Private Context-aware Recommendation of Points of Interest: An Initial Investigation

Daniele Riboni Claudio Bettini
Universita' degli Studi di Milano, D.I.Co., EveryWare Lab
via Comelico, 39, I-20135 Milano, Italy
Email: {daniele.riboni,claudio.bettini}@unimi.it

***Abstract*—Several context-aware mobile recommender systems have been recently proposed to suggest points of interest (POIs). Ideally, a user of these systems should not be allowed to know the preferred POIs of another user, since they reveal sensitive information like political opinions, religious beliefs, or sexual orientations. Unfortunately, existing POI recommender systems do not provide any formal guarantee of privacy. In this paper, we report an initial investigation of this challenging research issue. We propose the use of differential privacy methods to extract statistics about users' preferences for POIs. Actual recommendations are generated by querying those statistics, in order to formally enforce privacy. We also present a high-level architecture to apply our methods.**

I. Introduction

People on the move frequently need to obtain customized information about places of interest, according to their context and preferences. For instance, a tourist visiting a city for the first time would like to obtain suggestions for places to visit that match her interests and current location. In order to address this need, various recommender systems for mobile users have been proposed in recent years; for instance, [1], [2], [3], [4], [5] among many others. Generally, those systems include a database of points of interest (POIs) belonging to different categories, such as restaurants, pubs, museums, cultural associations, and so on. Users access the recommender system through their smartphone. Each user can both recommend her preferred POIs among the ones in the system database, and ask for POI recommendations, based on her current context (location, interest categories).

In this paper, we report an initial investigation of the challenging research issue of formally guaranteeing privacy for users of a POIs recommender system. Indeed, a user's preferred POIs is private information: based on a person's preferred POIs, it may be easy to infer sensitive data such as her political opinions, religious beliefs, or sexual orientations. However, to the best of our knowledge, existing mobile recommender systems do not provide any formal privacy guarantee. For the sake of this work, we assume that the recommender system is trusted. On the contrary, the users of the recommender system are possible adversaries, that submit queries in order to reconstruct the preferred POIs of a target individual. The adversaries may have external knowledge about the target user, and use it to reconstruct her POIs. The following example shows a possible attack. The adversary's reasoning mechanism is illustrated in Figure 1.

Example 1: Suppose that a user Bob communicated his preferred POIs to the recommender system. An inquisitive friend of his, Alice, knows that Bob spends all his Saturday nights somewhere in a nearby city, Helltown, and she wants to discover in which place he actually goes. She decides to get this information by querying the recommender system. Hence, she submits a fictitious query to the system (step 1 in Figure 1), asking for recommendations by users living near Bob, and having his profile (age, gender, education, ...). Among the recommendations received by Alice (step 2), only one refers to a POI located in Helltown; it is the recommendation of the Devil's Dance night club (the "outlier" POI shown in Figure 1). Hence, exploiting her background knowledge (step 3), Alice derives with high confidence that Devil's Dance is a POI of Bob's.

The above example refers to an application logic in which the recommender system suggests POIs based on users' profile. However, as shown in [6], [7], [8], [9], privacy threats also exist for recommender systems based on different application logics. The following example illustrates a possible attack, when recommendations are based on the correlation among people's preferences for POIs.

