Exploring Dependency for Query Privacy Protection in Location-based Services

Xihui Chen*
Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg
xihui.chen@uni.lu

Jun Pang
Faculty of Science, Technology and Communication, University of Luxembourg
jun.pang@uni.lu

ABSTRACT

Location-based services have been enduring a fast development for almost fifteen years. Due to the lack of proper privacy protection, especially in the early stage of the development, an enormous amount of user request records have been collected. This exposes potential threats to users' privacy as new contextual information can be extracted from such records. In this paper, we study *query dependency* which can be derived from users' request history, and investigate its impact on users' query privacy.

To achieve our goal, we present an approach to compute the probability for a user to issue a query, by taking into account both user's query dependency and observed requests. We propose new metrics incorporating query dependency for query privacy, and adapt spatial generalisation algorithms in the literature to generate requests satisfying users' privacy requirements expressed in the new metrics. Through experiments, we evaluate the impact of query dependency on query privacy and show that our proposed metrics and algorithms are effective and efficient for practical applications.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, measurement

Keywords

Location based services, dependency, query privacy, anonymity, metrics, generalisation algorithms

1. INTRODUCTION

Location-based services (LBSs) are services customised according to users' locations. In the last fifteen years, LBSs have endured a great growth, especially after GPS-enabled devices such as smartphones became popular. A location-based request contains the issuer's location and a *query* – the type of information of interest, e.g., where the nearest Chinese restaurants are. In spite of the great convenience brought to users' daily life, LBSs also lead to users' privacy concerns when they send LBS requests. In the literature, two major privacy concerns in LBSs have been studied – *location privacy* and *query privacy* [13] in terms of the types of sensitive information. The former is related to the disclosure of exact locations while query privacy, the focus of our paper, concerns the disclosure of queries.

The basic idea to protect users' query privacy in LBSs is to break the link between user identities and requests [3]. However, in the context of LBSs, removing or replacing identities with pseudonyms has been proved insufficient. With contextual information such as address books, users can still be identified as queries are usually made from fixed locations such as home or offices. In this case, users' spatial and temporal information can serve as *quasi-identifiers*. Anonymisation techniques from other research areas such as sensitive data release [13] are thus introduced, including k-anonymity and its different extensions (e.g., ℓ-diversity and t-closeness [18, 19]). Locations or time are replaced with regions or periods so that a certain number of users share the same quasi-identifier with the real issuer. The calculation of the regions or periods is termed as *generalisation* or *cloaking*. Since in practice LBS providers are usually required to offer immediate responses, throughout the paper, temporal generalisation is out of the scope. A request is called *generalised* if the location is generalised and the user identity is removed.

When the adversary has access to more contextual information, new privacy risks will emerge. For instance, "outlier" attacks are found on existing generalisation algorithms when their implementation is made public [21]. Privacy in LBSs related to context revealing has been recognised as *context-aware privacy* [22] and many types of contextual information have been studied in the literature. For example, Mascetti et al. [4] propose the concept of *historical k-anonymity* which protects against attacks where the adversary can learn a trace of requests issued by the same user.

Our motivations. The first generation of commercial LBSs were launched after the E911 mandate in 1996. From then on, LBSs have evolved from simple single-target finder to diverse, proactive and multi-target services [2]. However, user privacy did not receive fair treatments from the very beginning. This enables LBS providers to accumulate a large amount of users' historical requests. What makes the situation worse is the shift of LBS providers from telecom operators (who were viewed as trusted entities) to open businesses such as Google Latitude, Foursquare, and MyTracks. This increases the risk of potential misuse of the accumulated records.

*Supported by the National Research Fund, Luxembourg (SECLOC 794361).

In this paper, we investigate what the adversary can obtain from users' historical requests and how to protect users against potential privacy attacks.

Users tend to have preference in arranging their daily activities [12]. This leads to a repetitive pattern in their requests, e.g., dependency between queries. For instance, a user often posts a check-in of a coffee house after lunch. The fact that users' frequent queries are usually restricted to a small set makes the extraction of query dependency easier and more precise. Users' query dependency can be abused and becomes a potential risk to users' query privacy. We illustrate this by a simple example. Bob issues a request about the nearest night clubs in a 2-anonymous region with Alice being the other user. Suppose the adversary has also learnt that Alice just issued a query about the nearest clinics and Bob queried about bars. As it is not common to ask clubs after clinics compared to bars, the adversary can infer that Bob is more probable to issue the request about night clubs. In this example, even if Alice and Bob share a similar profile, the dependency between queries obviously breaks 2-anonymity for all users in the region who are supposed to be equally likely to issue the request. As far as we know, we are the first to explore *query dependency* for privacy protection in LBSs.

Our contributions. In this paper, we accomplish two tasks – to show the impact of query dependency on users' query privacy and to design new privacy protection mechanisms. For the first task, we model a user's query dependency as a Markov chain and propose a simple method to derive it from the request history (Sect. 4). Then we present an approach from the perspective of the adversary to refine his view on possible issuers based on query dependency (Sect. 5). Our approach also makes use of users' observed requests, a type of dynamic contextual information that keeps changing. For the second task, we propose new query privacy metrics (Sect. 6), which allow users to express their privacy requirements and, more importantly, can take into account query dependency. We then adapt spatial generalisation algorithms in the literature to handle user privacy requirements by designing a method to update users' real-time proabibilities of issuing a query (Sect. 7). Through experiments, we show that when the adversary can explore query dependency, query privacy should be carefully addressed and our proposed protection is both effective and efficient in practice (Sect. 8).

2. RELATED WORK

k-anonymity & query privacy. The concept of k-anonymity [25] was originally proposed in the field of database privacy and then introduced for privacy protection in LBSs by Gruteser and Grunwald [13]. Because of its simplicity, k-anonymity has been widely studied in the last decades. For instance, Tan et al. [31] define *information leakage* to quantify location information revealed in spatial cloaking. Xue et al. [32] introduce *location diversity* to ensure that generalised regions contain at least ℓ semantic locations. Kalnis et al. propose a novel cloaking – Hilbert cloak which is the first proved method enforcing reciprocity [17]. However, deeper understanding of k-anonymity reveals its drawbacks in preserving location privacy. Shokri et al. [30] evaluate k-anonymity in different scenarios in terms of the adversary's background knowledge. They conclude that spatial cloaking (e.g., k-anonymity) is only effective for protecting query privacy but not location privacy.

Context-aware privacy analysis. It has been recognised that the effectiveness of spatial cloaking can be compromised when the adversary has access to additional contextual information, e.g., user profiles which have many interpretations in the literature. Shokri et al. [29] use mobility patterns and propose a probabilistic framework to learn users' whereabouts from anonymised and generalised traces. Personal information (e.g., gender, job, salary) is usually available on the Internet, e.g., online social networks such as Facebook and LinkedIn, and can serve as user profiles as well. Shin et al. [27, 28] propose metrics based on k-anonymity by restricting levels of similarity among users in generalised regions in terms of their profiles. Chen and Pang [7] use the same information but obtain probability distributions over users for issuing queries, which allows them to measure query privacy in several new ways.

The contextual information (e.g., mobility pattern, user information) explored in the above papers does not change during their analysis and thus can be considered as *static*. Whereas, in practice, contexts can be *dynamic* as well, e.g., users' whereabouts and observed requests. In the literature, two types of requests have been studied – *associated requests* [4, 3, 8] and *recurrent requests* [23].

Requests are associated once they are recognised as issued by a same (anonymous) user, which can be achieved for example by multi-target tracking techniques [14] or probabilistic reasoning [29]. In this case, the intersection of all requests' anonymity set helps the adversary reduce the number of possible issuers. To handle such privacy threats, Bettini et al. [4, 3] introduce *historical k-anonymity*, which is then extended for continuous LBSs by Dewri et al. [8]. Historical k-anonymity aims to guarantee that associated requests share at least k fixed users in the generalised regions.

Requests are recurrent when they are issued at the same time. For the recurrent requests containing the same query, if they are also from the same region, the protection of spatial cloaking, e.g., k-anonymity, is degraded for query privacy. For instance, in the extreme case, when all users in a region send an identical query, no user has query privacy. Riboni et al. [23] identify the threat and make use of t-closeness to guarantee that the distance between the distribution over the queries from an issuer's generalised region and that of the whole region is below a threshold. Dewri et al. [9] consider a scenario in continuous LBSs which have both associated and recurrent requests . The adversary only learns the regions and time of requests issued by an anonymous user. When recurrent requests are considered, Dewri et al. propose m-invariance to ensure that in addition to k fixed users in the generalised regions, at least m fixed queries are generated from each region.

