A More Secure Spatial Decompositions Algorithm via Indefeasible Laplace Noise in Differential Privacy

Xiaocui Li[1], Yangtao Wang[1], Xinyu Zhang[2], Ke Zhou[1(✉)], and Chunhua Li[1]

[1] Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, China
{LXC,ytwbruce,k.zhou,li.chunhua}@hust.edu.cn
[2] School of Computer, Wuhan University, Wuhan, China
zhangxinyu@whu.edu.cn

Abstract. Spatial decompositions are often used in the statistics of location information. For security, current works split the whole domain into sub-domains recursively to generate a hierarchical private tree and add Laplace noise to each node's points count, as called differentially private spatial decompositions. However Laplace distribution is symmetric about the origin, the mean of a large number of queries may cancel the Laplace noise. In private tree, the point count of intermediate nodes may be real since the summation of all its descendants may cancel the Laplace noise and reveal privacy. Moreover, existing algorithms add noises to all nodes of the private tree which leads to higher noise cost, and the maximum depth h of the tree is not intuitive for users. To address these problems, we propose a more secure algorithm which avoids canceling Laplace noise. That splits the domains depending on its real point count, and only adds indefeasible Laplace noise to leaves. The ith randomly selected leaf of one intermediate node is added noise by $\frac{(\beta-i+1)+1+\beta}{(\beta-i+1)+\beta}Lap(\lambda)$. We also replace h with a more intuitive split unit u. The experiment results show that our algorithm performs better both on synthetic and real datasets with higher security and data utility, and the noise cost is highly decreased.

Keywords: Indefeasible Laplace noise · Low noise cost
Differential privacy · Spatial decompositions

1 Introduction

In the era of Big Data, A variety of data mining algorithms and prediction strategies [1–3] were developed to analyze users' behavior habits, which brings heavily privacy threats to users. In many investigations, the statistic of location information is needed for different academic research, such as the distribution of endangered species, the distribution of hotel occupancy in a certain city, the trip distribution of occupied taxis, and so on. That can help biological scientists,

G. Gan et al. (Eds.): ADMA 2018, LNAI 11323, pp. 211–223, 2018.
https://doi.org/10.1007/978-3-030-05090-0_19

government, business decision-makers etc. make the correct and effective decision through some recommendation algorithms [4,5].

The existing approaches obtain statistic of location information via spatial decomposition. The process of spatial decompositions is that given a data set D of tuples in domain Ω, recursively decompose Ω into a set of sub-domains if the point count of current domain is larger than the given threshold θ. When the recursive termination condition is reached, a Hierarchical spatial decompositions tree is generated. Through retrieving this tree, the location information in a certain region can be obtained. Actually, we only hope users get the statistic information of database but not individual as individual's location may leak one's privacy.

However the spatial decompositions tree may reveal individuals' privacy, simply, some leaves of this tree may only contain one tuple. Some purpose oriented adversary can filch most individuals' privacy by various technical means such as data mining algorithms via retrieving the spatial decompositions tree. When publishing database, the data owner needs to perturb the information to achieve preserving privacy [6], which is known as preserving privacy data publishing [7,8]. It is hoped that users get the statistical information of the database as a whole but not individuals' information when users query the database.

Dwork et al. first gave the notion of differential privacy [9]: deleting an element from a statistical database should not substantially increase the risk of the record owner's privacy. Consequently, Dwork proposed a theoretical framework [10–12] called ε-differential privacy, and proved that the *Laplace mechanism* [13] can achieve differential privacy for numerical queries. In the last decade, differential privacy was applied in many algorithms, such as histogram-based data publishing [14], batch query [15], decision tree [16] and so on. In 2016, Apple Inc. was planning to adopt the differential privacy to preserve the users' privacies, and it is the first time that the differential privacy algorithm has been applied in practical applications.

