Adjustable Location Privacy-Preserving Nearest Neighbor Query Method

Linfeng Xie[1,2](✉), Zhigang Feng[1,2], Cong Ji[1,2], and Yongjin Zhu[1,2]

[1] Jiangsu Frontier Electric Technology Co., LTD,
Nanjing 211189, Jiangsu, China
15905166617@139.com
[2] School of Computer Science and Engineering,
Southeast University, Nanjing 211189, China

Abstract. Location-based services facilitate the daily life of the people, nevertheless, they also bring about the problem of privacy preserving. Privacy preserving methods without anonymity server, for example, Coprivacy, attract increasing concerning from researchers for their simple and reliable structure and the avoidance of high cost of communication and computing resulting from the using of cloaking area. The drawbacks of Coprivacy are the high cost of communication and computing and the uncontrollability during query period. A feedback based incremental nearest neighbor query method (FINN) is propose to solve the problem. The user sends feedback to the server according to the query, and the server chooses POIs to send to the user according to the feedback. Theoretical analysis and experimental results show that FINN can improve the performance of the system significantly while ensures user's anonymous requirements.

Keywords: Location-based services · Location privacy preserving · Feedback information · Incremental nearest neighbor query

1 Introduction

The rapid development of mobile communication and spatial positioning technology has promoted the rise of location-based services (LBS). K nearest neighbor query is an important query service of location services. It refers to finding K target objects (POI, point of interest) nearest to a query's current location, such as finding K restaurants or gas stations nearest to the query. This service requires the querier to provide the service provider with its exact location to obtain the query results. Real-time location information contains user behavior patterns. With the increasing attention to individual information security, the security of user location has been paid more and more attention. Sending location to service providers may lead to the leakage of privacy information such as identity and behavior patterns of individual users. How to realize k-nearest neighbor query without revealing the location privacy of individual users has become a hot topic in the field of privacy-sensitive location services.

At present, the main idea of location privacy preserving query is to hide the location of the inquirer and submit the hidden location and query request to the LBS server.

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 467–479, 2019.
https://doi.org/10.1007/978-3-030-30952-7_46

The server completes the query processing of the hidden location information, and feeds the query results back to the initiator for filtering the target results. The main hiding technologies include spatial obfuscation [10, 11], data transformation [12–14], location perturbation and Private Information Retrieval [5, 13, 14]. Spatial obfuscation enlarges the location of the query to a generalized region containing the location and submits it to the LBS server. The query results are screened out by the query or trusted third party from the returned candidate solutions. Data transformation achieves the privacy protection of the query location by transforming the location of the query and the target object into another data space for query processing. In location disturbance, the query is directed to L. BS servers submit query requests for specific false locations until they return the results satisfying their query accuracy and privacy security requirements. Most PIR technologies are based on the quadratic congruence problem, and use location keys to provide strong privacy protection intensity, but there are problems such as large computational load and high communication cost. These methods have different emphasis on privacy protection intensity, query accuracy and query processing performance. The specific comparison is shown in Table 1.

Table 1. Comparison of location privacy protection query technologies

	Spatial cloaking	Data transformation	Location perturbation	PIR
Protecting strength	Controllable	Controllable	Uncontrollable	Controllable
Query cost	Higher	Higher	High	High
Query accuracy	Exact	Not exact	Exact	Exact
Dependence on the trusted third party	Yes	Yes	No	Yes

From the perspective of query mode architecture, spatial cloaking and data transformation technology mostly rely on trusted third party (acting as anonymizing servers) to participate in query processing in online or offline mode [1–4]. There are difficulties in implementation and the trusted third party is inclined to be the bottlenecks of the query system. Location perturbation technology uses the mode of direct interaction between query client and LBS server to realize privacy protection query, which has the advantage of not depending on trusted third party. The false point method is proposed in ref [6] and ref [7], which uses the false position point instead of the user's real location to initiate the query. SpaceTwist method is proposed in ref [8], in which the client specifies a location as the anchor node, and the server performs incremental nearest neighbor query on the anchor node until the critical condition is satisfied. Privacy protection query method Coprivacy [9] is proposed based on SpaceTwist. It implements K anonymity by querying the user to form an anonymous group, and can support privacy preserving query satisfying K anonymity.

