

Tell Me What You Want and I Will Tell Others Where You Have Been

Anthony Quattrone, Elham Naghizade, Lars Kulik, Egemen Tanin
The University of Melbourne
{anthony.quattrone,enaghi,lkulik,etanin}@unimelb.edu.au

ABSTRACT

Trajectory data does not only show the location of users over a period of time, but also reveals a high level of detail regarding their lifestyle, preferences and habits. Hence, it is highly susceptible to privacy concerns. Trajectory privacy has become a key research topic when sharing/exchanging trajectory datasets. Most existing studies focus on protecting trajectory data through obfuscating, anonymising or perturbing the data with the aim to maximize user privacy. Although such approaches appear plausible, our work suggests that precise trajectory information can be inferred even from other sources of data. We consider the case in which a location service provider only shares POI query results of users with third parties instead of exchanging users' raw trajectory data to preserve privacy. We develop an inference algorithm and show that it can effectively approximate original trajectories using solely the POI query results.

Categories and Subject Descriptors

I.5 [Pattern Recognition]: Models; H.2.8 [Database Applications]: Data Mining

Keywords

Location Privacy, Inference Attacks

1. INTRODUCTION

As smartphones become ubiquitous tools of our everyday life, the growth of mobile applications has led to the generation and storage of massive amounts of location data. This data may be shared in social networks or exchanged among companies. Many applications, e.g., traffic management, urban management and geomarketing, gain substantial benefit through mining such sources of trajectory data. However, sharing this data in a raw format with third parties may incur serious privacy threats. Particularly, if inappropriately protected, such data may turn into powerful means of privacy invasions [3, 13, 19] such as location-based spams, physical threats, or inference attacks [7]. Findings of a recent study indicates that a large number of Location Based Service (LBS) users

are concerned about their privacy, which places a great impediment to sharing trajectory data and the growth of LBSs in general.

Various approaches in the literature focus on preserving privacy prior to publishing/sharing trajectory data [4, 6, 9, 17, 18]. The authors in [1, 15] propose publishing a *k-anonymous* trajectory dataset to make a user indistinguishable from $k - 1$ other users [16]. Other studies adopt cloaking/obfuscation techniques to coarsen the spatial and/or temporal features of a trajectory [8] before publishing it. The authors in [11] propose an approach that adds dummy trajectories to keep the data private.

To motivate our work we consider the case of an LBS provider who sanitises a dataset by omitting all sensitive attributes from the dataset [5]. In other words, instead of anonymizing or obfuscating trajectory data, the LBS provider removes all GPS tracks of its users along with their identity, presuming this will provide maximum privacy. This means that only the *results* of user issued queries, which is generated and *owned* by the Location Service Provider (LSP), remain in the database when sharing it.

To identify the risks of such a plausibly bullet-proof practice, we consider a scenario where users continuously request the closest POIs to their position. For instance, a user issues a query like "Where is the nearest gas station to my path?" or "Send me the closest Italian restaurant?". If detailed GPS tracks are removed from the database, the LBS provider may lead others to believe that it is safe to exchange the POI query results – a set of POIs ordered based on querying time.

In order to demonstrate the vulnerability of sharing LSP query results with a third party, we develop an algorithm to perform an indirect inference attack on query results. The algorithm infers individuals' trajectories using the response of LSP to the requested POIs. The trajectory reconstruction is performed without using GPS tracks captured by the LSP but instead, the requested POIs sequence. As background knowledge we assume the availability of road types, and either their *edge centrality* or *edge frequency*, and *maximum velocity bound*, and basic transportation information about the area.

To the best of our knowledge, our work is the first to attack trajectory privacy without directly using trajectory information (either fine-grained or coarse-grained). The accuracy of our approach suggests that indirectly inferred paths are sufficiently precise to raise serious privacy concerns. By demonstrating what indirect trajectory inference attacks can achieve, we show that there is an urgent need to address the privacy implications of LBSs when exchanging trajectory datasets even when the provider omits the sensitive location data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'14, November 3–7, 2014, Shanghai, China.

Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2661829.2661866>.

2. RELATED WORK

Generally, analysing data in order to gain knowledge about a subject in an adversarial manner is known as an “inference attack” [12]. A key work to highlight the potential of location data in predicting users’ movement behavior was given in [2]. Ashbrook et al. have used GPS data of mobile users to determine significant places being visited by them [2]. To underline the risks of leaked location data [12] used real GPS data of 172 subjects and found each person’s home location with a median error of about 60 meters. Finally, [12] identified people based on their pseudonymous location tracks using simple algorithms and a free Web service. In contrast, our algorithm can estimate a user’s trajectory without using the GPS tracks stored by the LSP.

In another study, [10] succeeded at reconstructing an unknown trajectory using its distance to a few fixed trajectories. In order to achieve this, the authors have introduced speed limit and known trajectories as a background knowledge used by an adversary. Similarly, assuming it is possible to observe a user’s movement behaviour in public places, or even inferring a couple of her visited spots using social networks, weblogs and etc. [14] used some snapshots of a user’s trajectory as adversary’s background knowledge. [14] proves that this “general world knowledge” can breach user’s privacy with a high probability regardless of how much attempt is being taken to anonymize or cloak her location data. These two works clearly show the potential risks stemming from combining the available background knowledge along with the mutual distances released for analytical purposes.

3. PROBLEM DEFINITION

3.1 Closest POIs Database

POI queries are a common applications of LBSs. An LBS user sends her current location accompanied with a query asking for her closest points of interest, e.g., the closest gas station, Italian restaurant, etc., and the LSP returns a set of points as query result. The query database records are in the form of $(ID, \mathcal{P}_{\mathcal{G}})$, where ID determines a user. $\mathcal{P}_{\mathcal{G}}$ is the result of successive POI queries along the user’s trip. In other words, $\mathcal{P}_{\mathcal{G}} = \{p_1, p_2, \dots, p_n\}$ where each p_i represents a the closest point of interest for the i_{th} query. Note that ID is not necessarily a user’s actual identity, but rather a unique identifier such as a pseudonym.

3.2 Adversary Model

An LSP may remove both user identity and location information as the sensitive attributes from the query database, presuming this would guarantee the user’s privacy. This supposedly anonymised database is then shared with third parties. We suppose that an adversary is any third party with whom this query database is shared. Except for the results database, we assume that the adversary has one of the following two types of background knowledge about a transportation network:

Edge Centrality: The adversary may consider each street as an edge in a graph and assume that the more central an edge is, i.e., how frequent an edge occurs in a set of candidate paths, the more likely it is that a user travels along it. Hence, edge centrality can be modelled introducing a weighting function. Generally, main roads are more likely to receive higher weights.

Edge Frequency: The adversary may have access to the trips users have taken in the past. In this model, we assume that the adversary will assign a higher weight to more frequently travelled edges. The count for a specific edge is incremented for every trip that it could be part of to account for the GPS error. In addition, we assume that an attacker also has the following information:

Maximum Velocity Bound: The adversary may also assume that there is a maximum velocity with which a user can travel between two subsequent time stamps. The velocity of a user at a given time can be estimated based on the maximum speed limit of a road.

3.3 Attack Success

GPS points do not uniquely identify the roads a user has taken, so the actual trip is generally not known. Thus, conventional trajectory similarity measures such as the Hausdorff distance and DTW are not suitable to assess the success of our algorithm. Take Figure 1a as an example, where the GPS logs do not overlap with the actual path due to the measurement error. Using linear interpolation between GPS points (the dashed line connecting points) does not provide a robust means of inferring the original path, as can be seen, this interpolation may barely overlap with the underlying road network. Even map matching cannot resolve such a situation because a GPS point cannot be uniquely mapped to a single edge.