In 2012, Cormode first applied the differential privacy to spatial decompositions quadtree [17], which took the approach of adding Laplace noise to each node of the spatial decompositions tree and publishing the noisy private tree to achieve the purpose of privacy preserving. Thereafter more excellent algorithms [19–21] emerged for spatial decompositions based on differential privacy, of which the PrivTree proposed by Zhang 2016 was especially noteworthy. In the algorithms proposed by Granham and Zhang, a private tree was generated through spatial decompositions. Each split of a node will generate 2^d children, where d is the dimension of the dataset, which finally outputs a 2^d-tree. For preserving privacy, each node of the 2^d-tree was added by Laplace noise. Users can obtain the answers to queries of the point count of a given area by retrieving the private tree.

1.1 Motivation and Contributions

In 2008, Dwork indicated that preserving privacy by adding Laplace noise into the true answer is delicate [10], as Laplace noise is symmetric about of the origin

and the same question is asked many times, the responses may be averaged, canceling out the noise. In spatial decompositions private tree, each intermediate node's point count equals the summation of all its descendants' point counts, that may reveal the intermediate node's privacy for the summation of all its descendants Laplace noise may be canceled.

Another problem is that the private tree with every node added Laplace noise makes domain splitting imprecise for the noise snowballs from root to leaf. Moreover, the depth of the private tree h, also the maximum depth of recursion should be predefined when splitting Ω. However, the choice of h can be challenging, since a smaller h will cause coarse splitting of Ω, while a larger h will lead to the depth of the private tree too high, as a result the noise added to the private tree will increase.

To address the limitations of above, we propose a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy. We first propose a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy, which only adds noise to leaves but not to intermediate nodes. We use a more intuitive threshold u, the minimal split unit to limit the maximum depth of the private tree. We add indefeasible noise to the ith randomly selected leaf child of each intermediate node of the private tree through multiplying Laplace noise by coefficient of $\frac{(\beta-i+1)+\beta+1}{(\beta-i+1)+\beta}$. It satisfies ε-differential privacy and the answer to a query will be not real because the added noise cannot be canceled. It is also proved that the noise cost is lower than existing algorithms. Finally, we conduct extensive experiments to demonstrate the performance of our algorithm.

2 Preliminaries

In this section, we give the background of differential privacy and introduce the problem of spatial decomposition.

2.1 Differential Privacy

Definition 1 *(Neighboring Datasets).* *The dataset D and D' are neighbors if D and D' differ in at most one element.*

Definition 2 *(ε-Differential Privacy).* *The randomized function* F *satisfies ε-differential privacy if, for any two neighboring datasets D and D' and for all output $S \in Range(\mathrm{F})$,*

$$\frac{Pr\left[F\left(D\right) \rightarrow S\right]}{Pr\left[F\left(D'\right) \rightarrow S\right]} \leq e^{\varepsilon} \tag{1}$$

where $Pr[\cdot]$ denotes the probability of an event.

Definition 3 *(Sensitivity).* *Let f be a function that maps a dataset* D *to a vector of real numbers. The global sensitivity of f is defined as:*

$$S\left(f\right) = \max_{D,D'} \frac{\left\|f\left(D\right) - f\left(D'\right)\right\|_1}{dis_{Ham(D,D')}} \tag{2}$$

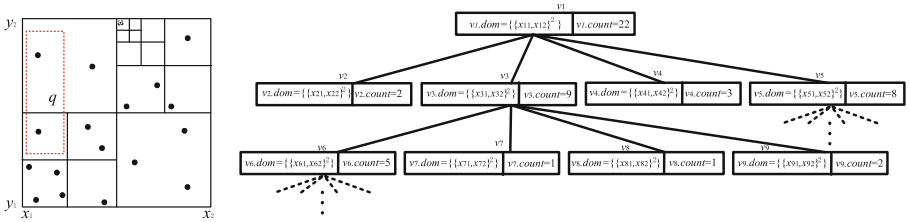
where $\|\cdot\|_1$ *denotes the L1 norm, and the* $dis_{Ham(\cdot)}$ *denotes the hamming distance.*