A short discussion. Associated requests are not always available. Uncertainty of the linkability between requests is inevitable in reality and should be carefully handled. Moreover, the previous research considers queries in historical requests as independent, and ignores users' issuing patterns over queries. Last but not least, we believe both static and dynamic contexts should be taken into account in the analysis of users' privacy in LBSs.

Our work differs from the related papers from the following perspectives: (i) we take into account dependency between successive queries in our privacy analysis; (ii) we infer a user's next query by his probability to issue the query calculated based his observed requests; (iii) we do not assume that a trace of requests are recognised, which makes our work more realistic.

3. PRELIMINARIES

In this section, we first present our formal framework in Sect. 3.1 and then define the adversary model in Sect. 3.2.

3.1 Formal framework

Let $\mathcal{U}$ be the set of all users, who are in fact the potential issuers of LBS requests. We use $\mathcal{L}$ to denote the set of all possible positions where a user can issue a request. The accuracy of any position $\ell \in \mathcal{L}$ is determined by the positioning devices used. We represent time as a totally ordered discrete set $\mathcal{T}$, whose granularity, e.g., minutes or seconds, is decided by LBS providers. The function $\mathit{whereis} : \mathcal{U} \times \mathcal{T} \to \mathcal{L}$ gives the exact position of a user at a given time. Thus, for any time $t \in \mathcal{T}$, users' spatial distribution is $dis_t = \{\langle u, \mathit{whereis}(u,t)\rangle \mid u \in \mathcal{U}\}$. Let $\mathcal{R} \subseteq 2^{\mathcal{L}}$ be the set of all regions with any size that could be included in generalised requests. With $\mathcal{Q}$ being the set of supported queries, an LBS request is in the form of $\langle u, \ell, t, q\rangle \in \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{Q}$. The corresponding generalised request has the form of $\langle r, t, q\rangle$, obtained by the request generalisation function $f : \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{Q} \to \mathcal{R} \times \mathcal{T} \times \mathcal{Q}$ – it removes the issuer's identity and replaces the exact location with a region $r \in \mathcal{R}$. We use function *req* to obtain the query of a (generalised) request (i.e., $\mathit{req}(\langle u, \ell, t, q\rangle)$=$q$ and $\mathit{req}(\langle r, t, q\rangle)$=$q$).

For each user u, we use a sequence to denote the requests that he has issued, i.e., $\mathcal{H}_u = (\langle u, \ell_1, t_1, q_1\rangle, \ldots, \langle u, \ell_n, t_n, q_n\rangle)$ with $t_i < t_{i+1}$ for all $i \in \{1, \ldots, n-1\}$ and $\mathcal{H}_u(i)$ is the ith request in $\mathcal{H}_u$. We call this sequence *user request history*, whose length is denoted as $\mathit{len}(\mathcal{H}_u)$.

After observing a generalised request at time t, the adversary adds it to a sequence, i.e., $\mathcal{O}_t = (\langle r_1, t_1, q_1\rangle, \ldots, \langle r_m, t_m, q_m\rangle)$ ($t_i < t_{i+1}$ for all $i \in \{1, \ldots, m–1\}$ and $t_m < t$). For the sake of simplicity, we do not consider recurrent queries, i.e., those elements in $\mathcal{O}_t$ with the same time-stamps. Furthermore, for each request in $\mathcal{O}_t$, the adversary calculates its anonymity set, i.e., all those users located in the generalised region. Thus, for each user, the adversary can maintain a sequence of generalised requests, whose anonymity sets contain this user as an element. We call this sequence an *observed request trace* and denote the one for user u up to time t as $\mathcal{O}_{u,t}$. It is obvious that with time passing, a user's observed request trace keeps growing. The length of $\mathcal{O}_{u,t}$ is denoted as $\mathit{len}(\mathcal{O}_{u,t})$. The difference between $\mathcal{H}_u$ and $\mathcal{O}_{u,t}$ is that the adversary is certain about the issuer of each request in $\mathcal{H}_u$ but uncertain about the issuers of the requests in $\mathcal{O}_{u,t}$. Tab. 1 summarises relevant notations used in this paper.

Table 1: Notations

Notations	Description
$\mathcal{Q}$	the set of supported queries
$\mathcal{U}$	the set of users
$\mathcal{L}$	the set of positions
$\mathcal{R}$	the set of regions
$\mathcal{T}$	the set of time granules
$\langle u, \ell, t, q\rangle$	a query q issued by user u at position ℓ at time t
$\langle r, t, q\rangle$	a generalised request by the anonymiser
$\mathcal{O}_{u,t}$	user u's observed request trace up to time t
$\mathcal{H}_u$	user u's request history
$\mathcal{V}^u$	user u's prior vector
$\mathcal{D}^u$	user u's dependency matrix
$p_u(q_i \mid q_j)$	the probability of user u issuing q_i after q_j
$p_u(q_i)$	the priori probability of user u issuing q_i
$p(u \mid \langle r, t, q\rangle)$	the probability of user u issuing the generalised query $\langle r, t, q\rangle$
dis_t	user spatial distribution at time t
$u\ell(r,t)$	the set of users located in region r at time t
$\mathit{req}(\langle r, t, q\rangle)$	the query of request $\langle r, t, q\rangle$

3.2 Adversary model

Privacy attacks and countermeasures should be categorised according to the model and aims of the adversary [3]. For query privacy, the aim of the adversary is to associate issuers to their queries. We use the following assumptions to define our adversary model.

Assumption 1. The adversary knows users' spatial distribution dis_t at any time t and the spatial generalisation algorithms. This is now a common assumption used in the literature and it makes a strong adversary which allows us to analyse users' query privacy in the worst case situations. We have this assumption based on the observation that uses may publish their positions in applications or issue requests at some known places , e.g., home/offices. The availability of dis_t enables the adversary to obtain the set of users located in any region r at time t, which is denoted as $u\ell(r,t)$. This assumption can be relaxed by introducing unidentified users whose positions are not part of the adversary's knowledge [3].

Assumption 2. As we have mentioned, LBS providers have collected users' request history. For each user u, we assume that the adversary has a user u's request history $\mathcal{H}_u$ for a sufficiently long period. Furthermore, the adversary maintains the up-to-date observed request trace $\mathcal{O}_{u,t}$ of every user u. For the sake of simplicity, we assume that $\mathcal{H}_u$ is complete, namely there do not exist any requests that are issued by u during the period but are not included in $\mathcal{H}_u$. To handle incomplete $\mathcal{H}_u$, we can use the approaches such as Gibbs sampling [24] to reconstruct the missing queries similarly to fill missing locations in mobility traces [29].

4. DERIVING QUERY DEPENDENCY

In this section, we present an approach to derive dependency between queries for a user from his request history. *Query dependency* can be used to predict a user's next query based on past queries that he has issued before. There also exists a special situation when a user has no past queries or the past queries have little impact on his future queries. In this case, we need to consider users' *a priori preference* on issuing queries. Both of these two types of information are calculated once and remains unchanged, and thus are classified as static.

4.1 Query dependency

We model query dependency with the assumption that the query that a user will issue next can only be affected by the last query that the user has issued (i.e., the Markov property). For a pair of queries q_i and q_j, the dependency of query q_j on q_i is denoted as the conditional probability $p_u(q_j \mid q_i)$. In other words, it is the probability for user u to issue query q_j after having issued query q_i (without issuing any other queries in between). The query dependency information of user u can thus be expressed as a *dependency matrix* – $\mathcal{D}^u$ of size $|\mathcal{Q}| \times |\mathcal{Q}|$ and $\mathcal{D}^u_{ij} = p_u(q_j \mid q_i)$.

To find dependent queries, we need to identify the successive requests. Intuitively, two requests are successive if there are no other requests between them in the request history. This simply means that $\mathcal{H}_u(i+1)$ is the successive query of $\mathcal{H}_u(i)$ for all $i < \mathit{len}(\mathcal{H}_u)$. All the occurrence of query q_j depending on q_i can be captured by the set of pairs of successive requests $\mathcal{C}_{i,j} = \{(\mathcal{H}_u(k), \mathcal{H}_u(k+1)) \mid \mathit{req}(\mathcal{H}_u(k)) = q_i \wedge \mathit{req}(\mathcal{H}_u(k+1)) = q_j, 0 < k < \mathit{len}(\mathcal{H}_u)\}$. Given a request history $\mathcal{H}_u$, the adversary can derive for a user u the dependency between any pair of queries using the sets $\mathcal{C}_{i,j}$ of successive requests. Furthermore, in this paper we make use of Lidstone's or additive smoothing [20] to ensure that there is no dependency of degree zero for q_j on q_i due to no occurrence of the pair (q_i, q_j) in the request history.

