Perosonalized Differentially Private Location Collection Method with Adaptive GPS Discretization

Huichuan Liu, Yong Zeng(✉), Jiale Liu, Zhihong Liu, Jianfeng Ma, and Xiaoyan Zhu

Xidian University, Xi'an 710126, Shaanxi, China
hc_liu@stu.xidian.edu.cn, {yzeng,liuzhihong,jfma,
xyzhu}@mail.xidian.edu.cn, liujialehenu@163.com

Abstract. In recent years, with the development of mobile terminals, geographic location has attracted the attention of many researchers because of its convenience in collection and its ability to reflect user profile. To protect user privacy, researchers have adopted local differential privacy in data collection process. However, most existing methods assume that location has already been discretized, which we found, if not done carefully, may introduces huge noise, lowering collected result utility. Thus in this paper, we design a differentially private location division module that could automatically discretize locations according to access density of each region. However, as the size of discretized regions may be large, if directly applying existing local differential privacy based attribute method, the overall utility of collected results may be completely destroyed. Thus, we further improve the optimized binary local hash method, based on personalized differential privacy, to collect user visit frequency of each discretized region. This solution improve the accuracy of the collected results while satisfying the privacy of the user's geographic location. Through experiments on synthetic and real data sets, this paper proves that the proposed method achieves higher accuracy than the best known method under the same privacy budget.

Keywords: Local differential privacy · Geographical location · Privacy security

1 Introduction

With the development of mobile Internet technology, various mobile platforms such as mobile phones, tablets, smart watches and other devices have brought many conveniences and joys to people's lives. Sensors such as Accelerometer, GPS, Gyroscope and Magnetometer could capture information about the user's surroundings and provide a richer and more interesting interface for human-computer interaction. Among them, geographic location sensing has been widely equipped on smart devices. As a form of information that could reflect the user's trajectory and lifestyle, it is widely used by major application service providers in the recommendation system to provide users with personalized advertisement.

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 175–190, 2020.
https://doi.org/10.1007/978-981-33-4922-3_13

However, due to the sensitivity of the geographic location itself, and the fact that background applications may collect user data at any time, the uploaded user trajectory data may reflect the user's sensitive information, such as the user's income, beliefs, daily habits, illness and other information [1]. Users may dislike their private data that could expose their activity being analyzed. Besides that, improper data management may result in the disclosure of user privacy data, thereby causing legal problems.

In order to ensure privacy of user uploaded data in analysis process, many researches have been conducted and most differential privacy based methods for solving privately analysis can mainly be divided into two categories. The first category [2–6] is to disturb the collected data before data sharing and publishing. This type mainly uses differential privacy settings. The other category [7–9] mainly focuses on the data collection process and disturbs the data before users upload their private data. Among them, the former category couldn't provide protection against database intrusions or application service providers' threats to user privacy. In reality, the database interface provided by the server is very likely to have problems. For example, in March 2018, a security breach on Facebook enables third-party application software to download unpublished private photos of users without permission, affecting up to 6.8 million users. It is conceivable that with the expansion of business and the growth of code volume, security vulnerabilities are inevitable. The privacy protection of the second category, which is based local differential privacy model, can also essentially prevent third-party analysts from threatening privacy, and it can also prevent the inappropriate use of user privacy data by the enterprise itself, so it has a stronger privacy protection. In this paper, we follow the second category research line and adopt a variant of local differential privacy as our privacy model.

Most existing attribute collection methods [10–12] assume that the user attributes to be collected are discrete, which means, for GPS data, the continuous GPS signal must be quantified before being applied to an existing collection method. But in fact, due to the non-uniformity of the geographical location itself, completely uniform density quantization without any knowledge of the whole user density distribution, will cause very low signal-to-noise ratio. In addition, in order to provide more fine-grained geographic location collection, the number of quantized geographic location areas is large, so local differential privacy based location collection methods would cause overwhelming noise, completely destroying the utility of the data collection results.