The *Laplace Mechanism* is the fundamental algorithm used in numerical function through adding *i.i.d.* noise into each output. The noise obeys *Laplace distribution* with the following probability density function:

$$Pr\,(x) = \frac{1}{2\lambda} e^{-\frac{|x|}{\lambda}} \tag{3}$$

2.2 Spatial Decompositions

Spatial decomposition was usually classified into data-dependent decomposition and data-independent decomposition. The data-dependent decomposition indicates that the partition of space is dependent on the input data. The most common data-dependent decomposition are KD-tree [22] and R-tree [23]. The data-independent decomposition indicates that the partition of the spatial nodes is independent of the input data. It is computed by splitting the space into two average parts on each coordinate. The best known is quadtree in two dimensions and 2^d-tree [24–26] in higher dimensions. In our algorithm, we use data-independent decomposition as the spatial decomposition.



(a) dataset D (b) The quadtree of D

Fig. 1. Spatial decomposition quadtree of a dataset D in 2-dimensional space Ω

Let D be a dataset consisting of data points in a d-dimensional space Ω. A spatial decomposition of D generates a 2^d-tree, which decomposes Ω into its sub-domains, along with partitioning of its data points into the leaves of the decomposition tree. In Fig. 1, (a) gives a *2*- dimensional dataset D containing 22 points, it recursively divides into four regions until the number of its data points is less than a given threshold. (b) is the generation of a 2^d-tree (quadtree with d=2) of D. The root of the quadtree v_1 corresponds to the region that covers the entire domain Ω, it has four child nodes v_2, v_3, v_4 and v_5 respectively corresponding to the four sub-domains of Ω. Each node of the quadtree consists of its corresponding domain and the number of data points contained in its region.

The 2^d-tree is widely used for querying the spatial region through up-down traversing the quadtree from the root.

3 Related Work

In this section we first review private spatial decompositions algorithms, then discuss their disadvantages and the dilemmas.

Cormode et al. first proposed the differentially private spatial decompositions through quadtree in 2012, as called DPSD. It takes the solution of publishing the perturbed version of the spatial decomposition quadtree, which adds Laplace noise to each node of quadtree to achieve ε-differential privacy. Algorithm 1 presents the main approach of DPSD. The input contains two thresholds h and θ, which respectively represent the maximum recursion depth and the decision value of split as discussed in Sect. 2.2, and $\overline{v_i.count}$ is the noisy version of the $v_i.count$.

Algorithm 1 DPSD(D,λ,θ,h)

```
initialize a quadtree T with a root node v_1;
set v_1.dom = Ω, and mark v_1 as visited ;
while there exists an unvisited node v_i do
    mark v_i as visited ;
    compute the number v_i.count of points in D that are contained in v_i.dom;
    compute the noisy version of v_i.count, v_i.count-bar = v_i.count + Lap(λ);
    if v_i.count-bar > θ and depth(v_i) < h − 1 then
        split v_i, and add its children to T;
        mark the children of v_i as unvisited;
return T;
```

Assume that D and D' are neighboring datasets. The quadtrees of D and D' have at most h nodes with different point count, and these h nodes form a path from the root to one leaf. That indicates that the sensitivity of quadtree is h, so the Algorithm 1 satisfies ε-differential privacy if $\lambda \geq \frac{h}{\varepsilon}$. In Algorithm 1 there is n Laplace noise added to the private tree, the noise cost of the private tree is $\sum_{i=1}^{n} |Lap(\lambda)|$. The main limitation of Algorithm 1 is that the privacy cost depends on the recursive depth h.

Zhang et al. in 2016 proposed PrivTree, in which biased count of v_i was given, $b_i.count = \max\{\theta - \delta, v_i.count - depth\,(v_i) \cdot \delta\}$. In PrivTree input h was replaced by a new parameter θ and the splitting only depends on the noisy version of $b_i.count$.

