University of Southern California
Viterbi School of Engineering
Department of Computer Science

Mechanisms for Co-Location Privacy

Nithin Krishna Ottilingam

Master of Science in Computer Science
University of Southern California, Los Angeles
Prof. Cyrus Shahabi, Chair

Submitted in part fulfillment of the requirements for the degree of
Master of Science in Computer Science, December 2017

Abstract

The ever-growing proliferation of location based services produces vast quantities of user location data that is susceptible to sensitive inferences. This work focuses on the information generated by simultaneous occurrences of individuals in the same geographic locale at roughly the same time, termed *co-locations*. Extracting sensitive information from co-location data, i.e. the social ties inferred from the physically-embedded social structure, is well studied. However, protecting users against such inference attacks has not received any attention. This work marks the first attempt at studying the problem of inhibiting sensitive inferences on co-location exposures. We present a general framework to co-location privacy that captures the privacy-utility trade-off in co-location information. We propose two techniques to co-location privacy: (i) inaccuracy and imprecision producing technique such as co-location k anonymity, and (ii) distortion techniques such as co-location perturbation. We investigate the behavior of the privacy mechanisms and qualitatively assess their results on real datasets.

The thesis by Nithin Krishna Ottilingam was *approved* by.

Prof. Aleksandra Korolova

Prof. Craig Knoblock

Prof. John Heidemann

Prof. Cyrus Shahabi *(Committee Chair)*

University of Southern California, Los Angeles, December 2017

Acknowledgement

I extend my gratitude to Professor Cyrus Shahabi. His inputs were pointed and insightful and kept us focused to problem at hand, *(which was often tough)*. I thank Ritesh my collaborator, for all the life lessons, grueling late night white board discussions, numerous take-outs and mortal kombat sessions on the campus play station. Finally my Aunt's family in Fremont CA, for all the love and tasty Indian food.

Dedication

To the various 'moonjis' for helping me get through grad school.

In no particular order:

Zam, A.Veriyan, Handi, Lava, Ranj, Chyut, Batty, AJ, Sindu, Raj, Kaza;

The IR-Dudes: Karanjeet, Madav, Thamme; and

The IT-Folk from back home.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The pervasive use of ever-advancing technologies in geographical positioning systems (GPS), Wi-Fi localization, and cellular location identification has produced powerful Location Based Services (LBS) that cover large populations. Location sharing has become a ubiquitous mechanism in most mobile devices. Geo-Social Network (GeoSN) applications enhanced with location capability provide rich functionality to their users in the form of recommendation services, and at the same time empower lucrative location-based advertising for businesses. However, as location systems record user movements automatically, they generate an enormous amount of potentially sensitive information. For example, if two people are in the same geographical locale at roughly the same time, it may imply a social connection between them [13]. In due course, these location-based services gather a wealthy source of structured information on their users' behavior in space and time [8, 16]. Of particular interest in this paper, is the location-embedded social structure that can be derived from *co-location* data. While such a physically embedded social network helps bridge the gap between the physical and virtual worlds, it poses new challenges to personal data and privacy protection due to the ever-evolving nature of abuses.

People generally do not place a high value on the privacy of their location information [11, 21, 19], however they remain unaware of associated privacy risks that undergird co-location data.

Researchers have well established the strong form of privacy leakage associated with information of people proximate in space and time. Several studies extract pairwise information about links in the underlying social network from sparse individual location information in the physical world [28, 13, 9]. In turn, other studies can predict future locations from the inferred social links [12, 37], thereby cementing, in essence, a complete knowledge of the interdependent social and spatio-temporal behavior of individuals. Even in cases where a targeted user does not disclose any location information, historical co-locations or co-location type information released by friends of the user (e.g. face-tagged photos) can significantly decrease the overall location privacy of the friend pair [27].

Irrespective of the potential privacy leakage, users of LBSs share their locations in order to request services from mobile applications. For example, Google offers street navigation, and recommendations of top rated restaurants and nearby attractions; all of which rely on accurate location updates generated by the users. Likewise, other applications such as Facebook, Yelp and Foursquare rely on location identification to power the check-in functionality at points-of-interest and to prompt users to rate and review the establishments. The accuracy of a user's location is important to both the quality of the service he receives and of the location-based advertising that finances the application. But at the same time, location information exposes the social ties embedded in the information about position of users and their friends at the same points-of-interest and at roughly the same time. Any distortion in the location information that may protect a user's co-location privacy will incur a loss in quality of results from spatial range queries(utility) that are at the core of most recommendation and advertisement services. These are conflicting goals inherent to any privacy mechanism and form the basis of our problem formulation.

While it is clear that co-location data is susceptible to highly sensitive inferences, none of the current GeoSNs support co-location privacy [36]. Recent revelations of the NSA's *PRISM* program exposes the mass surveillance of user movement using the location data acquired from Microsoft, Google, Facebook and Apple[17]. Moreover, comprehensive details in the NSA white paper reveal

the *Co-Traveler* program that entails the lookup of unknown associates of suspected targets by recording people whose movements intersect in space and time [17]. The urgency of the problem is brought into clear focus when co-location information gathered by tracking who you are traveling with or meeting, is accepted as evidence of wrongdoing and facilitates prosecution in high courts of the United States [34]. We anticipate that as the privacy leaks associated to co-location exposures become more apparent and have real-world consequences, public awareness and opinion will motivate this topic further. Thus, the focus of this paper is to enable users and business partners of location based social networks to provide recommendation and advertisement services from location data, while ensuring co-location privacy of the users. To the best of our knowledge, this paper marks the first attempt to propose and evaluate different solutions to protect against inference attacks on co-location data.

Our contributions include the following:

1. We introduce the problem of co-location privacy and conduct a systematic investigation of the problem. We assume the attacker's goal is to identify the true co-locations of everyone after which he would nefariously profit from a multitude of associated socio-contextual inferences.

2. We quantify privacy as the loss in effectiveness of an adversary's inference attack by conducting tests that mimic what an attacker would do with a large volume of location data from several individuals, assuming he has defeated any encryption or access control on the data.

3. We give a general framework to co-location privacy and define functions that capture the privacy and the utility of the mechanisms. We first adapt the widely popular Gaussian noise based methods to co-location privacy.

4. We observe that such a simple method suffers from large variance in privacy levels due to the skewed nature of data distribution in user check-in data. We observe a strong level of privacy in dense regions due to above average distortion relative to the data density, and a

severely limited protection in sparse regions.

5. We introduce an adaptive distortion technique that adjusts the magnitude of noise based on the distribution of the user locations in space and time, to remedy the problems encountered. Using these techniques we improve both, the privacy of users' co-location data and the utility of the data to location-based advertising services.

6. We propose a syntactic notion of co-location privacy guarantee called co-location k anonymity. Although requiring a significant level of data distortion, co-location anonymity guarantees a privacy level of $1/k$.

7. We qualitatively assess the results of all mechanisms and discuss their ability to capture different application requirements due to their inherent strengths and weaknesses.

The remainder of this work is structured as follows. In Chapter 2, we survey the related work and discuss the challenges to co-location data privacy as opposed to location privacy. In Chapter 3, we define and formalize the system setting. In Section 4.1, we present the basic mechanisms to achieve co-location privacy and improve upon it in Sections 4.2 and Section 4.3. In Chapter 5, we exhaustively evaluate the performance of our techniques on real-world datasets (described in Section 5.1). Finally, we conclude the paper, discuss methods and suggest directions for the future work in Chapter 6.

Chapter 2

Related Work

We overview related work on Inference attacks on co-location data (Section 2.1), and Location privacy mechanisms (Section 2.2).

2.1 Inferences on co-location data

Individuals that are proximate in time and space exhibit a strongly correlated social structure. Eagle et. al. [13] compare observational data from mobile phones with self-reported survey data to accurately infer 95% of the friendships that exhibit distinctive spatio-temporal patterns in their physical proximity and calling patterns. Crandall et. al. [9] develop a probabilistic model to investigate the extent to which social ties between people can be inferred from co-locations. The authors find that even a very small number of co-locations can result in a high empirical likelihood of a social tie if the distinction between a seemingly random interaction of two entities in space and time (a co-incidence), and an intentional face-to-face meeting between friends (a co-occurrence) is captured. Cranshaw et al. [10] formulate the problem of discovering social ties as a classification problem and extract a large number of features such as the spatial and temporal range of the set of co-locations, location diversity and specificity, and structural properties to train the friendship

predictor. Pham et. al. [28] propose an entropy-based model (EBM) to estimate the social strength value between all pairs of users that is subsequently translated to binary friendship connections using a learned threshold parameter. They extract two factors for each pair of users to train their model; (i) the entropy of the distribution of co-occurrences across locations, and (ii) the weighted sum of the frequency of co-occurrences at locations normalized by the popularity of the location. Although they achieve an astonishing accuracy of 96.5% even in the sparse check-in data of GeoSNs, the total percentage of users for whom friendships are inferred (recall) remains to be fairly low. However, note that in the context of privacy concerns, a moderately large absolute number of affected individuals can represent a significant effect, even if most of the population is not implicated [9].

Moreover, recent research indicates that discovery of social ties can have far reaching implications on overall privacy of a user. Given a small fraction of the users, their social connections, and a seed set with a few of their location updates, Jurgens [20] propose a method that accurately infers locations for nearly all of individuals by spatially propagating location assignments through the social network. Following this observation, Olteanu et. al. [27] try to quantify the impact of indirect forms of co-location information on overall location privacy. The authors postulate that enough indirect co-locations can be procured from various sources including automatic face recognition on geo-tagged photos, users who connect from the same IP address (and hence attached to the same Internet access point) and Bluetooth-enabled device sniffing. The authors of the paper note a decrease by up to 62% in median location privacy when co-locations are considered with only a single friend of the target user. Additionally, even in situations when a target user chooses to hide his true location, his location privacy is reduced by up to 21% due to the information reported by other users. These revelations clearly intensify the need for co-location privacy, especially in location-based social networks that facilitate some control over a user's location privacy, all the while imposing no restrictions on co-locations publicized by his friends.

Although protecting against inference attacks that uncover social ties is an important problem,

we focus on the more general problem of protecting against all types of attacks on co-location data, beyond those just on the location-embedded social network. More precisely, the discovery of social ties is found to rely on a special set of features [28, 10] and it may be possible that privacy-preserving mechanisms that specifically exploit those feature perform better than a more general method. However, in this paper we abstract away from an application specific formulation and instead consider all co-locations equally important to a user's privacy.

2.2 Location privacy and obfuscation mechanisms

Field of location privacy has been a very active area of research in past years. The literature can be classified into the following broad categories: (i) Location hiding, (ii) Fake Location injection, (iii) Location cloaking (adding confusion) and (iv) Location perturbation (adding noise).

