Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 8, August 2019, pp. 2145-2152
DOI: 10.23940/ijpe.19.08.p14.21452152

Adaptive Grid Decomposition Algorithm based on Standard Deviation Circle Radius

Guoqiang Zhou[a,b,*], Xiulian Tang[a], and Shui Qin[a]

[a]*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210000, China*
[b]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210000, China*

Abstract

The differential privacy preservation model based on spatial dataset meshing has been widely concerned, but the distribution characteristics of the dataset and user's query granularity are often ignored or not fully considered in the partitioning of the dataset. Aiming at deficiencies in existing mesh-based algorithms, a standard deviation circle radius adaptive grid decomposition (SDCAG) algorithm is proposed. Firstly, the standard deviation circle radius is introduced to quantitatively represent the distribution characteristics of datasets in order to calculate privacy preservation requirements. Secondly, filtering and bucketing are used to reduce the noise error. Finally, the improved query precision is implemented based on the post-processing. Experiments on the NYC dataset, the Beijing dataset, and the Checkin dataset show that the SDCAG algorithm is superior to similar algorithms in terms of query performance.

Keywords: differential privacy; spatial dataset; standard deviation circle radius; adaptive grid; post-processing

(Submitted on June 10, 2019; Revised on July 21, 2019; Accepted on August 15, 2019)

1. Introduction

Providing location-based services by perceiving and collecting the spatial location information of users via mobile terminals is an important part of the future Internet industry. The leakage of the user's geographical location information may lead to the disclosure of personal information such as travel modes and lifestyle. Therefore, privacy preservation for spatial datasets has been a challenge.

A variety of privacy preservation models have proposed for spatial datasets [1-4]. The differential privacy preservation model [3] resists the background knowledge attack suffered by the traditional privacy preservation algorithms and carries out rigorous mathematical proofs and quantitative representation of the privacy leakage risk, so it is widely used.

Spatial datasets based on differential privacy generally divide datasets by partitioning [4], such as the data structure of the index (grid, Quad-tree, b-ary tree, etc.), and then add noise to each divided unit. The UG [4] algorithm divides spatial datasets with uniform grids. Although the partitioning granularity of the dataset is reasonably set, the sparsity of the dataset distribution is not considered. Aiming at this problem, the AG [4] algorithm adopts a strategy of top-down adaptive partitioning based on high-level partitioning granularity, but it does not give heuristic rules to distinguish data dense and sparse boundaries, and the user's query granularity is not considered. The STAG [5] algorithm performs three-layer adaptive meshing on the dataset, taking into account the sparseness of the dataset and the query granularity. However, in the noise-adding phase, the noise of uniform scale is added to the same-level grids without considering that the privacy preservation requirements of grids with different distribution characteristics are often different. It can be found that the current research on spatial dataset meshing partitioning based on differential privacy does not fully consider the distribution characteristics of spatial datasets and the user's query granularity, resulting in large noise errors that reduce the availability and query accuracy of datasets.

* Corresponding author.
E-mail address: winniay@163.com

Based on the above analysis, this paper proposes a standard deviation circle radius adaptive grid decomposition (SDCAG) algorithm, which describes the privacy preservation requirement of each grid by quantitatively describing the distribution characteristics of the dataset, so that the privacy budget is allocated reasonably. Aiming at the user's query granularity, the noise error is reduced by filtering and binning in the second layer meshing, and the query precision of the range query is improved by post-processing. Eventually, noise is added on demand, noise errors are reduced, and query accuracy and dataset availability are improved.