Both of these two algorithms consider the split problem by the noisy version of each node of the private tree, which will lead to higher privacy cost of the private tree. In PrivTree, the choice of δ in PrivTree is a difficulty as $b_i.count$ at least equals $\theta - \delta$. If δ is very small, $b_i.count$ plus Lap(λ) may easily be larger than δ, which results in unnecessary splitting.

4 A More Secure Indefeasible Laplace Noise Spatial Decompositions Algorithm

In this section, we propose a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy, the InLN_DPSD which can avoid the Laplace noise to be canceled and has lower noise cost.

4.1 Low Noise Cost Private Tree for Spatial Decompositions

According to the above-mentioned analysis, we propose a low noise cost private tree based differential privacy algorithm.

We modified Algorithm 1 with a new parameter minimal split unit u instead of h, which is more intuitive for split termination, and whether a node should be split depends on its real point count $v_i.count$ instead of $\overline{v_i.count}$. We replace the $\overline{v_i.count} > \theta$ and $depth(v_i) < h-1$ by $v_i.count > \theta$ and $\frac{\Omega}{2^{depth(v_i)*d}} > u$ of Algorithm 1 in line 7, which indicates that if $v_i.dom$, $\frac{\Omega}{2^{depth(v_i)*d}}$ is smaller than u, even if $v_i.count > \theta$, stop splitting v_i, where d is the dimensionality of Ω and $depth(v_i)$ denotes the depth of v_i in the private tree. The private tree T generated by the modified algorithm has all leaves with Laplace noise, while all intermediate nodes do not. For privacy preserving data publishing, we also get the point count of intermediate nodes by summing up the count of the leaves under it.

In order to compare with Algorithm 1, we make the same assumption that D and D' are neighboring datasets differing only by one element. There are at most $\max(depth(v_i))$ nodes with different point count in quadtrees of D and D', and these nodes must form a path from the root to the one leaf. In particular, we let $\max(depth(v_i))$ equal h. Then,

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} = \sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + Lap(\lambda) > \theta]}{Pr[v_i'.count + Lap(\lambda) > \theta]} + ln\frac{Pr[v_h.count + Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + Lap(\lambda) = \overline{v_h.count}]}$$

Although the first h-1 intermediate nodes do not add Laplace noise, the D and D' can also approximately output a same T satisfying ε-differential privacy if $\lambda > \frac{h}{\varepsilon}$. Every node v_i for any $i \in [1, h-1]$ is the ancestor of v_h and $v_h.dom \subset v_i.dom$, that means $v_i.count$ has the same Laplace noise with the leaf's.

For any $i \in (1, h-1)$ with $v_i.count \leq \theta$, $ln\frac{Pr[v_i.count+Lap(\lambda)>\theta]}{Pr[v_i'.count+Lap(\lambda)>\theta]}$ equals $\frac{1}{\lambda}$. Otherwise it is less than $\frac{1}{\lambda}$, which has been proved by Zhang et al. in [19]. So,

$$\sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + Lap(\lambda) > \theta]}{Pr[v_i'.count + Lap(\lambda) > \theta]} \leq \frac{h-1}{\lambda} \tag{4}$$

$$ln\frac{Pr[v_h.count + Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + Lap(\lambda) = \overline{v_h.count}]} = \frac{1}{\lambda} \tag{5}$$

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} \leq \frac{h}{\lambda} \leq \varepsilon \tag{6}$$

Assume that there are n points in T, m is the number of the intermediate nodes, t is the number of the leaf nodes, and β is the fanout of the T. The total noise added to this private tree T is $\sum_{i=1}^{t} |Lap(\lambda)|$. It then can be seen that the noise cost of modified algorithm is smaller than the private tree generated by Algorithm 1, which is $\sum_{i=1}^{n} |Lap(\lambda)|$.