This paper proposes a new geographic location collection method. The method is divided into two modules, each of which takes exclusive user sets as input. The first module is a location division module, which is responsible for sending location-related query requests to users in the corresponding user set. On the premise of localized differential privacy, the location area is divided, in the form of quadtree, to establish a quantitative level of location. The second module is the location collection module. It collected the its users' disturbed location set on the division results of the first module, and estimate the true user location distribution as the final result. The main innovations of our method are as follows:

Adaptive location discretization. Unlike the previous work, the method in this paper does not need to assume that the input geographical location are discrete. We propose a local differential privacy based method that can interactively make queries to users and could adaptively discretize the GPS data according to the user access density of each

region. This module divides the area as finely as possible while ensuring the signal-to-noise ratio of the collected result, which balances the grainess of region and signal-to-noise ratio.

Adoption of personalized differential privacy. In our experiments, we found that the geographic location collection scheme that conforms to local differential privacy introduces a lot of noise and makes the overall utility of the final collection results low. Therefore, we adopt the personalized local differential privacy model and modified existing attribute collection algorithms, achieving collection result with higher utility.

2 Related Work

Since local differential privacy needs to disturb user data before the user uploads the data, a mechanism that conforms to local differential privacy generally runs on the user side. Local differential privacy will disturb each user's data, and the variance of the noise of the aggregate result is proportional to the number of samples. In order to avoid noise overwhelming the real signal results, the data collection method that conforms to local differential privacy will only count the frequency of frequent item sets. In order to reduce the impact of noise on the data collection process, and to optimize the communication overhead and computational efficiency, researchers have conducted a lot of researches on the implementation of data collection mechanisms that conform to local differential privacy. Here we briefly introduce the design of methods that have inspired our work.

In 2014, a statistical method RAPPOR that conforms to local differential privacy is proposed. This method encodes the user's attribute set through the bloom filter and randomly disturbs all bits of the bloom filter. On the basis of local differential privacy, the disturbed bloom filter is uploaded to the data collector. On the collector side, the collector sums the set times of all bits of the bloom filter uploaded by all users, and use the least square method to estimate the frequency of occurrence of each attribute. In 2016, RAPPOR [8] was further improved, no longer need to assume that user attributes belong to a known limited set, so that RAPPOR can count the frequency of frequent occurrences of arbitrary unknown attributes. Their improved method is comprised of two modules. The first module is the same as the original RAPPOR method, using bloom filter results to estimate the frequency of attributes. The second module is used to calculate attribute sets that belong to frequent items. It cuts the string encoding of all attribute names into multiple fixed-length character segments, and uses the expected maximum algorithm to estimate the probability of occurrence of all character segment pairs. The connection of the character combination is stored in a graph. Each character segment corresponds to a node in the graph. When the occurrence probability of the character segment pair exceeds a certain threshold, the two nodes are connected. Since all character segments of each frequent element must also be frequent, fully connected subgraphs of a specific length in the graph then correspond to frequent item sets. Finally, the first module could estimate the frequency of items in the frequent attribute set.

In 2015, a local differential privacy based method—binary local hashing method [9] is proposed, which is completely different from RAPPOR and based on the principle of compressed sensing theory. This method randomly generates a ± 1 vector with a fixed length of m for each attribute of the user attribute set, and uses this vector as the binary

representation of the attribute. Since the expectation of two different vector dot product is 0, and the dot product of the same vector is m, the method randomizes the input vector while keeping the expectation of each value in the vector unchanged, and then sums all the uploaded user vector And by dot multiplying the sum vector with any representation vector of an attribute, we can get an unbiased estimate of the frequency of the attribute.

In 2017, researchers [10] summarized methods such as random response, RAPPOR, and binary local hash method, and proposed an error analysis framework for automatically optimizing random response probability parameters. But these two methods can only estimate the attribute frequency of a known and limited set, and cannot deal with the unknown or unlimited number of attribute sets.

In 2018, a frequent item set discovery framework, called PrivTrie [11], based on prefix trees was proposed. They believed that the reason RAPPOR improved method [8] has excessive computational overhead and sensitivity to noise interference, is that graph is not suitable for storing the relationship between character segments. Therefore, they propose to use the prefix tree structure to describe the coupling relationship between character segments. In addition, their paper proposes a method that can make the same query to users of different branches of the prefix tree at the same time and still ensure differential privacy security. It can make more query requests to a limited set of users, thereby improving the accuracy of estimated attribute frequency.