The basic mechanism of *Location Hiding* is constructed in the following way: For each location check-in of a user, with parameter λ as the probability of hiding the check-in, the privacy mechanism removes it from being published. Simply put, with probability λ, a given location will be hidden from adversarial observation. In [31], Shokri et. al. conclude that on-its-own the protection under this scheme is insufficient. Additionally, usage of location hiding in real-world applications is severely limited, since it is often necessary to reveal some location information in order to request services. Consequently, we do not evaluate location hiding in this paper. *Fake Location injection* is yet another seemingly straightforward privacy mechanism that publishes fake location reports interleaved or along with the true locations of the users[7][24]. There are few simple techniques proposed so far: adding independently selected fake locations drawn from the population's location distribution [31], generating dummy locations at random as a random walk on the grid [39], and very recently, fabricating entirely synthetic traces that mimic real mobility patterns in terms of transition probability across activities (e.g. working, driving, staying at home). The effectiveness of this technique in protecting user's privacy highly depends on the resemblance of

fake information to reality, otherwise an attacker may filter out all but the real data. This has been an especially difficult problem, and until recently existing approaches did not evaluate how plausible and privacy-preserving their synthesized traces were. In [7], Bindschaedler et. al. show that if an adversary exploits the geographic and semantic similarity in mobility patterns across all users, it is possible for him to filter out a significant portion of the fake location information. The authors demonstrate that methods such as i.i.d location sampling [31] and sampling locations from a random walk on a grid with uniform probability [39] fail to protect location privacy against inference attacks. Finally, when the utility of synthetic data is discussed, there can be many side effects to having dummy locations dispersed among real ones. For example, in applications that require active user participation, serving dummy users, and processing dummy messages, is a waste of resources. In pervasive computing environments, dummy users might have to control physical objects—opening and closing doors, for example—or purchase services with electronic cash [5]. Similarly, for data-mining tasks it is important to preserve general statistics about human mobility profiles. All these intricacies apply as-is to the co-location domain and go beyond just that. For example, consider two users with unrealistic location traces, a fabricated co-location between them—regardless of its realism—might be easily identified as fake based on the improbability of their individual location traces. In this paper, due to the complexity of generating realistic dummy user locations and the equally difficult task of measuring their protection against inference attacks, a technically sound solution, that in addition to generating realistic user location updates, also introduces fake yet seemingly realistic co-locations between users that appear close in the physical space, is outside the scope of this paper. However, we include the discussion here for the sake of completeness. While methods such as Location Hiding and Fake Co-location injection are straightforward in their implementation, their application is usually limited to specialized queries. Therefore, in this paper we do not evaluate them.

The first attempt at an intuitive definition to location privacy came in the form of a simple reduction in accuracy of location updates along spatial and/or temporal dimensions; i.e. a *Location Cloak* that obscures the exact location. In the case of mobile clients that expose their location

to the LBS, location k-anonymity obfuscates the actual location from which a user query is made by constructing cloaking regions (i.e. a coarser grained spatial range) that contain the locations of k-anonymous users. This insures that the adversary will have uncertainty in matching the mobile client to his exact location, due to his identity being indistinguishable from the $k-1$ other. Gkoulalas et. al [15] in their survey paper on location privacy make the distinction between *Space–dependent* and *Data–dependent* methods of constructing the cloaking region. Gruteser and Grunwald [18] propose a *Space–dependent cloaking* method that partition the area into a quadtree and generates the region of anonymity by retrieving the users in each cell of the grid (starting from the cell of the client at the lowest level of the tree and moving to levels above that cover a wider area) until at least k users are found. They note that time stamps on location measurements could be similarly ambiguous. On the other hand, Kalnis et. al. [22] propose a *Data–dependent* cloaking method, which given a query from the mobile client, finds his $k-1$ closest users, and produces the cloaking region as the minimum bounding rectangle or circle that encloses them. While [18] a system-wide static K, Gedik et. al. [14] consider allowing users to specify their own values of K, thus enabling mobile users to specify varying privacy protection requirements under different contexts. While this notion of privacy anonymized the user's identity among K others, it provided widely varying amount of location privacy, when measured in terms of reduction of location resolution. Xu et. al. [38] tweaked the definition slightly to allow a user to express his privacy requirement by specifying a public region, which the user would feel comfortable if the region is reported as her location.

Early methods motivated by the desire to protect a statistical database against compromise, considered the effect of Gaussian noise on security of numerical as well as nonnumerical sensitive fields; ranging from distorting all values in the database [35], distorting results of a database query [4], to selectively distorting values aimed at certain applications, e.g. in the context of privacy preserving data mining [1]. In recent years, the same notion is extended to spatio-temporal databases, wherein the intuition is that if location data is noisy, it will not be useful for inferring the actual location of the users. [3] and [31] evaluate obfuscation technique that reduces the precision of

a location by dropping the low-order bits of the x and y coordinate. Krumm [23] empirically evaluate the effect of adding noise to check-in location of users on protecting their identity from being discovered through a simple Web search. Very recently application of differential privacy to location protection has seen significant advances. Andrés et. al. [2] introduce the notion of *geo-indistinguishability* within a area. Intuitively, what it means is that, from an attackers perspective, the released location of a user reveals no more information about his whereabouts than the knowledge of his presence within a specified discretized radii r around a point-of-interest. To enable guarantees under their own definition of location privacy, they utilize a 2D Laplacian random noise on the location data.

Chapter 3

System Framework

In this section, we lay the foundation for our discussion on co-location privacy. First, we present the privacy objective of a Co-location privacy Mechanism that obfuscates co-location information. To enable a meaningful evaluation of privacy mechanisms, we define data quality measures that quantify the tradeoff between utility and privacy that is inherent to privacy-preserving systems. Any entity that has access to the published data consisting of the obfuscated user locations is considered as the adversary (or the observer) who has the potential to perform sensitive analysis on the data. The performance of the adversary and his success in recovering the original co-locations is also precisely measured to gauge the level of privacy. The summary of notations used are given in table 3.1.

3.1 System setting

We consider $U = u_1, u_2, ..., u_N$ a set of N users that are part of the location-based service. When a user sends a request to the service provider, a minimum of following information is recorded: the user's identifier, his position as a coordinate (latitude and longitude), and the time of request. This information is captured as a *user id, position, time* triple denoted as $\langle u, p, t \rangle$, also called a

Notation	Definition
$c < p, t, u >$	Check-in of user u at time t at location with co-ordinates p
$\|c_i.p, c_j.p\|$	Euclidean Spatial distance between c_i and c_j
$\|c_i.t, c_j.t\|$	Absolute temporal difference between check-in times of c_i and c_j
MAX_s	Spatial noise bound
MAX_t	Temporal noise bound
Δ_s	Maximum spatial distance between a pair of check-ins for a co-location
Δ_t	Maximum temporal distance between a pair of check-ins for a co-location

Table 3.1: Basic Notations

check-in for simplicity. The position of a user in space is considered to be discrete and the set of positions at which a user may be seen is $P = 1, 2, ..., P$ [1]. Likewise, time is considered to be discrete, and the set of time instants when the position of users may be observed is $T = 1, ..., T$. The history record of check-ins of a user i is represented as a vector $C_i = c_i^1, c_i^2, ..., c_i^j$. The set of check-ins of all the users in the system is denoted as $C = C_1 \cup C_2 \cup ... \cup C_N$. For simplicity, we use the notations $c.p$, $c.t$ and $c.u_{id}$ to denote the position, time and user id of a check-in c.

Furthermore, check-ins of two users are said to have co-located if they are in the same geographical locale at roughly the same time. More precisely, given a spatial distance Δ_s and a temporal distance Δ_t, check-ins of user u and user v are said to have co-located if they are within Δ_s spatial distance and Δ_t temporal distance to each other. The distance function used to characterize a co-location are orthogonal to this study. In this paper, we use the Euclidean distance to measure spatial distance (i.e. $\|c_u^i.p, c_v^j.p\| <= \Delta_s$) and the absolute difference in time to capture the temporal distances (i.e. $|c_u^i.t - c_v^j.t| <= \Delta_t$). The set of all co-locations is represented as $CL = cl_1, cl_2, cl_3 \ldots cl_M$, wherein an element cl_i of type (c_u^i, c_v^j) denotes a co-location between check-ins c_u^i and c_v^j. Figure 1

[1] The total number of positions are limited by the maximum precision of the data type used to represent the geographical coordinate and/or the precision of the device used to record the coordinates

illustrates the computation of all co-locations from a given set of check-ins C.

Algorithm 1 Generating all Co-Locations in the published data

```
1: procedure GENERATE CO-LOCATIONS(C)
2:     CL ← {}
3:     for c_i in C do
4:         for c_j in C do
5:             if ||c_i.p, c_j.p|| <= Δ_s and |c_i.t, c_j.t| <= Δ_t and c_i.u_id ≠ c_j.u_id then
6:                 CL ← (c_i, c_j)
7:     return CL
```

Lastly, we model the data as an undirected graph $G = (C, CL)$, where a node $c_u^i \in C$ represents a user's check-in and an edge $cl = (c_u^i, c_v^j) \in CL$ indicates a co-location between $c_u^i, c_v^j \in C$. The co-location network is the input to a Co-location Privacy preserving mechanism. In this paper, we study the case where a co-location of a user is independent of other events which belong to that user or other users. Following the distinction made by [30] between sporadic and continuous location exposures, In this paper we consider the former and assume no time dependance of one location update on another. Common uses of Location-sharing services involve search for nearby points-of-interest by the users, and potentially the flipside, i.e. the receipt of advertisements when in proximity to relevant businesses. In these situations, the location updates (and likewise the co-location updates) of a user are sparsely distributed over time, and thus can be reasonably assumed to be independent [2].

[2]On the contrary, if the users' locations are continuously exposed, a spatio-temporal correlation can be observed from one location update to the next [33]. These correlations can be exploited by an adversary to attack more successfully the location privacy (and potentially co-location privacy) of users [30, 31]. Mechanisms that address these concerns, however, are outside the scope of this paper.