4.2 The Full Private β-tree with Indefeasible Laplace Noise

There is a deficiency of the low noise cost private tree, in which the privacy of the intermediate node is delicate. The intermediate nodes are noisy with the integrated Laplace noises of the leaves under it. The noise of an intermediate node which has β leaves equals $\sum_{i=1}^{k} Lap(\lambda)$ which may be 0 with a very high probability as Laplace is symmetric about the origin. As a result the intermediate node's point count approximately to be real.

We make the same assumption as the modified algorithms. In particular, we assume a private tree T is a full β-tree and each leaf has $\beta - 1$ brother leaves. Let $L = \{l_1, l_2, \cdots, l_\beta\}$ be the set of leaves of one intermediate node, $|L| = \beta$. Each time we randomly select one leaf l_i from the set L and add Laplace noise $\frac{i+1+\beta}{i+\beta} Lap(\lambda)$ into its point count, then update $L = L \setminus \{l_i\}$. Repeat the above operation until $L = \Phi$. The noise of intermediate node which has β leaves equals $\sum_{i=1}^{k} \frac{i+1+\beta}{i+\beta} Lap(\lambda) \neq 0$, therefore the noise of intermediate node will not be cancelled out.

$$\overline{v_i.count} = v_i.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) \tag{7}$$

Lemma 1.

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} = \sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) > \theta]}{Pr[v'_i.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) > \theta]}$$
$$+ln\frac{Pr[v_h.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) = \overline{v_h.count}]}{Pr[v'_h.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) = \overline{v_h.count}]} < \frac{h}{\lambda}$$

The Laplace noise is proportional to $S(f)$ and inversely proportional to ε. While $S(f)$ and ε are fixed values, increasing the noise with proportion of $\frac{i+1+\beta}{i+\beta}$ can also satisfy the ε-differential privacy.

Lemma 2. *The total noise cost of the private tree generated by improved algorithm is* $\frac{t}{\beta} \sum_{i=1}^{\beta} |\frac{i+1+\beta}{i+\beta} Lap(\lambda)|$, *smaller than the noise cost of private tree generated by Algorithm 1,which is* $\sum_{i=1}^{n} |Lap(\lambda)|$.

4.3 The General Private β-tree with Modified Indefeasible Laplace Noise

In practice, the spatial decomposition tree is not a full β-tree, but a general β-tree which each intermediate node has the same fanout β. We modify our strategy by adding noise into the ith selected leaf child with $\frac{(\beta-i+1)+1+\beta}{(\beta-i+1)+\beta} Lap(\lambda)$, causing its first selected leaf child with noise of $\frac{\beta+1+\beta}{\beta+\beta} Lap(\lambda)$.

It can be computed that $m = (t-1)/(\beta-1)$. We define the function $f(k) = \frac{k+\beta+1}{k+\beta}$, for any $k \in [1, \beta]$. $f(k)$ is a monotonically decreasing function, that is $f(i) > f(i+1)$ for any $i \in [1, \beta-1]$.

Lemma 3. *Assume that there are x_k leaves which are kth selected and add noises with $f(k)Lap(\lambda)$ in the β-tree, for any $k \in [1, \beta]$. The total noise cost of the private tree $\sum_{k=1}^{\beta} |x_k f(k) Lap(\lambda)| = (x_1 f(1) + \cdots x_\beta f(\beta))|Lap(\lambda)|$ (and $\sum_{k=1}^{\beta} x_k = t$) is smaller than the noise cost of the full β-tree, which has the same leaves count and intermediate nodes count with the general β-tree. Besides, the noise cost of full β-tree can be symbolically simplified as $\frac{t}{\beta} \sum_{k=1}^{\beta} |f(k)Lap(\lambda)|$.*

4.4 The Spatial Decomposition Algorithm InLN_DPSD

Our technique for private spatial decompositions is presented in Algorithm 2. We first generate a β-tree applying low noise cost private tree for spatial decompositions(line 1∼8). We split the node according to the real point count $v_i.count$ and the algorithm terminates depending on the minimum unit u, which is more intuitive than the maximal depth of recursion h(line 6).

