Differentially Private Trajectory Analysis for Points-of-Interest Recommendation

Chao Li
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
Email: chl205@pitt.edu

Balaji Palanisamy
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
Email: bpalan@pitt.edu

James Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
Email: jjoshi@pitt.edu

***Abstract*—Ubiquitous deployment of low-cost mobile positioning devices and the widespread use of high-speed wireless networks enable massive collection of large-scale trajectory data of individuals moving on road networks. Trajectory data mining finds numerous applications including understanding users' historical travel preferences and recommending places of interest to new visitors. Privacy-preserving trajectory mining is an important and challenging problem as exposure of sensitive location information in the trajectories can directly invade the location privacy of the users associated with the trajectories. In this paper, we propose a differentially private trajectory analysis algorithm for points-of-interest recommendation to users that aims at maximizing the accuracy of the recommendation results while protecting the privacy of the exposed trajectories with differential privacy guarantees. Our algorithm first transforms the raw trajectory dataset into a bipartite graph with nodes representing the users and the points-of-interest and the edges representing the visits made by the users to the locations, and then extracts the association matrix representing the bipartite graph to inject carefully calibrated noise to meet ϵ-differential privacy guarantees. A post-processing of the perturbed association matrix is performed to suppress noise prior to performing a Hyperlink-Induced Topic Search (HITS) on the transformed data that generates an ordered list of recommended points-of-interest. Extensive experiments on a real trajectory dataset show that our algorithm is efficient, scalable and demonstrates high recommendation accuracy while meeting the required differential privacy guarantees.**

I. Introduction

Ubiquitous deployment of mobile positioning devices and the widespread use of high-speed wireless networks enable massive collection of large-scale trajectory data of individuals moving on road networks. The rapid proliferation of low-cost GPS-supported mobile devices enables a wide range of location-based service (LBS) applications including location-based social networks [10], [30], location-based advertising [2], [22], location-based information sharing [9], [24] and navigation applications [28], [32]. A trajectory represents a sequence of location information formed by a series of $(latitude, longitude, timestamp)$ triple that captures a variety of travel information of the users including user's movement pattern [17], travel paths [36] and travel destination [34], [35]. Each travel destination in a trajectory reveals that the user has made a visit to the place. In addition to this information, a trajectory may also include temporal information such as the visiting times of the users. While trajectories of an individual mobile user can be analyzed to understand her personal travel recommendations comprehensively, aggregate analysis of historical trajectory data belonging to different mobile users can provide more generalized travel recommendations such as 'Where are the top-10 points-of-interest in a given city?', 'Which shopping mall is the most popular in this area?' and 'Which users have frequently visited this restaurant?'.

Although historical personal trajectory data provide immense information to generate accurate and useful points-of-interest recommendation, the exposure of the sensitive trajectory information can pose significant privacy risks that can invade the location privacy of the users. In particular, the location information of the travel destination, represented as a two-dimensional geographical region, is often associated with a semantic meaning, such as a university, a shopping mall or a hospital. The disclosure of the association between a mobile user and such a location may reveal private information about the health conditions, life style and social and political beliefs of the user. For example, if an adversary infers the association between a user and a treatment center, the health state of the user may be revealed.

Privacy-preserving trajectory mining is an important and challenging problem as exposure of sensitive location information in the trajectories can directly invade the location privacy of the users associated with the trajectories. Differential privacy [12], [13], as a state-of-the-art privacy paradigm, provides a model to quantify the disclosure risks by ensuring that the published statistical data does not depend on the presence or absence of an individual record in the dataset. By carefully applying differential privacy mechanisms [12], [13], [25] on trajectory data, the personal trajectory information, such as the travel destination of the users, can be protected from the malicious or curious inference from the recommendation results, thus protecting the location privacy of the mobile users. In this paper, we propose a differentially private trajectory analysis algorithm for points-of-interest recommendation to users that aims at maximizing the accuracy of the recommendation results while protecting the privacy of the exposed trajectories with differential privacy guarantees. Our algorithm first transforms the raw trajectory dataset into a bipartite graph with nodes representing the users and the points-of-interest

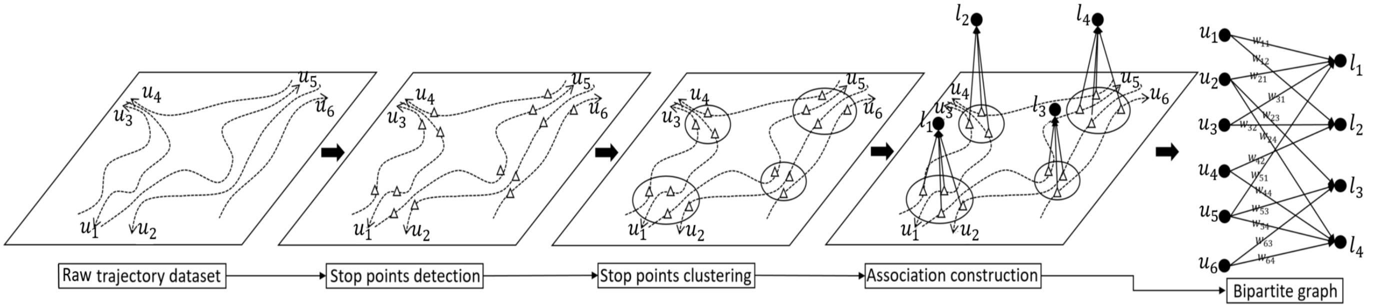


Fig. 1: User-location bipartite graph construction

and the edges representing the visits made by the users to the locations, and then extracts the association matrix representing the bipartite graph to inject carefully calibrated noise to meet ϵ-differential privacy guarantees. A post-processing of the perturbed association matrix is performed to suppress noise prior to performing a Hyperlink-Induced Topic Search (HITS) on the transformed data that generates an ordered list of recommended points-of-interest. We perform extensive experiments on the *Geolife* GPS trajectory dataset [34], [35], [36] which contains 17621 trajectories collected from 182 users for five years. Our results show that the proposed algorithm is efficient, scalable and demonstrates good recommendation accuracy while guaranteeing differential privacy.