3.2 Co-location privacy preserving mechanism CoPPM

The data publisher implements a global co-location privacy preserving mechanism that obfuscates the co-location information in order to deteriorate the performance of an adversary's attack. Given a co-location $cl = (c_u^i, c_v^j)$, the purpose of the mechanism is to apply distortion or introduce confusion to the exposed positions $c.p'$ and timestamps $c.t'$ of the constituent check-ins $c \in \{c_u^i, c_v^j\}$[3]. More precisely, given $G = (C, CL)$, the CoPPM uses a mapping function μ to construct another graph $G' = (C', CL')$, i.e. $\mu(G) = G'$. We consider several potential methods to protect against inference attacks on co-location privacy data as follows:

- Co-location anonymization (adding confusion)
- Co-location perturbation (adding noise)

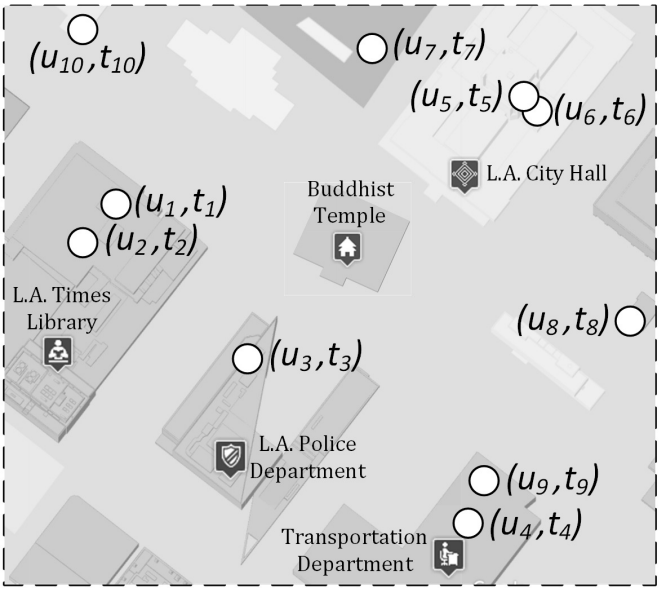


Figure 3.1: Running Example of a co-location network

Figure 3.1 illustrates the methods through a running example, wherein, for the sake of simplicity, two check-ins are assumed to be co-located if they are in the same place (library, city hall, etc)

[3]Note that obfuscating, at random, either check-in of the co-location pair may suffice to obfuscate the co-location successfully (i.e. no longer satisfy Δ_s and Δ_t constraints

and are checked-in at consecutive timestamps (i.e. $t_1 - t_2 \leq \delta_t$, but $t_4, t_9 > \delta_t$). Accordingly, there are two co-locations: $cl_1 = (u_1, u_2)$ and $cl_2 = (u_5, u_6)$. Without going into precise details, we give some examples of co-location mechanisms in the particular example. An instance of Co-location anonymity could potentially move u_3's check-in such that it appears to have co-located with u_1 and u_2. This would guarantee that the true co-location cl_1, is indistinguishable from the now co-located pairs (u_1, u_3) and $(u_2, u_3$. We formally define co-location k anonymity in Section 4.3. Co-location perturbation adds noise to the spatial and temporal values of a check-in; for instance, moving check-in of user u_2 to the Temple would essentially break the co-location cl_1 by violating spatial Δ_s and temporal Δ_t distance constraints. Each method is discussed in depth and corresponding processing algorithms are presented in Section 4.2.

While the data publisher wants to do his best at protecting the co-locations from potential inferences, the perturbation of the information published leads to a degradation of the quality of queries that can be performed on the obfuscated data. Consequently, there is a trade-off between the level of privacy that the data publisher wishes to guarantee and the service quality loss that he will have to accept. We define the loss of quality as a linear combination of the spatial distance between the real location and the reported location, and the temporal difference between the real timestamp and the reported timestamp [4]. The function captures the deterioration in the utility of spatio-temporal queries that are common to location-based recommendation and advertisement services. For an obfuscated check-in c_u^i of user u, the quality loss QL of a query that references him is defined as

$$QL_u^i = \alpha \cdot \frac{\left\| c_u^i.p, c_u^i.p' \right\|}{MAX_S} + (1 - \alpha) \cdot \frac{|c_u^i.t, c_u^i.t'|}{MAX_T} \tag{3.1}$$

where α is a weighing parameter which trades-off the magnitude of spatial distortion for temporal distortion. For example, given a fixed value of quality loss, if $\alpha = 0.33$ then for the same absolute value of spatial distortion to a check-in, the temporal distortion applied is double that of when

[4]The semantics of these distances depend on the LBS under consideration, and also on the user's specific service-quality expectations. For example, in a road network based application, one might use the Manhattan distance (i.e. L_1 norm) or the network distance to measure the spatial quality loss.

α is 0.5. The value of α compensates for the difference in sensitivity of the spatial and temporal dimensions towards producing a desired level of privacy. Factors MAX_S and MAX_T normalize the spatial and temporal distances in the range [0,1]. These normalizing factors are typically set to the maximum distortion that can be applied to a co-location. We experimentally determine the parameters and also discuss their qualitative effect on the privacy mechanism in Chapter 5. We also report the total quality deterioration averaged over all co-locations.

3.3 Adversary Attacks and Background knowledge

In order to evaluate our mechanisms, we must model the adversary knowledge to a reasonable degree. For example, it is popularly accepted that the perturbation mechanism is transparent to the adversary. However, consider a scenario in which the adversary has access to an alternate information source that may make identification of private, individual-level co-location data deducible. In other words, an attacker with a smaller set of potential victims could afford more time-consuming means of compiling co-location data on them by physically staking out their neighborhood or manually inspecting their locations and meeting with other users. Nevertheless, our attacks are limited to computation. In reality, there may be a variety of adversaries with varying degree of prior knowledge.

We assume that the adversary obtains and constructs his knowledge based on statistical analysis of the observed locations of the users and of the complete knowledge of the implementation of the protection mechanism. Hence, he can use the information leaked by the privacy mechanism to reduce his uncertainty about the user's true location. We quantify the user's location privacy as the adversary's error in his inference attack, i.e. the distortion in the reconstructed co-location. Put simply, the fraction of true locations that are missed by the adversary is our privacy metric. We represent as RCL the set of co-locations that are reconstructed by the adversary. Naturally, the adversary will see a distortion in his inference attack proportional to the level of obfuscation

administered by the privacy mechanism. We define Inference Accuracy as the fraction of correct co-location instances among all the inferred instances, and Inference Recall as the fraction of correct co-locations that have been retrieved over total co-location instances in the data. More precisely,

$$\text{Inference Accuracy } IA = \frac{CL \cap RCL}{RCL}$$

$$\text{Inference Recall } IR = \frac{CL \cap RCL}{CL}$$

In sections 4.1 to 4.2, we propose methods to preserve co-location privacy and present corresponding processing algorithms. We summarize our results and compare their strength and weakness in Chapter 5.

Chapter 4

Co-location Privacy Preserving Mechanisms

We first present a baseline gaussian perturbation approach and an adversary strategy for efficient restoration, an adaptive perturbation approach which addresses the pitfalls of the baseline method and finally a syntatic approach using k-anonymity. In the following sections we formulate each of these methods.

4.1 Naïve Co-location Perturbation

Gaussian perturbation mechanism. As a baseline we implement Gaussian perturbation to distort co-location information. It is interesting to note that in the case of co-locations, distorting the position of just one check-in in the co-location pair suffices in breaking the relation. The mechanism is constructed in the following way: For a randomly selected check-in $c \in cl \wedge cl \in CL$, we generate a spatial noise vector and a temporal scalar noise magnitude each derived from the Gaussian normal distribution. The spatial noise vector (2-Dimensional) is generated with a random uniform direction over $[0, 2\pi)$ and a Gaussian-distributed magnitude from (N, σ_s^2). A negative

magnitude reverses the direction of the noise vector. The temporal scalar noise (1-Dimensional) is generated as a Gaussian-distributed magnitude from (N, σ_t^2). Finally, the position of check-in $c.p$ is transformed along the spatial noise vector and the timestamp $c.t$ is distorted by the magnitude of temporal scalar noise.

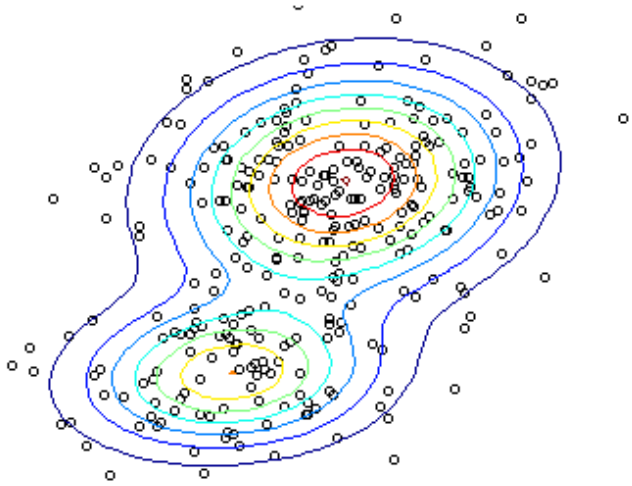


Figure 4.1: 2-D Gaussian Mixture Distribution

Restoration. In order to evaluate the potential privacy provided by the Gaussian perturbation mechanism, we implement a co-location restoration scheme that given the distorted co-location network G', reconstructs the original network G. The restoration scheme exploits key observations in the data distribution of the datasets, such as the fact that the number of co-locations at any place (e.g. home < restaurant < shopping mall) is directly correlated to the unique number of users that checked-in there (also called *diversity*)[10]. We also note that spatial Gaussian noise across all places in the data creates a Gaussian mixture distribution between these places [6]. Figure 4.1 illustrates this concept. The algorithm visits places in decreasing order of their diversity. For each check-in at a place, it initiates a range query that finds the next closest place, say at a spatio-temporal combined distance r (computed according to equation 4.1. All check-ins that fall in this range are assumed to have originated from the location of the nearest neighbor query, hence are restored to its original co-location pair. Figure 2 gives the pseudocode of the restoration scheme. We do not claim to simulate the worstcase scenario wherein an attacker has complete background knowledge, and hence can give an accurate estimate of expected privacy level. This is achieved

when an attacker implements the optimal Bayes inference algorithm[32], which is formulated to minimize the expected user privacy. The optimal inference attack is found as the solution to a linear program, which is at present, intractable for large geo-social datasets. On the other hand, our naïve but efficient restoration heuristic performs to an acceptable degree, albeit it overestimates the privacy of perturbation mechanism. The restoration scheme also enables us to compare the baseline perturbation mechanism to our other methods, and provides insight into the behavior of the Gaussian perturbation function in skewed LBSNs.

Algorithm 2 Gaussian restoration

```
1:  procedure RESTORE-GAUSSIAN(G')
2:      C' ← {}
3:      for c in C ordered by getDiversity(c.p) do
4:          C' ← c
5:          while True do
6:              h ← and getNextNearestNeighbor(c.p,c.t)
7:              if getDiversity(h.p) = 0 then break
8:              h.p ← c.p
9:              h.t ← c.t
10:             C' ← h
11:     return C'
```

Shortcomings. It is apparent that if the co-location data is noisy, it becomes difficult to infer the actual co-location of the users. However, due to the skewed nature of the check-in distribution in most location based networks, the performance of Gaussian noise mechanism varies significantly across the data distribution. Figure 4.2 presents the histogram of normalized spatio-temporal distance to the nearest neighbors for every check-in. For any fixed value of parameters σ_s and σ_t input to the Gaussian perturbation mechanism, the distortion applied to check-ins that have nearest neighbors closer than the added spatio-temporal noise will see better privacy than those who have nearest neighbors very far as compared to the magnitude of noise. Put simply, in sparse

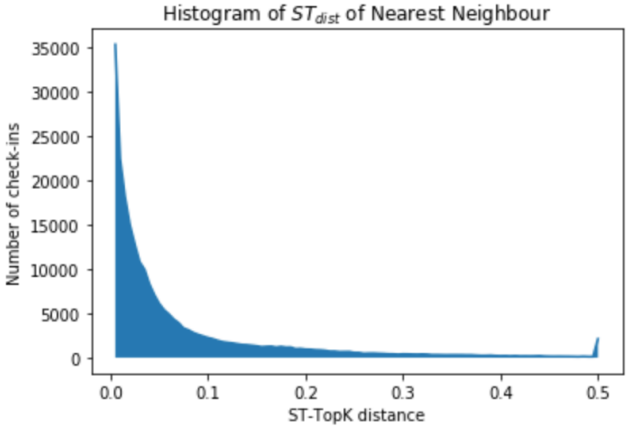


Figure 4.2: Histogram of normalized ST_{dist} to 1-NN

areas (i.e. say $ST_{dist} > 0.5$) the added noise is not sufficient, while in dense areas noise is excessive. We experimentally confirm this behavior in three real datasets in Chapter 5. To reconcile with the discussed issues, we present a mechanism which adapts the magnitude of noise according to the data properties of the considered point.