Existing location perturbation based privacy-preserving query methods, just as the representative method SpaceTwist of them, have the following shortcomings.

(1) The number of iteration queries between the query client and the LBS server is uncontrollable. It results in large computation and communication overhead.
(2) The query client lacks the regulation mechanism for query efficiency and privacy security, which makes it difficult to meet the personalized regulation requirements of query efficiency and privacy security.

To solve the above problems, a feedback-based incremental nearest neighbor query method (FINN) is proposed. Based on location perturbation, the query client provides feedback information to the LBS server in each iteration of query between the query client and the LBS server to realize the existence of the LBS server. Guided query processing reduces the communication and computing overhead of location perturbation method, and realizes the adjustable query efficiency and location privacy protection effect.

The main contributions of this paper list as follows:

(1) A feedback angle-based control mechanism for query efficiency and privacy protection intensity is proposed to realize the dynamic and controllable adjustment of the query processing process of the server.
(2) By using feedback angle control mechanism, a nearest neighbor query method based on location perturbation to protect location privacy, FINN, is proposed to dynamically adjust query processing efficiency and privacy protection intensity.
(3) FINN method is realized, and experiments are devised to verify the effectiveness of the solution.

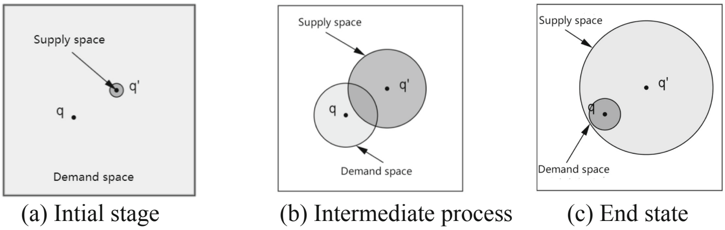
The organizational structure of the paper is as follows: Sect. 2 summarizes the related work and expounds the existing problems of SpaceTwist method. Section 3 gives the basic idea of FINN method and the calculation method of key parameter feedback angle; Sect. 4 elaborates the execution process of FINN method on client and LBS server respectively. Sections 5 and 6 carries out theoretical analysis and experiment on FINN method, respectively. Finally, it summarizes the full text and looks forward to the next step.

2 Related Work and Problem Description

Trusted third party is easy to become the bottleneck of system performance and the target of attacker's concentrated attack, so the location perturbation method which does not rely on trusted third party has its unique advantages.

2.1 SpaceTwist Method

SpaceTwist algorithm uses the query client to iteratively initiate queries about false locations (anchors). By analyzing the geometric relationship between the POI of the false locations feedback by the LBS server and the real locations of the queriers, it decides whether to continue the query process until the target query results are obtained. The algorithm introduces the concepts of query demand space γ and supply space τ. The demand space is a circular region with the center of the query's real location and the radius of its distance from current k nearest neighbor POI, and the supply space is a circular region with the center of anchor and the radius of its distance from the nearest return POI. In order to reduce communication overhead, several POIs are transmitted in one message. The querier maintains a heap W_k record of the k-nearest neighbor currently known by the querier. As shown in Fig. 1, q represents the querier and q′ represents the anchor. The specific process is: (1) the demand space and the supply space are initialized to ∞ and 0, respectively; (2) the server continuously makes incremental nearest neighbor queries on the anchor and sends the query results to the querier; (3) when the query keeps judging whether the supply space fully contains the demand space, if execution (2) is not included. The query ends.



(a) Intial stage (b) Intermediate process (c) End state

Fig. 1. Demand space and supply space in SpaceTwist

2.2 Problem Description

Location perturbation-based method SpaceTwist has the disadvantage of high communication and computational overhead, and the intensity of location privacy protection and query efficiency can not be adjusted. How to dynamically adjust the intensity of privacy protection and query efficiency, and reduce the computational and communication overhead caused by unpredictable pseudo-location iteration queries are the main problems to be solved.