The rest of the paper is organized as follows: We first discuss the related work in Section II. Then, in Section III, we present the definitions of trajectory processing and differential privacy and the model to transfer a raw trajectory dataset to a user-location bipartite graph. In Section IV, we introduce our privacy goal and explain the proposed differentially private trajectory analysis algorithm for travel recommendation. We experimentally evaluate our algorithm in Section V under varying differential privacy budgets, global sensitivity levels and database scale. Finally, we conclude in Section VI.

II. Related work

Location privacy has been an active research area for a long time. In the past, the location privacy protection mechanisms mainly focused on prevention by protectively processing and perturbing the location information prior to disclosure. Depending on processing location data discretely or continuously, the location privacy protection mechanisms can be roughly classified to location data perturbation techniques represented by [16], [23], [27], [31], and trajectory inference prevention techniques represented by [3], [4], [5], [6]. The latter can be further broken down into trajectory perturbation [6] and Mix-zone [3], [4], [5] techniques. However, all these techniques assume to restrict the background knowledge of the adversary, which fails to provide strong and quantifiable privacy guarantee.

Differential privacy [12], [13], as a state-of-the-art privacy paradigm, provides a model to quantify the disclosure risks by ensuring that the published statistical data does not depend on the presence or absence of an individual record in the dataset. Differential privacy can dispense with the restriction of the adversary background knowledge and quantify the privacy in a mathematically provable manner. By carefully applying differential privacy mechanisms [12], [13], [25] to the trajectory data, the personal trajectory information can be protected from malicious inference from the statistical outputs. Usually, the raw trajectory dataset is first transferred to special data structures, such as Prefix tree [8], [19] or N-gram [7]. Then, the differential privacy protection mechanisms (e.g. Laplace Mechanism [12], Exponential Mechanism [25]) inject noises to the data structures before releasing them for further processing. To the best of our knowledge, the work presented in this paper is the first differential privacy protection mechanism aimed at processing trajectories modeled as bipartite graphs to generate accurate travel recommendation while protecting the location privacy of the users in the trajectories.

III. Concepts and Model

In this section, we first present the background concepts and the bipartite graph model used to model raw trajectory datasets and the user-location associations in the dataset. We then discuss the differential privacy model and mechanisms for achieving differential privacy.

A. User-location bipartite graph representation

The trajectory dataset analysis typically consists of two major components, namely trajectory preprocessing and trajectory analysis [33]. Depending on the objective of the analysis, the output of trajectory preprocessing can be organized as graphs [34], matrix [35] or tensors [29]. For the purpose of points-of-interest recommendation considered in this work, the raw trajectory dataset is transformed to be processed as a bipartite graph [33]. A bipartite graph can be represented by a graph, $G = (U, L, E)$, containing $m = |U|$ nodes on the left side, $n = |L|$ nodes on the right side and a set of edges $E \subseteq U \times L$ between the two sets of nodes. This structure naturally meets the objectives of the points-of-interest recommendation problem as the left nodes and right nodes can respectively represent users and point-of-interest (POI) locations to be recommended. Here an edge $e_{ij} = (u_i, l_j) \in E$ indicates that the user u_i visited the POI location l_j. In addition, we expect to know the frequency of visit between user u_i and location l_j modeled as the edge weight w_{ij} of the edge e_{ij}.

To construct such a user-location bipartite graph from the raw trajectory dataset, we follow a sequence of three steps as shown in Figure 1. We start from the raw trajectory dataset:

Definition 1 (RAW TRAJECTORY DATASET). *The raw trajectory dataset $RTD = \{(u_i, TD_i)|1 \leq i \leq m\}$ contains the trajectory data (TD_i) for m users (u_i). The trajectory data $TD_i = \{(x_j, y_j, t_j)|1 \leq j \leq k_i\}$ of user u_i is formed by k_i triple, consisting of latitude x_j, longitude y_j and timestamp t_j ($t_j < t_{j+1}$).*

In the example shown in Figure 1, we find the trajectory data for users u_1 to u_6 as six dotted lines as the trajectory data is always discretely captured by GPS devices. In the first step, we identify the stop points for all the users. A stop point is defined as a spatial region that the trajectory data fluctuates within a distance threshold D_t for at least a time threshold T_t.

Definition 2 (STOP POINTS SET). *The stop points set $SPS = \{(u_i, SP_i)|1 \leq i \leq m\}$ contains the stop points information (SP_i) for the m users (u_i). The stop points information $SP_i = \{sp_j = (x_j, y_j)|1 \leq j \leq p_i\}$ for user u_i is formed by p_i stop points. A stop point sp is detected when a subset of sequential triple of TD_i, $\{(x_j, y_j, t_j)|a \leq j \leq b\}$ follows $\forall a < j \leq b$, $\sqrt{(x_j - x_a)^2 + (y_j - y_a)^2} \leq D_t$, $\sqrt{(x_{b+1} - x_a)^2 + (y_{b+1} - y_a)^2} > D_t$, $t_b - t_a \geq T_t$.*

In Figure 1, the stop points are identified as triangles. Subsequently in the second step, these stop points (triangles) are clustered through well-known clustering techniques such as k-means [15], DBSCAN [14] or OPTICS [1] clustering algorithms. These clustered stop points implicitly recommend those regions covered by the clusters as attractive places as multiple users in the RTD dataset have historically visited (stopped at) them. As can be seen in Figure 1, the stop points of the six users form four clusters, represented by l_1 to l_4. We denote each cluster as a location as we need to assign a geographically semantic meaning to the cluster for travel recommendation. In practice, these locations can be represented by the landmarks (e.g. tourist attractions, shopping malls) within the clusters.