4.2 Adaptive Co-location Perturbation

Adaptive co-location perturbation mechanism adds a varying degree of noise to each co-located pair that is dependent on the distribution of its nearest neighbors. Figure 3 presents the pseudocode for the adaptive perturbation mechanism. Given as input a parameter b from the set $0, 1, \ldots N$, the mechanism iterates through connected component $S' \in S$ and for each checkin $c \in S'$, finds its b nearest neighbors. Then, with uniform probability distribution on the set of the b selected neighbors together with the check-in under consideration, moves the check-in to be co-located with a random point. In short, the check-in is perturbed to one of its b nearest neighbors with equal probability of $1/b$.

Algorithm 3 Adaptive Co-location perturbation

```
1: procedure ADAPTIVE-PERTURB(G = (C, CL),b)
2:     S ← getConnectedComponents(G)
3:     for S' = (V, E) in S do
4:         for check-in c in S' do
5:             bNN ← getBNearestNeighbor(c.p, c.t, b)
6:             i ← U{0, 1, 2..b}
7:             c.p ← i − NN.p
8:             c.t ← i − NN.t
```

4.3 Co-location k Anonymity

The Co-location anonymity mechanism extends existing notion of k-anonymity in relational and spatial databases to co-location data. The key idea of our approach is to "hide the co-location of the user inside a crowd", which can be seen as the spatio-temporal equivalent of k-anonymity. The goal is to ultimately create imprecision in the re-identification probability of the adversary. Formally, we define co-location anonymity as follows:

Definition 1. *Given a co-location network $G = (C, CL)$, a co-location $cl \in CL$ is said to be k-anonymous if it is spatio-temporally indistinguishable to $k-1$ other co-locations.*

The co-location k anonymity metric guarantees that the re-identification probability is at most $1/k$. Consider the running example in Figure 3.1, wherein the check-in $c_1 = (u_1, t_1)$ is co-located with $c_2 = (u_2, t_2)$ at the Library. For the sake of simplicity let's only consider the spatial dimension. For this instance, a possible cloaking region that guarantees 3-anonymity is the minimum bounding circle that encompasses a nearby check-in, say $c_3 = (u_3, t_3)$. This implies that the original true co-location $cl_1 = (c_1, c_2)$ is indistinguishable from the now additionally observed $k-1$ other co-locations, i.e. $cl_2 = (c_2, c_3)$ and $cl_3 = (c_1, t_3)$. In other words, there is a plausible deniability for a co-location between any check-in pair within the cloaking region, thus safeguarding co-location

information of the users. In practice, the data publisher would select a value of k commensurate with the re-identification probability they are willing to tolerate (i.e. a threshold risk). Higher values of k imply a lower probability of re-identification, but also more distortion to the data, and hence greater quality loss due to k-anonymization.

There exist several challenges to guaranteeing k-anonymity across the co-location network $G = (C, CL)$. First, is the presence of connected components [1] in the network. Figure 4.3(a) illustrates some examples of connected components that may exist in the co-location network. Connected components are formed as a direct consequence of the co-location generation algorithm in Figure 1. Therefore, our algorithm must consider more than just individual co-locations in isolation. The second challenge as hinted above, is the extension of the simple spatial cloaking to the 3-dimensional case that additionally considers the temporal dimension. This problem can be overcome by using a distance function that combines both spatial and temporal distances. We utilize a linear combination function akin to the quality loss function in equation 3.1. Given two check-ins c_u^i and c_v^j, the spatio-temporal (ST) distance between them is defined as

$$ST_{dist}(c_u^i, c_v^j) = \frac{\alpha.\left\|c_u^i, c_v^j\right\|}{MAX_S} + \frac{(1-\alpha).|c_u^i.t, c_u^i.t'|}{MAX_T} \tag{4.1}$$

Factors MAX_S and MAX_T normalize the spatial and temporal distances in the range [0,1]. They can be understood to be the maximum temporal and spatial resolutions the data publisher is willing to tolerate when applying k-anonymity preserving mechanism.

Figure 4 presents the pseudocode for the co-location k-anonymity algorithm, which takes as input the co-location network G and the parameter k. Sets S and R are initialized to be empty, and used to store the connected components and the nearest neighbors, respectively. The algorithm first computes all connected components[2] S' in the co-location network $G = (V, E)$ and stores

[1] A connected component (or just component) of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph.

[2] It is straightforward to compute the connected components of a graph in linear time using either breadth-first search or depth-first search.

them in set S. For each connected component $S' \in S$, it computes the center of the Minimum Bounding Circle (MBC), and then applies spatial and temporal transformation to each check-in $c \in S'$ in order to move it to the center. Simply put, the transformation step converts each connected component into a clique at the center of the component. This step may create new co-locations amongst the members of the components that were previously further away than δ_s and δ_t distance (e.g. see third row of Fig 4.3(a)). Next, given the value of k, the size $|S'|$ of the connected component $S' = (V, E)$, and the number of edges $|E|$, the algorithm computes the minimum number h of points required to ensure k-annonymity within component S'. h is computed by solving the following quadratic equation:

$$\underbrace{(k-1)\cdot|E|}_{\substack{\text{(i) total edges}\\ \text{required at } S'}} = \underbrace{{}^{|V|}C_2 - |E|}_{\substack{\text{(ii) new edges}\\ \text{within set } V}} + \overbrace{{}^{h}C_2}^{\substack{\text{(iii) edges in}\\ \text{clique of } h \text{ points}}} + \underbrace{|V|\cdot h}_{\substack{\text{(iv) edges b/w}\\ \text{set } V \text{ and } h \text{ points}}} \tag{4.2}$$

Variable h is the final size of set H, which stores the h nearest neighbors to the center of the component. Given h, the algorithm incrementally obtains the next nearest neighbor to the center of MBC(S') and adds him to set H until $|H| = h$ check-ins of unique users not in the component S' are found. Finally, all points in S' are transformed to the center of MBC(S').

As an example consider 2-anonymization (i.e. $k = 2$) of the component consisting of three users in Figure 4.3a. There are a total of three edges between the three users, hence $|V| = 3$ and $|E| = 3$. Recall that a 2-anonymization ensures that an adversary with access to the published data cannot ascertain weather any co-located pair is true with confidence greater than $1/k$. Accordingly, in this instance of 2-anonymization, we must ensure that the three true edges are indistinguishable within an observed six. Therefore h is calculated to be 1 according to equation 4.2 as follows; new edges needed to create necessary anonymity is (i) = 3, the new edges formed within set V after transformation to the center of $MBC(S')$ (ii) = 0, the edges that would form within the h points of set H when brought to the center of $MBC(S')$ (iii) = 0, and the new co-locations formed as a result of moving the points in H to $MBC(S')$ (iv) = 3. Similarly, for the connected component

with four users and three edges (i.e. $|V| = 4$, $|E| = 3$), (i) = 3, (ii) = 3, (iii) = 0, and (iv) = 0; implying that no additional edges beyond those created by the transformation to the center of MBC(S') are needed.

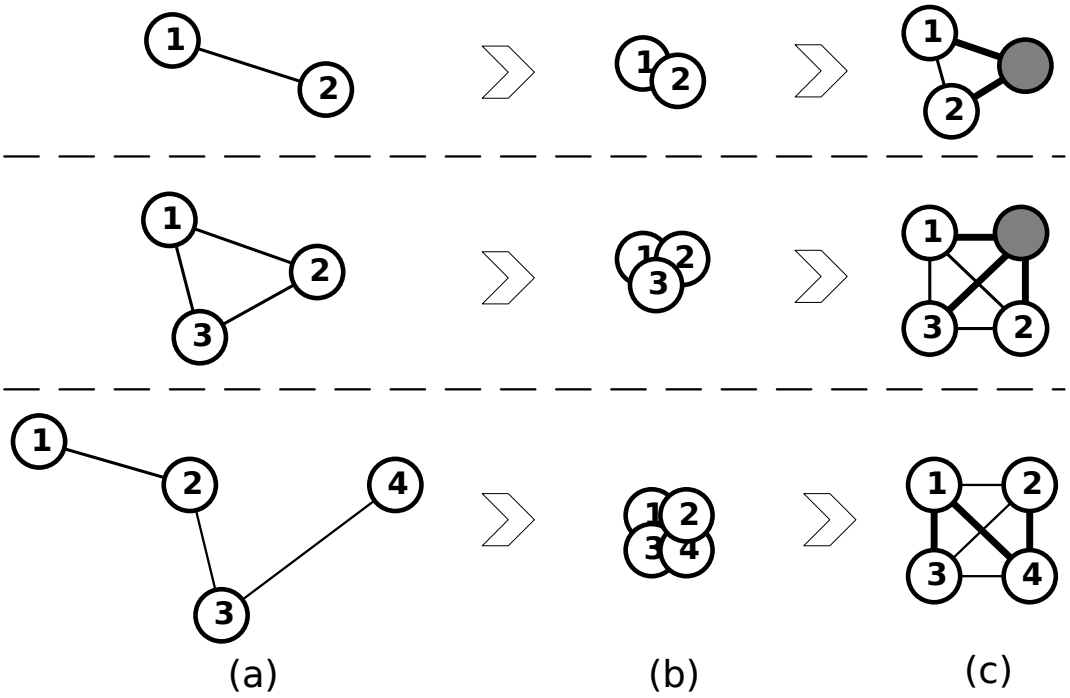


Figure 4.3: (a) Connected components in G (b) components transformed to center of MBC (c) k-anonymized components

Algorithm 4 Co-location k-anonymity

1: **procedure** k-ANONYMIZE($G = (C, CL)$,k)
2: $S \leftarrow$ getConnectedComponents(G)
3: **for** $S' = (V, E)$ in S **do**
4: $(p, t) \leftarrow$ center of MBC(S')
5: **for** check-in c in S' **do**
6: $c.p \leftarrow p, \quad c.t \leftarrow t$
7: $H \leftarrow \emptyset$
8: $h \leftarrow \left\lceil (0.5 - \cdot|V|) + 0.5 \cdot \sqrt{8 \cdot k \cdot |E| + 1} \right\rceil$
9: $c \leftarrow$ getNextNearestNeighbor(p,t)
10: **while** $c.u_id \notin V$ AND $h < t$ **do**
11: $H \leftarrow c$
12: $c \leftarrow$ getNextNearestNeighbor(p,t)
13: **for** check-in c in H **do**
14: $c.p \leftarrow p, c.t \leftarrow t$

Chapter 5

Experiments

5.1 Datasets

We utilize 3 datasets with varying level of spatial and temporal density, to study the degree of protection offered by our method against inference attacks on user co-location. This demonstrates the effectiveness of our methods. Table 5.1 describes some statistics about the datasets.