In addition, in 2016, researchers [12] first applied the concept of local differential privacy to the field of geographic location collection research, and its scheme adopted a binary local hash method for location data collection. As the direct use of localized differential privacy would result in low signal-to-noise ratio, researchers proposed the concept of personalized local differential privacy, which is different from local differential privacy in that the new concept only requires that the probability distribution on the user-specified attributes are approximate rather than the whole attribute set. In addition, the scheme assumes that all geographic locations have been quantified as discrete areas. This scheme is a geographic location collection scheme based on the concept of local differential privacy derivation, which is known to have high data utility. Therefore, we use this work as a comparison to verify the utility of the data collection results of our work, and in paper, we refer to it as PSDA.

3 System Overview

In order to guarantee the user's data privacy during data collection, our method adopts the local differential privacy [13] as the privacy protection model. The principle of localized differential privacy is to randomly disturb the user's data before uploading it. After the collector collects a certain number of users' disturbed data, the collector then estimates the distribution of real users. There are mainly two problems in the scheme design:

(1) Suppose the size of the user set to be collected is N, the noise magnitude added by local differential privacy is orders of, and the noise added by centralized differential privacy is generally a constant. Therefore, compared to centralized differential privacy based method, data collection methods that conform to local differential privacy need to be designed to ensure that the attribute whose frequency is to be estimated must be frequent. As a result, before estimating the frequency of geographic

location access, our method first needs to calculate the frequent item sets, and the process of calculating frequent item sets also needs to satisfy the local differential privacy.

(2) There are huge differences in user attitudes towards privacy. On the one hand, capturing this difference meets the personalized privacy requirement; on the other hand, it adaptively reduces the magnitude of added noise. Therefore, in our method, it is necessary to adopt a privacy concept that can reflect the privacy protection needs of different users according to the characteristics of geographic location data, so as to improve the availability of data.

In response to the problem in (1), our method first divides the user set into two disjoint set, the first set is used to calculate frequent itemsets of geographic location. As original GPS data is continuous, and there is a certain unevenness in the distribution, so first of all, it is necessary to quantify the continuous geographic location into discrete areas, and adjust the quantization granularity of different areas according to each area's user access frequency. More fine-grained quantification need to be performed on the area with higher user access frequency; the second user set is used to collect the disturbed frequency of user visits in each geographical area, and estimate the true geographic distribution of users.

In response to the problem in (2), our method adopts the concept of personalized local differential privacy, using the tree structure to organize the calculated frequent area sets, and allows users to personalize their privacy requirement, which can greatly improve the accuracy of the estimation result.

In terms of system architecture, this chapter is divided into a geographic location division module and a geographic location collection module. The relationship between these two modules is shown in Fig. 1.

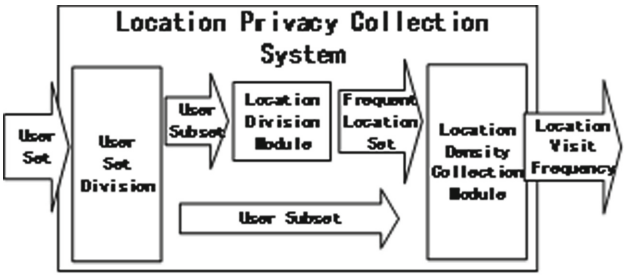

Fig. 1. Architecture of our location privacy collection method

4 Module Design

4.1 Location Division Module

This section introduces the design of the geographical location division module. The map division method used in our method uses the quadtree division method adopted by

previous researchers [4, 14, 15], and the division method is shown in Fig. 2. The largest square represents the entire map. By recursively dividing the map with a quadtree, multiple quantization areas are obtained. In the location density collection module, the results of the map division will be used to quantify the continuous geographic location of the user, and then the data collection method for discrete attributes can be adopted.