Example 2: Suppose that Bob communicated to the rec-

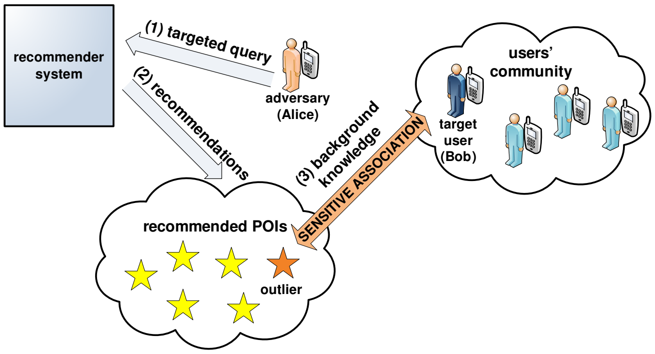

Figure 1. Attack to discover a user's preferred POIs

ommender system his preferred POIs, which are rather uncommon: the “Eccentric dish” restaurant (POI A), the “Strange clothing” boutique (POI B), the “Museum of queer things” (POI C), and the “Devil’s Dance” night club (POI D). Bob publishes on his preferred social network site that he liked POIs A, B and C, but he wants to keep secret his preference for POI D. However, his friend Alice, suspecting that Bob frequents the ill-famed Devil’s Dance night club, submits a query to the recommender system, fictitiously declaring that D is her preferred POI. The recommender system responds that people liking D, also liked POIs A, B and C. Since the latter are uncommon POIs, and Bob publicly expressed a preference for them, Alice derives that, with high probability, D is also a preferred POI of Bob’s.

In this paper, we introduce the *POI-Ti-Dico* system, which provides recommendations based on user’s location, interests, and preferences for POIs expressed by individuals having the same profile of the user. *POI-Ti-Dico* applies differential privacy [10] methods to extract privacy-conscious statistics about POI preferences expressed by users. Generally speaking, differential privacy guarantees that the probability distribution of those statistics does not change significantly whether an individual’s preferred POIs are present or not in the knowledge base. Actual recommendations are generated by querying those differentially-private statistics, in order to formally guarantee users’ privacy.

The rest of the paper is structured as follows. Section II discusses related work. Section III illustrates the overall architecture of *POI-Ti-Dico*, its algorithms, and shows how users’ privacy is protected. Section IV concludes the paper.

II. Related work

In this section, we review existing mobile recommender systems, showing the lack of methods to enforce users’ privacy. Then, we illustrate proposed privacy-preservation techniques for generic recommender systems.

A. Mobile recommender systems

Many techniques to provide recommendations of items of interest have been proposed in the literature [11]. However, most existing techniques are targeted to items that do not have a spatial characterization, like movies or books. Hence, location and context are not taken into account by those systems. Techniques to include context-aware features into recommender systems have been investigated by Adomavicius and Tuzhilinin [12]. However, while those techniques take into account the user’s context, they do not consider the items’ location; hence, they cannot be seamlessly applied to localized resources like points of interest.

The mobility of users, as well as the spatial characterization of possible resources of interest, claim for recommender systems specifically targeted to mobile computing. Horozov et al. proposed *GeoWhiz* [1], a mobile recommender system based on collaborative filtering. GeoWhiz exploits correlations among the preferences of people living in proximity to suggest possible restaurants of interest. A recommender system for generic POIs has been proposed by Kang et al. in [2]. In that work, recommendations are generated by first selecting a set of POIs based on user’s profile; then, selected POIs are ranked based on opinions of users having similar preferences. More recently, Gavalas and Kenteris proposed a mobile recommender system to assist tourists in choosing places to visit [4]. Recommendations are based on context-aware ratings of users having a similar profile; context data such as time, location, and weather are considered for producing recommendations. Baltrunas et al. proposed *ReRex* [5], a customizable system for context-aware recommendations of POIs, providing an automatic explanation for the received recommendations.

Even if the above mentioned methods involve the release of sensitive user information (context data and preferred POIs), no technique to enforce users’ privacy is proposed in those works. A technique for privacy-conscious recommendation of POIs has been presented by Sato et al. in [3]. That technique is based on the obfuscation of users’ interests, by applying a perturbation to the users/interests matrix. However, the proposed method is not supported by formal privacy guarantees, especially in the presence of external knowledge available to an adversary. On the contrary, our technique, being based on the use of differential privacy methods, provides strong privacy guarantees even in the presence of external background knowledge.

B. Privacy in generic recommender systems

Since most recommender systems rely on users’ sensitive information, several techniques have been proposed to incorporate privacy protection into those systems. Canny proposed in [13] the use of homomorphic encryption to aggregate individual users’ ratings of items. A similar system, using randomized perturbation instead of encryption, has been proposed by Polat and Du in [14]. Secure multiparty computation has been used by Aïmeur et al. in *Alambic* [15] to achieve essentially the same goal of the above systems. Those and similar techniques avoid the disclosure of sensitive microdata to untrusted recommender systems; however, they do not protect against background knowledge attacks, like the one illustrated in Example 1.