The first dataset, the Gowalla-Small dataset is a subset of user check-in data collected from Gowalla (a location based social networking website) by SNAP. For the purpose of experimentation and analysis we only utilize check-ins within the United States. The filtered Gowalla-Small data contains 3,669,249 check-ins. This constitutes check-ins from 54,551 unique users at 673,774 locations from Feb. 2009 - Oct. 2010.

The second dataset, the Gowalla-Big dataset was aggregated by using the Gowalla API to download all user check-in data before Jun. 2011. We use a filtered subset of this dataset which only contains check-ins within the United States. It contains 18,290,199 user check-ins, constituting check-ins from 127,477 users across 1,214,943 locations.

The third dataset, the CRESDA-Shanghai dataset collected by China Center for Resources Satellite

Dataset	Temporal density	Spatial density	Densest city	Check-in count
Gowalla-Small	38.28	1,227	Austin	333,525
Gowalla-Big	95.13	5,810	Austin	1,579,163
CRESDA-Shanghai	868.62	9,090	Shanghai	7,858,442

Table 5.1: Dataset statistics; Temporal Density: Check-ins per hour in the densest city, Spatial density: Check-ins per sq. mile in the densest city, Check-in count: Number of Check-ins in the densest city

Data and Applications(CRESDA) contains social media check-in records extracted from users across places in Shanghai collected during the yearlong period from September 2011 to September 2012.

The check-ins in the Gowalla datasets span across all of the united states, however the ones CRESDA-Shanghai are limited to the city of Shanghai, hence we compare them at a city scale. The average number of check-ins per square mile in the densest city is a rough measure of spatial density and the average number of check-ins across the densest city at any given hour as a rough measure of temporal density. The datasets arranged in the increasing order of spatio-temporal density are : Gowalla-Small < Gowalla-Big < CRESDA-Shanghai.

5.1.1 Co-Locations

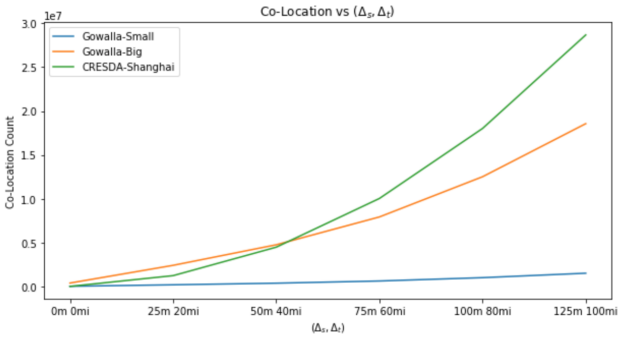
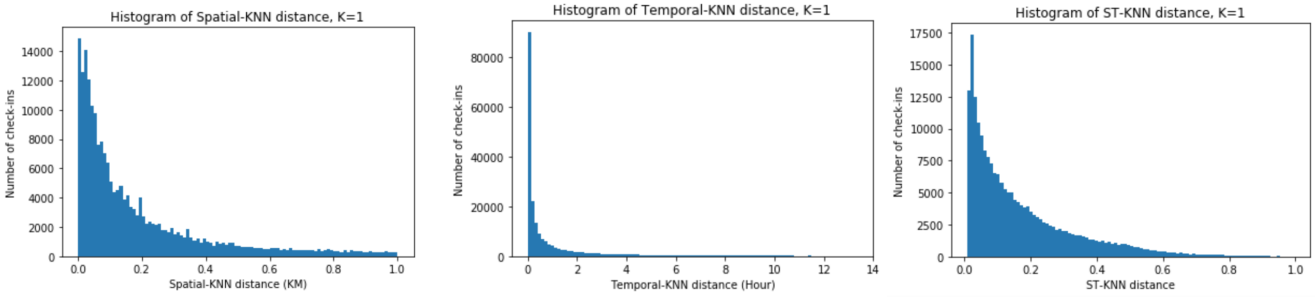


Figure 5.1: (Δ_s, Δ_t) vs Co-Location count

Δ_s and Δ_t (Defined in Table 3.1) constitute the maximum spatial and temporal distance between a check-in pair to be considered a co-location. Figure 5.1 is a plot between (Δ_s, Δ_t) on the X-Axis and the number of co-locations in the data set normalized by the number of total check-ins in the dataset, on the Y-Axis. We note that, 1) The number of co-locations grows exponentially with linear increase in (Δ_s, Δ_t), across all 3 datasets. 2) Datasets with a higher spatio-temporal density show a faster rate of co-location growth with increase in (Δ_s, Δ_t).

5.1.2 Nearest Neighbors

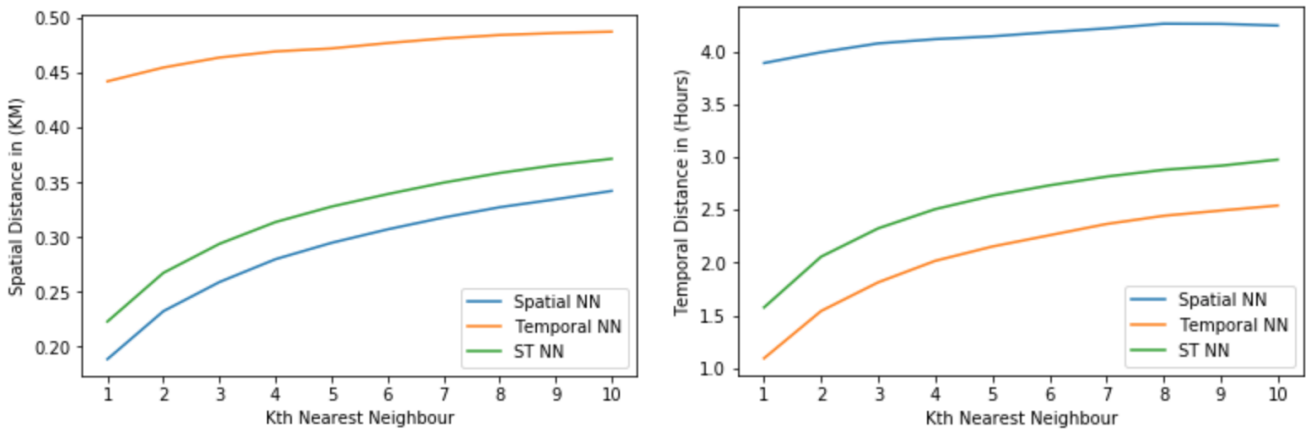


(a) Spatial NN Distance (b) Temporal NN Distance (c) ST_{dist} to ST NN

Figure 5.2: Histograms of KNN Distances (Gowalla-Small)

Figure 5.13 illustrates the histogram distribution of the first nearest neighbor(NN) distances from the Gowalla-Small dataset. The neighbor distances reveal the data density of a region. Check-ins in sparse regions will have a distant neighbors, check-ins in dense regions will have a neighbors in close proximity. The distributions of spatial first NN distances (Figure 5.2a), temporal first NN distances (Figure 5.2b) and ST_{dist} first NN distances (Figure 5.2b) follow a power law distribution; indicating most check-ins like in close proximity to each other (spatially and temporally), and a few lie farther away from each other.

Figure 5.3a shows the mean spatial distance(in KM) of the Kth NN in the Spatial Dimension, Temporal Dimension and ST Dimension. Figure 5.3b shows the mean temporal distance (in Hours) of the Kth Nearest Neighbor in the Spatial Dimension, Temporal Dimension and ST Dimension.



(a) Mean Spatial Distance

(b) Mean Temporal Distance

Figure 5.3: Mean Distances vs K (Gowalla-Small)

The mean spatial distance to the first spatial nearest neighbor is 180m and to the first temporal nearest neighbor is 440m. We see that on average the temporally nearest neighbors are much further away spatially. The mean temporal distance to the first temporal nearest neighbor is 65 minutes and to the first spatial nearest neighbor is 225 minutes. Similarly we see that spatially nearest neighbors on average are much further away temporally. Thus, closeness in the spatial dimension does not guarantee closeness in the temporal dimension and vice versa. We see that our notion of ST_{dist} between check-ins ensures that nearest neighbors are close both spatially and temporally. The nearest neighbor based on ST_{dist} is on average 220m and 90 minutes away.

5.2 Parameters and Implementation

We store each check-in (represented by it's ID and coordinates) in a regular 15000 × 15000 grid, which produces a resolution of around 100 m^2 at the cell. We perform Range queries in a straightforward manner using the grid, whereas we adopt the CPM [26] algorithm for answering kNN queries. We also use a hash table on the user ID, to facilitate fast execution of Get-Co-Location method. Since we only compute distances between check-in within the same city, we use euclidian

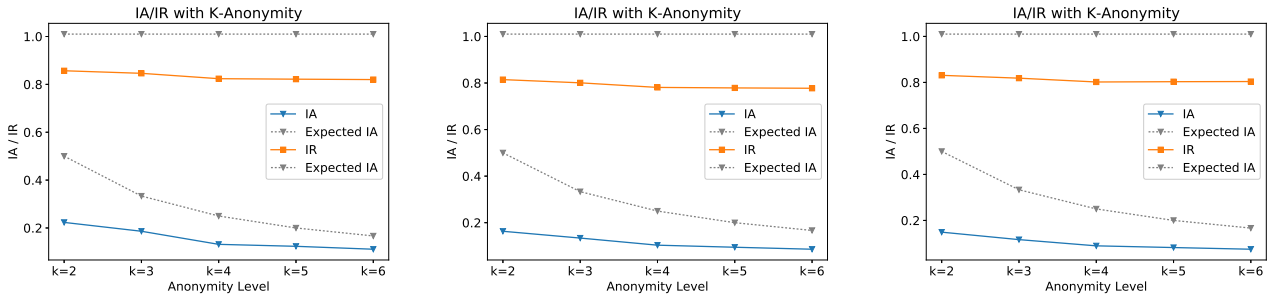
distance to proximate the distance between the latitude,longitudes of check-ins. All methods were implemented in C++, under Linux Ubuntu on a machine with Intel Core 2 Duo processor, with a 2.33GHz CPU and 16GB RAM. We set Δ_s to 25 meters and Δ_t to 20 minutes according to the generally accepted value in literature. For the Gowalla-Small, Gowalla-Big datasets we find that 99 percentile of co-located check-ins have a ST nearest neighbor within 5km and 48 hours. We thus set MAX_s to 5 km and MAX_t 48 hours and discard 1 percentile of co-locations in extremely sparse regions. For the CRESDA-Shanghai dataset we set MAX_s to 500m and MAX_t 6 hours.

