Privacy Preserving Publication of Locations Based on Delaunay Triangulation*

Jun Luo[1,2], Jinfei Liu[3], and Li Xiong[3]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2] Huawei Noah's Ark Laboratory, Hong Kong, China
[3] Department of Mathematics & Computer Science, Emory University, Atlanta, USA
jun.luo@siat.ac.cn, jinfei.liu@emory.edu,
lxiong@mathcs.emory.edu

Abstract. The pervasive usage of LBS (Location Based Services) has caused serious risk of personal privacy. In order to preserve the privacy of locations, only the anonymized or perturbated data are published. At the same time, the data mining results for the perturbated data should keep as close as possible to the data mining results for the original data. In this paper, we propose a novel perturbation method such that the Delaunay triangulation of the perturbated data is the same as that of the original data. Theoretically, the Delaunay triangulation of point data set presents the neighborhood relationships of points. Many data mining algorithms strongly depend on the neighborhood relationships of points. Our method is proved to be effective and efficient by performing several popular data mining algorithms such as KNN, K-means, DBSCAN.

Keywords: Privacy Preserving, Location, Delaunay Triangulation, Data Mining.

1 Introduction

Due to the rapid development of location sensing technology such as GPS, WiFi, GSM and so on, huge amount of location data through GPS and mobile devices are produced every day. The flood of location data provides the numerous opportunities for data mining applications and geo-social networking applications. For example, mining the trajectories of floating car data (FCD) in a city could help predict the traffic congestion and improve urban planning. For individual driver, we can analyze his/her driving behavior to improve his/her profit. Moreover, many popular mobile device applications are based on locations such as FourSquare[1], Google Latitude[2], *etc*.

However, exposing location data of individuals could cause serious risk of personal privacy. For example, Buchin *et al.* [1] provide an algorithm to find the commuting pattern for an individual by clustering his/her daily trajectories. With some extra information such as timestamps, we can easily identify his/her home and working place.

* This research has been partially supported by NSF of China under project 11271351 and AFOSR under grant FA9550-12-1-0240.

[1] https://foursquare.com/

[2] http://www.google.com/latitude

V.S. Tseng et al. (Eds.): PAKDD 2014, Part I, LNAI 8443, pp. 594–605, 2014.

Another real world example about the risk of revealing location information in social networking application is that thieves planned home invasions according to user's location information and his/her planned activity published in Facebook[3].

The goals of data mining and privacy protection are often conflicting with each other and can not be satisfied simultaneously. On the one hand, we want to mine meaningful results from the dataset which requires the dataset to be precise. On the other hand, in order to protect privacy, we usually need to modify the original dataset. There are two widely adopted ways for modifying location data:

- cloaking, which hides a user's location in a larger region.
- perturbation, which is accomplished by transforming one original value to another new value.

A widely used privacy principle for cloaking is k-anonymity. A dataset is said to satisfy k-anonymity [14] [13] if each record is indistinguishable from at least $k-1$ other records with respect to certain identifying attributes. In the context of location cloaking, k-anonymity guarantees that given a query, an attack based on the query location cannot identify the query source with probability larger than $1/k$ among other $k-1$ users. The location anonymizer removes the ID of the user and transforms his location by a k-anonymizing spatial region (k-ASR or ASR), which is an area that encloses the user that issued the query, and at least $k-1$ other users [7]. The cloaking based on k-anonymity has the following shortcomings:

1. The location point is cloaked to a region. This causes trouble for point based data mining algorithms such as k-means [9], Density Based Spatial Clustering of Applications with Noise (DBSCAN [4]), K-Nearest Neighbor (KNN [2]) and so on.
2. The result of cloaking based on k-anonymity could be a very large region even for a rather small k if some points are in a sparse region, which could sacrifice the accuracy of data mining results.
3. k-anonymity only protects the anonymity of the users, but may in fact disclose the location of the user, if the cloaked area is small.