Recently, the problem of defending privacy against background knowledge attacks has been addressed by applying *differential privacy* [10] methods. Generally speaking, differential privacy guarantees that the probability distribution of query answers (e.g., recommendations obtained by users) is the same, irrespective of whether or not a user’s data is present in the knowledge base. Formally, a randomized computation C satisfies ϵ-differential privacy if, for any possible datasets A and B that differ in at most one record, and any subset S of possible outcomes of C:

$$\mathbf{Pr}[(C(A) \in S)] \ \leq \ \exp(\epsilon) \ \times \ \mathbf{Pr}[(C(B) \in S)], \qquad (1)$$

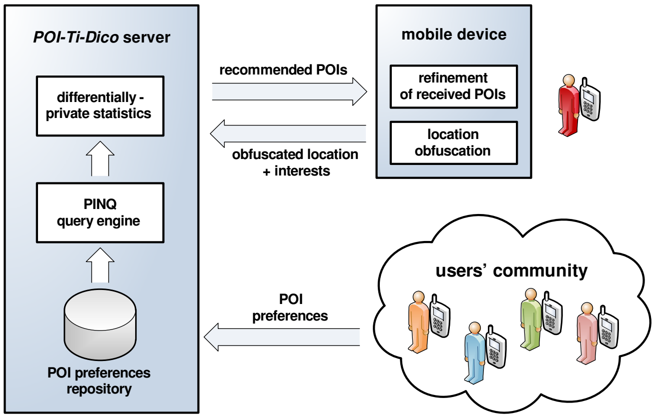
POI-Ti-Dico server
differentially - private statistics
PINQ query engine
POI preferences repository
recommended POIs
obfuscated location + interests
mobile device
refinement of received POIs
location obfuscation
users' community
POI preferences

Figure 2. System architecture

user_ID	stereotype_ID	poi_1	poi_2	poi_3	...	poi_n
1	4	*null*	1	*null*	...	1
2	7	1	*null*	*null*	...	*null*
3	4	*null*	*null*	*null*	...	1
...	...	...	...	...	...	...

Table I
LOGICAL REPRESENTATION OF THE POI RECOMMENDATION REPOSITORY

stereotype_ID	poi_1	poi_2	poi_3	...	poi_n
1	43.785	-1.831	3.163	...	15.756
2	1.834	163.063	4.472	...	26.401
3	-0.041	29.551	83.937	...	-1.031
...	...	...	...	...	...

Table II
DIFFERENTIALLY-PRIVATE STATISTICS

where **Pr**[] represents the probability distribution of query answers. As a consequence, irrespectively of the background knowledge available to an adversary, the inference about the presence of a single record is bounded by the factor $\exp(\epsilon)$.

McSherry showed in [7] that many existing recommendation techniques can be adapted to enforce differential privacy, without significantly degrading the quality of recommendations. The technique proposed in that paper can be applied to recommender systems based on item/item similarity: a user receives recommendations by users that share a preference for the same items. For instance, a user that bought the whole series of Harry Potter's novels is likely to receive recommendations by other users that bought Harry Potter's books. While the proposed method provides formal protection against background knowledge attacks, we argue that it is not suitable to recommendations of POIs, since it does not take into account the spatial characterization of items. For instance, it is very likely that a tourist travelling to a foreign city for the first time has no preferred POIs in common with people who traveled or live in that city. Hence, no recommendations would be produced using the above mentioned technique. In order to overcome this problem, in our system we provide recommendations based on users' profile, instead of preference for common items. By applying differential privacy methods, we formally guarantee privacy against background knowledge attacks.

III. THE *POI-Ti-Dico* SYSTEM

In this section, we describe the *POI-Ti-Dico* system and algorithms, and we discuss the achieved privacy guarantees.

A. Architecture

The overall system architecture is shown in Figure 2. Each user belongs to a given *stereotype* [16]; i.e., a semantic abstraction of profile data such as age, gender, education level, etc. Stereotypes are organized in a hierarchy. Users indicate their stereotype when they register to the *POI-Ti-Dico* service. The server includes a database of POIs. Each POI belongs to one or more categories, like restaurants, pubs, museums, theaters, shops, etc.