5.3 Experimental results

For each privacy mechanism we evaluate its effect on Co-Location Inference Accuracy (IA) and Inference Recall (IR) across datasets. To understand the behavior of the functions across the skewed distribution of the geo-social network, we plot the privacy of co-locations based on the density distribution of points around it. We order the co-locations based on their ST_{dist} to the nearest neighbor and bucket them into 100 equal sized bins. For each bin we present the average IA, which enables us to gauge the variance in privacy offered by our methods across varying data densities in the graph. Across datasets, for the same levels of avg. quality loss, we compare the level of privacy offered by the various methods.

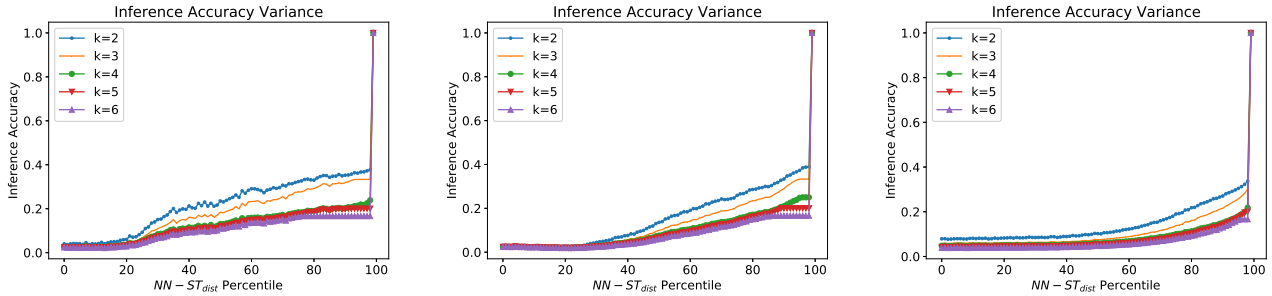
5.3.1 Co-location k-Anonymity

Figure 5.4 demonstrates the IA and IR of varying levels of k-anonymization. k-anonymity guarantees a maximum inference accuracy of $1/k$, which is the expected value of IA (plotted as the dotted line). However we find that the observed inference accuracy is less than $1/k$. This can be explained using first row Fig 4.3(a). Consider a co-location $cl = (c_u, c_v)$ between the two users. A 2-anonymity only needs one other co-location to make cl 2-anonymous. Suppose the k-anonymity



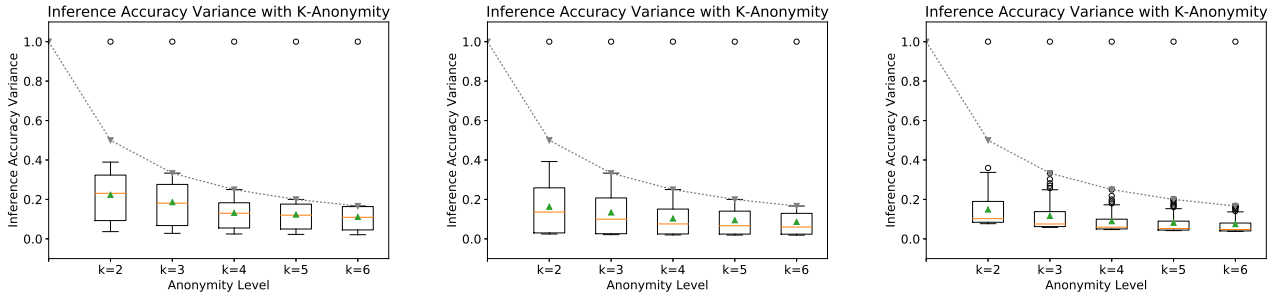
(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.4: *Co-Location k-Anonymity* - k vs IA/IR



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.5: *Co-Location k-Anonymity* - Average IA per co-location bin



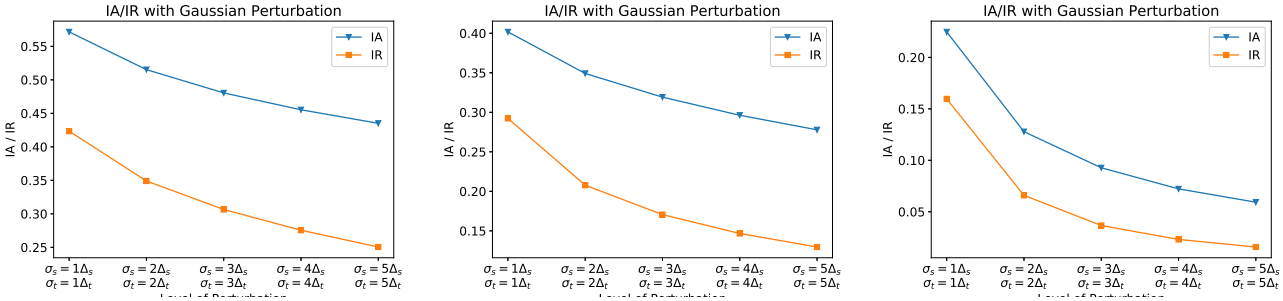
(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.6: *Co-Location k-Anonymity* - Variance of IA

procedure brings one of cl's nearest neighbor c_w to its location, and in this step creates new co-locations at cl. This procedure satisfies the two anonymity requirement but inadvertently also 3-anonymizes the original co-location due to the pairwise nature of co-location formation. This

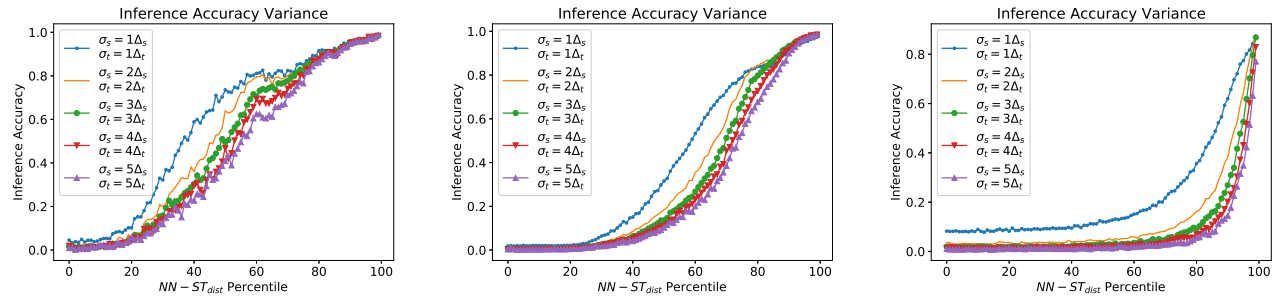
results is noticeably less Inference Accuracy than expected. Finally, the sharp bump in the last bin of Figure 5.5 is a result of parameters MAX_s and MAX_t that control the maximum temporal and spatial deterioration acceptable to the data publisher. MAX_s and MAX_t are set to discard 1 percentile of co-locations in very sparse regions. As a result, these co-locations are not anonymized and hence the adversary can observe them with complete accuracy.

5.3.2 Naive Gaussian Perturbation



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

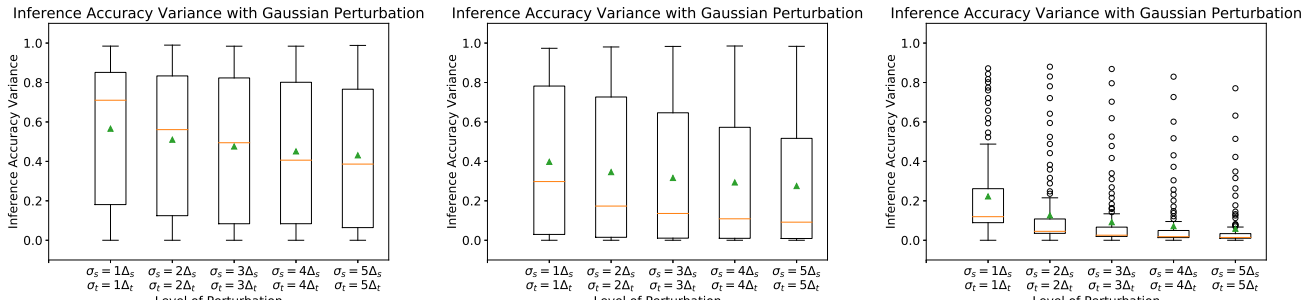
Figure 5.7: *Gaussian Perturbation* - Δ_s, Δ_t vs IA/IR



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.8: *Gaussian Perturbation* - Average IA per co-location bin

Figure 5.7 presents the IA and IR for varying magnitude of noise as controlled by the parameters σ_s and σ_t, which are input to the Gaussian noise function. We increment parameters σ_s and σ_t as a multiple of the co-location constraint. Parameter $\sigma_s = n_o\Delta_s$ and $\sigma_t = n_o\Delta_t$, where n_o ranges



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.9: *Gaussian Perturbation* - Variance of IA

from 1 to 5. It is straightforward to see that with increasing magnitude of noise, successfully reconstructing perturbed co-location becomes increasingly difficult.

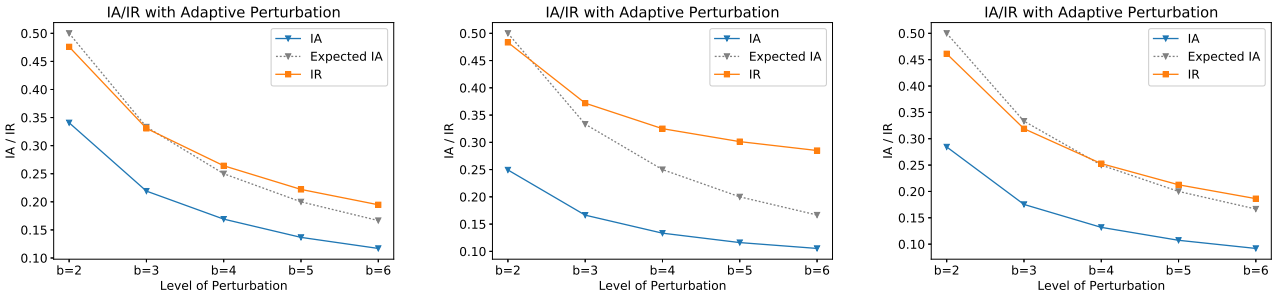
Consider the Gowalla data sets, although on adding enough noise Gaussian perturbation scheme achieves a reasonable average privacy, however there exists a large variance in the privacy across the density distribution of the data. Figure 5.8a,5.8b demonstrates this behavior, wherein the inference accuracy of co-locations in bins 0-20 is at-most 15% whereas the inference accuracy in bins 80-100 is at least 80%. Figure 5.9a,5.9b illustrates a high variance in IA for varying magnitude of noise. Regardless of the amount of distortion, the minimum/maximum amount IA for a co-location bucket are 0/1 indicated by the inner fence of the box plot. We see this trend mildly visible in the CRESDA-Shanghai data set too (Shown in Figure 5.8c), co-locations in sparse regions have low IA and ones in very sparse regions (90-100 percentile) have increasingly higher IA (greater than 0.5). However the variance in IA is much less significant.

