of Data*, 2009, pp. 139–152.

[18] A. Khoshgozaran and C. Shahabi, "Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy," in *Proceedings of the 10th International Conference on Advances in Spatial and Temporal Databases*, 2007, pp. 239–257.

[19] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, "Location Privacy: Going beyond k-Anonymity, Cloaking and Anonymizers," *Journal of Knowledge and Information Systems*, vol. 26, no. 3, pp. 435–465, 2011.

[20] S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest Neighbor Search with Strong Location Privacy," *VLDB Endowment*, vol. 3, no. 1-2, pp. 619–629, 2010.