Finally, in the third step, we need to construct the associations between the users and the locations to build the user-location bipartite graph. We can connect each stop point with its associated location with an arrow line pointing to the location, which denotes that the user of this stop point has visited the location once. Actually, a user may have more than one stop points within one cluster, which indicates that this user visited this location multiple times. This information, called frequency of visit, is important for travel recommendation since a more frequent visit can implicitly represent a stronger recommendation. Therefore, if we denote an edge in the user-location bipartite graph to mean that the user visited the location, we can apply the frequency of visit as the weight of the edge to indicate that the user has visited the location multiple times. Based on this assumption, we define the user-location bipartite graph as:

Definition 3 (USER-LOCATION BIPARTITE GRAPH). *The user-location bipartite graph $ULBG = (U, L, E)$ consists of the left set of user nodes $U = \{u_i|1 \leq i \leq m\}$, the right set of location nodes $L = \{l_j|1 \leq j \leq n\}$ and the set of visits represented as edges $E = \{e_{ij} = (u_i, l_j, w_{ij})|1 \leq i \leq m, 1 \leq j \leq n\} \subseteq U \times L$, where w_{ij} is the frequency of the visit.*

We next introduce the notion of differential privacy and the mechanisms required to achieve differential privacy guarantees in a dataset.

B. Differential privacy

Differential privacy is a classical privacy definition [12] that makes conservative assumptions about the adversary's background knowledge and bounds the allowable error in a quantified manner. In general, differential privacy is designed to protect a single individual's privacy by considering adjacent data sets which differ only in one record. Before presenting the formal definition of ϵ-differential privacy, we first define the notion of adjacent datasets in the context of differential privacy. A data set D can be considered as a subset of records from the universe U, represented by $D \in \mathbb{N}^{|U|}$, where $\mathbb{N}$ stands for the non-negative set and D_i is the number of element i in $\mathbb{N}$. For example, if $U = \{a, b, c\}$, $D = \{a, b, c\}$ can be represented as $\{1, 1, 1\}$ as it contains each element of U once. Similarly, $D' = \{a, c\}$ can be represented as $\{1, 0, 1\}$ as it does not contain b. Based on this representation, it is appropriate to use l_1 distance (Manhattan distance) to measure the distance between data sets.

Definition 4 (DIFFERENTIAL PRIVACY [12]). *A randomized algorithm $\mathcal{A}$ guarantees ϵ-differential privacy if for all adjacent datasets D_1 and D_2 differing by at most one record, and for all possible results $\mathcal{S} \subseteq Range(\mathcal{A})$,*

$$Pr[\mathcal{A}(D_1) = \mathcal{S}] \leq e^{\epsilon} \times Pr[\mathcal{A}(D_2) = \mathcal{S}]$$

where the probability space is over the randomness of $\mathcal{A}$.

In other words, the possible results of the randomized algorithm, given a dataset and a query, can form a distribution and differential privacy guarantees that the change of the distribution for two input databases differing in one record is bounded by a threshold. Many randomized algorithms have been proposed to guarantee differential privacy [12], [25]. In our work, we use the most commonly used differential privacy mechanism namely the Laplace Mechanism [12]. Given a data set D, a function f and the budget ϵ, the Laplace Mechanism first calculates the actual $f(D)$ and then perturbs this true answer by adding a carefully calibrated noise [12]. The noise is calculated based on a Laplace random variable, with the variance $\lambda = \triangle f/\epsilon$, where $\triangle f$ is the l_1 sensitivity, which is defined as follows.

Definition 5 (l_1 SENSITIVITY [13]). *Given a function $f : \mathbb{N}^{|U|} \rightarrow \mathbb{R}^d$, the l_1 sensitivity is measured as:*

$$\triangle f = \max_{\substack{D_1, D_2 \in \mathbb{N}^{|U|} \\ ||D_1 - D_2||_1 = 1}} ||f(D_1) - f(D_2)||_1$$

where $||f(D_1) - f(D_2)||_1 = |f(D_1) - f(D_2)|$ is the Manhattan Distance and U stands for the record universe.

In other words, l_1 sensitivity measures the maximum impact that can be caused by changing a single record in a dataset. It

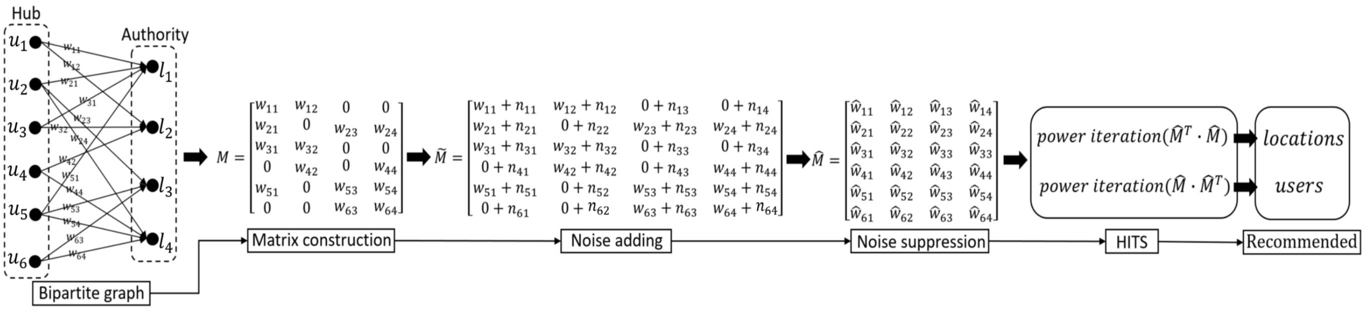


Fig. 2: Differentially private bipartite graph mining

is only related to the function f itself, but independent of the data sets.