Put simply, the results verify our intuition that any fixed magnitude of distortion will leave parts of data unprotected, since dense areas get unnecessarily excessive amounts of noise while sparse areas receive an insufficient amount of noise. The variance is affected by the proportion of the dataset which remains un-protected. This depends on level of perturbation with respect the spatio-temporal density of the dataset.

The mean spatial and temporal distance of the nearest ST neighbor in the Gowalla-Small dataset

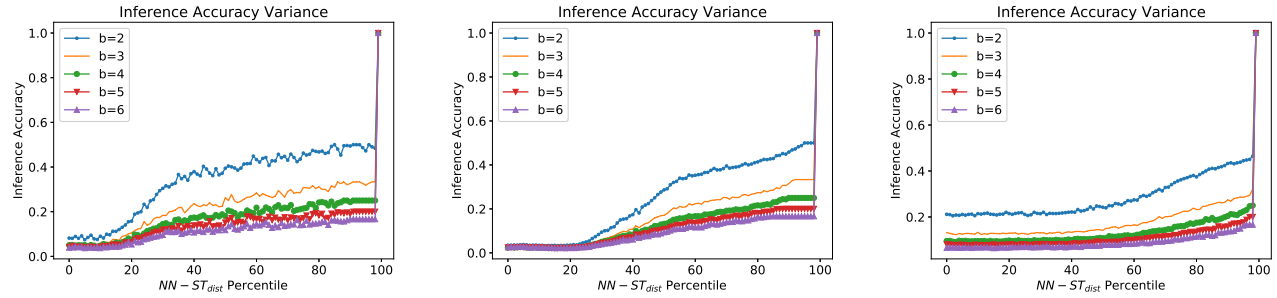
are around 450m and 3hours. Thus perturbing co-locations with a mean gaussian perturbation radius of 20m and 16mi ($\sigma_s = 1.\Delta_s$, $\sigma_t = 1.\Delta_t$) leaves a significant portion of the dataset unprotected (around 50 percentile of co-locations have IA more than 70%). The mean spatial and temporal ST nearest neighbor distances in the Shanghai dataset are around 40m and 30mi, for which the perturbation radius is sufficient (only 10 percentile of co-locations have IA more than 70%).

5.3.3 Adaptive Perturbation



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

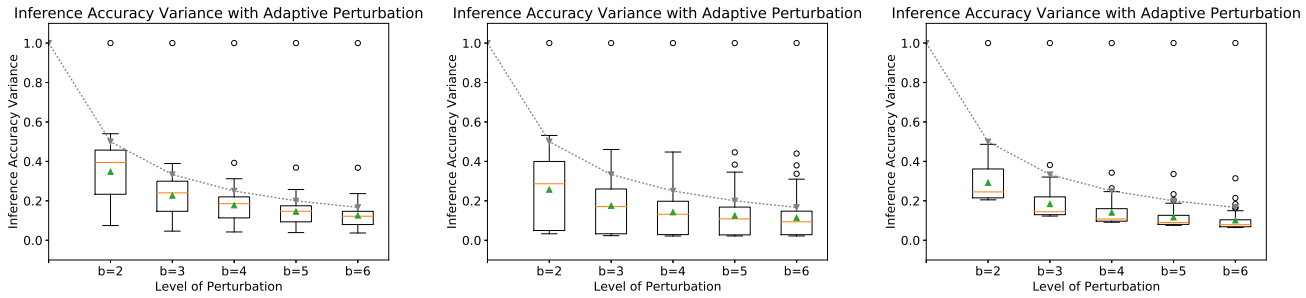
Figure 5.10: *Adaptive Perturbation* - b vs IA/IR



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.11: *Adaptive Perturbation* - Average IA per co-location bin

Figure 5.10 presents the value of IR and IA for varying degree of distortion $b = 2, 3, 4, 5, 6$. The expected value of IA for a given value of b is $1/b$ (plotted as a dashed line) because an observed co-location is potentially correct with probability $1/b$. It is not surprising that as the degree of

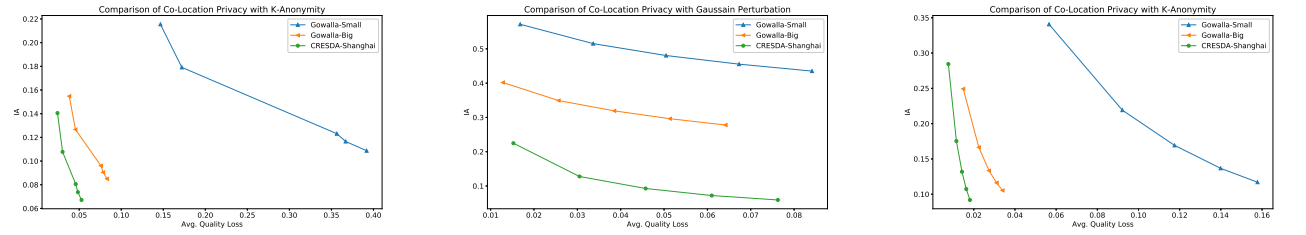


(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.12: *Adaptive Perturbation* - Variance of IA

distortion is increased the observed co-locations are more likely to be incorrect. However, the difference between the expected accuracy and the observed accuracy is significant. Figure 5.11 helps us explain the results. In dense regions, given a pair of co-locations in close proximity to each other, there is a high likelihood of them having several nearest neighbors in common. This can potentially generate many false co-locations at the place of the common nearest neighbors. Figure 5.12 depicts the variance in the inference accuracy for varying level of distortion b. While adaptive perturbation on average provides a better privacy level than the expected $1/b$, the topmost fence of the box-plot indicates that this is not a guaranteed bound, unlike the case of co-location k-anonymity.

5.3.4 Comparison of Privacy between datasets

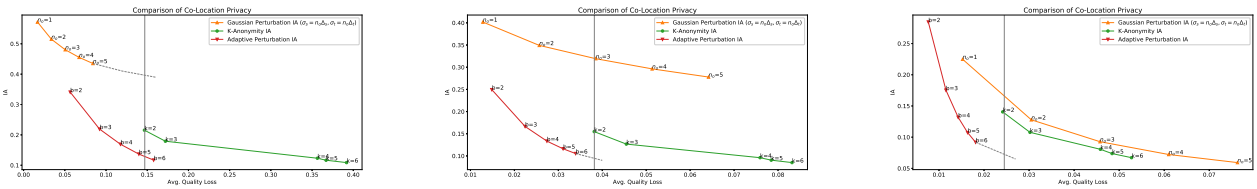


(a) K-Anonymity (b) Gaussian Perturbation (c) Adaptive Perturbation

Figure 5.13: Comparison of IA from Co-PPM Methods across datasets

We find that the data density a given dataset significantly affects the performance of Co-PPM mechanisms for identical levels of quality loss, higher the data density greater the privacy. This trend is clearly visible in the case of Gaussian perturbation as the average quality loss scales are comparable across datasets (illustrated in 5.13b). Through extrapolation we see that this trend is consistent across other methods as-well. We also note that sparse datasets require significant perturbation to achieve similar levels of privacy as dense ones. Consider Figure 5.13a, Gowalla-Small needs to be perturbed around 8-times as much as CRESDA-Shanghai to limit the adversary's IA to around 10%.

5.3.5 Comparison of Privacy between methods



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.14: Comparison of IA between Co-PPM Methods

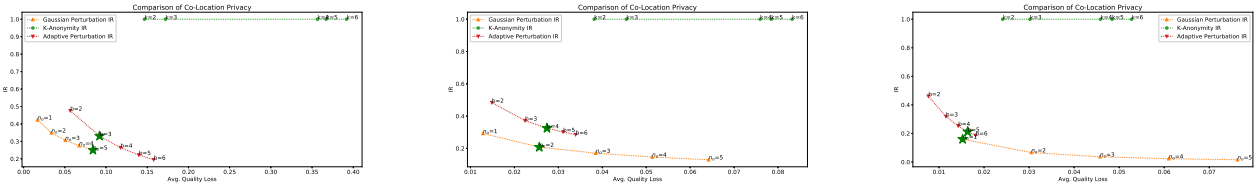
Figure 5.14 demonstrates the Inference Accuracy for the various Co-PPM methods with a varying magnitude of Quality Loss. We extrapolate our measurements (indicated by the dotted line) to compare across our methods at a fixed value of quality loss (indicated by the solid vertical line). Across all 3 datasets, Adaptive perturbation method clearly outperforms its Gaussian counterpart and k-anonymity in terms of IA (lower is better).

Adaptive vs k-anonymity: For comparable noise levels, we see a lower adversary IA (around 10% lesser, across all 3 datasets). Since k-anonymity method does not break apart any co-locations, the recall is considerably larger than that of the Adaptive mechanism, thus implying that a larger population is at risk. k-anonymity results in the perturbation of h (Algorithm 4, line 8) check-ins

Dataset	Method	IA	IR	F1	Inter-Quartile Variance
Gowalla-Small	Gaussian $n_0 = 5$	0.45	0.25	0.32	0.70
Gowalla-Small	Adaptive $b = 3$	0.22	0.35	0.27	0.10
Gowalla-Big	Gaussian $n_0 = 2$	0.35	0.20	0.25	0.70
Gowalla-Big	Adaptive $b = 4$	0.13	0.35	0.18	0.20
CRESDA-Shanghai	Gaussian $n_0 = 1$	0.22	0.21	0.215	0.20
CRESDA-Shanghai	Adaptive $b = 5$	0.10	0.16	0.12	0.04

Table 5.2: Comparison of Inter-Quartile Variance between Gaussian and Adaptive Perturbation schemes for comparable levels of quality loss

to hide a co-location, however the adaptive approach results in the perturbation of a single check-in to one of b locations at random. Thus on average adds results in far lesser Avg. Quality Loss.