Perturbation [11] [15] has a long history in statistical disclosure control due to its simplicity, efficiency and ability to preserve statistical information. The method proposed in this paper for privacy preserving data mining of location data falls in the category of perturbation. Possible transformation functions include scaling, rotating, translation, and their combinations [8]. All previous works on perturbation used the same transformation function for all points. This is not reasonable for nonuniform distributed points (almost all real life location data are nonuniform distribution). For example, one way is to perturb each point p by randomly moving it to another point p' inside a disk with radius r and center p. In this way, if the distance between two points is less than $2r$, after perturbation, those two points could change topological relationship (left point becomes right point and right point becomes left point). This could cause serious problem for point based data mining algorithms since the results of most of those algorithms depend on the relative topological relationships.

[3] `http://www.wmur.com/Police-Thieves-Robbed-Homes-Based-On-Facebook-Social-Media-Sites/-/9858568/11861116/-/139tmu4/-/index.html`

Delaunay triangulation has been widely used in computer graphics, GIS, motion planning and so on. The Delaunay triangulation of a discrete point set P corresponds to the dual graph of the Voronoi diagram for P [3]. The Voronoi diagram of n points is the partition of plane into n subdivisions (called cells) such that each cell contains one point (called site) and the closest site for all point in one cell is the site in that cell. The relationship between Delaunay triangulation and Voronoi diagram is shown in Figure 1: there is an edge in Delaunay triangulation between two sites if and only if the two cells corresponding to those two sites have common edge in Voronoi diagram (in other words, two cells are neighbors). Since the Voronoi diagram captures the topological relationships of points, the Delaunay triangulation also presents the neighborhood relationships of points. Intuitively, if we can keep the Delaunay triangulation unchanged after perturbation, the results of data mining algorithms based on neighborhood relationships of points will not change or have a very small change. That is the key idea for our perturbation method. On the other hand, we want to maximize the size of the perturbation region, i.e. the uncertainty of the user's exact location, for maximum privacy protection given the constraint of maintaining Delaunay triangulation. The formal definition of the problem we want to solve is as follows:

Problem 1. Given a set of n points $S = \{p_1, p_2, ..., p_n\}$, we want to compute a continuous region R_i (perturbation region) as large as possible for each point such that for any point $p'_i \in R_i$, the topology structure of the Delaunay triangulation of the new point set $S' = \{p'_1, p'_2, ..., p'_n\}$, denoted as $DT(S')$, is the same as that of the original point set S, denoted as $DT(S)$. We use $DT(S) \sim DT(S')$ to denote the topology structure of $DT(S')$ is the same as the topology structure of $DT(S)$.

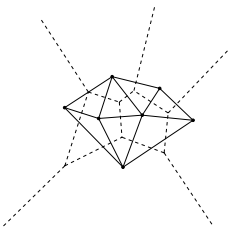


Fig. 1. Delaunay triangulation (solid lines) is the dual graph of Voronoi diagram (dashed lines)

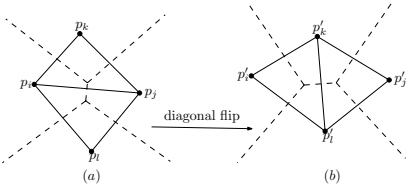


Fig. 2. Illustration of diagonal flip from (a) to (b)

The contributions of our paper are three-fold:

1. We present a novel perturbation method that guarantees the Delaunay triangulation of the point set does not change, which means we can guarantee the utility of privacy preserving data ming algorithms.
2. In our method, the perturbation region for all points are not uniform. Each point has its own distinctive perturbation region that depends on the surrounding situation of that point. Basically, if a point is located in a dense area, its perturbation region will be much smaller than the point in a sparse area. This is different from existing perturbation method that applies a uniform perturbation to each point.