2. Related Work

In the research of dataset publishing based on differential privacy, methods such as histogram publishing [6-7], sampling-filtering [8], and partitioning [4] publishing have emerged. Partitioning is a common method to divide spatial datasets based on differential privacy. The early spatial dataset partitioning methods use uniform grids to divide the spatial dataset, which reasonably set the partition granularity of the dataset, such as the UG [5] and DP-Where [10] algorithms. However, there is a definite possibility that the noise error appears in sparse units due to excessive division of the algorithm and the uniform error occurs in dense units due to insufficient division of the algorithm. For the sparsity of data distribution, the AG [5] algorithm adopts a strategy of top-down adaptive partitioning based on high level partitioning granularity to equalize the noise error and the uniform error, but there is no heuristic consideration for the distribution of spatial data, which may lead to too much noise added to local area, and the query accuracy is not enough. The STAG [6] algorithm performs three-layer adaptive meshing on the dataset based on Bernoulli Random sampling, which effectively takes into account large-scale spatial data, data skewness and query precision. However, in the noise addition phase, noise with uniform proportion is added to grids of the same level without taking into account differences among the privacy preservation requirements of the grids with different distribution characteristics. The Quad-Post [10] and QuadTree [11] algorithms introduce Quad-tree to partition spatial datasets, and the FA [12] algorithm combines Fibonacci and Quad-trees to allocate privacy budgets. Although these algorithms can reasonably allocate privacy budgets, they do not consider the uneven distribution of the spatial dataset that will cause a large even error. In addition, they control the noise by the depth of the tree; the larger the depth of the tree, the larger the noise error. In order to avoid dependency on the depth of the tree, the PrivTree [13] algorithm uses the offset value of the node count to reduce the noise error, but it still does not take into account the user's query granularity. The H_b [14] algorithm introduces the b-ary tree to hierarchically divide the dataset and processes the query result through constraint reasoning to improve the query precision. However, it also depends on the depth of the tree to control the noise and does not take into account the user's query granularity.

3. Methodology

In this section, this paper proposes a standard deviation circle radius adaptive grid decomposition algorithm (SDCAG). First, the specific flow of the algorithm is shown, and then the implementation details of the key steps of the algorithm are explained.

3.1. SDCAG

The SDCAG algorithm first allocates the privacy budget on demand based on the spatial dataset distribution feature. Then, for the user's query granularity, it introduces filtering and bucketing to the second layer meshing to reduce the noise error. Finally, it constrains the data to improve the query accuracy of the dataset range query.

The steps for the SDCAG algorithm are as follows:

Step 1 Data Processing: First, each record in a given spatial dataset *D* is regarded as a data point in two-dimensional coordinates and is mapped into a two-dimensional coordinate area by the longitude and latitude of the record. Finally, a rectangular area containing these points, namely, research objects, is obtained.

Step 2 Data Division: The rectangle obtained by data processing adaptively is divided into two layers of grids.

Step 3 Distribution of Privacy Budget: In the data division, the standard deviation circle radius of each grid is first calculated. Then, the proportion of the standard deviation circle radius of each grid in the layer grid is calculated, that is, privacy preservation requirements. Finally, privacy budgets are allocated on demand based on privacy preservation requirements.

Step 4 Filtering and Bucketing: Filtering mainly considers the influence of the 0 value. If the raw count of the grid is

0, the noise is directly set to 0. The similar grids are divided into the same bucket by granularity in the second layer grid.

Step 5 Adding Noise: After Steps 3 and 4, the corresponding laplace noise is added to the bucket according to the allocated privacy budget.

Step 6 Result Release: The processed dataset with noise count is released.

3.2. Division of Datasets

In this paper, datasets are divided into two layers according to [15]. The first layer is divided into coarse grained $m_1 \times m_1$, determined by the distribution characteristics of the spatial dataset in spatial dataset meshing, and the second layer is adaptively divided into fine grained $m_2 \times m_2$. The coarse grained m_1 is:

$$m_1 = \max(10, \left\lceil \frac{1}{4}\sqrt{\frac{N \times \varepsilon}{C_1}} \right\rceil) \tag{1}$$

In Equation (1), N represents the number of samples of the spatial dataset and ε represents the total privacy budget. C_1 is a constant and is taken as $\sqrt{2}$ in this paper.

Following the privacy allocation and noise addition processing after the first layer of data division, the second layer adaptive $m_2 \times m_2$ is further divided, and the fine grained m_2 is:

$$m_2 = \left\lceil \sqrt{\frac{\tilde{x}_i \times \varepsilon_i}{C_2}} \right\rceil \tag{2}$$

In Equation (2), $\tilde{x}_i$ represents the noise count of the i^{th} grid of the first layer, ε_i is the privacy budget of the i^{th} grid, and C_2 is the constant $C_1 / 2$. The final value of C_2 in [13] is taken as $\sqrt{2}$ to avoid excessive division.