Algorithm 2 InLN_DPSD(D,λ,θ,u)

```
1: initialize a quadtree T with a root node v_1;
2: set v_1.dom = Ω, and mark v_1 as visited ;
3: while there exists an unvisited node v_i do
4:    mark v_i as visited ;
5:    computer the number v_i.count of points in D that are contained in v_i.dom;
6:    if v_i.count > θ and Ω / 2^(depth(v_i)*d) > u then
7:       split v_i, and add its children to T;
8:       mark the children of v_i as unvisited;
9: for each i ∈ [1, n] do
10:   if isleaf(v_i) == 0 then
11:      initialize a children set L = φ;
12:      add all v_i's children to L;
13:      for each j ∈ [1, β], k = β do
14:         randomly select v_j from L;
15:         if isleaf(v_j) == 1 then
16:            v_j.count = v_j.count + (k+β+1)/(k+β) * Lap(λ);
17:            k − −;
18:         L = L \ v_j;
19: return T;
```

Then, add noise to the leaves of the β-tree (line 9∼18), which is the biggest innovation of InLN_DPSD. We first give the notion of indefeasible Laplace noise, which multiplies Laplace noise with coefficient $\frac{(\beta-i+1)+\beta+1}{(\beta-i+1)+\beta}$, where i indicates the ith leaf randomly selected from the intermediate node's children, and β is the fanout of the private tree.

In the process of adding indefeasible noise, we check all nodes of the β-tree. If the node is not a leaf, add noise to its children who are leaves. An intermediate node's child leaves are added to Laplace noise multiplied with coefficient $\frac{k+\beta+1}{k+\beta}$, where k decreases from β to 1. Usually k may be greater than 1 for not all the β children are leaves, and k only is decreased by 1 when a child leaf is processed(line 15∼17). It has been proved in Sect. 4.2 that when the private tree was added to Laplace noise multiplied $\frac{k+\beta+1}{k+\beta}$ where k decreases from β, its total noise cost is less than the DPSD algorithm. Otherwise if the k increases from 1 to β, the noise cost is larger than existing algorithms'.

5 Experiment

This section we evaluate InLN_DPSD against the state-of-the-art algorithms of spatial decompositions based on differential privacy.

5.1 Competing Methods and Testing Datasets

CompetingMethods. To evaluate the efficacy of the proposed approaches, we compare InLN_DPSD with the DPSD and PrivTree.

TestingDatasets. In the experiments we employ one synthetic two-dimensional dataset and two real spatial datasets. SD contains 1 million location information; Beijing[1] is two-dimensional real dataset which contains 15 million records of pickup locations of Beijing taxis; NYC[2], a four-dimensional real dataset containing one hundred million records of pickup and drop-off locations of NYC taxis in 2013. The distribution of these datasets is shown in Fig. 2.

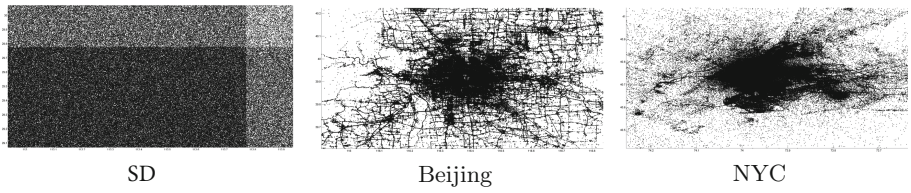


Fig. 2. The distribution of each dataset

[1] http://research.microsoft.com/apps/pubs/?id=152883.
[2] http://publish.illinois.edu/dbwork/open-data/.

5.2 Evaluation Measures

We run each algorithm on every dataset to evaluate their performances. We respectively generate ten thousand queries on the region covering [0.1%, 1%), [1%, 10%), [10%, 100%) of the dataset domain and get their answers with each algorithm. For evaluating the query accuracy, we define the relative error RE [17,18] as the measure accuracy of a perturbed answer $\overline{q(D)}$ to a query q by its real answer $q(D)$.

$$RE(\overline{q(D)}) = \frac{|\overline{q(D)} - q(D)|}{\max\{q(D), \Delta\}}$$

where Δ is a smoothing factor [27] set to 1% of the dataset cardinality n. We repeat each experiment 1000 times and report the average relative error of each method for each query set.