Definition 6 (Laplace Mechanism [12]). *Given a function $f : \mathbb{N}^{|U|} \to \mathbb{R}^d$, a budget ϵ and a data set D, for each output,*

$$\mathcal{A}_{LM}(D, f, \epsilon) = f(D) + Lap(\triangle f/\epsilon)$$

where $Lap(\triangle f/\epsilon)$ is a random variable sampled from the Laplace distribution with 0 mean and $\triangle f/\epsilon$ variance.

We next discuss the proposed differentially private bipartite graph analysis techniques for trajectory analysis that employs a Laplace Mechanism to achieve differential privacy guarantees in the exposed points-of-interest recommendation.

IV. Differentially Private Trajectory Analysis

The proposed differentially private trajectory analysis technique analyzes the user-location bipartite graph obtained from the raw trajectories to generate a list of ordered recommended locations (points-of-interest) and another list of ordered recommended users based on their frequency of visits to a location. We start from presenting the privacy goal, namely what is identified as a 'record' in Definition 4 that needs to be protected by the differential privacy mechanism. We then present the proposed differentially private algorithm for bipartite graph analysis that generates the two recommendation lists from the user-location bipartite graph.

A. Privacy goal

By considering the user-location bipartite graph as a dataset, a 'record' in Definition 4 has the form '$< u_i, l_j >$', representing user u_i visited location l_j once, namely a stop point defined in Definition 2. Therefore, an edge $e_{ij} = (u_i, l_j, w_{ij})$ in the bipartite graph, which can be considered as a group of w_{ij} of $(u_i, l_j, 1)$, results in w_{ij} of same $< u_i, l_j >$ records in the dataset. The entire bipartite graph can be transferred to a dataset with $\sum_{i=1,j=1}^{m,n} w_{ij}$ records. Then, by setting the global l_1 sensitivity $\triangle f$ in Definition 5 to different values, we can protect different levels of differential privacy. We define $\triangle f \in [1, w_{max}]$, where w_{max} represents the maximum weight among the edges. The proposed differentially private trajectory analysis algorithm with sensitivity $\triangle f \in [1, w_{max}]$ can thus protect differential privacy of a group of $\triangle f$ records. In other words, the differential privacy of any edge $e_{ij} = (u_i, l_j, w_{ij})$ in the bipartite graph with $w_{ij} \leq \triangle f$ can be protected. When $\triangle f = 1$, the edges with $w_{ij} = 1$, namely the visits that only happened once, can be protected. When $\triangle f = w_{max}$, all the edges in the bipartite graph corresponding to the association between each pair of user and location, can be protected. To sum up, a larger $\triangle f$ results in more noisy edges in the bipartite graph to be protected but results in lower recommendation results due to higher injected noise. We will evaluate the varying $\triangle f$ later in Section V.

B. Differentially private Points-of-Interest Recommendation

Among the recommendation algorithms [11], [20], [26], [35], the one that fits the bipartite graph structure best is the HITS-based algorithm [35]. Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm originally designed for web pages rating [21]. It defines a hub as a web page with many links pointing to other web pages and an authority as a web page pointed by many other web pages. It assumes a good hub points to many good authorities and a good authority is pointed by many good hubs. In the user-location bipartite graph, edges point to locations from users. Therefore, by considering users and locations as hubs and authorities respectively, we can apply HITS algorithm to score every user and location. A user with higher score represents a more experienced user who has more knowledge about the given city and a location with higher score indicates a more popular place that is worth being visited. Therefore, with the user-location bipartite graph as input, we expect the algorithm to output the user and location lists ordered by the scores while preserving differential privacy. We present a pseudocode of the differentially private mining algorithm in Algorithm 1 and illustrate the process in Figure 2.

The differentially private mining algorithm consists of four steps, namely matrix construction (line 1-2), noise addition (line 3-9), noise suppression (line 10) and HITS (line 11-12). In the first step, the bipartite graph structure is transformed into an association matrix M. The matrix M has $m = |U|$ rows and $n = |L|$ columns. Each entry in M is denoted by the weight w_{ij} of the edge between user u_i and location l_j. If user u_i never visited location l_j, w_{ij} is set to 0.

Then, in the second step, each entry w_{ij} is perturbed using noise calculated by Laplace Mechanism [12] to become $\widetilde{w}_{ij} = w_{ij} + Lap(\frac{\triangle f}{\epsilon})$ so that M becomes $\widetilde{M}$. Each noise is a random variable sampled from Laplace distribution with variance $\frac{sensitivity}{budget}$. The privacy budget ϵ is typically considered to be 1 in many differential privacy settings [12], [13]. A

Algorithm 1: Differentially private bipartite graph mining

```
Input  : User-location bipartite graph $ULBG = (U, L, E)$,
         expected privacy level $lv \in [1, w_{max}]$, budget $\epsilon$.
Output: Recommended authority (locations) list $A$,
         recommended hub (users) list $H$.
Initialize matrix $M[m = |U|][n = |L|]$, $\widetilde{M}[m][n]$, $\widehat{M}[m][n]$
  with 0s;
Transfer $ULBG$ to matrix $M$ by filling $M$ with
  $\{w_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$;
$\triangle f = lv$;
$\delta = \frac{\triangle f}{\epsilon}$;
for $i = 1$ to $m$ do
    for $i = 1$ to $n$ do
        $\widetilde{M}[i][j] = M[i][j] + Lap(\delta)$;
    end
end
$\widehat{M} = Sup(\widetilde{M})$;
$A = powerIte(\widehat{M}^T \cdot \widehat{M})$;
$H = powerIte(\widehat{M} \cdot \widehat{M}^T)$;
```

smaller ϵ indicates that the difference between the statistical query replies caused by the change of one record in the dataset is smaller, thus providing better privacy protection. We will evaluate the impact of different values of ϵ in Section V. In our algorithm, all the entries share the same privacy budget ϵ because of their independence and the composition property of differential privacy:

Theorem 1 (COMPOSITION THEOREM [13]). *Let $\mathcal{A}_i$ be ϵ_i-differential private algorithms applying to independent datasets D_i for $i \in [1, k]$. Then their combination $\mathcal{A}_{\sum_{i=1}^k}$ is $max(\epsilon_i)$-differential private.*

The sensitivity $\triangle f$ can be selected from the range $[1, w_{max}]$. However, we note that the noise can be significantly high without further post-processing. Due to higher noises, the accuracy of recommendation can actually become too low to be acceptable. Therefore, in the third step, we apply consistency constraints to post-process the matrix from $\widetilde{M}$ to $\widehat{M}$ to suppress noise and improve the recommendation accuracy.