3.3. Distribution of Privacy Budget

After grid partitioning of datasets, the privacy budget should be allocated to the grid. In order to improve the deficiency of the traditional grid partitioning algorithm in adding uniform scale noise to the grid, the SDCAG algorithm quantitatively represents the distribution characteristics of data by calculating the dispersion degree of the grid partitioned and further calculates the privacy preservation requirements of the grid. Finally, it adds the corresponding noise by the privacy preservation requirements to reduce the noise error.

In classical statistics, the standard deviation describes the deviation of the observed value from the mean, while the radius of the standard deviation circle describes the spatial deviation of the data point from the mean center in the spatial point pattern analysis. By describing the degree of dispersion of the grid by the standard deviation circle radius, the distribution characteristics of the grid can be quantitatively represented, and the proportion of the standard deviation circle radius in the layer grid can be used to indicate the privacy preservation requirements of the data.

The formula for calculating the standard deviation circle radius is:

$$r = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}{n-2}} \tag{3}$$

In Equation (3), $\bar{x}$, $\bar{y}$ represents the average of the coordinate values distributed in a spatial dataset point, x_i, y_i represents the horizontal and vertical coordinate values of each point, and n represents the number of points in the space.

The formula for calculating the requirement of privacy preservation is:

$$\rho_i = \frac{r_i}{sum_j} \tag{4}$$

In Equation (4), r_i represents the standard deviation circle radius of the i^{th} region obtained according to Equation (3) and sum_j represents the sum of the standard deviation circle radii of all regions of the j^{th} layer including the i^{th} region.

After the first layer of meshing, the calculation formula of the privacy budget ε_i allocated according to the privacy preservation requirement is:

$$\varepsilon_i = \rho_i \times \varepsilon \tag{5}$$

3.4. Filtration and Bucket Processing

The SDCAG algorithm filters and buckets the grid in the second layer to improve large query errors existing in the query granularity of the second layer grid.

First, fine grained m_2 partitioning is performed on the first layer of the grid $\tilde{x}_i$. Then, the filtering operation is performed, mainly for the influence of a large number of zero values, and for the grid whose original true count is 0, the noise is directly set to 0. After that, the divided $m_2 \times m_2$ grids are loaded into the bucket corresponding to $V_{m_2 \times m_2}$ according to the $step = x_i / m_2 \times m_2$. Finally, add noise $\tilde{V}_k = \frac{(|bucket| + Lap(1/\varepsilon_i))}{bucket.size()}$ to the grids in the bucket, where |*bucket*| is the number of data loaded in the grid in the bucket, ε_i is the privacy budget of the grid $\tilde{x}_i$, and *bucket.size*() is the number of grids in the bucket. Finally, return to the grid $\tilde{x}_i$ after the noise processing.

The pseudo-code of the filter-bucket algorithm is as follows:

Algorithm 1 Filter-bucket

Input: Grid $\tilde{x}_i$; privacy budget ε_i

Output: Grid $\tilde{x}_i$

1: $m_2 = \left\lceil \sqrt{\frac{\tilde{x}_i \times \varepsilon_i}{C_2}} \right\rceil$
2: $V_{m_2 \times m_2} = split(\tilde{x}_i, m_2)$; //divide the grid $\tilde{x}_i$ into $m_2 \times m_2$ cells $V_{m_2 \times m_2}$ according to m_2
3: $V'_{m_2 \times m_2} = filter(V_{m_2 \times m_2})$
4: $step = x_i / m_2 \times m_2$;
5: **for** (i=0; i< x_i ; i+=*step*) **do**
6: LinkList *bucket*;
7: **for** j=1 to $|V'_{m_2 \times m_2}|$ **do**
8: **if** ($|V'_j| >$ i && $|V'_j| \le$ i + step) **do**
9: add $|V'_j|$ to bucket
10: **end if**
11: **end for**
12: **for** (k=1;k<= *bucket.size*() ;k++) **do**
13: $\tilde{V}_k = \frac{(|bucket| + Lap(1/\varepsilon_i))}{bucket.size()}$;
14: **end for**
15: **end for**
16: **return** $\tilde{x}_i$