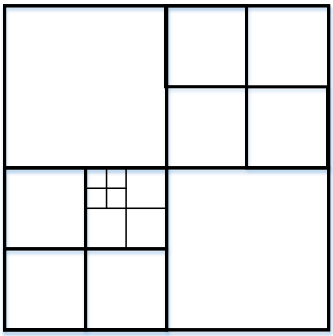

Fig. 2. Schematic diagram of geographical location division method based on quadtree

Because the local differential privacy based method can only collect the frequency of frequent itemsets, it is necessary to ensure that the frequency of user access in each sub-region finally obtained is higher than a certain threshold to reduce the impact of noise. Therefore, the problems solved in this section are summarized as follows: Under the limitation of local differential privacy, the map is reasonably segmented using a limited set of user data, so that the number of users in each sub-region is higher than a certain threshold and as close as possible to the threshold.

Before introducing the scheme, first we introduce the data structure used in the algorithm. The TreeNode structure is used to store tree node information, where Cell represents the area corresponding to the tree node, children represents the child nodes of the tree node, number represents the number of users who select the node as a geographic protection area. As our Location Density Module exploits personalized differential privacy, user_index is used to store the set of users who designate this TreeNode as their privacy protection area. Count is used to save the weights of the four child nodes of the node, and parent represents the parent node of the node.

```
struct TreeNode {
    Cell c
    TreeNode* [] children
    int number
    int[] users_index
    CellCount count
    TreeNode* parent
}
```

The algorithm for segmenting the map is shown in Algorithm 1. It draws on the design of Privtrie [11], which was designed for calculating frequent discrete attribute,

and we modified the algorithm process to make it adaptively calculating discretization level of continuous GPS data.

Algorithm 1：DivideTreeNode(**rt**, ***D***, ε, batch_size)

Algorithm 1 : DivideTreeNode(rt, D, ε, batch_size)

Input: the tree root node rt，user subset D, local differential privacy budget ε, batch_size
Output: map division tree rooted with rt
1: F=∅
2: CS= set of four sub-areas of rt
3: count=0
4: while D!=∅:
5: choose batch_size users from D, represented as G
6: delete G elements from D
7: $F = F \cup G$
8: for every user u in G do
9: count+=IsInCell(r.Cell,u.Location, ε)
10: if evaluate(count, F.size) > threshold then
11: for every cell cnode in CS do
12: root.Children.append(DivideTreeNode(cnode,D, ε))
13: break
14: return root

Lines 1–3 are the initialization of parameters. Lines 5–7 indicate that batch_size users are randomly sampled from the set of users assigned to the current node. In the 9–10 line, IsInCell is used to simulate the process of making a query request to the sampled user, and the implementation of the IsInCell function is given in Algorithm 2. Line 10 simulates the process that the data collector uses the evaluate function to remove noise and determine whether the frequency of user access to the node is a certain threshold. We choose max(, 0.001|D|) as threshold, among which, means the variance of evaluate result. Since the evaluate result follows normal distribution, its variance could be calculated easily. If evaluate result is greater than the threshold, then in line 12, corresponding areas to the child nodes are further recursively divided; if it is less, return to line 5, adds more users, and repeat the process of lines 7–13 until D is the empty set.

Algorithm 2 : IsInCell(Cell c, Location *l*, double ε)

Input: Location Area $\boldsymbol{c}$, user location $\boldsymbol{l}$, local differential privacy budget ε

Output: 0 or 1

1: sample from the distribution, and get the result **b**

$$\Pr[output = 1] = \begin{cases} \mathrm{p} = \dfrac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}} + 1}, & \text{if } l \in \mathrm{c} \\ \mathrm{q} = \dfrac{1}{e^{\frac{\varepsilon}{2}} + 1}, & \text{if } l \notin \mathrm{c} \end{cases}$$

2: **return b**

The information collection process given in Algorithm 2 exploits the randomized response mechanism, which has been proved to satisfy local differential privacy [7]. We simply show the proof of local differential privacy here.