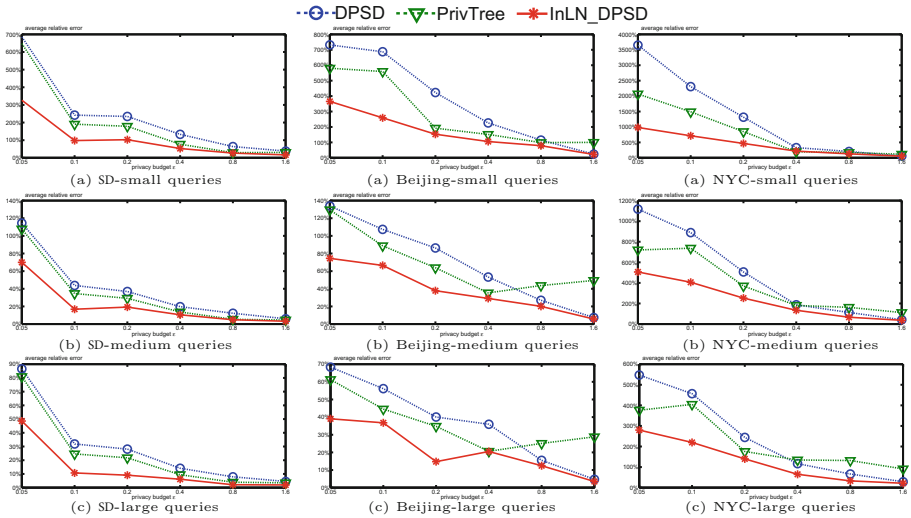


Fig. 3. The results of average relative error of range count queries on each dataset

5.3 Results and Analyses

Figure 3 shows the average relative error of the three different range queries on each dataset applying every algorithm with different privacy budget ε. From the results we can see:

(1) The relative error of InLN_DPSD algorithm is always smaller than the other two algorithms. That shows our algorithm outperforms.
(2) The relative error of NYC dataset is much larger than SD and Beijing datasets. The more skewed the dataset is, the larger relative error it has.

(3) The relative error increases as privacy budget ε increases. Since the noise added to dataset is inversely proportional to ε.
(4) The relative error is inversely proportional to the query area. That indicates the larger the query area is, the more accurate the answer is. If the query area is much smaller, even to one tuple's location, the relative error will be very large, which preserves the individual's privacy.

Table 1. The number of Laplace noises added to the private tree

	DPSD	PrivTree	InLN_DPSD
SD	736449	755261	227677
Beijing	405269	371705	145216
NYC	130209	109041	63527

Table 1 shows the number of noises added to the private tree in different datasets when applying these three algorithms. From the table we can see that the InLN_DPSD algorithm adds fewer noises to the private tree.

6 Conclusion

In this paper, we study the problem of spatial decompositions based differential privacy. The existing algorithm's total noise cost is too high and the recursion depth h of the private tree is hard to choose. Most important of all, the traditional differential privacy is delicate as the Laplace distribution is symmetric about the origin, the sum of several Laplace noises may be 0 so that the added noises may be canceled and the privacy may be compromised. We take the strategy of only adding indefeasible Laplace noises to leaves, the noises will not be canceled and the domain splitting will be more accurate which leads to higher data utility and the total noise cost of the private tree is highly decreased. The predefined h is replaced by minimal split unit u, which is more intuitive for users. Based on this strategy we propose InLN_DPSD, a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy. The experiment results show that InLN_DPSD outperforms the DPSD and PrivTree both in a synthetic dataset and real dataset.