3.5. Post-Processing

In order to improve the query accuracy of datasets, a post-processing method is proposed based on [12]: the grid $\tilde{x}_i$ in the

first layer is divided into $m_2 \times m_2$ cells, the noise count is denoted by $\tilde{V}_1, \tilde{V}_2, \cdots, \tilde{V}_{m_2 \times m_2}$, and then the post-processing of $\tilde{x}_i$ is:

$$\tilde{x}_i = \frac{((m_2 \times m_2)\tilde{x}_i + \sum_{j=1}^{m_2 \times m_2} \tilde{V}_j)}{m_2 \times m_2 + 1} \tag{6}$$

For instance, $x_3 = 3$, and after adding noise, $\tilde{x}_3 = 4$. The grid is divided into 2×2 subgrids $\tilde{V}_1$, $\tilde{V}_2$, $\tilde{V}_3$, $\tilde{V}_4$, and the four noise counts are 1, 1, 0, and 0, respectively. After the post constraint, $\tilde{x}_3 = 3.4$.

4. Experiments and Analysis

4.1. Experimental Datasets

To verify the impact of the SDCAG algorithm on the query accuracy, experiments are performed using the NYC dataset, the Beijing dataset, and the Checkin dataset. The NYC dataset contains the geographic coordinates of the journey and departure of New York taxis in 2010. The Beijing dataset consists of the geographic coordinates of the 10,357 taxis in Beijing in a week in February 2008. The Checkin dataset comes from Gowalla, a location-based social networking site that includes location points, registration time, and location ID. The visualization of the three datasets mapped to two-dimensional plane coordinates is shown in Figure 1.

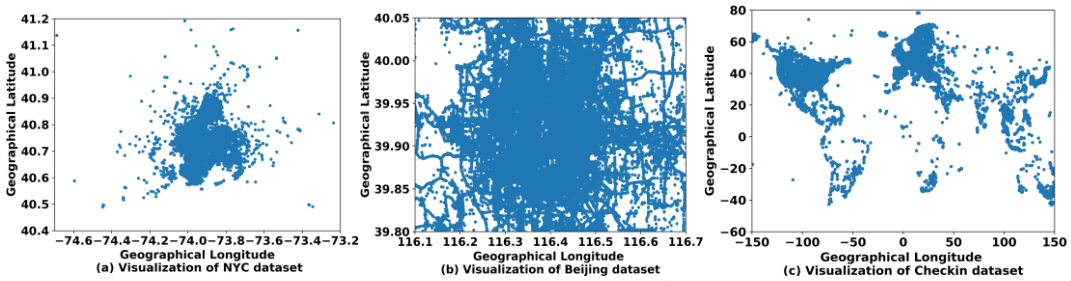


Figure 1. Visualization result graphs of three datasets

4.2. Experimental Evaluation

In order to measure the merits of spatial dataset meshing algorithms, the relative error is introduced to measure the range query accuracy of spatial dataset meshing algorithms.

$$Error(Q) = \frac{|Q(D) - Q(\tilde{D})|}{\max(Q(D), |D| \times 0.1\%)} \tag{7}$$

In Equation (7), $Q(D)$ is the result of Q's real range query on the D dataset, $Q(\tilde{D})$ represents the noise result on the D dataset, and $|D|$ represents the size of the D dataset. $\max(Q(D), |D| \times 0.1\%)$ indicates that when $Q(D)$ =0, the function value is $|D| \times 0.1\%$.

4.3. Experimental Approach

In order to make the experimental results comparable, two classical privacy preservation algorithms UG and AG are selected and applied to the above experimental data and compared with the SDCAG algorithm. In the experiment, six query rectangles of different sizes $Q = \{q_1, \cdots, q_6\}$ are set up, as shown in Table 1. Under three different privacy budgets of 0.1, 0.5, and 1, different range queries are experimented. For each query rectangle, 500 queries are generated randomly, and the average value of queries is calculated. The privacy preservation performance of each algorithm is compared by calculating the relative errors of the query results of each algorithm model. It can be explained that the offset of query results between publishing datasets and original datasets is smaller, the noise error in adding noise phase is smaller, and the query accuracy and availability of datasets are higher if the relative error of the algorithm is smaller. That is to say, the privacy preservation

effect of the algorithm is better.