There are four situations here, which are:

$$\frac{\Pr[output(l) = 1]}{\Pr[output(l') = 1]} \begin{cases} 1, & if\ l \in c\ \ and\ \ l' \in c' \\ e^{\frac{\varepsilon}{2}}, & if\ l \in c\ \ and\ \ l' \notin c' \\ e^{-\frac{\varepsilon}{2}}, & if\ l \notin c\ \ and\ \ l' \in c' \\ 1, & f\ l \notin c\ \ and\ \ l' \notin c' \end{cases}$$

Thus we can easily see that each IsInCell algorithm satisfies 0.5ε-local differential privacy. Furthermore, in algorithm 1, every user sent bit vector contains at most one 1-bit, and all others 0-bit, so algorithm 1 satisfies ε-local differential privacy. On the server side, The implementation of evaluate function is

$$\text{evaluate}(count, n) = \frac{count - n \cdot q}{p - q}$$

Finally, the algorithm given in Algorithm 1 can get the quadtree corresponding to the map area division, and the leaf nodes in the tree have a one-to-one correspondence with each quantized area.

4.2 Personalized Location Privacy

Since the map has been recursively divided into multiple areas, and the areas are in a tree-like, hierarchical relationship, our method allows users to specify their privacy protection areas. Note that user-specified privacy protection areas are considered not to be private data and it could be obtained directly by the server. Assume that the geographical division module divides the map as shown in Fig. 3.

In order to reduce the error caused by quantization, the user's location data will only be quantized to any element in the set of leaf nodes, in our example, {2, 3, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17} numbered nodes corresponding areas. Assume that a user is quantified to area 11, he is allowed to choose his privacy protection level in 4 levels of differential privacy protection. The numbers of the privacy protection areas

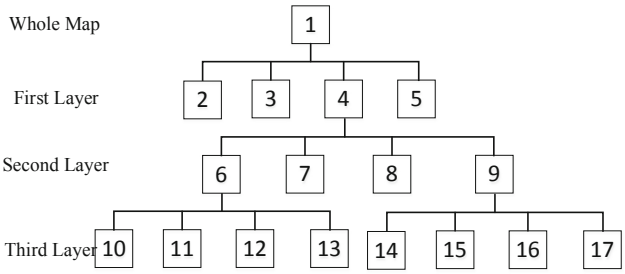

Fig. 3. Example of map division result

corresponding to the four levels are 11, 6, 4, 1, respectively, that is, a user can choose any ancestor node of his location node as his privacy protection area.

For example, when the user selects area 4 as its privacy protection area, according to the definition of personalized local differential privacy, we needs to ensure that on all leaf nodes under area 4, including {7, 8, 10, 11, 12, 13, 14, 15, 16, 17}, local differential privacy needs to be satisfied. The advantage of personalized differential privacy is that the user's data collection process only needs to ensure the differential property in the privacy protection area specified by the user, which doesn't need to consider the probability distribution on the locations outside the privacy protection area, in this example, {2, 3, 5} area.

4.3 Location Density Collection Module

Since the privacy protection areas designated by users are different, firstly, users are divided according to their designated differential privacy protection areas, and a data collection method is called individually for each user subset. This section introduces the design of location collection module.

This module uses the improved method of the binary local hash method proposed by researchers [9, 10, 12] and in order to improve the utility of collection results, this module exploit personalized differential privacy model. Assuming that each user to be collected has designated his privacy protection area, suppose the geographic location of a user u is u.l and privacy protection area is u.L. The collection process is shown in Algorithm 3.

Algorithm 3 : Location Collection Process

Input: Collector designated parameter **m** and **g**, quantization location set **D**, user set **U**

Output: All quantization locations' estimated frequency

1: d=|D|

2: collector generate a $m \times d$ sized matrix M, each item in matrix is randomly chosen from {1,2,3,…,g}, and each column corresponds to a location

3: collector initializes a zero matrix $\boldsymbol{z}$, sized $m \times g$

4: collector initializes a $\boldsymbol{d}$ sized zero vector f , to save all locations' estimated frequency

5: **for** every user u in U **do**

6: collector randomly generates a number j from {1,2,3,…,m}

7: collector sends j-th row of M to user u

8: user u computes r=LocalRandomize(u.l, u.L, $M_{j,.}$), and sends r to collector

9: collector computes z[j][r]= z[j][r]+1

10: **for** every location l in D **do**

11: f_l = EstimzteFrequency($M_{.,l}$, z)

12: **return** f

In the first step, the collector generates a random matrix. It should be noted that this matrix does not need to be kept secret. It can be obtained by sharing the key between the data collector and the user and generated from a random stream, which reduces communication overhead of sending the j-th row of matrix M in the row 7. The matrix z in the second step is used to save the user's aggregate statistical results. Steps 6 to 9 are basically the same as the binary local hash mechanism [9, 12]. The difference is that the return value r of LocalRandomize in our method is no longer, but a value in {1, 2, 3,…, g}. Corresponding to that, in step 7, our method takes r as an index, add 1 to the r-th column of the j-th row of the aggregate result z.