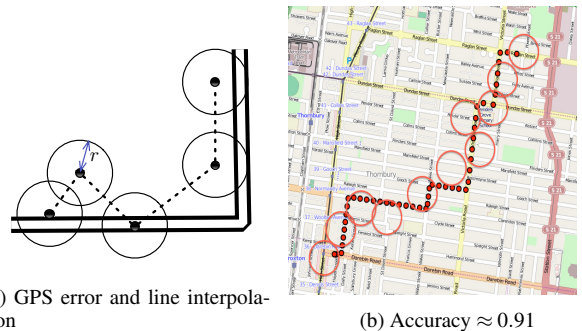


Figure 1: Proximity circles.

To address this issue, we determine the closeness of an inferred path to the original one through proximity circles. We draw circles with r radius around the location of each GPS point (Figure 1a) and estimate the attack success based on the circles. More specifically, the overlapping percentage of the inferred path within r meters to the original path determines the success of the inference attack. This is measured by the number of proximity circles that are visited by the inferred path segments divided by the total number of circles.

Figure 1b shows the example of an inferred path as well as the proximity circles representing the original GPS logs. The inference accuracy is measured using the above equation and is 0.91. This metric is useful in evaluating how well the algorithm works in identifying the band a user is travelling along.

4. INDIRECT TRAJECTORY INFERENCE ALGORITHM

Our proposed inference algorithm utilizes the location of query results to generate a Voronoi diagram, which for a given set of points, $p_i \in \mathcal{P}$, divides the space into a number of cells (regions) such that all the points in any cell, c_i , are closer to the corresponding p_i than to any other p . Voronoi diagrams are widely applied in nearest POIs problems. For the set \mathcal{P}_T , we generate a Voronoi diagram V , and retrieve a set of *candidate paths* that travel from one Voronoi region to the next. However, since Voronoi cells can be quite large in many areas, the number of these candidate paths is not restrictive enough to reconstruct a unique trajectory and needs to be further reduced. Therefore, maximum velocity bound and edge centrality have been employed in this work to infer the most likely path that was taken by the users.

4.1 Indirect Path Generation

To generate our initial paths, we propose an incremental search algorithm that iterates through each pair of points. On the first iteration, the Voronoi edge between the first and second point is determined and every path segment that intersects with the Voronoi edge is retrieved. This is illustrated on the left of Figure 2. All streets that intersect the Voronoi edge between p_1 and p_2 are retrieved. The initial retrieved path segments are considered as the starting path segments for the candidate path(s). The initial segments are

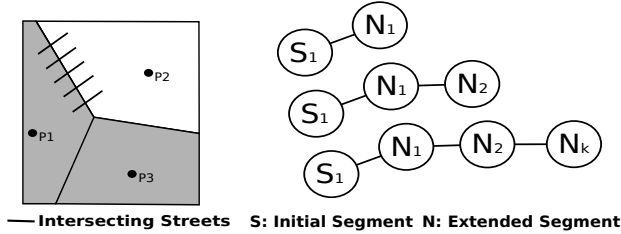


Figure 2: Generation and Extension of Initial Segments.

then passed to a function which retrieves all the connected path segments to the initial one and add them to the respective candidate path (Figure 2). Once a set of paths is generated, the last path segment of each path is checked to ensure that it is in the destination Voronoi cell, otherwise the entire candidate path is discarded.

In addition, assuming a maximum velocity bound, max_v , we can compute the maximum distance, d , that a user could have travelled between two consecutive timestamps. Therefore, the length of the path is checked and only if its difference with d is less than a threshold, δ , it is added to the set of candidate paths. Since the beginning of a path segment is not necessarily the start point of the trajectory, we assume the generated path may be slightly longer than d , i.e., δ longer.

4.2 Candidate Path Selection

To select a single path from the set of generated candidate paths we derived a weighting function to rank the paths based on edge centrality, and then select the top ranked candidate path as the path representing the user's trajectory.

The weighting function counts the frequency of path segment occurrence in each candidate path and stores this value in a hashtable. The weight for each candidate path is calculated by the summation of each path segment length divided by the length of the whole candidate path which is then multiplied by the frequency value of each path segment stored in the hashtable divided by the total number of candidate paths.

$$w_{path} = \sum_{i=0}^n \frac{pathSegmentFreq_i}{n} \times \frac{pathSegmentLength_i}{totalLength_i} \quad (1)$$

The weighting function returns the path that contains the greatest overlap with other paths in the candidate path set. Therefore, edges that are more commonly used in a set of candidate paths are favoured over edges that are not. Provided users take fairly direct routes to their destination, this weighting function works well.