Table 1. Parameter information about datasets

Datasets	Actual size	Sample size	Query size
NYC	11145410	1000000	$q_i = 0.05(i+1) \times 0.05(i+1),\ i \in [1,6]$
Beijing	15000000	1000000	$q_i = 0.02i \times 0.02i,\ i \in [1,6]$
Checkin	6442890	680000	$q_i = 10(i+1) \times 10(i+1),\ i \in [1,6]$

4.4. Analysis of Experimental Results

The relative errors of the UG algorithm, the AG algorithm, and the SDCAG algorithm under different query ranges Q on the three datasets are calculated.

4.4.1. NYC Dataset

Figures 2(a)-(c) show the relative error values of each query range when the privacy budget is fixed. It can be seen that the performance of SDCAG is far superior to that of the other algorithms. When $\varepsilon = 0.1$ and the query size is q_3, the query accuracy of SDCAG is nearly ten times that of UG and AG. When $\varepsilon = 0.5$ and the query size is q_1, the query accuracy of SDCAG is nearly seven times that of UG and four times that of AG. When $\varepsilon = 1$ and the query size is q_4, the query accuracy of SDCAG is more than five times that of UG and nearly four times that of AG. Figures 2(d)-(i) illustrate the query accuracy with a fixed query range privacy budget ε from 0.1 to 1. It can be seen that the larger the privacy budget, the smaller the relative error, and the query accuracy of SDCAG still significantly outperforms UG and AG. The reason is that the distribution of the dataset is sparse, and SDCAG takes into account the distribution of the dataset by using the standard deviation circle radius and reasonably allocates the privacy budget.