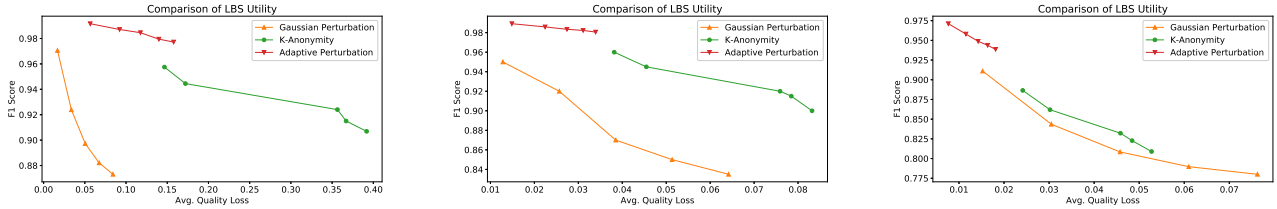


(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.15: Comparison of IR between Co-PPM Methods

Adaptive vs Naive Gaussian: Naive Gaussian perturbation results in a lower IR than the Adaptive scheme (lower proportion of the population is exposed), though the IA is much higher. Thus it's not possible to weigh one approach over the other through accuracy metrics. However when the variance in its privacy is considered, it becomes apparent that the method fails to preserve privacy of a significant portion of the population. For comparable levels quality loss (illustrated by star symbols in Figure 5.15), Table 5.2 lists the adversary's accuracy metrics. Considering the F1-Score(i.e. the harmonic mean of IA and IR) of the adversary's inference and the interquartile-variance in the IA Adaptive perturbation clearly out-performs naive gaussian as well.

5.3.6 Comparison of LBS Utility between methods



(a) Gowalla-Small (b) Gowalla-Big (c) CRESDA-Shanghai

Figure 5.16: Comparison of Utility between Co-PPM Methods

To demonstrate the utility of released locations, we present the F1 score (i.e. the harmonic mean of precision and recall) of a spatio-temporal range query at a set of sample query points, which are randomly selected subset of the location updates. Figure 5.16 presents the F1 score of the range utility and the associated quality loss. We compare our methods pairwise for similar quality loss. Gaussian perturbation mechanism performs poorly for range queries, since it usually applies excessive amounts of noise in dense areas. On the other hand, Adaptive perturbation mechanism adds just enough noise to create the necessary confusion, as a result the distorted check-ins do not fall too far away from the originating location. Though k-anonymity also perturbs check-ins within the spatio-temporal vicinity, multiple check-ins are disturbed in order to hide a single co-location. This results in higher quality loss and lower utility when compared to the Adaptive scheme.

Chapter 6

Conclusion

In this work, we introduced the problem of co-location privacy and conducted a systematic investigation of the problem. We presented a general framework to protect the privacy of co-locations. We adapted the popular gaussian noise based location privacy preserving method to protect co-locations. We proposed two new methods (adaptive perturbation and k-anonymity) to address the weakness of the baseline gaussian noise method and improve on it's privacy and utility.

6.1 Discussion

The problem of co-location privacy is new, and this the first work to formulate the problem and propose heuristic and syntactic-privacy guaranteeing methods. While the privacy preserving mechanisms are inspired by current literature on location privacy methods, their application to co-locations is not straightforward due to the pairwise dynamic inherent in co-locations. We performed extensive preliminary studies in which we attempted to apply existing location privacy preserving mechanisms like (i) Location hiding, (ii) Fake Location injection, (iii) Location cloaking (adding confusion) and (iv) Location perturbation (adding noise), to protect co-locations.

Co-Location hiding (publishing a subset of co-locations) is a straight-forward mechanism for preserving co-location privacy but has apparent shortcomings; while the methods guarantees that a hidden co-location remains completely private, it provides no protection to co-locations which are not covered. In the context of privacy concerns, even a moderately large absolute number of affected individuals can represent a significant privacy risk. Injection of fake co-locations seems simple at the outset but require the injected location traces to simulate actual user behavior, which has proven to be hard and out of the scope of this work.

k-Anonymity, a form of cloaking on the other hand is built-in with privacy guarantees but at the cost of high total data quality loss. This deterioration is an inverse function of the data density of a region thus location traces in dense regions with significant utility aren't deteriorated as-much. Thus the utility of LBS services remain high. Anonymity does not fine-grained control over data quality deterioration like the perturbation techniques. We see that the amount of total quality loss is a step function of the data distribution. Anonymity is, not viable in the provision of an on-line service when the identification of users is required [25]. In this case, a solution to protect the privacy of users consists in decreasing the accuracy of co-location information [29].

Naive gaussian perturbation or noise addition offers reasonable average privacy. Gaussian noise has been used extensively as a means for privacy in statistical and location databases. We thus adopt it as a baseline. Perturbation methods can also be extended to an live services with an incoming data-stream. However it suffers from a high variance in privacy levels, as it fails to protect co-locations from an attacker with knowledge of data distribution. The proposed adaptive perturbation improvement fixes the variance issues while minimizing the total quality loss; but perturbation schemes in general fail to offer any privacy guarantees.

6.2 Limitations

In this work we have explored the problem of co-location privacy with several assumptions and limitations on adversary knowledge. We recognize several correlations that under-gird a user's location and his co-location information. Some preliminary studies to help is enumerate these dimensional correlations.

Singular Dimensional correlations:

1. Temporal correlation of location: Check-ins at a particular location will have a known temporal distribution. Example, Check-ins at a diner which operates from 9AM-9PM, will have peak hours at noon and in the evening.

2. Temporal correlation of user: There exists temporal seasonality in a user's movements. Example, Morning person vs Night Person.

3. Temporal correlation of a user's check-in: There exists order in a user's movement through the day. For example, a person traveling along a trajectory is likely to remain along that path.

Bi-Dimensional correlations:

1. User-User Correlation: User co-locates with other users he's historically co-located with. Example user is more likely to co-locate with people in his social group rather than a random person.

2. User-user temporal-correlation: User meets a work-friend in morning, and his significant other in the evening.

3. User-user spatial-correlation: User meets his work Friend usually at work-place and not at any other location.

We realized that when attributes are correlated, it is not sufficient to independently add noise to the different attributes using (for example) the Laplacian mechanism of differential privacy [2] mechanism design rather than a simple composition of mechanisms for scalar-valued outcomes. We do not claim to simulate the worst-case scenario wherein an attacker has a complete background knowledge, and hence can give an accurate estimate of expected privacy level. There are several potential inference attacks on location privacy. Some attacks may be ported to co-location privacy as is while others may require heavy modifications. Addressing these concerns is however, outside the scope of this work. We hope this work motivates research towards the design of more intricate Inference attacks as well as theoretically sound privacy mechanisms.

6.3 Future Work

The proposed methods consider all co-locations equally informative. However in the context of inference of sensitive social-connections between users, this is not the case. (eg) Happenstance co-locations (i.e) user pairs who have co-located just once are not significant [28]. We wish to improve our current methods by controlling the quality loss of a co-location based on it's relative informativeness. We also hope to extend the Co-PPM framework to a live model, which handles incoming location trace streams.

Bibliography

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 439–450, New York, NY, USA, 2000. ACM.

[2] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.

[3] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati. An obfuscation-based approach for protecting location privacy. *IEEE Transactions on Dependable and Secure Computing*, 8(1):13–27, 2011.

[4] L. L. Beck. A security machanism for statistical database. *ACM Transactions on Database Systems (TODS)*, 5(3):316–3338, 1980.

[5] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[6] J. A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[7] V. Bindschaedler and R. Shokri. Synthesizing plausible privacy-preserving location traces. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 546–563. IEEE, 2016.

[8] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of modern physics*, 81(2):591, 2009.

[9] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441, 2010.

[10] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.

[11] G. Danezis, S. Lewis, and R. J. Anderson. How much is location privacy worth? In *WEIS*, volume 5. Citeseer, 2005.

[12] C. A. Davis Jr, G. L. Pappa, D. R. R. de Oliveira, and F. de L Arcanjo. Inferring the location of twitter messages based on user relationships. *Transactions in GIS*, 15(6):735–751, 2011.

[13] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences*, 106(36):15274–15278, 2009.

[14] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. Technical report, Georgia Institute of Technology, 2004.

[15] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios. Providing k-anonymity in location based services. *ACM SIGKDD explorations newsletter*, 12(1):3–10, 2010.

[16] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[17] G. Greenwald and E. MacAskill. Nsa prism program taps in to user data of apple, google and others. *The Guardian*, 7(6):1–43, 2013.

[18] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[19] G. Iachello, I. Smith, S. Consolvo, G. D. Abowd, J. Hughes, J. Howard, F. Potter, J. Scott, T. Sohn, J. Hightower, et al. Control, deception, and communication: Evaluating the deployment of a location-enhanced messaging service. In *International Conference on Ubiquitous Computing*, pages 213–231. Springer, 2005.

[20] D. Jurgens. That's what friends are for: Inferring location in online social media platforms based on social relationships. *ICWSM*, 13:273–282, 2013.

[21] E. Kaasinen. User needs for location-aware mobile services. *Personal and ubiquitous computing*, 7(1):70–79, 2003.

[22] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12):1719–1733, 2007.

[23] J. Krumm. Inference attacks on location tracks. *Pervasive computing*, pages 127–143, 2007.

[24] J. Krumm. Realistic driving trips for location privacy. *Pervasive Computing*, pages 25–41, 2009.

[25] M. Langheinrich. Privacy by design—principles of privacy-aware ubiquitous systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 273–291. Springer, 2001.

[26] K. Mouratidis, D. Papadias, and M. Hadjieleftheriou. Conceptual partitioning: An efficient method for continuous nearest neighbor monitoring. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 634–645. ACM, 2005.

[27] A.-M. Olteanu, K. Huguenin, R. Shokri, and J.-P. Hubaux. Quantifying the effect of co-location information on location privacy. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 184–203. Springer, 2014.

[28] H. Pham, C. Shahabi, and Y. Liu. Ebm: an entropy-based model to infer social strength from spatiotemporal data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 265–276. ACM, 2013.

[29] P. Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[30] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec. Quantifying location privacy: the case of sporadic location exposure. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 57–76. Springer, 2011.

[31] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Security and privacy (sp), 2011 ieee symposium on*, pages 247–262. IEEE, 2011.

[32] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 617–627. ACM, 2012.

[33] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[34] R. M. Thompson. *United States v. Jones: GPS Monitoring, Property, and Privacy*. Congressional Research Service, 2012.

[35] J. F. Traub, Y. Yemini, and H. Woźniakowski. The statistical security of a statistical database. *ACM Transactions on Database Systems (TODS)*, 9(4):672–679, 1984.

[36] C. R. Vicente, D. Freni, C. Bettini, and C. S. Jensen. Location-related privacy in geo-social networks. *IEEE Internet Computing*, 15(3):20–27, 2011.

[37] D. Xu, P. Cui, W. Zhu, and S. Yang. Find you from your friends: Graph-based residence location prediction for users in social media. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.

[38] T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 348–357. ACM, 2009.

[39] T.-H. You, W.-C. Peng, and W.-C. Lee. Protecting moving trajectories with dummies. In *Mobile Data Management, 2007 International Conference on*, pages 278–282. IEEE, 2007.