5. EXPERIMENTS

5.1 Dataset

In our work we employ the GeoLife trajectory dataset¹ to evaluate the performance, i.e., accuracy, of our inference approach. The

¹www.research.microsoft.com/en-us/projects/geolife

GeoLife dataset consists of more than 17,000 trajectories that have been collected by 182 individuals over three years. We focused on a smaller part of the city of Beijing and retrieved those trajectories that fully reside inside this part. In total we ran our inference algorithm on 279 routes.

5.2 Implementation

We generate random POIs in the city of Beijing and stored them in a database as our \mathcal{P}_T . These POIs are generated uniformly inside the boundary of Beijing and they are then mapped to their closest road segment. In order to evaluate the performance of our inference algorithm for different scenarios, we create four POI batches of 400, 800, 1600 and 3200 points to reflect dense and sparse areas. For example, 1600 POIs equate to 4 POIs per square kilometer.

In the road network, there are many path combinations a user can travel along to get to a destination. This leads to a large search space and an inefficient search process. To reduce the search complexity, we employ a pruning method that discards any combinations that terminates at the same road while expanding paths between Voronoi cells. Moreover, due to the geometric nature of Voronoi diagrams, individual Voronoi cells can become quite small and may create a case where some paths overshoot the cell and lead to zero candidate paths. In order to compensate for this case, the algorithm allows paths to continue to expand even if they did not terminate in the current Voronoi cell. However a path is finally removed if it does not end in the next Voronoi cell in the next iteration. In our

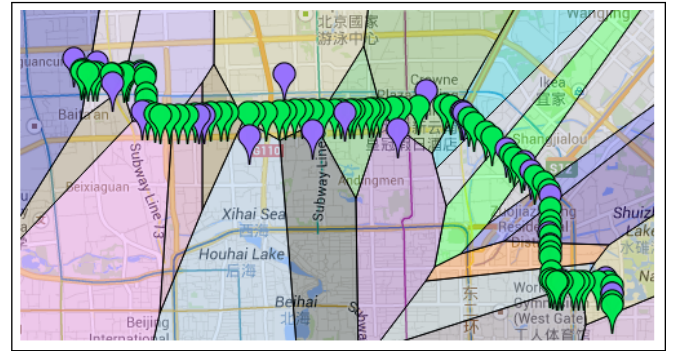


Figure 3: Voronoi Diagram

implementation we use the OpenStreetMap² data to generate the road network graph. A web interface is also constructed to visually view the data using PHP, Javascript and the Google Maps API. Javascript is used to implement the trajectory inference algorithm and an OpenStack cloud computing environment is utilised to run the inference algorithm. Moreover, the Bower-Watson algorithm is employed to compute the Voronoi diagrams. An example GeoLife path is also illustrated in Figure 3, where the purple (darker) flags illustrate POIs along the path and their respective Voronoi cells. The green (lighter) flags show the original GPS logs.

5.3 Experimental Results

We measure the accuracy of our inference algorithm as the number of proximity circles that are visited by the inferred path divided by the total number of circles (Section 3.3). To estimate the inference accuracy of our approach, we consider varying radii, r , in meter to generate proximity circles, where $r \in \{10, 50, 100, 250, 500\}$. Table 2 shows the performance of our inference algorithm for varying POI densities. Although the overall accuracy is low for very

²www.openstreetmap.org

POI	$r = 50m$	$r = 100m$	$r = 250m$	$r = 500m$
400	27.73	39.10	51.83	64.74
800	35.10	47.97	61.31	73.76
1600	39.00	53.90	69.63	80.84
3200	36.32	49.74	64.38	75.37

Table 1: Experimental results using edge centrality.