Formally, let λ be the smoothing parameter which is usually set to 1. The dependency $p_u(q_j \mid q_i)$ is calculated as follows:

$$p_u(q_j \mid q_i) = \frac{|\mathcal{C}_{i,j}| + \lambda}{\sum_{q_k \in \mathcal{Q}} |\mathcal{C}_{i,k}| + |\mathcal{Q}| \cdot \lambda}.$$

4.2 A priori preference

There are many cases that a query does not depend on its past queries. For example, users may issue an LBS query for the first time or accidentally for an emergent need. In such cases, the best the adversary can do is to apply users' *a priori* preference to find possible issuers of the query.

We model the *a priori* preference of a user u as a distribution over the set of queries denoted as $\mathcal{V}^u$. The distribution indicates the probability of the user to issue a query. For query $q_i \in \mathcal{Q}$, $\mathcal{V}^u_i = p_u(q_i)$ and $\sum_{q_i \in \mathcal{Q}} p_u(q_i)$=1.

There are many sources of information reflecting users' *a priori* preference. Users' personal information such as hobbies and occupation have been discussed and shown effective in assessing users' preference [27, 28, 7]. Moreover, a user's request history also reflects his preference. Thus, we estimate a user's *a priori* preference (i.e., $\mathcal{V}^u$) by combining his request history ($\mathcal{H}_u$) and his personal information. For users' personal information, we apply the approach of Chen and Pang [7] where weights are assigned to different types of information as well as their values according to their correlation to a query. Let $\mathcal{P}_u$ be user u's personal information. The preference of user u for query q_i with respect to $\mathcal{P}_u$ is denoted as $p_u(q_i \,|\, \mathcal{P}_u)$. Moreover, let $p_u(q_i \,|\, \mathcal{H}_u)$ be the likelihood for user u to issue q_i based on his request history, we can use the frequency of the occurrence of the query in the request history to estimate $p_u(q_i \,|\, \mathcal{H}_u)$:

$$p_u(q_i \,|\, \mathcal{H}_u) = \frac{|\{\mathcal{H}_u(k) \,|\, \mathit{req}(\mathcal{H}_u(k)) = q_i\}|}{\mathit{len}(\mathcal{H}_u)}.$$

The two distributions evaluate a user's *a priori* preference on next queries from two different perspectives. An agreement between them is needed. This is equivalent to aggregate expert probability judgements [5]. We use *linear opinion pool aggregation* which is empirically effective and has been widely applied in practice [1]. By assigning a weight to each distribution, i.e., $w_{\mathcal{P}}$ and $w_{\mathcal{H}}$ with $w_{\mathcal{P}} + w_{\mathcal{H}} = 1$, we can calculate $p_u(q_i)$ as follows:

$$p_u(q_i) = w_{\mathcal{P}} \cdot p_u(q_i \,|\, \mathcal{P}_u) + w_{\mathcal{H}} \cdot p_u(q_i \,|\, \mathcal{H}_u).$$

Remark. The way we model users' query dependency and *a priori* preference has some restrictions. For instance, we do not consider the influence of other factors such as time – usually a user's behaviours in weekdays are different from weekends. Our approach can be extended by distinguishing the request history at different time periods. We have also assumed that a query is only dependent on its immediate previous query. This restriction can be lifted by considering, e.g., the last k historical queries. However, deriving such dependency from $\mathcal{H}_u$ might not be as efficient and accurate as the derivation of $\mathcal{D}^u$.

5. QUERY PRIVACY ANALYSIS

In this section, we present an analysis of the possible issuers of a given generalised request from the adversary's point of view by considering both static contexts (query dependency and *a priori* preference) and dynamic contexts (observed request traces). We use a posterior probability distribution over users to represent the results of the analysis. Recall that the trace of observed generalised requests up to time t is denoted by $\mathcal{O}_t$. For a generalised request $\langle r, t, q\rangle$ and a user u, the corresponding posterior probability distributions is defined as $p(u \,|\, \langle r, t, q\rangle, \mathcal{O}_t)$. Since the static contexts do not change during the analysis, we do not include them in the definition explicitly.

Through the Bayesian rule we have:

$$\begin{aligned} p(u \,|\, \langle r, t, q\rangle, \mathcal{O}_t) &= \frac{p(\langle r, t, q\rangle \,|\, u, \mathcal{O}_t)}{p(\langle r, t, q\rangle, \mathcal{O}_t)} \\ &= \frac{p(\langle r, t, q\rangle \,|\, u, \mathcal{O}_t) \cdot p(u \,|\, \mathcal{O}_t) \cdot p(\mathcal{O}_t)}{\sum_{u'} p(\langle r, t, q\rangle \,|\, u', \mathcal{O}_t) \cdot p(u' \,|\, \mathcal{O}_t) \cdot p(\mathcal{O}_t)}. \end{aligned}$$

There are three new distributions. The distribution $p(\mathcal{O}_t)$ measures the probability of generating the observed request trace $\mathcal{O}_t$. It is difficult to evaluate its value. Whereas, since it appears in both the numerator and the denominator, we can eliminate it from the formula. The distribution $p(u \,|\, \mathcal{O}_t)$ is the probability of user u to issue a request at time t based on the observed requests. As we have no information about the distribution, it is assumed to uniform according to the principle of maximum entropy [15, 16], which leads to $p(u|\mathcal{O}_t) = p(u'|\mathcal{O}_t)$ ($\forall u' \in \mathcal{U}$). Thus, the posterior distribution can be simplified as:

$$p(u \,|\, \langle r, t, q\rangle, \mathcal{O}_t) = \frac{p(\langle r, t, q\rangle \,|\, u, \mathcal{O}_t)}{\sum_{u' \in \mathcal{U}} p(\langle r, t, q\rangle \,|\, u', \mathcal{O}_t)} \qquad (1)$$

The probability $p(\langle r, t, q\rangle \mid u, \mathcal{O}_t)$ indicates the probability for the user u to issue the generalised request $\langle r, t, q\rangle$ in terms of his observed request trace. As a generalised algorithm (see Sect. 7) always outputs a request with a region including the issuer, only the users located in the region may have issued the request. Thus, for any user u' out of region r at time t, we have $p(\langle r, t, q\rangle \,|\, u', \mathcal{O}_t) =$ 0 for any $q \in \mathcal{Q}$. Furthermore, because of the independence between users with regard to issuing requests, other users' request history has no influence on the next query of the user. Thus we have $p(\langle r, t, q\rangle \,|\, u, \mathcal{O}_t) = p(\langle r, t, q\rangle \,|\, u, \mathcal{O}_{u,t})$ for $u \in \mathcal{U}$.

The size of $\mathcal{O}_{u,t}$ is a key factor determining the accuracy and the complexity of the calculation of $p(\langle r, t, q\rangle \mid u, \mathcal{O}_{u,t})$. Recall that $\mathcal{O}_{u,t}$ consists of all the observed requests that may be issued by user u up to time t. Intuitively, the longer $\mathcal{O}_{u,t}$ is, more computational overhead is required for getting $p(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$. It is impractical to consider the complete $\mathcal{O}_{u,t}$ during the calculation. Instead, we fix a *history window* which consists of the latest n observed requests of user u (i.e., $n \leq \mathit{len}(\mathcal{O}_{u,t})$). Therefore, our problem can be reformulated as to compute $p^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$, indicating the distribution is based on last n observed requests.

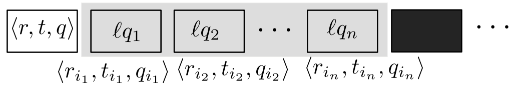

$\langle r, t, q\rangle$
ℓq_1
ℓq_2
$\cdots$
ℓq_n
$\cdots$
$\langle r_{i_1}, t_{i_1}, q_{i_1}\rangle$
$\langle r_{i_2}, t_{i_2}, q_{i_2}\rangle$
$\langle r_{i_n}, t_{i_n}, q_{i_n}\rangle$

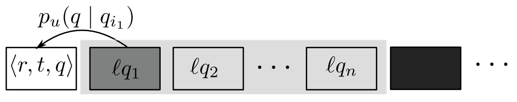
Figure 1: A history window of n observed requests.

In Fig. 1, we show an example of a history window. It has n observed requests, $\langle r_{i_1}, t_{i_1}, q_{i_1}\rangle, \ldots, \langle r_{i_n}, t_{i_n}, q_{i_n}\rangle$ with $t_{i_j} > t_{i_{j-1}}$ ($j > 1$). Let $\ell q_j(\mathcal{O}_{u,t})$ be the jth latest observed request in $\mathcal{O}_{u,t}$, whose query is $\mathit{req}(\ell q_j(\mathcal{O}_{u,t})) = q_{i_j}$. In the following discussion, we simply write ℓq_j if $\mathcal{O}_{u,t}$ is clear from the context. It is obvious that ℓq_1 is the latest observed request of user u.