3. Since our method is point perturbation, it can be used in any point based data mining algorithms.

The rest of this paper is organized as follows. Section 2 presents the algorithms of our perturbation method and the analysis of the algorithms. Section 3 covers comprehensive experiment evaluations. Finally, section 4 concludes and points out future directions for research.

2 Algorithms for Computing Perturbation Area R_i and Perturbated Point p'_i.

In this section, we explain how to compute the perturbation area R_i for each point p_i and how to pertubate p_i to p'_i. First we solve Problem 1 for the simple case of four points in section 2.1 and then give the solution for n points case in section 2.2. In section 2.3, we show how to compute p'_i based on R_i.

2.1 A Simple Case: Four Points

The reason why we choose four points is that four points is the simplest case that could have diagonal flip[4]. The analysis of $n > 4$ cases are based on the analysis of the simplest case.

Suppose we only have four points $S = \{p_i, p_j, p_k, p_l\}$ and they are in convex position. Without loss of generality, we assume $\overline{p_i p_j}$ is the edge (diagonal) of Delaunay triangulation (see figure 2(a)). Now the problem is to find a value r such that if $d(p_i, p'_i) \leq r$, $d(p_j, p'_j) \leq r$, $d(p_k, p'_k) \leq r$ and $d(p_l, p'_l) \leq r$, then $DT(S) \sim DT(S')$ where $S' = \{p'_i, p'_j, p'_k, p'_l\}$ and $d(.,.)$ is the Euclidean distance between two points. Note the continuous region R_i for each point p_i is the disk with the center p_i and radius r.

In order to let $DT(S) \sim DT(S')$, we need to prevent the diagonal flip case (see figure 2). That means we need to compute the largest r such that there is no diagonal flip (or to compute the smallest r that causes diagonal flip). Therefore, for two adjacent triangular faces $p_i p_j p_k$ and $p_i p_j p_l$, we want to compute the smallest r that makes the four points cocircular. This is equivalent to compute the circle that minimizes the maximum distance to the four points, or to compute the annulus of minimum width containing the points. It has been shown [5] [12], that the annulus of minimum width containing a set of points has two points on the inner circle and two points on the outer circle, interlacing angle-wise as seen from the center of the annulus (see figure 3). Therefore, the center will be the intersection of the bisectors of $\overline{p_i p_j}$ and $\overline{p_k p_l}$. Obviously the value of r can be computed in $O(1)$ time.

2.2 n Points

For n points set S, there is simple solution based on the above algorithm. First we can compute the $DT(S)$. For each pair of adjacent triangle faces, we compute one radius as

[4] Diagonal flip means the topology structure or neighborhood relationship is changed (see figure 2). For more details, please refer to the computation geometry book [3].

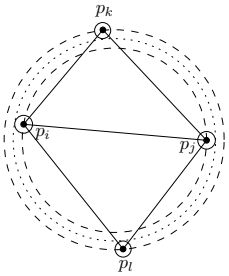


Fig. 3. The annulus of minimum width containing the four points

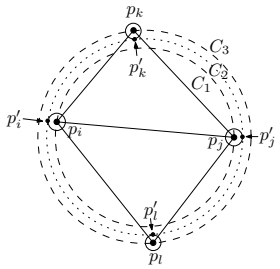


Fig. 4. The worst case scenario for four points

above such that no diagonal flip happens if four points are perturbated within the circle with that radius. Then for each point, there will be several radii. After computing all radii for all pairs of adjacent triangle faces, we get the smallest radius r. Since this is the smallest radius among all radii, no diagonal flip happens if all points are perturbated within the circle with radius r. In this way, all points have the uniform perturbation radius. But actually, except for those four points which produces the smallest radius r, all other points could be perturbated in a larger area. Notice for a point $p \in S$ with m incident edges in $DT(S)$, there are at most $2m$ ways such that p is a vertex of one pair of adjacent triangle faces, which means there are at most $2m$ radii for p. We can get the smallest radius from those $2m$ radii for the radius of p. In this way, we can still guarantee there is no diagonal flip.