Theorem 2 (POST-PROCESSING [13]). *Let $\mathcal{A}$ be a ϵ-differentially private algorithm and g be an arbitrary function. Then $g(\mathcal{A})$ is also ϵ-differentially private.*

We propose three consistency constraints, named zero consistency constraint(zero-CC), up consistency constraint (up-CC) and down consistency constraint(down-CC). In zero-CC, since all the entries w_{ij} in M are non-negative, the negative $\widetilde{w}_{ij}$ in $\widetilde{M}$ after perturbation is constrained to be $\widehat{w}_{ij} = 0$ in $\widehat{M}$ to ensure consistency while the non-negative $\widetilde{w}_{ij}$ keeps same in $\widehat{M}$ as $\widehat{w}_{ij} = \widetilde{w}_{ij}$. The up-CC and down-CC follow the isotonic regression [18]. Specifically, in up-CC, the entries w_{ij} in M are sorted from 0 to w_{max} in non-decreasing order. After adding noises to the non-decreasing list of entries, the perturbed entries $\widetilde{w}_{ij}$ should also keep the non-decreasing order to ensure consistency. If one perturbed entry is smaller than the one before it in the list, this perturbed entry will be adjusted to be equal to the one before it, thus making the list non-decreasing. After adjusting all $\widetilde{w}_{ij}$ to $\widehat{w}_{ij}$, the non-decreasing list is transferred back to matrix, namely $\widehat{M}$. The down-CC is similar to up-CC, but the list is constrained to follow a non-increasing order from w_{max} to 0.

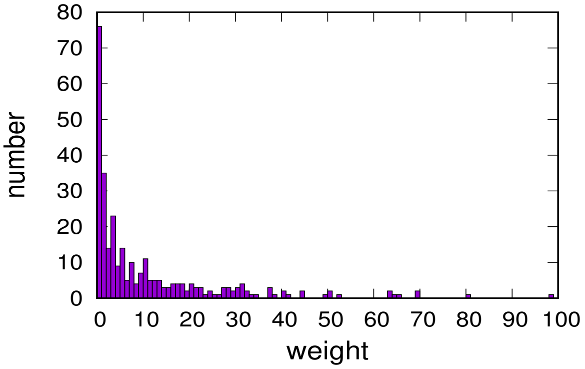


Fig. 3: Edge weight distribution

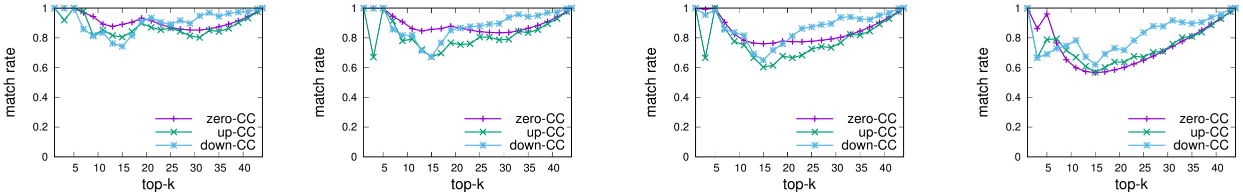
Finally, after generating $\widehat{M}$, we apply the HITS algorithm to calculate the recommended authority (locations) list A and the recommended hub (users) list H. Precisely, we can initialize A and H to be vectors of 1s with size n and m respectively. Then, by applying power iteration method [35], we can get the eigenvectors of $\widehat{M}^T \cdot \widehat{M}$ and $\widehat{M} \cdot \widehat{M}^T$ as the final A and H respectively. We refer the interested readers to [21] for more details on HITS.

V. EXPERIMENTAL EVALUATION

In this section, we experimentally evaluate the performance offered by the proposed differentially private trajectory analysis algorithm. Before reporting our results, we first present our experimental setup.

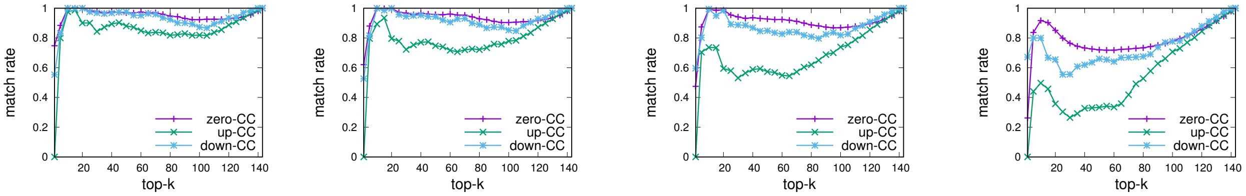
A. Experimental setup

Our experiments were programmed in Java language and implemented on an Intel Core i7 2.70GHz PC with 16GB RAM. In our experiments, we apply the *Geolife* GPS trajectory dataset [34], [35], [36], which contains 17621 trajectories collected from 182 users for five years. We first follow the user-location bipartite graph construction scheme shown in Figure 1 to process the raw trajectory dataset and generate the user-location bipartite graph. The sizes of node sets, edge set and stop point set are shown in Table I (some users are abundant during clustering). Each edge in the user-location bipartite graph is assigned a weight w representing the number of visits. The distribution of weights among the 316 edges is shown in Figure 3 (we just show the part for $w \leq 100$) and the statistics of weights is shown in Table II. As can be seen, most edges have small weights, indicating that users have many rarely visited locations. This suggests the intuition behind our privacy goal. That is, we can inject little noise to protect most of the edges in the bipartite graph. Typically, these protected edges have lower weights but higher sensitivity.