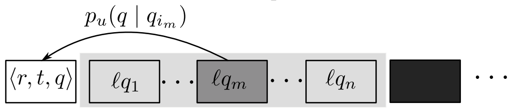
Once $p^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$ is calculated, it is then added into the adversary's knowledge. Therefore, for a past request $\langle r', t', q'\rangle$ in $\mathcal{O}_{u,t}$ ($t' < t$), the adversary has the probability $p(u \,|\, \langle r', t', q'\rangle, \mathcal{O}'_t)$. In the sequel, we simply denote it as $p(u \mid \langle r', t', q'\rangle)$ in cases without any confusion.

The key to calculate the distribution is to determine the user's latest request. Whereas, it is uncertain which is his latest one in the history window. To handle this uncertainty, we distinguish three cases which are depicted in Fig. 2.

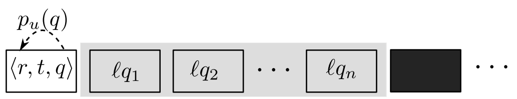
1. User u has issued both the last request in the history window (i.e., ℓq_1, see Fig. 2a) and the current request (i.e., $\langle r, t, q\rangle$).



(a) The latest request is ℓq_1.



(b) The latest request is $\ell q_m (m \in (1,n])$.



(c) The latest request is not in the history window.

Figure 2: The three cases.

Considering query dependence, the probability of this case is

$$p_u(u \,|\, \ell q_1) \cdot p_u(q \,|\, q_{i_1}).$$

2. User u has issued the current request $\langle r,t,q\rangle$ and his latest request is ℓq_m $(1 < m \leq n)$ (see Fig. 2b). The probability of ℓq_m being the latest request is the production of the probability that the last $m-1$ requests are *not* issued by u and the probability that u has issued ℓq_m, i.e., $p(u \mid \ell q_m) \cdot \prod_{j=1}^{m-1}(1 - p(u \mid \ell q_j))$. Considering query dependence, the probability of this case is

$$p_u(q \,|\, q_{i_m}) \cdot p(u \,|\, \ell q_m) \cdot \prod_{j=1}^{m-1}(1 - p(u \,|\, \ell q_j)).$$

3. User u did not issue any request in the history window (see Fig. 2c). In this case, we suppose that the user issued the current request according to his *a priori* preference, i.e., $p_u(q)$. Based on the probability that the user's latest request is outside of the history window as $\prod_{j=1}^{n}(1 - p(u \mid \ell q_j))$, the probability of this case is

$$p_u(q) \cdot \prod_{j=1}^{n}(1 - p(u \,|\, \ell q_j)).$$

We sum up the above three probabilities to compute the probability for user u in region r at time t to issue q when a history window of size n is considered:

$$\begin{aligned} &p^n(\langle r,t,q\rangle \,|\, u, \mathcal{O}_{u,t}) = \qquad (2)\\ &\quad p(u \,|\, \ell q_1) \cdot p_u(q \,|\, req(\ell q_1))\\ &\quad + \sum_{m=2}^{n} p(u \,|\, \ell q_m) \cdot p_u(q \,|\, req(\ell q_m)) \cdot \prod_{j=1}^{m-1}(1 - p(u \,|\, \ell q_j))\\ &\quad + p_u(q) \cdot \prod_{j=1}^{n}(1 - p(u \,|\, \ell q_j)). \end{aligned}$$

We use the following example with $n = 2$ to show the calculation.

EXAMPLE 1. *Suppose the last two requests are $\langle r'', t'', q''\rangle$ and $\langle r', t', q'\rangle$ with $t'' < t' < t$ in $\mathcal{O}_{u,t}$. Let $\langle r,t,q\rangle$ be an observed request. Then for user u, the probability that he issues the request is computed as follows:*

$$\begin{aligned} &p^2(\langle r,t,q\rangle \,|\, u, \mathcal{O}_{u,t}) =\\ &\quad p_u(q \,|\, q') \cdot p(u \,|\, \langle r', t', q'\rangle)\\ &\quad + \big(1 - p(u \,|\, \langle r', t', q'\rangle)\big) \cdot p(u \,|\, \langle r'', t'', q''\rangle) \cdot p_u(q \,|\, q'')\\ &\quad + \big(1 - p(u \,|\, \langle r', t', q'\rangle)\big) \cdot \big(1 - p(u \,|\, \langle r'', t'', q''\rangle)\big) \cdot p_u(q). \end{aligned}$$

It is clear that the calculation of Eq. 2 combines the static contextual information, i.e., users' *a priori* preference on queries and the dependency between queries, with the dynamic contextual information, i.e., observed request traces.

6. MEASURING QUERY PRIVACY

To protect users' query privacy, we follow the principle that users should be able to express their privacy requirements as it is unlikely to have absolute privacy in the context of spatial anonymisation. In this paper, we consider users' query privacy well-preserved if the spatial generalisation algorithms can generate regions meeting their privacy requirements. The calculation of the probability distribution over users in Sect. 5 provides us a way to measure query privacy through the uncertainty of the adversary. A number of metrics for query privacy are proposed in [7] where it is assumed that the adversary has access to users' profiles. In this paper, we extend two of them, i.e., k-ABS and β-EBA, by taking query dependence into account. The other metrics can be extended similarly.

Query dependent k-ABS. Intuitively, this requirement is satisfied if at least k users are grouped together in the generalised region and they have close posterior probabilities to issue the given request. Let $\| \, p_1, p_2 \, \|$ be the distance between two probabilities p_1 and p_2, and ϵ be the maximum distance allowed between users' posterior probabilities. Recall that f is the request generalisation function. Given a history window of size n, the metric *query-dependent k-ABS* can be defined as follows:

DEFINITION 1. *Let $\langle u, whereis(u,t), t, q\rangle$ be a request of u and $\langle r,t,q\rangle$ be the corresponding generalised request. The issuer u is* query dependent k-approximate beyond suspicious *if*

$$\begin{aligned} &|\{u' \in u\ell(r,t) \;|\; \| \, p^n(u \,|\, \langle r,t,q\rangle, \mathcal{O}_t), p^n(u' \,|\, \langle r,t,q\rangle, \mathcal{O}_t) \, \| < \epsilon\\ &\qquad \wedge f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle\}| \geq k \end{aligned}$$

Query dependent β-EBA. This metric utilises the notion of entropy from information theory to measure the uncertainty of the adversary about the issuer of a request. Let variable U denote the issuer of request $\langle r,t,q\rangle$ and n be the size of the history window. When query dependency is considered as part of the adversary's knowledge, the adversary's uncertainty of the issuer of $\langle r,t,q\rangle$ can be measured by the following entropy:

$$\begin{aligned} H^n(U \,|\, \langle r,t,q\rangle) = -\sum_{u' \in u\ell(r,t)} &p^n(u' \,|\, \langle r,t,q\rangle, \mathcal{O}_t) \cdot\\ &\log p^n(u' \,|\, \langle r,t,q\rangle, \mathcal{O}_t). \end{aligned}$$

Thus, we can define *query-dependent β-EBA* as follows:

DEFINITION 2. *Let $\beta > 0$, $\langle u, whereis(u,t), t, q\rangle \in Q$ be a request and $\langle r,t,q\rangle$ the corresponding generalised request. The issuer u is* query-dependent β-entropy based anonymous *if for all $u' \in u\ell(r,t)$,*

$$H^n(U \,|\, \langle r,t,q\rangle) \geq \beta \;\wedge\; f(\langle u', whereis(u',t), t, q\rangle) = \langle r,t,q\rangle.$$

Remark. When users use these metrics to express their privacy requirements, at least three elements should be provided – a metric, the parameter values of the chosen metric, and the history window's size. However, in practice it is difficult and cumbersome for a user to give exact values to these elements, as this requires them to understand the meaning of each parameter and the corresponding implication on privacy protection. To avoid this situation it is better to provide a list of privacy levels, e.g., from *low* to *very high*. Each level corresponds to a setting of privacy parameters. For example, a user's privacy requirement can be represented as $\langle \mathsf{kABS}, \mathit{high}\rangle$, which is then transformed into $\langle \mathsf{kABS}, (0.05, 10), 5\rangle$. This ensures that whenever a request is successfully generalised, the region has 10 users with similar posterior probabilities to the issuer's, after taking into account the last 5 observed requests. Furthermore, the distance between two such users' posterior probabilities is bounded by 0.05. In practice, the transformation can be made automatic and embedded in the request generalisation process.

7. AN GENERALISATION ALGORITHM

In this section, we focus on the spatial generalisation procedures, which can generate regions satisfying users' privacy requirements expressed in the metrics as defined in Sect. 6.