Preferences for POIs, expressed by the users of the community, are stored in a local repository of the *POI-Ti-Dico* server. Of course, the same POI may receive preferences from multiple users. The logical representation of the POI preference repository is shown in Table I. The first column field stores the ID of the user. The second column field stores her stereotype. The following column fields (one for each POI) have value 1 if the user expressed a preference for that POI; *null* otherwise. Since the repository is extremely sparse, users' preferences for POIs are internally stored in a more compact form than the one shown in Table I.

In order to extract statistics from the POI preference repository, we use the PINQ [17] query engine, which enforces ϵ-differential privacy by adding random noise to aggregate query answers. Random noise is drawn from a symmetric exponential (Laplace) distribution, with scale parameter ϵ. For each stereotype S, we count the number of preferences expressed by users of stereotype S for each POI in the repository. Table II shows how the derived statistics are stored by the server. Note that the result of some count queries may be negative, due to the presence of random noise. User IDs do not appear in those statistics, that are actually queried to produce POIs recommendations.

The spatial domain of the service is partitioned into a discrete number of non-overlapping regions, called *spatial granules*. Hence, each POI belongs to one and only one spatial granule. Figure 3 shows the spatial granules that partition the city center of Milano. When a user of stereotype S asks for POI recommendations, she specifies her current categories of interests I (e.g., pubs and discos), and the spatial granule G in which she is currently located. The use of spatial granules avoids the release of precise users' location to the recommender system, for privacy reasons.

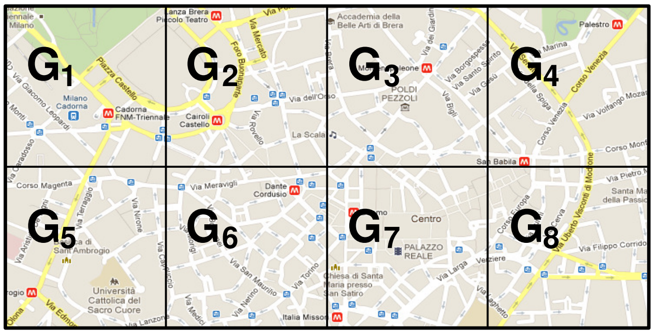

Figure 3. The eight spatial granularities for the city center of Milano

```
Input: The original POI recommendation repository
       prefDB; the set of stereotypes $S$; the set of
       POIs $P$; the privacy budget $b$.
Output: The differentially-private statistics DPS.
1  DPS-extraction(prefDB, $S, P, b$)
2  begin
3      $\overline{b} = \frac{b}{|P|}$
4      forall stereotype $s \in S$ do
5          $U_s$ = prefDB.getTuples($s$)
6          agent = new PINQAgentBudget($b$)
7          $\overline{U_s}$ = new PINQueryable<ArrayList>($U_s$,
           agent)
8          forall poi $p \in P$ do
9              $u_p$ = from $u$ in $\overline{U_s}$ where $u$.Contains($p$)
               select $u$
10             DPS[$s$][$p$] = $u_p$.NoisyCount($\overline{b}$)
11         end
12     end
13     return DPS
14 end
```

Algorithm 1: Extraction of differentially-private statistics

Then, the *POI-Ti-Dico* server queries the differentially-private statistics to retrieve the top-k POIs for the user; i.e., those POIs in G that received the highest number of preferences from users of stereotype S, and belonging to an interest category in I. If too few POIs satisfy the required conditions, the query is smoothed by generalizing its spatial extent, until the desired number of POIs is obtained.