The implementation of LocalRandomize and EstimateFrequency are shown in Algorithm 4 and Algorithm 5 respectively.

Algorithm 4 : LocalRandomize

Input: user location l, user designated privacy protection area L, j-th row R of matrix M, location quantization set D

Output: disturbed user location index from {1,2,3,…,g}

1: e=R[l]

2**:** user randomizes z following the distribution,and get the result v

$$\Pr[v=z]=\begin{cases}\dfrac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, & z=e\\ \dfrac{1}{e^{\varepsilon}+g-1}, & z\neq e\end{cases}$$

3: return v

Since for every user, randomized response mechanism is invoked, and the proof is the same as Algorithm 2.

Algorithm 5 : EstimateFrequency

Input: the location encoding *c*, aggregate matrix *z*, user number *N* that designate the location as their privacy protection area

Output: the location's estimated visit frequency

1: $p = \frac{e^{\varepsilon}}{e^{\varepsilon} + g - 1}$, $q = \frac{1}{g}$

2: count=0

3: **for** i=0;i<c.size;i++:

4: count+=z[i][c[i]]

5: **return** $\frac{\text{count} - N \cdot q}{p - q}$

The basic idea of the frequency estimation process in Algorithm 5 is the same as the randomize response mechanism. The difference is that the user aggregation result here is a matrix instead of a vector. Since each column of the random matrix generated by the collector can be regarded as a encoding of a location area, each element is randomly chosen from {1, 2,…, g}. So when estimating the frequency, only the same indexed aggregation value as the target encoding needs to be count. So in line 4, we first take the value of the column c[i], and use c[i] as index to take the corresponding aggregation frequency value in z. After eliminating the bias in line 5, we can get the estimated frequency of the target attribute.

It should be noted that in our method, Location Collection Process needs to be invoked for every set of users that designate the same privacy protection area. But this wouldn't be a efficiency bottleneck, because every user still only needs to participates in one collection. After all users location data has been collected, add the estimated results in each collection and then the total corresponds to the location's real visit frequency.

5 Experimental Validation

5.1 Experiment Setup

In our experiment, we use Brinkhoff [16] and the Portugal taxi trajectory dataset as the users' location data set.

Brinkhoff is trajectory generator that has been widely adopted as benchmark [17, 18]. It takes the map in the real world as the input, and establishes a trajectory generator, which can generate trajectory data sets of any size according to the characteristics specified by the user. In the experiment, the German Oldenberg is used as the map, and a total of 1,000,000 trajectory data are generated as the trajectory data set of the experiment.

Protugal taxi trajectory dataset was drawn from the ECML/PKDD 2015, and we randomly chose 1,000,000 trajectory data from original 1,673,686 trajectories.

Since the goal of our method is to collect the users' geographic location data as accurately as possible, we compare the collected user location distribution with real user

data distribution to evaluate the geographic location collection method proposed in this paper. The evaluation indicators adopted in this article are the same as PSDA work and are as follows:

(1) KL distance. We calculate the distribution of the original data set on the geographical location division results, and then calculate the distribution of the collected location access probability distribution. In order to measure the distance between the two distributions, KL divergence is used as the evaluation metric.
(2) The accuracy of top-K areas with the highest density. We calculate the K locations with the highest frequency of density in the original data set, then calculate the K locations with the highest frequency of access in the estimation result, and calculate the accuracy of the estimation result.

5.2 Experiment Results

The performance of this scheme and PSDA scheme on the KL distance evaluation index on different data sets is shown in Fig. 4.