Basically, there are two ways to implement generalisation algorithms – *centralised* and *distributed*. A centralised structure relies on a trusted agent, the *anonymiser*, to collect users' requests and anonymise them before sending them to the LBS servers, while in a distributed implementation users cooperate with each other to construct a generalised region [10, 26]. The centralised framework is easy to implement and well-studied in the literature while the distributed framework requires more communication between collaborators and security analysis, e.g., with respect to *insiders*, is not well studied. Because of its simplicity and efficiency, we decide to choose the centralised framework although the trust in the anonymiser is needed. In this centralised structure, users send their positions to the anonymiser, who will generalise a request according to the issuer's privacy requirement.

The two algorithms kABS and uniformDP proposed in [7] protect users' query privacy against the adversary which has users' *a priori* preference on queries as part of his knowledge. The main idea is to compute an anonymity set of users based on the posterior probability of each user to issue the given request. This methodology is generic but the algorithms are designed specially for user profiles, which do not change over time. Whereas, due to query dependency, we need to handle users' observed request traces, which change over time. We start with a brief introduction to the algorithms in [7] and then show how to handle observed request traces.

kABS & uniformDP. The former copes with requirements in terms of k-ABS while the later is a uniform algorithm for the other metrics proposed in [7]. Both algorithms take users' real-time spatial distribution and (static) user a priori preferences as inputs and output a generalised region (if possible). Algorithm kABS first calls a clustering algorithm, i.e., K-means, to cluster users with similar profiles, and then uses existing k-anonymity generalisation algorithms to calculate regions. On the other hand, uniformDP iteratively splits a region into two sub-regions until it is not possible to have a partition such that both of the sub-regions satisfy the issuer's privacy requirement.

Our algorithm. If a request $\langle r, t, q\rangle$ of user u satisfies query-dependent k-ABS, the region r must contain at least another k-1 users with posterior probabilities close to $p^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)$. From Eq. 1, we can see that such users (e.g., u') also have close probabilities to issue the request in terms of their observed requests (i.e., $p^n(\langle r, t, q\rangle \mid u', \mathcal{O}_t)$). Thus, we can apply the idea of the algorithm kABS to find a region with at least k users with similar probability with respects to observed requests.

There are two extensions needed to adapt kABS. First, the probability $p^n(\langle r, t, q\rangle \mid u', \mathcal{O}_t)$ is related to the generalised request, which seems not available before the generalisation. However, an interesting feature of the calculation in Eq. 2 is that for user u, given a time t, for any two regions r and r' such that $\mathit{whereis}(u, t) \in r \cap r'$, we have $p^n(\langle r, t, q\rangle \mid u, \mathcal{O}_t) = p^n(\langle r', t, q\rangle \mid u, \mathcal{O}_t)$. So we can obtain the probability before generalising the request by computing $p^n(\langle r_{ori}, t, q\rangle \mid u, \mathcal{O}_t)$ where r_{ori} is the whole initial region. Second, since observed request traces are part of the computation of users' posterior probabilities, the probability $p^n(\langle r_{ori}, t, q\rangle \mid u, \mathcal{O}_t)$ for all $u \in \mathcal{U}$ has to be updated for any received request. This requires the algorithm to maintain users' status, including their observed request traces (i.e., $\mathcal{O}_{u,t}$) and the corresponding posterior probabilities (i.e., $p(u \mid \langle r, t, q\rangle)$ from the view of the adversary.

Algorithm 1 An algorithm for spatial generalisation.

```
1: FUNCTION: QD-AreaGen
2: INPUT: $\langle u, \mathit{whereis}(u,t), t, q\rangle, \mathit{dis}(t), \mathcal{O}_t, R_u$
3: OUTPUT: A region $r$ that satisfies $k$-ABS
4:
5: $R'_u = \mathit{transformReq}(R_u)$;
6: $n = \mathit{getWindowSize}(R'_u)$;
7: $\mathcal{M} = \emptyset$;
8: for $u' \in \mathit{u\ell}(r_{ori}, t)$ do
9:     calculate $p^n(\langle r_{ori}, t, q\rangle \mid u', \mathcal{O}_t)$;
10:    $\mathcal{M} = \mathcal{M} \cup \{\langle p^n(\langle r_{ori}, t, q\rangle \mid u', \mathcal{O}_t), u'\rangle\}$;
11: end for
12:
13: if $\mathit{getMetric}(R_u) = \mathsf{kABS}$ then
14:     $(\epsilon, k) = \mathit{getRequirement}(R'_u)$;
15:     $r = \mathsf{kABS}(u, \mathit{dis}(t), \mathcal{M}, \epsilon, k)$;
16: else if $\mathit{getMetric}(R_u) = \mathsf{EBA}$ then
17:     $\beta = \mathit{getRequirement}(R'_u)$;
18:     $r = \mathsf{uniformDP}(u, \mathit{dis}(t), \mathcal{M}, \beta)$;
19: end if
20:
21: if $r \neq \emptyset$ then
22:     for $u' \in \mathit{u\ell}(r, t)$ do
23:         $\mathcal{O}_{u',t} = \mathcal{O}_{u',t} \cup \{\langle r, t, q\rangle\}$;
24:         $p(u' \mid \langle r, t, q\rangle) = p^n(u' \mid \langle r, t, q\rangle, \mathcal{O}_t)$;
25:     end for
26: end if
27: return $r$
```

In order to find a region satisfying query-dependent β-EBA, we apply the same idea of area splitting – we adapt uniformDP by giving the probabilities $p^n(\langle r_{ori}, t, q\rangle \mid u', \mathcal{O}_t)$ ($u' \in \mathcal{U}$) as input instead of users' *a priori* preference. Similar to the algorithm designed for query-dependent k-ABS, users' status (i.e., observed request traces and the corresponding posterior probabilities) needs to be dynamically updated.

We use Alg. 1 to describe the our spatial generalisation algorithm when query dependency is considered. The algorithm takes a request $\langle u, \mathit{whereis}(u, t), t, q\rangle$ as input and outputs a region r satisfying the requirement R_u based on users' whereabouts $\mathit{dis}(t)$ and observed requests $\mathcal{O}_t$. The user's privacy requirement R_u is first transformed into R'_u by function $\mathit{transformReq}(R_u)$ (line 5) based on a mapping table so R'_u consists of an exact parameter setting that will be used in the generalisation. For requirements using k-ABS and β-EBA, R'_u is of the form of $\langle \mathsf{kABS}, (\epsilon, k), n\rangle$ and $\langle \mathsf{EBA}, \beta, n\rangle$, respectively. Function $\mathit{getWindowSize}(R'_u)$ returns the size of the history window in R'_u and $\mathit{getMetric}(R'_u)$ gives the type of metric used. Recall that r_{ori} is the whole initial region under our consideration. For the reason discussed above, the probabilities $p^n(\langle r_{ori}, t, q\rangle \mid u', \mathcal{O}_t)$ ($u' \in \mathcal{U}$) is calculated (line 9) and the results are stored in set $\mathcal{M}$.

If query-dependent k-ABS is used, then we call function kABS with a distance parameter (i.e., ϵ) and an anonymity degree (i.e., k), which can be extracted from R'_u (line 14). As our implementation of kABS uses the clustering algorithm K-means, we use ϵ to estimate the number of clusters (i.e., K). If the privacy requirement is expressed using other metrics, e.g., query dependent β-EBA, uniformDP is called (line 18).

If a valid region is found, then for each user located in it, we need to update his status (line 22-25). First, his observed request sequence is updated (line 23) as they are considered as possible issuers after the generalised request is issued. Second, each user's posterior probability being the issuer is assigned (line 24) which will be used for future requests (line 9).

Note that it is time-consuming to update users' probabilities to issue the request in terms of their observed requests before generalising each request, because it considers all users in the whole area. During implementation, we observe that in each generalisation process, only a small fraction of users are concerned. Thus, for each request, we first calculate a smaller region (e.g., with 100 users) containing the issuer by k-anonymity generalisation algorithms and then execute Alg. 1 on this region. In this way, the computational overhead can be largely reduced.

8. EXPERIMENTAL RESULTS

We conduct experiments to evaluate our work from two aspects. First, we compare issuers' posterior probabilities *with* vs. *without* adding query dependency to the adversary's knowledge. In this way, we illustrate the privacy risks caused by request history and extracted query dependency. Second, we implement and test our algorithm presented in Sect. 7 on a sample dataset to show the effectiveness of our new privacy metrics (see Sect. 6).