Furthermore, we can improve the perturbation area for each point because in above analysis, we assume the worst case scenario. For example, in Figure 4, suppose the inner circle is C_1 and the outer circle is C_3 for the minimum width annulus containing four points p_i, p_j, p_k, p_l. Let the median axis of the annulus be the circle C_2. Let four disks with radius r_1 and center points p_i, p_j, p_k, p_l be $D_i^{r_1}, D_j^{r_1}, D_k^{r_1}, D_l^{r_1}$ and the tangent points between $D_i^{r_1}, D_j^{r_1}, D_k^{r_1}, D_l^{r_1}$ and C_2 be p'_i, p'_j, p'_k, p'_l. Then only when p_i, p_j, p_k, p_l move to p'_i, p'_j, p'_k, p'_l respectively, there is diagonal flip. In other words, p_i, p_j can not move out of C_2 and p_k, p_l can not move into C_2. Now if there is another disk $D_k^{r_2}$ for point p_k where $r_2 > r_1$ which is produced by other pair of adjacent triangles, then we know p_k can move inside $D_k^{r_2} \setminus D_2 = D_k^{r_2} \cap \overline{D_2}$ such that there is no diagonal flip (see Figure 5), where D_2 is the disk corresponding to C_2 and $\overline{D_2}$ is the area outside of D_2. Similarly, we can compute the larger perturbation area for p_i, p_j, p_l. There is only one small difference for p_i, p_j since p_i, p_j are in D_2 while p_k, p_l are outside of D_2. Therefore we need to use $D_i^{r_3} \cap D_2$ instead of $D_i^{r_3} \cap \overline{D_2}$ where r_3 is the larger radius than r_1 for p_i which is produced by other pair of adjacent triangles related to p_i.

For a point p, if there are m cocentric disks with center p and radii $r_1, r_2, ..., r_m$ and their corresponding media axis disks are $D_1, D_2, ..., D_m$, suppose $r_1 \leq r_2 \leq ... \leq r_m$ and the largest disk with radius r_m and center p is D, then the perturbation area R for p is

$$R = D \cap D_1(or\ \overline{D_1}) \cap D_2(or\ \overline{D_2}) \cap ... \cap D_m(or\ \overline{D_m})$$

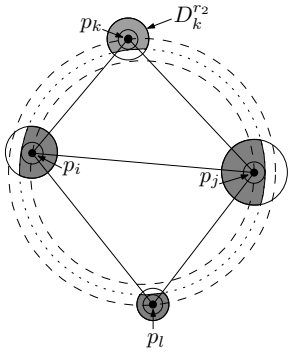


Fig. 5. The larger perturbation area (shaded area)

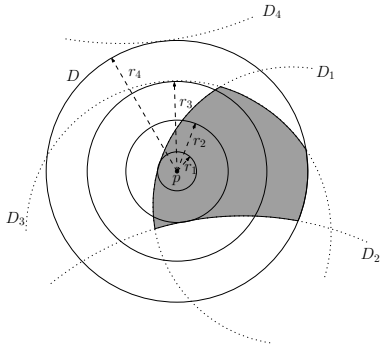


Fig. 6. The perturbation area (shaded area) for p

If $p \in D_i$, then we use D_i in above formula, otherwise $\overline{D_i}$ is used. See Figure 6 for example. The perturbation area is much larger than the smallest disk with radius R.

The algorithm for computing R_i is shown in Algorithm 1.