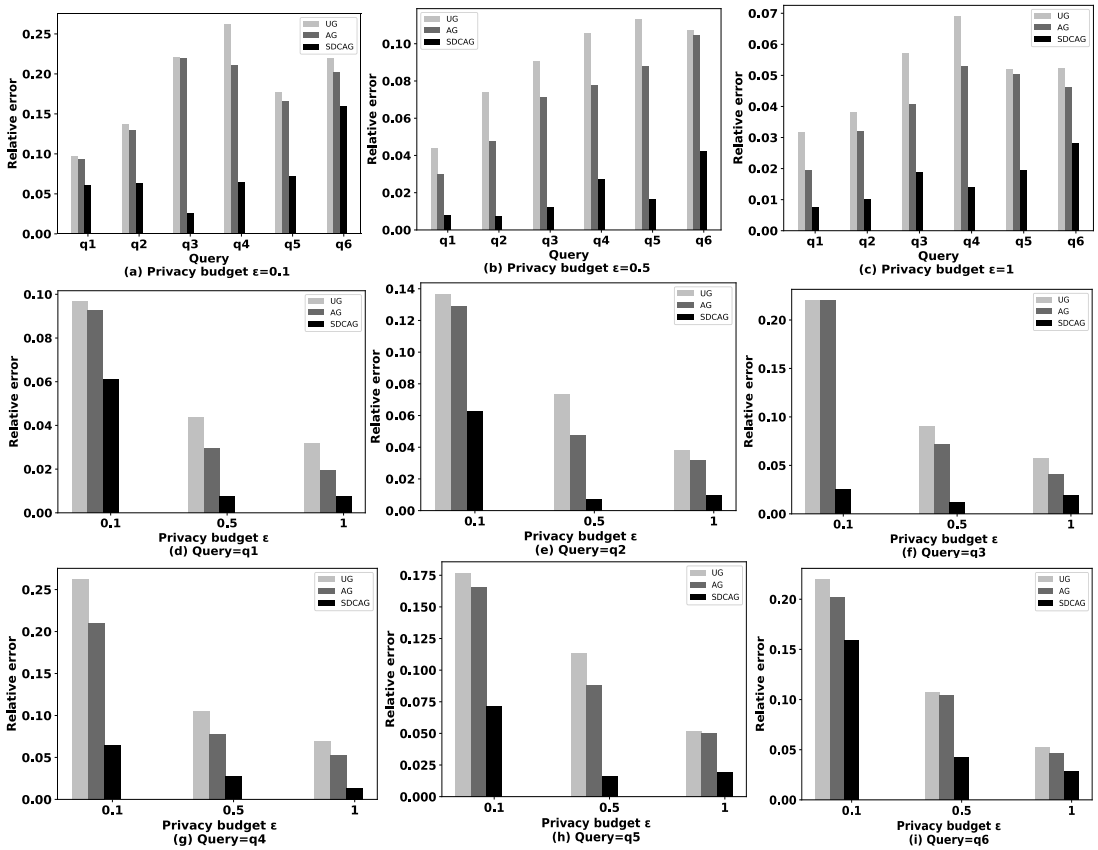


Figure 2. Results of range queries on NYC datasets

4.4.2. Beijing Dataset

Figures 3(a)-(c) show the relative error values of each query range when the privacy budget is fixed. It can be seen that the larger the query range, the smaller the query accuracy. The difference of the query accuracy between SDCAG and AG is not very significant, but when $\varepsilon = 0.5$ and the