To conduct the experiments, we first construct a mobility dataset using the moving object generator [6]. This dataset consists of the positions of $38,500$ users travelling in a period with 50 discrete time points. This dataset contains users' spatial distributions. Second, we construct the dataset of users' requests. For a number of active users who would issue requests in the period, we simulate a trace of requests for each active user according to his query dependency matrix and *a priori* preference on queries. Specifically, we assume 6 types of queries for users to choose from. This makes users' *a priori* preference around 17% on average. As our purpose is to evaluate the privacy risk caused by query dependency and the effectiveness of the algorithm, we assume query dependency matrix available and generate it by a random procedure. Users' *a priori* preference is assessed in a similar way.

Throughout our experiments, we use one mobility dataset but generate many request datasets with different number of active users so as to evaluate its influence on query privacy. Our simulation is implemented with Java and run on a Linux laptop with 2.67 Ghz Intel Core (TM) and 4GB memory.

Impact of query dependency on users' posterior probabilities. To measure the privacy risk caused by query dependency, we compare users' posterior probabilities in two attack scenarios when k-anonymity spatial generalisation is used. In one scenario, the adversary only learns users' *a priori* preference while in the other, users' query dependency is added.

Let $p_{pf}(u \mid \langle r, t, q\rangle)$ be the issuer's posterior probability when only user u's *a priori* preference is considered. We use Δp to measure the changes of the posterior probability after query dependency is added, which is defined as follows:

$$\Delta p = \frac{|\,p_{pf}(u \mid \langle r, t, q\rangle) - p^n(u \mid \langle r, t, q\rangle, \mathcal{O}_t)\,|}{p_{pf}(u \mid \langle r, t, q\rangle)}.$$

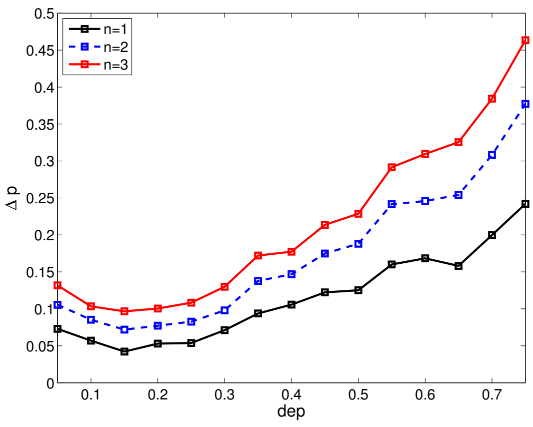
Fig. 3 shows how Δp changes according to (1) different values of history window size n, (2) different strengths of query dependency and (3) the number of active users in the LBS. The results are obtained by a simulation with $8,000$ requests. We divide requests into clusters according to the query dependency of the issuers when sending the requests by an interval of 0.05, and use $dep = 0.05x$ ($1 \leq x \leq 20$) to denote the maximum query dependency allowed in the clusters. For example, if $dep = 0.15$, the issuer of any request in the cluster has a dependency between 0.1 and 0.15. Fig. 3a depicts the average Δp of generalised requests in clusters satisfying k-anonymity with $k = 10$ and with 2.6% of the users being active. Based on the results, we have three observations. First of all, larger history windows lead to big changes in users' posterior probabilities. In our simulation, the average value of Δp increases by 53% and 24% when n grows from 1 to 2 and from 2 to 3, respectively (see Fig. 3a). Second, the average value of Δp increases when query dependency of two successive queries gets stronger (see Fig. 3a). The curves reach their lowest points when dep is about 0.15. This is due to the fact that users' average *a priori* preference on each type of queries ($p_u(q_i)$) is around 17%. The little difference between $p_u(q_i \mid q_{i-1})$ and $p_u(q_i)$ eliminates the influence of query dependency. Third, Δp decreases when there are more active users issuing LBS requests, but the influence becomes smaller with larger history windows. Fig. 3b shows that the average Δp decreases by 30%, 24% and 19% for $n = 1, 2, 3$, respectively, when the percentage of active users increases from 2.5% to 7.5%. This is because more active users in the LBS lead to more observed requests added into users' observed request traces and mixed with users' real requests, while bigger history windows have larger chances to include users' real requests. In general, from Fig. 3, we can conclude that exploring query dependency does greatly decrease the adversary's uncertainty about the real issuers.

Effectiveness of the new privacy metrics. Through experiments, we discuss the features of privacy metrics in terms of (1) area of the generalised regions and (2) issuers' posterior probabilities. We set the percentage of active users to 2.6% and only use the first $1,000$ requests after $8,000$ requests have been observed. Each number shown in the following discussion is an average of the $1,000$ samples. To compare the two metrics presented in Sect. 6, we define a normalised value $norm$: $norm$=k for query-dependent k-ABS, while $norm$=2^{β} for query-dependent β-EBA.

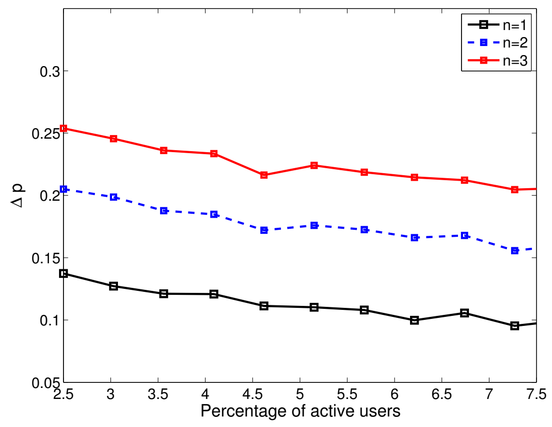
From the above discussion, we know that users can have better their query privacy with larger history windows. Fig. 4 shows how issuers' posterior probabilities and the area of generalised regions change according to the normalised value $norm$ and the history window size n. Note that when n= 0, the generalisation algorithm only considers users' *a priori* preference.

For query-dependent k-ABS, issuers' posterior probabilities are about $\frac{1}{k}$ as the generalised regions have at least k users with similar posterior probabilities. However, after taking a closer look, we can find that a larger n leads to a larger distance to $\frac{1}{k}$. This is because larger history windows make the issuers' posterior probabilities more different from the others, which in turn makes it more difficult to find users with similar posterior probabilities. This also explains why the corresponding generalised regions become larger with larger history windows as shown in Fig. 4b.

For query-dependent β-EBA, issuers' posterior probabilities can

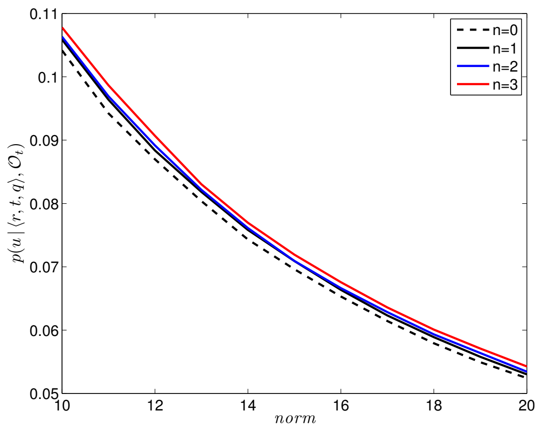


(a) Δp vs. $p(q_i \,|\, q_{i-1})$ and n.

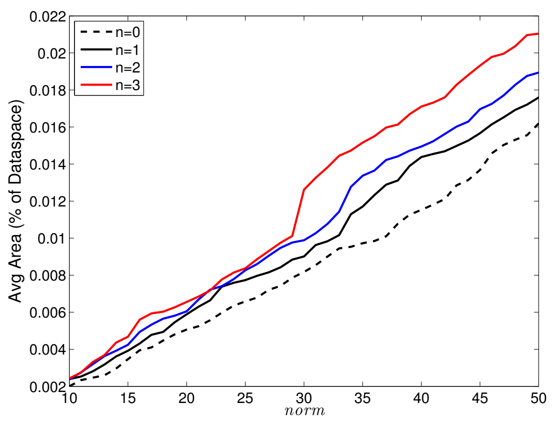


(b) Δp vs. #active users and n.

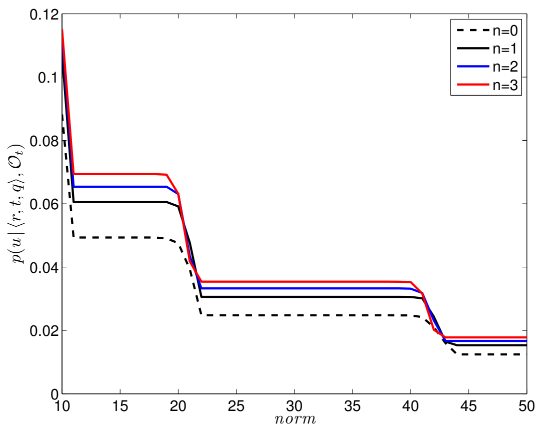
Figure 3: Impact of query dependency and the number of active users on Δp.