Algorithm 1. Algorithm for computing R_i

Require:
The original point set $S = p_1, p_2, ..., p_n$;
Ensure:
The cloaking regions $R_1, R_2, ..., R_n$;

Compute DT(S);
FOR each pair of adjacent triangles in DT(S) with vertices p_i, p_j, p_k, p_l and diagonal $\overline{p_i p_j}$
Compute the minimum width annulus containing p_i, p_j, p_k, p_l and the median axis disk D_{median};
Compute corresponding perturbation disks $D_i^r, D_j^r, D_k^r, D_l^r$ with the same radius r;
ENDFOR
FOR each point p_i
Sort the m cocentric perturbation disks around p_i according their radius and get the largest radius disk D;
Get the m median axis disks $D_1, D_2, ..., D_m$ corresponding to m cocentric perturbation disks;
Compute $R_i = D \cap D_1(or\ \overline{D_1}) \cap D_2(or\ \overline{D_2}) \cap ... \cap D_m(or\ \overline{D_m})$;
ENDFOR

Theorem 1. *For any point p, the perturbation area R is larger than* 0.

Proof. Since $R = D \cap D_1(or\ \overline{D_1}) \cap D_2(or\ \overline{D_2}) \cap ... \cap D_m(or\ \overline{D_m})$ $(i = 1, ..., m)$, and $p \in D$ and $p \in D_i(or\ \overline{D_i})$, then at least $p \in R$. Actually, R at least includes the smallest disk with radius r_1. □

Theorem 2. *The perturbation areas $R_1, R_2, ..., R_n$ for a set of n points $S = \{p_1, p_2, ..., p_n\}$ can be computed in $O(n^2)$ time.*

Proof. The Delaunay triangulation of S can be computed in $O(n \log n)$ time [3]. For each pair of adjacent faces in $DT(S)$, the perturbation disks and corresponding one medial axis disk for four points can be computed in $O(1)$ time. Since there are $O(n)$ pairs of adjacent faces, there are $O(n)$ perturbation disks and $O(n)$ medial axis disks.

Each R_i is one face of arrangement of $O(n)$ perturbation disks and $O(n)$ medial axis disks. The arrangement for $O(n)$ disks can be computed in $O(n \log n+k)$ expected time by using randomized incremental algorithm [6], where k is the number of intersection points in the arrangement. In worst case, $k = O(n^2)$. But in reality, k could be much smaller (even linear, see Figure 11(b)). □

Figure 7 shows the perturbation area for 12 points. Figure 8 shows the Delaunay triangulations for original 12 points (solid lines) and for perturbated 12 points (dashed lines).

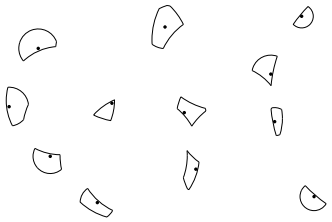


Fig. 7. The perturbation area for 12 points

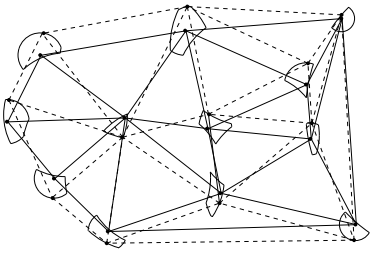


Fig. 8. The Delaunay triangulations for original 12 points (solid lines) and for perturbated 12 points (dashed lines)

2.3 Point Perturbation

After get perturbation region R_i, we need to perturbate p_i inside R_i for publication. In order to achieve maximum privacy protection, we perturbate p_i to the boundary of R_i as follows: choose a random angle $\theta \in [0, 2\pi]$, shoot a ray with angle θ and find the intersection point p_i' of this ray with boundary of R_i, and let p_i' be the perturbated point (see Figure 9).

3 Experimental Evaluation

3.1 Data Set

We use three datasets from University of Eastern Finland's clustering datasets[5]: Flame (240 points), R15 (600 points) and Jain (373 points) for testing our method's utility on KNN, K-means and DBSCAN respectively. The reason we choose those three algorithms is that KNN is the most popular neighborhood querying algorithm in database, K-means and DBSCAN are two popular clustering algorithms in data mining. The other important reason is that the results of those three algorithms all depends on the neighborhood relationship between points. We also generate 29 synthetic datasets with the number of points ranging from 100 to 100000 for testing the performance of our perturbation algorithm.