References

1. Yin, H., Chen, H., Sun, X., et al.: SPTF: a scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In: IEEE International Conference on Data Mining, pp. 585–594. IEEE Press, New Orleans (2017)
2. Chen, H., Yin, H., Wang, W., et al.: PME: projected metric embedding on heterogeneous networks for link prediction. In: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1177–1186. ACM Press, London (2018)
3. Chen, T., Yin, H., Chen, H., et al.: TADA: trend alignment with dual-attention multi-task recurrent neural networks for sales prediction. In: IEEE International Conference on Data Mining. IEEE Press, Singapore (2018)
4. Yin, H., Wang, W., Wang, H., et al.: Spatial-aware hierarchical collaborative deep learning for POI recommendation. IEEE Trans. Knowl. Data Eng. **29**(11), 2537–2551 (2017)
5. Yin, H., Sun, Y., Cui, B., et al.: LCARS: a location-content-aware recommender system. In: 19th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 221–229. IEEE Press, Chicago (2013)
6. Friedman, A., Schuster, A.: Data mining with differential privacy. In: 16th International Conference on Knowledge Discovery and Data Mining, pp. 493–502. ACM Press, Washington (2010)
7. Fung, B.C.M.: Privacy-preserving data publishing. ACM Comput. Surv. **42**(4), 1–53 (2010)
8. Hardt, M., Ligett, K., Mcsherry, F.: A simple and practical algorithm for differentially private data release. In: Advances in Neural Information Processing Systems, pp. 2339–2347 (2010)
9. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
10. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
11. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
12. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
13. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
14. Xu, J., Zhang, Z., Xiao, X., et al.: Differentially private histogram publication. In: 29th IEEE International Conference on Data Engineering, pp. 32–43. IEEE Press, Brisbane (2013)
15. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. In: 26th IEEE International Conference on Data Engineering, pp. 225–236. IEEE Press (2010)
16. Mohammed, N., Chen, R., Fung, B.C.M., et al.: Differentially private data release for data mining. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 493–501. ACM press (2011)
17. Cormode, G., Procopiuc, C., Srivastava, D., et al.: Differentially private spatial decompositions. In: 28th IEEE International Conference on Data Engineering, pp. 20–31. IEEE Press, Washington (2012)

18. Li, N., Yang, W., Qardaji, W.: Differentially private grids for geospatial data. In: 28th IEEE International Conference on Data Engineering, pp. 757–768. IEEE Press, Washington (2012)
19. Zhang, J., Xiao, X., Xie, X.: PrivTree: a differentially private algorithm for hierarchical decompositions. In: 35th ACM Conference on Management of Data, pp. 155–170. ACM Press, San Franciso (2016)
20. Zhang, J., Cormode, G., et al.: PrivBayes: private data release via Bayesian networks. In: 33th ACM Conference on Management of Data, pp. 1423–1434. ACM Press, Utah (2014)
21. Zhang, J., Cormode, G., et al.: Private release of graph statistics using ladder functions. In: 34th ACM Conference on Management of Data, pp. 731–745. ACM Press, Melbourne (2015)
22. Miller, F.P., Vandome, A.F., Mcbrewster, J.: KD-tree (2009)
23. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: International Conference on Management of Data 1984, pp. 47–57. ACM Press, Massachusetts (1984)
24. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. In: The 25th ACM Symposium on Theory of Computing, pp. 226–234 (1993)
25. Demaine, E.D., Mozes, S., Rossman, B., et al.: An optimal decomposition algorithm for tree edit distance. ACM Trans. Algorithms **6**(1), 1–19 (2007)
26. Li, B., et al.: Dynamic reverse furthest neighbor querying algorithm of moving objects. In: Li, J., Li, X., Wang, S., Li, J., Sheng, Q.Z. (eds.) ADMA 2016. LNCS (LNAI), vol. 10086, pp. 266–279. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49586-6_18
27. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Trans. Knowl. Data Eng. **23**(8), 1200–1214 (2011)