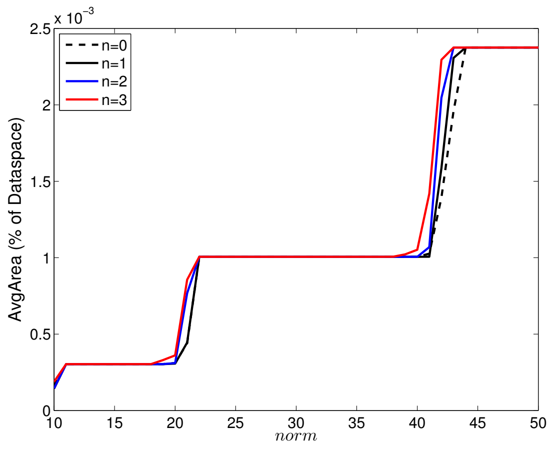
(a) $p(u \,|\, \langle r,t,q\rangle, \mathcal{O}_t)$ vs. n (k-ABS).



(b) Average area vs. n (k-ABS).



(c) $p(u \,|\, \langle r,t,q\rangle, \mathcal{O}_t)$ vs. n (β-EBA).

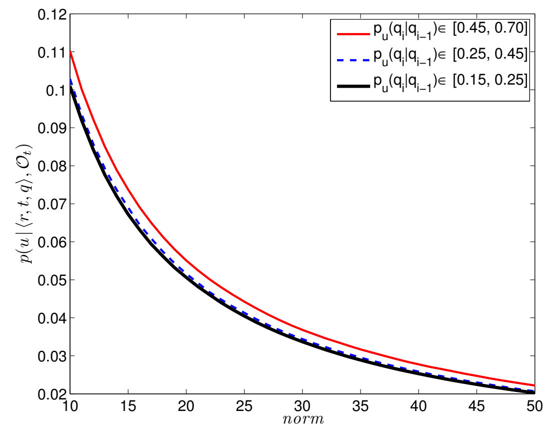


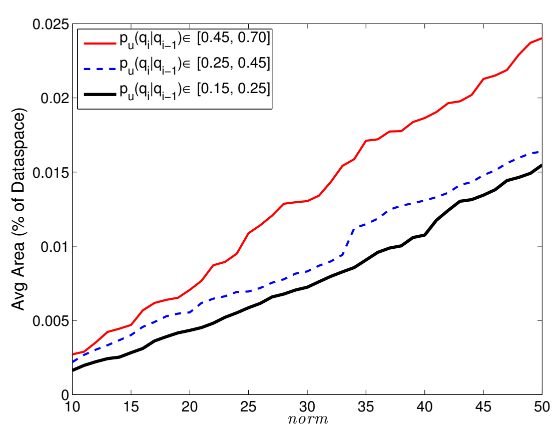
(d) Average area vs. n (β-EBA).

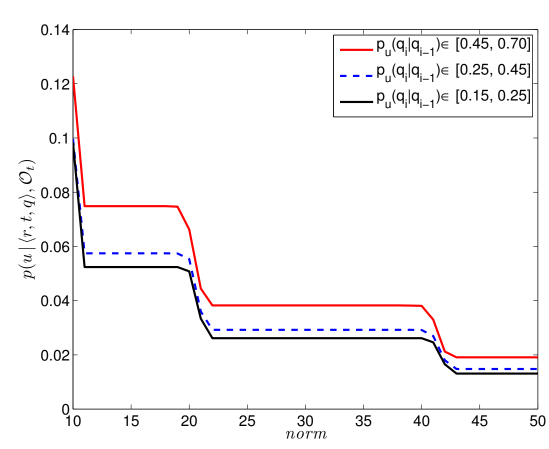
Figure 4: Impact of history window size n.

remain almost unchanged in some segments of the curves. The projection of the middle point of such a segment on axis $norm$ has an logarithm of integer, such as 16 and 32 (see in Fig. 4c). Similar to query-dependent k-ABS, larger history windows increase the issuers' posterior probabilities, which leads to smaller entropy. This can be seen from Fig. 4d where the generalised regions of larger n double their sizes earlier than the regions of smaller n.

We can also observe from Fig. 4 that for the same value of $norm$, although the metric β-EBA cannot always ensure issuers' posterior probabilities as close to $\frac{1}{k}$ as k-ABS (see Fig. 4a and Fig. 4c), the corresponding area of generalised regions is about ten times smaller (see Fig. 4b and Fig. 4d). Since bigger regions lead to worse quality


(a) $p(u \mid \langle r, t, q \rangle, \mathcal{O}_t)$ vs. $p(q_i \mid q_{i-1})$ (k-ABS).


(b) Average area vs. $p(q_i \mid q_{i-1})$ (k-ABS).


(c) $p(u \mid \langle r, t, q \rangle, \mathcal{O}_t)$ vs. $p(q_i \mid q_{i-1})$ (β-EBA).

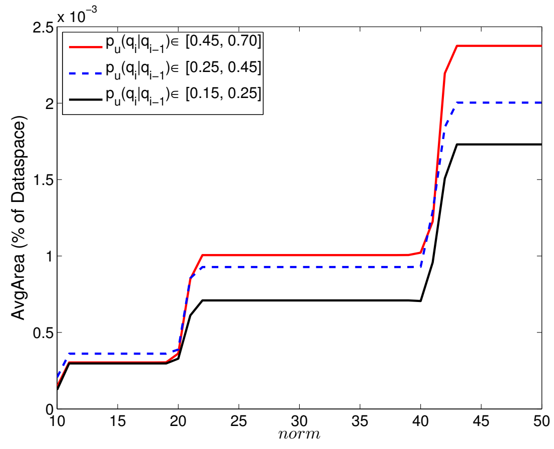

(d) Average area vs. $p(q_i \mid q_{i-1})$ (β-EBA).

Figure 5: Impact of dependency $p(q_i \mid q_{i-1})$.

of service, this indicates that a balance between privacy protection and quality of services needs to be considered in practice.

The protection of issuers' privacy varies with issuers' query dependency. Fig. 5 plots posterior probabilities and average area of generalised regions for issuers with different levels of query dependency. The results are collected with the history window size n=3. Our general observation is that issuers with larger dependencies have bigger posterior probabilities and larger generalised regions.

Tab. 2 summarises the corresponding average increases (in percentage) for issuers with high (≥ 0.45) and medium ($0.25 - 0.45$) dependencies, when compared with those with low dependencies (≤ 0.25). The table shows that posterior probabilities of the issuers, when β-EBA is used, are more sensitive to the degree of dependency (43.1% increase for high-level dependency), while the generalised regions are more sensitive to dependency (62.9% increase for high-level dependency) when k-ABS is used.

Table 2: Increases in posterior probabilities and average area of generalised regions.

	k-ABS		β-EBA	
	medium	high	medium	high
Posterior Prob.	2.1%	9.5%	11.1%	43.1%
Avg Area	21.3%	62.9%	23.3%	30.1%

Performance of the proposed generalisation algorithm. In Fig. 6, we present the performance of Alg. 1 when dealing with query-dependent privacy metrics (k-ABS and β-EBA). For the sake of comparison, we also show in Fig. 6 the performance of the original algorithms (k-ABS-ori and β-EBA-ori) in [7]. The computation time recorded is the average time per request based on executions with the same 100 requests.

As discussed in Sect. 7, it is necessary to update the status of each user, i.e., their observed request traces and the corresponding posterior probabilities. This is time-consuming, especially when the initial region is huge and contains a large number of users. In our implementation, we reduce the computation overhead by restricting the size of initial regions. The number of users located in an initial region is fixed as ten times as many as what users require for. For instance, for k-ABS, if k=10, then we first call k-anonymity generalisation algorithm to get an initial region with 100 users. As the generalisation algorithm is deterministic, which means for any user in a generalised region, it always returns the same region. Thus, our new algorithm Alg. 1 does not suffer from the "outlier" problem identified in the literature [21].

From Fig. 6, we can see that the computation time increases as $norm$ gets bigger. This is because the algorithm has to consider larger initial regions and more users are involved in the calculation of dependency-based posterior probabilities. For β-EBA, about 20ms are needed when $norm$=50, while k-ABS requires more time (around 35ms) as the K-means clustering algorithm is executed

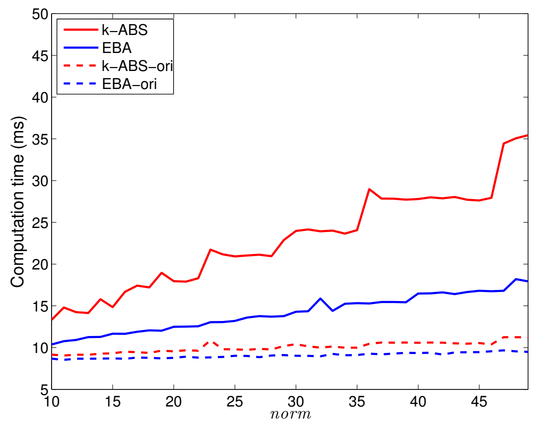


Figure 6: Average computational time (history window $n = 3$).

first to find similar users. When compared to the original algorithms, the computation time of Alg. 1 increases by about two times for β-EBA and about four times for k-ABS when $norm$=50.