[5] `http://cs.joensuu.fi/sipu/datasets/`

3.2 Utility Results

The definition of utility varies for different privacy preserving data mining algorithms. In this paper, we make use of the two well-known criteria, i.e., the precision and recall rates [10]. We will give the definitions of precision and recall for privacy preserving KNN, K-means, DBSCAN in the following three subsections and also give the experimental results respectively.

Utility of Privacy Preserving KNN. For KNN, the precision and recall rate is defined as follows:

$$Precision_{KNN} = \frac{||Q(p_i) \cap Q(p_{i'})||}{||Q(p_{i'})||}$$

$$Recall_{KNN} = \frac{||Q(p_i) \cap Q(p_{i'})||}{||Q(p_i)||}$$

where $Q(p_i)$ is the set of points that are the K nearest neighbors of p_i, $p_{i'}$ is the perturbated point of p_i. Note $Precision_{KNN} = Recall_{KNN}$ since $||Q(p_{i'})|| = ||Q(p_i)||$ for fixed K. Therefore we only report precision for KNN. We perform 100 rounds of perturbations and query KNN on each point and get the average value of precision:

$$AP_{KNN} = \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} \frac{||Q(p_i) \cap Q(p_{i'}^j)||}{||Q(p_{i'}^j)||}}{m \times n}$$

where n is the number of query times (in this paper, we perform query on each point, therefore $n = 240$ for Flame dataset) and m is the number of rounds of perturbation ($m = 100$ in this paper), $p_{i'}^j$ is the perturbated point of p_i after the jth round of perturbation. Therefore AP_{KNN} is the average precision rate per query. We test $Precision_{KNN}$ for $K = 1, ..., 100$. For comparison we also perform the **uniform** purturbation with uniform distance r_{max} where

$$r_{max} = \max_i \{d(p_i, p_i') | p_i' \in \partial R_i\}$$

Figure 10 shows the precision rate of two perturbation methods for KNN algorithm. We can see precision rate based on Delaunay triangulation AP_{KNN}^{DT} is always better than that of uniform perturbation $AP_{KNN}^{Uniform}$. When $K < 10$, AP_{KNN}^{DT} is 20 percent higher than $AP_{KNN}^{Uniform}$. When K becomes larger, $AP_{KNN}^{DT} - AP_{KNN}^{Uniform}$ becomes smaller. $AP_{KNN}^{DT} - AP_{KNN}^{Uniform} = 3.93\%$ when $K = 100$. Also when K become larger, both $AP_{KNN}^{Uniform}$ and AP_{KNN}^{DT} seem to converge. This is reasonable since the K nearest neighbors may only change for the far away points (for example, the Kth nearest point) no matter how the points are perturbated.

Utility of Privacy Preserving K-means. For K-means, we adopt B-CUBED algorithm to compute the precision and recall rate of each point for clustering results:

$$Precision_{K-means}^{p_i} = \frac{||C_i \cap C_{i'}||}{||C_{i'}||}$$

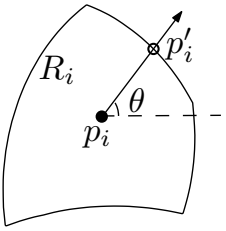


Fig. 9. Illustration of perturbation from p_i to p'_i

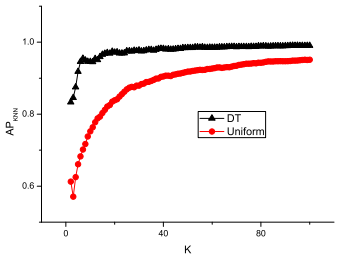


Fig. 10. The utility of privacy preserving KNN. x axis is K and y axis is AP_{KNN}. Red circle curve is for uniform perturbation and black triangle curve is for Delaunay triangulation based perturbation.