(a) $\epsilon = 1$ (b) $\epsilon = 0.7$ (c) $\epsilon = 0.4$ (d) $\epsilon = 0.1$

Fig. 4: Recommended authority list A with varying privacy budget ϵ



(a) $\epsilon = 1$ (b) $\epsilon = 0.7$ (c) $\epsilon = 0.4$ (d) $\epsilon = 0.1$

Fig. 5: Recommended hub list H with varying privacy budget ϵ

Users	143
Locations	44
Stop points	8017
Edges	316

TABLE I: No. of

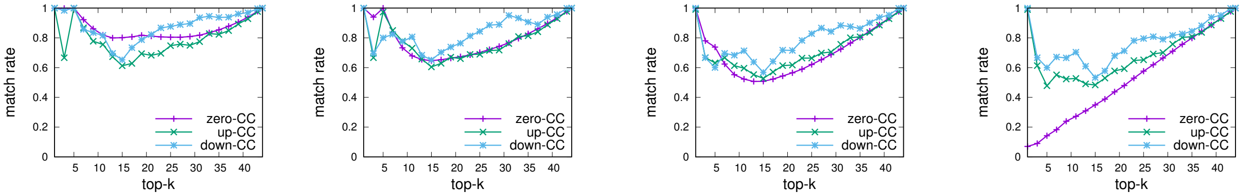
First quartile	2
Median	5
Third quartile	18
Max	591

TABLE II: w statistics

B. *Experimental results*

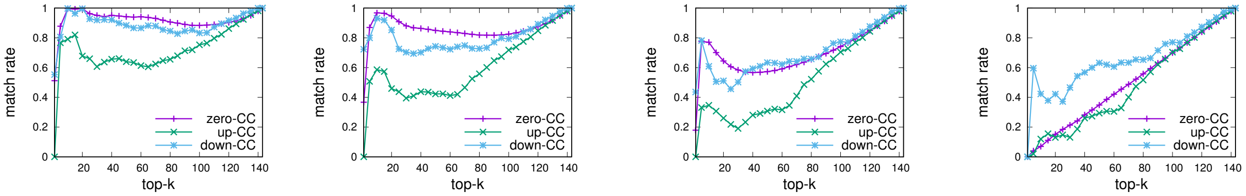
The goal of our experiments is to evaluate the performance of the proposed differentially private user-location bipartite graph analysis algorithm under various privacy budgets and sensitivity values. That is, we evaluate the accuracy of the recommendation results while simultaneously meeting the differential privacy guarantees. We define *top-k match rate* to measure the recommendation results. If we denote the recommended authority (location) list as $A = \{a_1, a_2, ..., a_n\}$ and the recommended hub (user) list as $H = \{h_1, h_2, ..., h_n\}$, where the elements with smaller index have higher score, and represent the top-k elements in the list as $A_k = \{a_1, a_2, ..., a_k\}$ and $H_k = \{h_1, h_2, ..., h_k\}$ respectively, the top-k match rate for A is $MR_k(A) = \frac{org(A_k) \bigcap noised(A_k)}{org(A_k)}$ and the top-k match rate for H is $MR_k(H) = \frac{org(H_k) \bigcap noised(H_k)}{org(H_k)}$, where $org()$ denotes the original lists without injected noise to protect differential privacy and $noised()$ stands for the differentially private lists with noise. In other words, for a query like 'show me top-5 recommended locations in this city', we expect the replied 5 locations do not change after adding noise. Our experiments consist of three components. The noise calibrated by the Laplace mechanism $Lap(\frac{\triangle f}{\epsilon})$ is affected by the two parameters, privacy budget ϵ and global sensitivity $\triangle f$. Therefore, in the first part, we adjust the privacy budget ϵ to observe the change of $MR_k(A)$ and $MR_k(H)$. Then, in the second part, we adjust the global sensitivity $\triangle f$ to evaluate the algorithm performance. Finally, to evaluate the scalability of the algorithm, we reduce the raw dataset scale by only using the first-100-day trajectory data and only using the first-90-user trajectory data respectively. In all the experiments, we evaluate the three consistency constraint schemes, denoted by 'zero-CC', 'up-CC' and 'down-CC'. For each experiment, we repeat 1000 times and show the average.

In the first set of experiments, we change the value of privacy budget ϵ from 1 to 0.7, 0.4 and 0.1 and show the results with varying k of $MR_k(A)$ and $MR_k(H)$ as Figure 4 and Figure 5 respectively. In differential privacy, a smaller privacy budget means the difference between query replies from two datasets differing in at most one record is smaller, which implies higher privacy requirement and requires more noise to guarantee better privacy protection. For this part, we fix the global sensitivity to be 1 to protect differential privacy of individual stop point or the 76 weight-1 edges. When $\epsilon = 1$, Figure 4(a) and Figure 5(a) show that for most values of k, $MR_k(A)$ and $MR_k(H)$ for all the three consistency constraint schemes are larger than 80%. If we keep choosing the consistency constraint scheme giving the best results, the match rates can even be larger than 90%. As $\epsilon = 1$ is typical in most differential privacy settings [12], [13], the 90% accuracy shows that our differentially private bipartite graph analysis algorithm for travel recommendation is practical and effective. In Figure 4(a), we can see that zero-CC provides best match rate when $k \leq 20$, which is defeated by down-CC for $k > 20$. In Figure 5(a), although down-CC is very close to zero-CC, zero-CC gives best results for nearly all the values of k. In both the two figures, up-CC shows worst performance. All the above observations can be explained by the principles of the three consistency constraint schemes and the features of the dataset. The matrix M (input of HITS algorithm) is sparse because most of its entries are 0, indicating no edge between corresponding user and location. In Figure 3, we see that most edges have very small weights and only a small number of edges have outstanding weights. Therefore, the entries in M can be divided into three classes, entries with high weights, entries with low weights and entries with 0 weights. After adding noise to each entry, the perturbed low-weight entries (e.g. $w = 1 + noise$) are almost indistinguishable from the perturbed 0-weight entries (e.g. $w = 0 + noise$), but the perturbed high-weight entries (e.g. $w = 100 + noise$) still