POI	$r = 50m$	$r = 100m$	$r = 250m$	$r = 500m$
400	32.15	44.52	58.09	71.31
800	38.03	51.93	65.85	77.79
1600	41.07	56.44	71.62	81.40
3200	37.97	52.45	67.70	77.97

Table 2: Experimental results using edge frequency.

small radius of 10 meters, our results show that for more realistic buffers and higher densities our approach is successful in accurately estimating a user trajectory and can achieve an average accuracy level beyond 80%. This shows with an increase in POI density and for urban areas with higher POI densities such as city centers, the inferred paths get closer to the original paths incurring higher privacy risks for a user.

In order to understand if access to the trip patterns of users can increase an attacker’s ability to infer a user’s original path, we ran experiments using edge frequency instead of edge centrality. The edge frequency (see Section 3.2) is computed on the basis of actually travelled trips. Our experiments show small gains in terms of accuracy but also demonstrate that this additional knowledge does not significantly improve an attacker’s ability to infer a user’s path. Our findings show that in either case the ability of an attacker to infer a user’s path based on POI information is high.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an algorithm that only employs a set of POIs to indirectly infer a user’s trajectory. Our results suggest that even coarse location information allows us to approximate a user’s trajectory within dense urban areas with high accuracy. Therefore, exchanging query results of LBS users instead of their tracks does not offer adequate privacy protection.

While our inference attacks are effective in areas with high POI densities, there are still a number of directions that are likely to make the overall inference strategy more effective. We assume in our work that for every request there is only the closest POI available, however, an LSP usually provides users with several POIs for a single request. This information could be encoded as a higher-order Voronoi diagram that leads to smaller cells and thus should enable a more refined attack strategy.

In our work we have used positive information, i.e., information directly shared by a location service provider. However, another location service provider (or adversary) could also have the information about all the POIs that were not revealed because they were not among the closest POIs. Since the overall number of POIs is much larger than the number of POIs returned as a query result, the underlying Voronoi diagram may result in smaller cells, which in turn should improve the accuracy of an inference attack algorithm. We are currently investigating these strategies.

7. REFERENCES

- [1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.
- [2] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *PerComp*, 7(5):275–286, 2003.
- [3] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky. Human mobility characterization from cellular network data. *Communications of the ACM*, 56(1):74–82, 2013.
- [4] A. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.
- [5] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *SIGKDD*, pages 70–78, New York, NY, USA, 2008.
- [6] M. L. Damiani, E. Bertino, and C. Silvestri. Protecting location privacy against spatial inferences: the probe approach. In *ACM SPRINGL*, pages 32 – 41, 2009.
- [7] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In H.-W. Gellersen, R. Want, and A. Schmidt, editors, *PerComp*, volume 3468, pages 152–170. 2005.
- [8] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, pages 31 – 42, 2003.
- [9] T. Hashem and L. Kulik. Don’t trust anyone: Privacy protection for location-based services. *Pervasive and Mobile Computing*, 7(1):44 – 59, 2011.
- [10] E. Kaplan, T. B. Pedersen, E. Savas, and Y. Saygin. Discovering private trajectories using background information. *Data and Knowledge Engineering*, 69(7):723 – 736, 2010.
- [11] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS*, pages 88–97, 2005.
- [12] J. Krumm. Inference attacks on location tracks. In *PerComp*, pages 127–143. 2007.
- [13] M. Lin, W.-J. Hsu, and Z. Q. Lee. Predictability of individuals’ mobility with high-resolution positioning data. In *UbiComp*, pages 381–390, 2012.
- [14] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao. Privacy vulnerability of published anonymous mobility traces. In *MobiCom*, pages 185 – 196, 2010.
- [15] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *ACM SPRINGL*, pages 52 – 61, 2008.
- [16] L. Sweeney. K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
- [17] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, pages 65 – 72, 2008.
- [18] M. Wernke, F. Durr, and K. Rothermel. Pshare: Position sharing for location privacy based on multi-secret sharing. In *PerCom*, pages 153–161, 2012.
- [19] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *MobiCom*, pages 145–156, 2011.