query size is q_3, the query accuracy of SDCAG is nearly twice that of UG and AG. Figures 3(d)-(i) illustrate the query accuracy with a fixed query range privacy budget ε from 0.1 to 1. It can be seen that the SDCAG algorithm has better performance than the other algorithms in terms of query accuracy, especially in the medium query range.

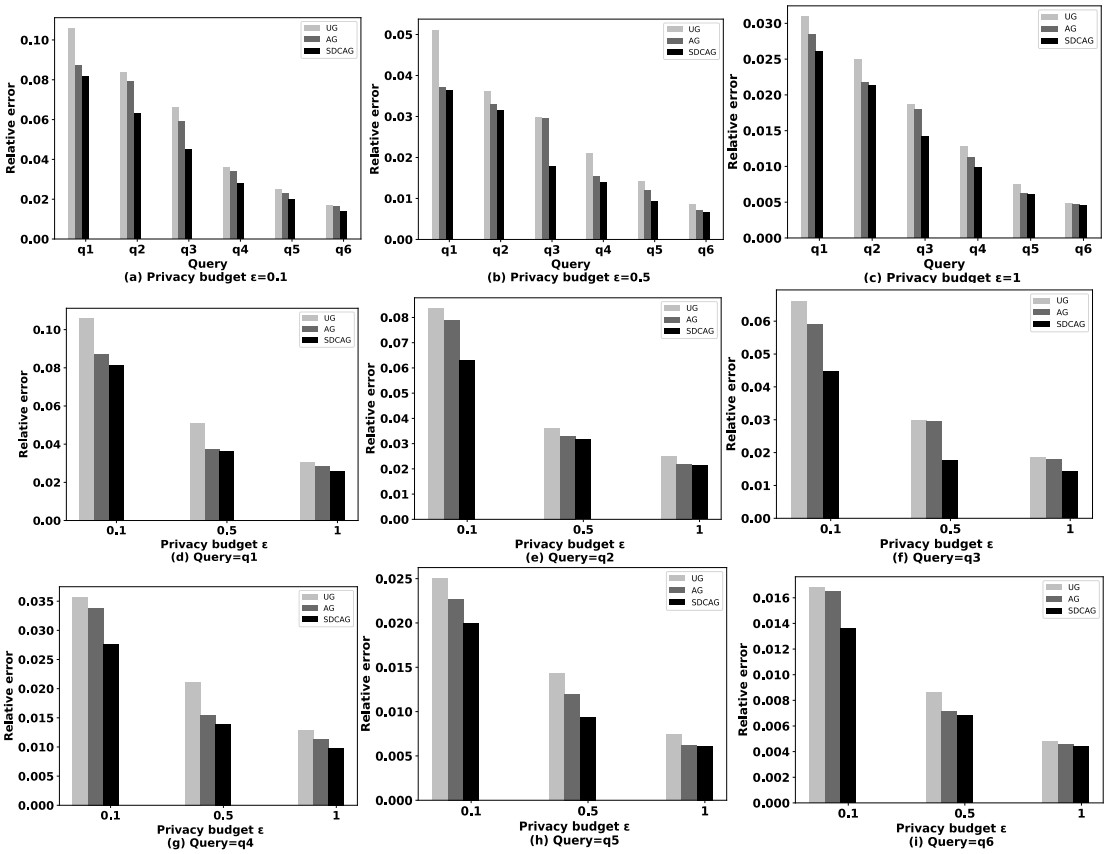
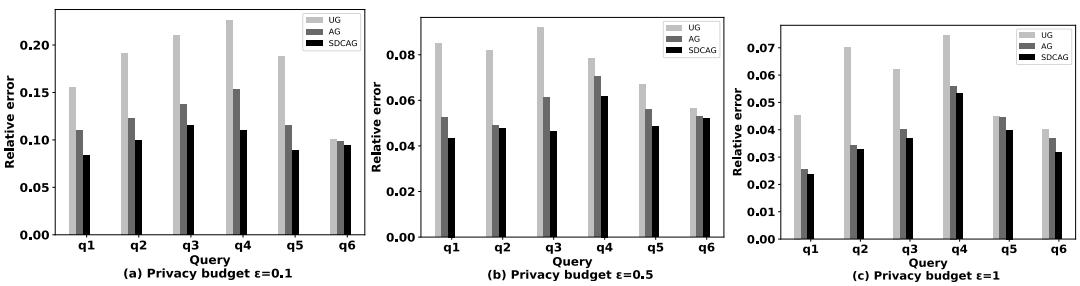