There are several ways to improve the efficiency of our implementation. For instance, we can use better data structures to main users' status. For practical applications, we can expect that with a powerful anonymiser our algorithm is efficient enough to handle concurrent requests and gives real-time responses to the users.

9. DISCUSSION AND CONCLUSION

In this paper, we have identified a new type of contextual information *query dependency* which has not been studied for query privacy in LBSs. To show its impact on users' query privacy, we presented an analysis, where the adversary explores query dependency to effectively reduce his uncertainty about the real issuers of requests. The analysis also makes use of observed request traces – a dynamic context. To protect query privacy against such an analysis, we first proposed new privacy metrics for users to express their privacy requirements precisely. Then we designed a new spatial generalisation algorithm to compute regions meeting users' privacy requirements. Through experiments, we have shown (1) enabling the adversary to have access users' query dependency does impose risk on query privacy; (2) the proposed metrics is effective to protect users' query privacy and (3) the generalisation algorithm is efficient for practical applications.

In this paper, we modelled query dependency with the Markov property, and ignored other influencing factors. For instance, the time interval between queries can also be explored by the adversary to further refine his view on possible issuers of an observed request. Regular transition time between two places has been studied in spatial and temporal databases as an important pattern in modelling users' mobility profiles [11]. We can expect to find similar patterns of the time intervals between LBS requests. Such temporal patterns can be considered as another support to predict users' behaviour, especially for issuing future requests. Furthermore, it is also useful to determine an appropriate size of history windows as the influence of past queries decreases as time passes.

10. REFERENCES

[1] Ariely, D., Au, W. T., Bender, R. H., Budescu, D. V., Dietz, C. B., Gu, H., Wallsten, T. S., and Zauberman, G. The effects of averaging subjective probability estimates between and within judges. *Journal of Experimental Psychology: Applied 6* (2000), 130–147.

[2] Bellavista, P., Küpper, A., and Helal, S. Location-based services: Back to the future. *IEEE Pervasive Computing 7*, 2 (2008), 85–89.

[3] Bettini, C., Mascetti, S., Wang, X. S., Freni, D., and Jajodia, S. Anonymity and historical k-anonymity in location-based services. In *Privacy in Location-Based Applications*, vol. 5599 of *LNCS*. Springer, 2009, pp. 1–30.

[4] Bettini, C., Wang, X. S., and Jajodia, S. Protecting privacy against location-based personal identification. In *Proc. 2nd VLDB Workshop on Secure Data Management* (2005), vol. 3674 of *LNCS*, Springer, pp. 185–199.

[5] Bolger, F., and Wright, G. Coherence and calibration in expert probability judgement. *Omega 21*, 6 (1993), 629–644.

[6] Brinkhoff, T. A framework for generating network-based moving objects. *GeoInformatica 6*, 2 (2002), 153–180.

[7] Chen, X., and Pang, J. Measuring query privacy in location-based services. In *Proc. 2nd ACM Conference on Data and Application Security and Privacy (CODASPY)* (2012), ACM, pp. 49–60.

[8] Dewri, R., Ray, I., Ray, I., and Whitley, D. On the formation of historically k-anonymous anonymity sets in a continuous LBS. In *Proc. 6th International Conference on Security and Privacy in Communication Networks (SecureComm)* (2010), vol. 50 of *LNCS*, Springer, pp. 71–88.

[9] Dewri, R., Ray, I., Ray, I., and Whitley, D. Query m-invariance: Preventing query disclosures in continuous location-based services. In *Proc. 11th International Conference on Mobile Data Management (MDM)* (2010), IEEE CS, pp. 95–104.

[10] Ghinita, G., Kalnis, P., and Skiadopoulos, S. PRIVE: anonymous location-based queries in distributed mobile systems. In *Proc. 16th International Conference on World Wide Web (WWW)* (2007), ACM Press, pp. 371–380.

[11] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. Trajectory pattern mining. In *Proc. 13th ACM International Conference on Knowledge Discovery and Data Mining (KDD)* (2007), ACM, pp. 330–339.

[12] González, M. C., Hidalgo, C. A., and Barabási, A.-L. Understanding individual human mobility patterns. *Nature 453* (2008), 779–782.

[13] Gruteser, M., and Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)* (2003), USENIX Association.

[14] Hoh, B., Gruteser, M., Xiong, H., and Alrabady, A. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proc. 14th ACM Conference on Computer and Communications Security (CCS)* (2007), ACM, pp. 161–171.

[15] Jaynes, E. T. Information theory and statistical mechanics. *Physical Review Series II 106*, 4 (1957), 620–630.

[16] Jaynes, E. T. Information theory and statistical mechanics ii. *Physical Review Series II 108*, 2 (1957), 171–190.

[17] Kalnis, P., Ghinita, G., Mouratidis, K., and Papadias, D. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering 19*, 12 (2007), 1719–1733.

[18] Li, N., Li, T., and Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In

Proc. 23rd International Conference on Data Engineering (ICDE) (2007), IEEE CS, pp. 106–115.

[19] MACHANAVAJJHALA, A., KIFER, D., GEHRKE, J., AND VENKITASUBRAMANIAM, M. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data 1*, 1 (2007).

[20] MANNING, C., AND SCHUÜTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, 1999.

[21] MASCETTI, S., BETTINI, C., FRENI, D., AND WANG, X. S. Spatial generalization algorithms for LBS privacy preservation. *Journal of Location Based Services 1*, 3 (2007), 179–207.

[22] RIBONI, D., PARESCHI, L., AND BETTINI, C. Privacy in georeferenced context-aware services: A survey. In *Proc. 1st International Workshop on Privacy in Location-Based Applications (PiLBA)* (2008), vol. 397 of *CEUR Workshop Proceedings*, CEUR.

[23] RIBONI, D., PARESCHI, L., BETTINI, C., AND JAJODIA, S. Preserving anonymity of recurrent location-based queries. In *Proc. 16th International Symposium on Temporal Representation and Reasoning (TIME)* (2009), IEEE CS, pp. 62–69.

[24] ROBERT, C., CELEUX, G., AND DIEBOLT, J. Bayesian estimation of hidden markov chains. *Statics & Probability Letters 16*, 1 (1993), 77–83.

[25] SAMARATI, P. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering 13*, 6 (2001), 1010–1027.

[26] SANTOS, F., HUMBERT, M., SHOKRI, R., AND HUBAUX, J.-P. Collaborative location privacy with rational users. In *Proc. 2nd International Conference on Decision and Game Theory for Security (GameSec)* (2011), vol. 7037 of *LNCS*, Springer, pp. 163–181.

[27] SHIN, H., ATLURI, V., AND VAIDYA, J. A profile anonymization model for privacy in a personalized location based service environment. In *Proc. 9th International Conference on Mobile Data Management (MDM)* (2008), IEEE CS, pp. 73–80.

[28] SHIN, H., ATLURI, V., AND VAIDYA, J. A profile anonymization model for location-based services. *Journal of Computer Security 19*, 5 (2011), 795–833.

[29] SHOKRI, R., THEODORAKOPOULOS, G., BOUDEC, J.-Y. L., AND HUBAUX, J.-P. Quantifying location privacy. In *Proc. 32nd IEEE Symposium on Security and Privacy (S&P)* (2011), IEEE CS, pp. 247–262.

[30] SHOKRI, R., TRONCOSO, C., DÍAZ, C., FREUDIGER, J., AND HUBAUX, J.-P. Unraveling an old cloak: k-anonymity for location privacy. In *Proc. 2010 ACM Workshop on Privacy in the Electronic Society (WPES)* (2010), ACM Press, pp. 115–118.

[31] TAN, K. W., LIN, Y., AND MOURATIDIS, K. Spatial cloaking revisited: Distinguishing information leakage from anonymity. In *Proc. 11th International Symposium on Spatial and Temporal Databases (SSTD)* (2009), vol. 5644 of *LNCS*, Springer, pp. 117–134.

[32] XUE, M., KALNIS, P., AND PUNG, H. K. Location diversity: Enhanced privacy protection in location based services. In *Proc. 4th International Symposium on Location and Context Awareness (LoCA)* (2009), vol. 5561 of *LNCS*, Springer, pp. 70–87.