$$Recall^{p_i}_{K-means} = \frac{||C_i \cap C_{i'}||}{||C_i||}$$

where $p_i \in C_i, p'_i \in C_{i'}$ and p'_i is the perturbated point of p_i. 100 rounds of perturbations are performed. Therefore we can get the average value of precision and recall rates per point:

$$AP_{K-means} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{||C_i \cap C_i^j||}{||C_i^j||}}{m \times n}$$

$$AR_{K-means} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{||C_i \cap C_i^j||}{||C_i||}}{m \times n}$$

where n is the number of points and m is the number of rounds of perturbation, $p_i \in C_i, p^j_{i'} \in C^j_{i'}$ and $p^j_{i'}$ is the perturbated point of p_i after the jth round of perturbation. In this paper, we use R15 dataset for K-means algorithms. Since we already know there are 15 clusters, we set $k = 15$ and get the results as shown in Table 1.

Table 1. Average precision and recall rate for 15-means clustering on R15 dataset

	DT	Uniform
AP	0.99951371	0.99419382
AR	0.99951316	0.9942013

We can see precision and recall rate of privacy preserving 15-means for DT based perturbation are better than those for uniform perturbation with uniform radius r_{max}. Since the precision and recall rate for uniform perturbation are already very high, it's very difficult to achieve higher precision and recall rate for DT based perturbation.

Utility of Privacy Preserving DBSCAN. For DBSCAN, the utility $Precision_{DBSCAN}$ is defined as the same as $Precision_{K-means}$ since they are both clustering algorithms. The difference is that we don't need to set the number of cluster K in DBSCAN. For Jain dataset, there are two clusters for original dataset. DBSCAN requires two parameters: ε and the minimum number of points required to form a cluster ($minPts$). In our experiments, we set $\varepsilon = 2.4$ and $minPts = 20$ such that DBSCAN algorithm produces two clusters over original Jain dataset. The results for the utility of uniform perturbation and DT based perturbation are shown in Table 2.

Table 2. Average precision and recall rate for DBSCAN clustering on Jain dataset

	DT	Uniform
AP	1	0.76515634
AR	1	0.9349843

We can see the points in each cluster keep exactly same after DT based perturbation while that could change a lot after uniform perturbation with uniform radius r_{max}.

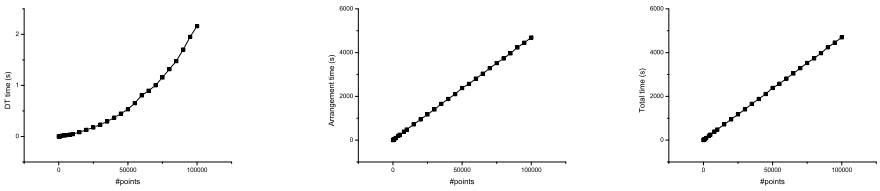
3.3 Privacy

We define the privacy for DT based perturbation as follows: $P = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} R_i^j}{m \times n}$ where R_i^j means the perturbation area for ith point after jth round of perturbation. Therefore P means the average perturbation area for one point. However, P depends on the whole area of point sets. In order to cancel out this effect, we define the privacy ratio P_r as follows: $P_r = P/Area(CH(S))$ where $CH(S)$ is the convex hull of point set S and $Area(CH(S))$ is the area of the convex hull of S. The experimental results of privacy for three datasets $R15, Jain, Flame$ are listed in Table 3.

Table 3. Privacy for DT based perturbation

	Jain	Flame	R15
P	0.0235319	0.0259528	0.000893764
$Area(CH(S))$	639.819	132.049	138.938
P_r	3.6779E-05	1.96539E-04	6.43283E-06

We can see the average perturbation areas for Jain and Flame datasets are similar while the average perturbation area for R15 dataset are much smaller which could be caused by the higher average neighboring point density for each point in R15 dataset. The other possible reason is there could be many degenerate cases in R15 dataset such that many four points groups are cocircular (or almost cocircular).