Figure 3. Results of range queries on Beijing dataset

4.4.3. Checkin Dataset

Figures 4(a)-(c) show relative error values for each query range when the privacy budget is fixed. It can be seen that the SDCAG query accuracy is significantly better than that of other algorithms. When $\varepsilon = 0.1$, the query accuracy of SDCAG is nearly twice that of UG and 1.5 times that of AG. Figures 4(d)-(i) illustrate the query accuracy with a fixed query range privacy budget ε from 0.1 to 1. The query accuracy of SDCAG is still superior to that of other methods. When the query range is q_3, the query accuracy of SDCAG is nearly twice as large as that of AG and UG. The reason is that the dataset has high sparsity, and SDCAG avoids large noise errors caused by sparsity through filtering and bucketing.



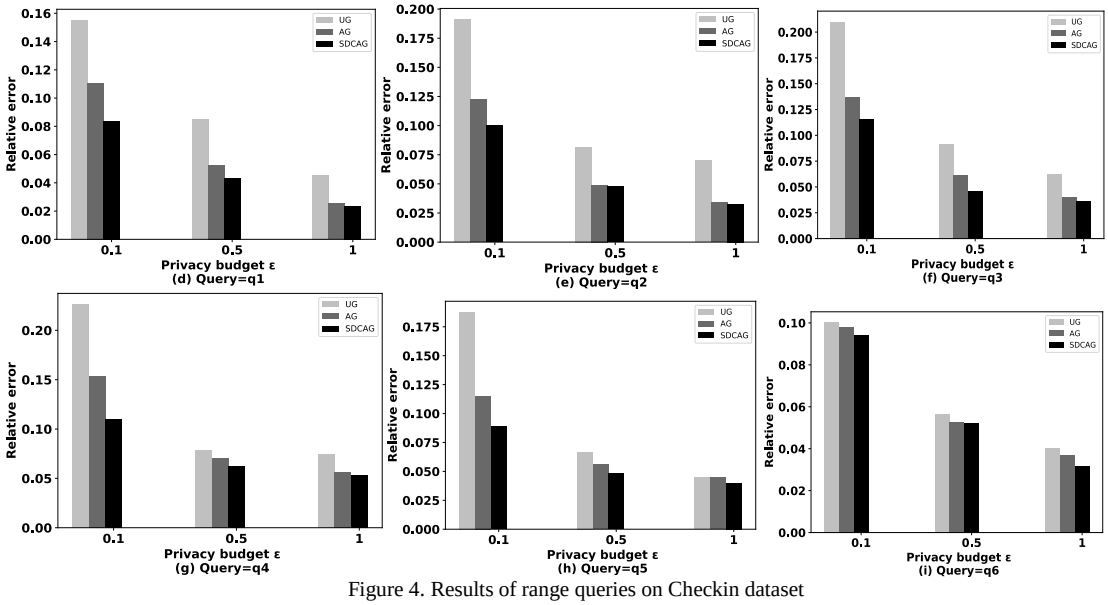


Figure 4. Results of range queries on Checkin dataset

5. Conclusions

For the current privacy preservation model based on spatial datasets, it is difficult for the classic meshing algorithms to meet its needs. The SDCAG algorithm distributes the privacy budget of the dataset reasonably by the standard deviation circle radius, uses filtering and bucket to reduce the noise error, and combines the post-processing to improve the query precision. SDCAG effectively improves the deficiencies in the differential privacy preservation algorithms based on mesh-based spatial datasets. The future work mainly considers the privacy preservation of multidimensional dynamic spatial datasets.

References

1. L. Sweeney, "*k*-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 10, No. 5, pp. 557-570, 2002
2. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-Diversity: Privacy Beyond k-Anonymity," in *Proceedings of 22nd International Conference on Data Engineering* (*ICDE'06*), pp. 24-24, 2006
3. N. Li, T. Li, and S. Venkatasubramanian. "*t*-Closeness: Privacy Beyond *k*-Anonymity and Diversity," in *Proceedings of 23rd International Conference on Data Engineering IEEE*, pp. 106-115, 2007
4. C. Dwork, "Differential Privacy," LNCS 4052: Lecture Notes in Computer Science, pp.1-12, Springer, Berlin, 2006
5. W. Qardaji, W. Yang, and N. Li, "Differentially Private Grids for Geospatial Data," in *Proceedings of* 29*th International Conference on Data Engineering* (*ICDE*), pp. 757-768, 2013
6. X. J. Zhang, K. Z. Jin, and X. F. Meng, "Private Spatial Decomposition with Adaptive Grid," *Journal of Computer Research and Development*, Vol. 55, No. 6, pp. 29-42, 2018
7. M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang, "Principled Evaluation of Differentially Private Algorithms using Dpbench," in *Proceedings of the 2016 International Conference on Management of Data ACM*, pp. 139-154, 2016
8. J. Xu, Z. J. Zhang, X. K. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially Private Histogram Publication," *The VLDB Journal*, Vol. 22, No. 6, pp. 797-822, 2013
9. G. Cormode, C. Procopiuc, D. Srivastava, and T. T. L. Tran, "Differentially Private Summaries for Sparse Data," in *Proceedings of International Conference on Database Theory ACM*, pp. 299-311, 2012
10. D. J. MIR, S. Isaacman, R. C ́aceres, M. Martonosi, and R. N. Wright, "Dp-Where: Differentially Private Modeling of Human Mobility," in *Proceedings of International Conference on Big Data*, pp. 580-588, 2013
11. G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially Private Spatial Decompositions", in *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, pp. 20-31, 2012
12. L. Y. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring Web Browsing Behavior with Differential Privacy," in *Proceedings of the 23rd International Conference on World Wide Web*, pp. 177-188, 2014
13. J. Wang, S. B. Liu, Y. K. Li, H. Cao, and M. J. Liu, "Differentially Private Spatial Decompositions for Geospatial Point Data," *China Communications*, Vol. 13, No. 4, pp. 97-107, 2016
14. J. Zhang, X. Xiao, and X. Xie, "Privtree: A Differentially Private Algorithm for Hierarchical Decompositions," in *Proceedings of the 2016 International Conference on Management of Data*, pp. 155-170, 2016
15. W. Qardaji, W. Yang, and N. Li, "Understanding Hierarchical Methods for Differentially Private Histograms," *Proceedings of the VLDB Endowment*, Vol. 6, No. 14, pp. 1954-1965, 2013
16. H. To, G. Ghinita, and C. Shahabi, "A Framework for Protecting Worker Location Privacy in Spatial Crowdsourcing," *Proceedings of the VLDB Endowment*, Vol. 7, No. 10, pp. 919-930, 2014