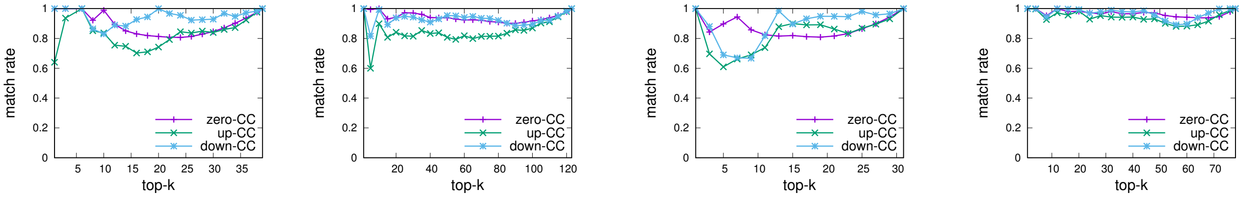
(a) $\triangle f = 2$ (b) $\triangle f = 5$ (c) $\triangle f = 18$ (d) $\triangle f = 591$

Fig. 6: Recommended authority list A with varying sensitivity $\triangle f$



(a) $\triangle f = 2$ (b) $\triangle f = 5$ (c) $\triangle f = 18$ (d) $\triangle f = 591$

Fig. 7: Recommended hub list H with varying sensitivity $\triangle f$



(a) $MR_k(A)$, 100 days (b) $MR_k(H)$, 100 days (c) $MR_k(A)$, 90 users (d) $MR_k(H)$, 90 users

Fig. 8: Varying scalability

have outstanding weights. In noise suppression, the zero-CC tends to make $w = 0 - noise$ to $w = 0$ and has little influence to the high-weight entries, low-weight entries and difference between low-weight entries and 0-weight entries. Both the up-CC and down-CC have greater influence to the high-weight entries and low-weight entries (makes their order changed). Also, the up-CC reduces the difference between low-weight entries and 0-weight entries while down-CC increases the difference by making $w = 0 + noise$ to $w = 0$. In HITS, if a user or location is linked by more edges with high weights, it has more chance to be recommended. In Figure 4(a), the locations can be divided into two sets. The first set of locations has higher scores determined by high-weight entries and low-weight entries while the second set of locations have lower scores determined by low-weight entries and 0-weight entries. Therefore, zero-CC, which has little influence to the high-weight entries and low-weight entries, dominates the first set while down-CC, which makes low-weight entries distinguishable from 0-weight entries, dominates the second set. The handover point between zero-CC and down-CC gradually decreases from 20 when $\epsilon = 1$ to 9 when $\epsilon = 0.1$ as shown in Figure 4(b) to 4(d). In Figure 5(a), unlike the locations, most users visited the very popular locations, so their scores are mainly affected by the high-weight entries and low-weight entries, which makes zero-CC to dominate the entire set. The reduction of ϵ from Figure 5(b) to 5(d) only makes the advantage of zero-CC more transparent. To sum up, we recommend zero-CC as default consistency scheme for $\triangle f = 1$ case.

In the second set of experiments, we adjust the global sensitivity $\triangle f$ from 2 to 5, 18 and 591 and show the results with varying k of $MR_k(A)$ and $MR_k(H)$ in Figure 6 and Figure 7 respectively. The selected sensitivities are the first quartile, median, third quartile and max of weight w in its distribution, shown in Table II and Figure 3, which can protect 1/4, 1/2, 3/4 and all edges in the bipartite graph respectively. In this part, the objective is to evaluate the algorithm performance with varying sensitivity and fixed $\epsilon = 1$. As can be seen, we can achieve about 80%, 60%, 50% and 40% match rate for all values of k to protect 1/4, 1/2, 3/4 and all edges. Surprisingly, our algorithm still achieves 40% match rate for our ultimate privacy goal. However, in most cases, we recommend setting $\triangle f = 1$ to protect the 23.4% weight-1 edges and have match rate higher than 90%. In addition, we find that when $\triangle f$ is large in Figure 6(c), 6(d), 7(c) and 7(d), the down-CC beats zero-CC. A big sensitivity results in a huge injected noise, which makes the perturbed high-weight entries start to be indistinguishable from the perturbed low-weight entries. The zero-CC has little influence to this while the down-CC can again make the perturbed high-weight entries start distinguishable from the perturbed low-weight entries start, thus dominating the entire set. To sum up, for very large sensitivity $\triangle f$, we recommend the down-CC.

Finally, in the last set of experiments, we change the scale of the dataset to evaluate the performance of our algorithm over databases with different size. We first limit the timestamp of trajectory data from 5 years to the first 100 days and show the results of $MR_k(A)$ and $MR_k(H)$ with $\epsilon = 1$ and

$\triangle f = 1$ in Figure 8(a) and Figure 8(b) respectively. The new user-location bipartite graph contains 123 users and 39 locations. As can be seen, both the $MR_k(A)$ and $MR_k(H)$ become worse, compared with Figure 4(a) and 5(a). Then, we limit the user number from 182 to 90 and show the results of $MR_k(A)$ and $MR_k(H)$ with $\epsilon = 1$ and $\triangle f = 1$ in Figure 8(c) and Figure 8(d) respectively. The new user-location bipartite graph contains 78 users and 31 locations. As can be seen, compared with Figure 4(a) and 5(a), the $MR_k(A)$ becomes worse while $MR_k(H)$ becomes slightly better. From the results, we can see that the data volume does impact the accuracy of recommendation. Specifically, the track of individuals for a longer period of time can help to improve the accuracy of recommendation in terms of both users and locations. However, a dataset with more users can only enhance the accuracy for location recommendation, but reduce the accuracy for user recommendation.