(a) (b) (c)

Fig. 11. x axis is the number of points and y axis is the running time of computing (a) Delaunay triangulations, (b) arrangement of perturbation areas, (c) total running time

3.4 Performance of Our Perturbation Algorithm

To test the performance of our perturbation algorithm, the 29 synthetic datasets are used with various number of points from 100 to 100000. We perform all experiments on our Thinkpad laptop with the following configurations: Ubuntu 13.04 operating system, 2 Gbyte of memory, 2.6GHz intel core i3 CPU. Since our perturbation algorithm mainly consists of two parts: computing Delaunay triangulation and computing the arrangement of perturbation areas, we give the results of running time for both parts and the total running time as well.

Figure 11(a) shows the running time for Delaunay triangulation. It almost fits the theoretical bound of $O(n \log n)$ running time.

Figure 11(b) shows the running time for computing the arrangement of perturbation areas. Although, theoretically the running time is $O(n^2)$ for the worst case, in practice, the running time is almost linear.

Figure 11(c) shows the total running time. Since the computation of perturbation areas is the dominant factor (for example, for 100000 points dataset, computing Delaunay triangulation only takes only around 2 seconds while computing the arrangement of perturbation areas takes 4690 seconds), the total running time is also linear. Therefore, in practice, our algorithm is very efficient.

4 Conclusions and Future Work

In this paper, we present a novel method to address the problem of location privacy. Our method achieves both high assurance privacy and good performance. Specifically, our method perturbates each point distinctively according to its own environment instead of uniform perturbation. In this way, the attackers are more difficult to guess original location. Furthermore our perturbation method guarantees that the Delaunay triangulation of point set does not change, which means we can guarantee the utility of privacy preserving data ming algorithms. We also develop a privacy model to analyze the degree of privacy protection and evaluate the utilities of our approach through three popular data mining algorithms. Extensive experiments show our method is effective and efficient for location based privacy preserving data mining.

Several promising directions for future work exist. First of all, the perturbation region or the perturbation distance for each point could be larger. We need to prove it theoretically and find the optimal values in terms of perturbation distance and area. Also we can apply our method on other data mining algorithms to see whether our method is a universal method for location based privacy preserving data mining algorithms. This is significant since many current methods are only good for one kind of data mining algorithm.

References

1. Buchin, K., Buchin, M., Gudmundsson, J., Löffler, M., Luo, J.: Detecting commuting patterns by clustering subtrajectories. Int. J. Comput. Geometry Appl. 21(3), 253–282 (2011)
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
3. De Berg, M., Cheong, O., Van Kreveld, M., Overmars, M.: Computational geometry: algorithms and applications. Springer-Verlag New York Incorporated (2008)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
5. Garcia-Lopez, J., Ramos, P.A.: Fitting a set of points by a circle. In: Symposium on Computational Geometry, pp. 139–146 (1997)
6. Goodman, J., O'Rourke, J.: Handbook of discrete and computational geometry. Chapman & Hall/CRC (2004)
7. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Trans. Knowl. Data Eng. 19(12), 1719–1733 (2007)
8. Lin, D., Bertino, E., Cheng, R., Prabhakar, S.: Location privacy in moving-object environments. Transactions on Data Privacy 2(1), 21–46 (2009)
9. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, California, USA, vol. 1, p. 14 (1967)
10. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
11. Rastogi, V., Hong, S., Suciu, D.: The boundary between privacy and utility in data publishing. In: VLDB, pp. 531–542 (2007)
12. Rivlin, T.: Approximation by circles. Computing 21(2), 93–104 (1979)
13. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 571–588 (2002)
14. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 557–570 (2002)
15. Xiao, X., Tao, Y., Chen, M.: Optimal random perturbation at multiple privacy levels. PVLDB 2(1), 814–825 (2009)