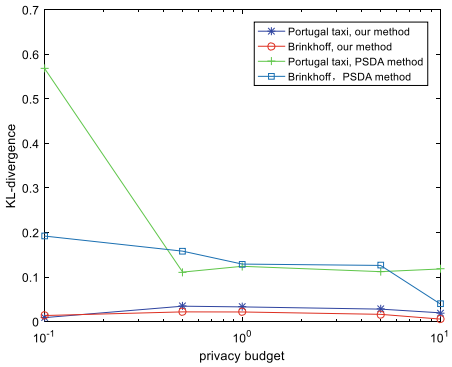

Fig. 4. KL divergence between original data set and collected results.

It can also be verified that under the same local differential privacy budget, out method could achieve lower KL divergence and higher top-K accuracy than PSDA method. In addition, it should be noted that in Fig. 5, when differential privacy budget, the geographical location is divided and the size of the division location set is less than $K = 100$, so the accuracy rate of the K regions with the highest access density is 100%. It can be seen that the accuracy of the experimental results in Fig. 6 does not increase with the increase in differential privacy budget. According to the analysis, there are two reasons for this phenomenon:

(1) The top-K indicator only cares about the frequency of the area with a larger frequency, and the collection result of the area with a higher frequency itself has higher signal-to-noise and is less affected by noise.

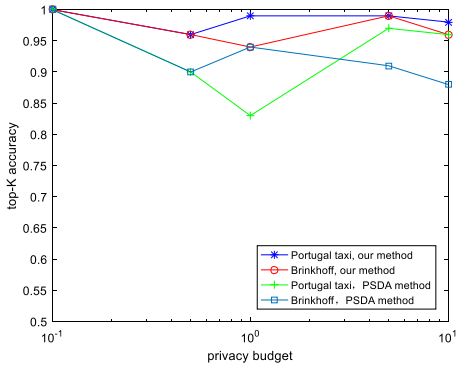

Fig. 5. Top-K accuracy of collected location results. K = 100.

(2) The location collection module also uses the result of the geographic location division module. As the differential privacy overhead increases, the variance of the noise also decreases, so the threshold of the leaf nodes in the division process also decreases. As a result, the leaf nodes are further divided, making the location set larger. In the experiments of the Portuguese taxi data set, the change of the size of the divided location set with the differential privacy budget is shown in Fig. 6.

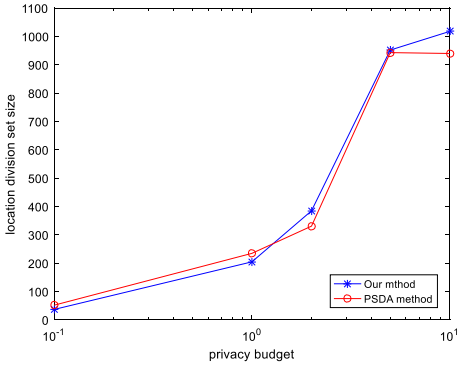

Fig. 6. Change of location division result size with differential privacy budget.

It can be seen from Fig. 6 that the size change of the location set obtained by this scheme and PSDA scheme is basically the same. When the differential privacy budget is low, the number of geographically divided areas is also low, which can compensate for the increase in noise, even if signal-to-noise ratio of each collected location density reduces. It should be noted that in the experiments corresponding to Fig. 4 and Fig. 5, PSDA scheme also has this effect, but because the noise amplitude of their method grows too fast, the change in the size of the location set is not fast enough to compensate for the increase of noise. Therefore, its accuracy shows a significant downward trend, which

also proves that the method proposed in our paper could achieve better collected results utility.

In order to further illustrate the influence of the original location set size and location division results size on the accuracy of the final collection results, experiments are carried out on different sizes of original datasets. The experimental results are shown in Table 1. Note that original data size's unit is million.