B. Algorithms

The pseudo-code of the algorithm for the extraction of differentially-private statistics is reported in Algorithm 1. The algorithm takes as input the original POI recommendation repository *prefDB*, the set S of stereotypes, the set P of POIs, and the privacy budget b; the latter corresponds to the ϵ parameter in formula (1) reported in Section II-B. The higher the privacy budget, the more information is released to a possible adversary. Since each query consumes part of the available privacy budget, at first (line 3) the actual budget $\overline{b}$ to be spent for each query is calculated by dividing the total budget by the number of queries to be submitted (one for each POI). After retrieving the original tuples U_s regarding users with stereotype s from the POI preference database (line 5), we instantiate a PINQ [17] agent to manage the privacy budget (line 6). The PINQ agent is in charge of guaranteeing that the budget is not exceeded when answering to multiple queries over the same dataset. In line 7, we instantiate an object $\overline{U_s}$ with the preference tuples, which can be queried in a differentially-private fashion according to the agent policies. Then (lines 9 and 10), we query $\overline{U_s}$ to count, applying differential privacy, the number of recommendation by users of stereotype s for each POI p; for each query, we spend a budget $\overline{b}$. We repeat this procedure for each stereotype in S (lines 4 to 12). Finally, the obtained differentially-private statistics *DPS* are returned.

```
Input: The differentially-private statistics DPS; the
       required number of recommendations $k$; the
       spatial granule $g$ that includes the user; the
       user's stereotype $s$; the user's interest categories
       $I$; the set of POIs $P$.
Output: The recommended POIs $R$.
1  POI-recommendations(DPS, $k, g, s, I, P$)
2  begin
3      $G = \{g\}$
4      $R = \emptyset$
5      while $|R| < k$ do
6          $C$ = $P$.selectPOIs($G$, $I$)
7          $R_C$ = DPS.getRecommendedPOIs($C$, $s$)
8          if $|R_C| < k$ then
9              $G = G \cup$ getContiguousGranules($G$)
10         end
11         else
12             $R = R_C$.getTop($k$, $s$)
13         end
14     end
15     return $R$
16 end
```

Algorithm 2: Retrieval of POIs recommendations

The above statistics are used by the *POI-recommendations* algorithm (Algorithm 2) to retrieve actual POIs recommendations upon user's request. The algorithm takes as input also the requested number k of recommendations, the spatial granule g that includes the user, her stereotype s, her interest categories I, and the set of POIs P. The set G of spatial granules in which POIs are searched is instantiated with g (line 3), and the set R of POIs to be recommended is instantiated with the empty set (line 4). Then (lines 5 to 14), the algorithm queries the *DPS* until the desired number of POIs is reached. At first, the initial candidate set C of POIs (i.e., those in G and belonging to at least one category in I) is retrieved (line 6). From that set, we select those POIs that received a large number of preferences from users of stereotype s, to create a new set R_c of POIs to be actually recommended to the user (line 7). If R_c contains too few POIs (lines 8 to 10), we enlarge the query region G to include contiguous spatial granules, and we repeat the algorithm from line 5. Otherwise (lines 11 to 13), we get the k POIs in R_c that received the highest number of preferences from users in s, and return them to the user.

C. Privacy protection

As anticipated, our defense against background knowledge attacks by malicious users of the recommender system is based on differential privacy. Algorithm 2 responds to users' requests by querying the differentially private statistics about POIs recommendations, extracted by Algorithm 1. Since those statistics are produced by a differentially-private query engine, they are essentially the same, irrespective of whether a user's data (i.e., her preferences for POIs) opt in or out the recommendation repository. This property guarantees that no personal information about a specific user can be gathered by an adversary by mining the query answers.

Even if in this work we assume that the recommender system is trusted, our solution includes a simple mechanism to enforce location privacy. Indeed, when asking for POIs recommendations, users may not want to disclose their exact location to the recommender system for privacy reasons. Location privacy is enforced in our system by obfuscating the user's position through the substitution, in the request generated by the user's client, of the user's exact location with the spatial granule in which it is included. The spatial granule determines the uncertainty of the recommender system about the exact location of the user. Our system can be trivially extended to support multiple granularities. Depending on her privacy preferences, the user can tune the level of achieved location privacy by choosing an appropriate granularity for obfuscating her location.

IV. Conclusions and future work

In this paper, we reported an initial investigation of the challenging research issue of providing formal privacy guarantees in a system for context-aware recommendation of POIs. We have proposed a technique based on differential privacy to extract statistics about personal preferences for POIs, and a mechanism for generating profile- and location-based recommendations from those statistics. Our solution also includes a simple form of location privacy based on the generalization of the exact user's location in service requests.