VI. Conclusion

In this paper, we propose a differentially private trajectory analysis algorithm for travel recommendation that aims at increasing the accuracy of the recommendation results while protecting the differential privacy of the trajectory data. The proposed approach transforms the raw trajectory dataset into a user-location bipartite graph and injects a carefully calibrated noise to meet the required differential privacy guarantees. We propose three consistency constraint schemes to suppress the noise added in the process which improves the accuracy of the obtained recommendation results. Our extensive experiments on a real trajectory dataset show that our algorithm is efficient, scalable and demonstrates good recommendation accuracy while meeting the required differential privacy guarantees.

Acknowledgment

This work was performed under a partial support by the National Science Foundation under the grant DGE-1438809.

References

[1] Ankerst, Mihael, *et al.* OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record*, Vol. 28. No. 2. ACM, 1999.

[2] Banerjee, Syagnik Sy, and Ruby Roy Dholakia. Mobile advertising: Does location based advertising work?. 2008.

[3] B. Palanisamy and L. Liu. Attack-resilient mix-zones over road networks: architecture and algorithms. *IEEE Transactions on Mobile Computing (TMC 2015)*, 14(3), 495-508.

[4] B. Palanisamy, L. Liu. Mobimix: Protecting location privacy with mix-zones over road networks. in *27th International Conference on Data Engineering (ICDE 2011)*, 494-505.

[5] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing (2003). in *Pervasive Computing*, 46-55.

[6] Cicek, A. Ercument, Mehmet Ercan Nergiz, and Yucel Saygin Ensuring location diversity in privacy-preserving spatio-temporal data publishing. *The VLDB Journal*, 23.4 (2014): 609-625.

[7] Chen, Rui, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. *Proceedings of the 2012 ACM conference on Computer and communications security*, ACM, 2012.

[8] Chen, Rui, et al. Differentially private transit data publication: a case study on the montreal transportation system. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2012.

[9] Cheng, Zhiyuan, *et al.* Exploring millions of footprints in location sharing services. *ICWSM 2011*, (2011): 81-88.

[10] Cho, Eunjoon, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. *Proceedings of the 17th ACM SIGKDD* , ACM, 2011.

[11] Dai, Jian, *et al.* Personalized route recommendation using big trajectory data. *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, IEEE, 2015.

[12] Dwork C, McSherry F, Nissim K, *et al.* Calibrating noise to sensitivity in private data analysis. *Theory of cryptography*, Springer Berlin Heidelberg, 265-284, 2006.

[13] Dwork C, Roth A. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4), 211-407, 2013.

[14] Ester, Martin, *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, Vol. 96. No. 34. 1996.

[15] Forgy, Edward W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21 (1965): 768-769.

[16] B. Gedik, L. Liu. A customizable k-anonymity model for protecting location privacy, 2004.

[17] Giannotti, Fosca, *et al.* Trajectory pattern mining. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2007.

[18] Hay, Michael, *et al.* Accurate estimation of the degree distribution of private networks. *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, IEEE, 2009.

[19] He, Xi, *et al.* Dpt: Differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment*, 8.11 (2015): 1154-1165.

[20] Ho, Shen-Shyang, and Shuhua Ruan. Preserving privacy for interesting location pattern mining from trajectory data. 2013.

[21] Kleinberg, Jon M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46.5 (1999): 604-632.

[22] Klmel, Bernhard, and Spiros Alexakis. Location based advertising. *Mobile Business*, 2002.

[23] Li Chao, and Balaji Palanisamy. ReverseCloak: Protecting Multi-level Location Privacy over Road Networks. in *Proc. of 24th ACM International Conference on Information and Knowledge Management (CIKM)*, 2015.

[24] Li, Nan, and Guanling Chen. Sharing location in online social networks. *IEEE network*, 24.5 (2010).

[25] McSherry, Frank, and Kunal Talwar. Mechanism design via differential privacy. *FOCS'07*, 48th Annual IEEE Symposium on. IEEE, 2007.

[26] McSherry, Frank, and Ilya Mironov. Differentially private recommender systems: building privacy into the net. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[27] M.F. Mokbel, C.Y. Chow, W.G. Aref. The new Casper:query processing for location services without compromising privacy. *VLDB Endowment (2006)*, 763-774.

[28] Schiller, Jochen, and Agns Voisard, eds. Location-based services. Elsevier. *Elsevier*, 2004.

[29] Wang, Yilun, Yu Zheng, and Yexiang Xue. Travel time estimation of a path using sparse trajectories. *Proceedings of the 20th ACM SIGKDD*, ACM, 2014.

[30] Ye, Mao, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. *Proceedings of the 18th SIGSPATIAL*, ACM, 2010.

[31] Z. Xiao, X. Meng, J. Xu. Quality aware privacy protection for location-based services. *Advances in Databases: Concepts, Systems and Applications (2007)*, Springer Berlin Heidelberg, 434-446.

[32] Zhao, Yilin. Vehicle location and navigation systems. 1997.

[33] Zheng, Yu. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6.3 (2015): 29.

[34] Zheng, Yu, *et al.* Mining interesting locations and travel sequences from GPS trajectories. *Proceedings of the 18th international conference on World wide web*, ACM, 2009.

[35] Zheng, Yu, and Xing Xie. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2.1 (2011): 2.

[36] Zheng, Yu, Xing Xie, and Wei-Ying Ma. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.*, 33.2 (2010): 32-39.