Table 1. Change of evaluation with dataset size (batch_size = 1000)

Original dataset size/million	0.2	0.4	0.6	0.8	1
KL divergence	0.0208	0.0639	0.0868	0.136	0.222
Top-K	0.93	0.96	0.97	0.98	0.97
Location division set size	133	538	1546	2653	5239

As can be seen from the results in Table 1, as the scale of the data set increases, the number of regions obtained by dividing the map by the location division module has increased significantly, and the relative proportion of the growth rate is far faster than the growth rate of the scale of the data set, resulting in that the signal-to-noise ratio averaged in each area is reduced. With the increase in the size of the data set, the KL divergence indicator showed a significant increase, but the top-k accuracy rate remained almost unchanged. The reason for this result is that the KL divergence represents the accuracy of the collection results of all regions, and the top-K accuracy represents the accuracy of the collection results of high-frequency sub-regions, so the latter itself is less affected by noise. In summary, it can be concluded that if the goal of collecting data only considers high-frequency attributes, the system can achieve high-precision collection results without special settings; if the data to be collected needs to consider the frequency of all attributes, we need to adjust the size of batch_size according to the size of the user set to be collected, so that the number of regions divided by the geographic location division module increases in proportion to the size of the data set, so as to ensure the relative stability of the signal-to-noise ratio.

6 Conclusion

In this paper, we explain the necessity of privately collecting user locations from the perspective of users and service providers, and then divides the private collection method into a location division module and a location density collection module, and explains functions and principles of the two modules. Finally, the utility and accuracy of the method are tested using the Brinkhoff trajectory generator and the Portugal taxi trajectory data set. The results shows that out method could achieve better utility than the best method known so far.

References

1. Fawaz, K., Feng, H., Shin, K.G.: Anatomization and protection of mobile apps' location privacy threats. In: 24th USENIX Security Symposium, pp. 753–768. USENIX (2015)
2. Chen, R., Fung, B., Desai, B.C.: Differentially private trajectory data publication. arXiv preprint arXiv:1112.2020 (2011)
3. Chen, R., Acs, G., Castelluccia, C.: Differentially private sequential data publication via variable-length n-grams. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 638–649. ACM (2012)
4. Zhang, J., Xiao, X., Xie, X.: PrivTree: a differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 International Conference on Management of Data, pp. 155–170. ACM (2016)
5. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C.M., Srivastava, D.: DPT: differentially private trajectory synthesis using hierarchical reference systems. In: Proceedings of the VLDB Endowment, pp. 1154–1165. Springer (2015)
6. Gursoy, M.E., Liu, L., Truex, S., Yu, L., Wei, W.: Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 196–211. ACM (2018)
7. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067. ACM (2014)
8. Fanti, G., Pihur, V., Erlingsson, Ú.: Building a rappor with the unknown: privacy-preserving learning of associations and data dictionaries. Proc. Priv. Enhanc. Technol. **2016**(3), 41–61 (2016)
9. Bassily, R., Smith, A.: Local, private, efficient protocols for succinct histograms. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 127–135. ACM (2015)
10. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: 26th USENIX Security Symposium, pp. 729–745. USENIX (2017)
11. Wang, N., et al.: PrivTrie: effective frequent term discovery under local differential privacy. In: 2018 IEEE 34th International Conference on Data Engineering, pp. 821–832. IEEE (2018)
12. Chen, R., Li, H., Qin, A.K., Kasiviswanathan, S.P., Jin, H.: Private spatial data aggregation in the local setting. In: 2016 IEEE 32nd International Conference on Data Engineering, pp. 289–300. IEEE (2016)
13. Warner, S.L.: Randomized response: a survey technique for eliminating evasive answer bias. J. Am. Stat. Assoc. **60**(309), 63–69 (1965)
14. Samet, H.: The quadtree and related hierarchical data structures. ACM Comput. Surv. **16**(2), 187–260 (1984)
15. Ho, S.S., Ruan, S.: Differential privacy for location pattern mining. In: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, pp. 17–24. ACM (2011)
16. Brinkhoff, T.: Generating network-based moving objects. In: Proceedings of the 12th International Conference on Scientific and Statistical Database Management, pp. 253–255. IEEE (2000)
17. Agarwal, P.K., Fox, K., Munagala, K., Nath, A., Pan, J., Taylor, E.: Subtrajectory clustering: models and algorithms. In: Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 75–87, May 2018
18. Orakzai, F., Calders, T., Pedersen, T.B.: k/2-hop: fast mining of convoy patterns with effective pruning. Proc. VLDB Endow. **12**(9), 948–960 (2019)