Future work includes the development of a prototype of the proposed technique, as an extension of the *POIsafe* [18] system for privacy-conscious sharing and retrieval of an extended form of POIs, called POIsmarts, which are the convergence between physical and virtual POIs (the latter being essentially Web resources related to physical spots). We will also perform extensive experiments about the quality of service achieved by our technique. Indeed, there is an obvious tradeoff between the allocated privacy budget and the precision of extracted statistics. A possible solution to improve the quality of service is to adopt a relaxed form of differential privacy, like the one proposed in [19]. The technique proposed in [19] could also be applied to provide users' preferences for POIs to an untrusted recommender system under the guarantees of differential privacy.

V. Acknowledgments

This work has been partially supported by the research fund "Dote ricercatori": FSE, Regione Lombardia.

References

[1] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments," in *Proc. of SAINT'06*. IEEE Comp. Soc., 2006, pp. 124–129.

[2] E. Kang, H. Kim, and J. Cho, "Personalization method for tourist point of interest (poi) recommendation," in *Proc. of KES'06*, ser. LNCS, vol. 4251. Springer, 2006, pp. 392–400.

[3] H. Sato, T. Inoue, H. Iwamoto, and N. Takahashi, "Virtual scent: Finding locations of interest in ambient intelligence environments," in *Proc. of PDCAT'09*. IEEE Comp. Soc., 2009, pp. 384–389.

[4] D. Gavalas and M. Kenteris, "A web-based pervasive recommendation system for mobile tourist guides," *Personal and Ubiquitous Computing*, vol. 15, no. 7, pp. 759–770, 2011.

[5] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, "Context-aware places of interest recommendations for mobile users," in *Proc. of DUXU'11*, ser. LNCS, vol. 6769. Springer, 2011, pp. 531–540.

[6] S. K. Lam, D. Frankowski, and J. Riedl, "Do you trust your recommendations? An exploration of security and privacy issues in recommender systems," in *Proc. of ETRICS'06*, ser. LNCS, vol. 3995. Springer, 2006, pp. 14–29.

[7] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proc. of ACM SIGKDD*. ACM, 2009, pp. 627–636.

[8] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, "You might also like: Privacy risks of collaborative filtering," in *IEEE Symposium on Security and Privacy*. IEEE Comp. Soc., 2011, pp. 231–246.

[9] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations - accurate or private?" *PVLDB*, vol. 4, no. 7, pp. 440–450, 2011.

[10] C. Dwork, "Differential privacy," in *Proc. of ICALP'06*, ser. LNCS, vol. 4052. Springer, 2006, pp. 1–12.

[11] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[12] ——, “Context-aware recommender systems,” in *Recommender Systems Handbook*. Springer, 2011, pp. 217–253.

[13] J. F. Canny, “Collaborative filtering with privacy,” in *IEEE Symposium on Security and Privacy*. IEEE Comp. Soc., 2002, pp. 45–57.

[14] H. Polat and W. Du, “Privacy-preserving collaborative filtering using randomized perturbation techniques,” in *Proc. of ICDM'03*. IEEE Comp. Soc., 2003, pp. 625–628.

[15] E. Aïmeur, G. Brassard, J. M. Fernandez, and F. S. M. Onana, “Alambic : a privacy-preserving recommender system for electronic commerce,” *Int. J. Inf. Sec.*, vol. 7, no. 5, pp. 307–334, 2008.

[16] E. Rich, “User Modeling via Stereotypes,” *Cognitive Science*, vol. 3, no. 4, pp. 329–354, 1979.

[17] F. McSherry, “Privacy integrated queries: an extensible platform for privacy-preserving data analysis,” *Commun. ACM*, vol. 53, no. 9, pp. 89–97, 2010.

[18] D. Riboni, L. Pareschi, and C. Bettini, “Integrating identity, location, and absence privacy in context-aware retrieval of points of interest,” in *Proc. of MDM'11*. IEEE Comp. Soc., 2011, pp. 135–140.

[19] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, “Our data, ourselves: Privacy via distributed noise generation,” in *Proc. of EUROCRYPT'06*, ser. LNCS, vol. 4004. Springer, 2006, pp. 486–503.