The high cost of query processing and the uncontrollable intensity of privacy protection in location perturbation-based query methods originate from the uncontrollable number of iteration query rounds and process between the query client and the LBS server. Considering that the query client can provide the LBS server with effective auxiliary information about iteration rounds, the unsupervised iteration can be changed into the guided iteration, and the dynamic strength of privacy protection and query efficiency can be realized.

3 Query Control Mechanism Based on the Feedback Angle

3.1 Basic Idea

In the SpaceTwist method, the LBS server iteratively queries the anchor until the critical condition is satisfied. Multiple rounds of communication are needed between the query and the LBS server, and the query needs multiple rounds of computation for the demand space and the supply space.

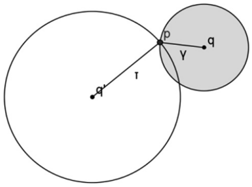


Fig. 2. Intermediate stage of SpaceTwist

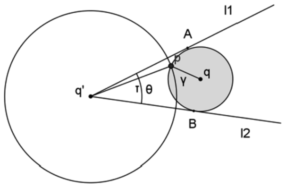


Fig. 3. Illustration of feedback angel

As shown in Fig. 2, the gray area represents the current requirement space. Assuming that the W_k is full at this time, that is, the current k-nearest neighbor has been found, then the real k-nearest neighbor must be in this grey area. Because the server does not know this information, it will continue to iterate incremental queries on the anchor. If the distribution of POI adjacent to the anchor is uneven and the POI distributed outside the grey area is more intensive, more communication and computing overhead will be generated. In addition, in this process, queriers can not dynamically adjust their privacy protection intensity and query efficiency to meet their personalized needs. The analysis shows that the reason for the above situation is that the server only knows the end condition of the query and has no additional information to guide the query processing, which makes the query process uncontrollable.

Definition 1 Feedback angle θ. Feedback angle is the angle between the two tangent lines tangent to the outermost circle of the demand space through anchor node.

As shown in Fig. 3, tangent lines l_1 and l_2 are made to the outermost circle of the demand space through anchor point q′, intersecting with the outermost circle of the demand space at points A and B, A q′B is the feedback angle.

The real k nearest neighbor of the query is in the grey area, and in the sector area where the anchor node is the vertex and the feedback angle is the angle. If the query node feeds back the information about this angle to the server, the server filters the query results that will be sent to the user according to the feedback angle, and only sends the query results that satisfy the specific conditions, it will reduce the communication overhead between the LBS server and the query, and thus reduce the computational overhead of the query. At the same time, the larger the feedback angle, the less likely the attacker will infer the real location of the query. Therefore, the query efficiency and privacy protection intensity can also be adjusted by adjusting the feedback angle.

3.2 Feedback Angle θ Computing

Without losing generality, the whole region space is assumed to be a rectangular region, with the point at the lower left corner of the region as the origin, the lower x-axis, and the left Y-axis as the coordinate system. For convenience, coordinates represent the longitude and latitude of nodes.

As shown in Fig. 4, there are two tangent points A and B. The anchor node q′ is taken as the end point and qM is made along the positive direction of the x-axis. The qM is parallel to the x-axis. θ_1 and θ_2 are the angles of q′M rotating counterclockwise to q′A and q′B, respectively. θ can be divided into two cases as shown in Fig. 4. The range of θ is $[\theta_1, \theta_2]$ and $[0, \theta_1] \cup [\theta_2, 2\pi]$.

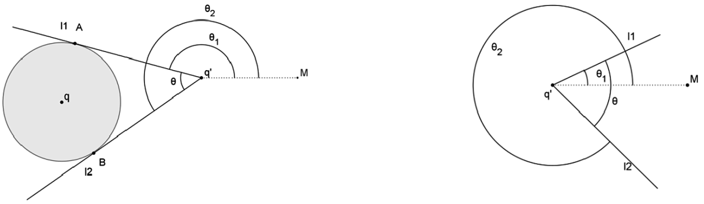


Fig. 4. The range of feedback angel θ

Assume that the attacker grasps the following information: (1) all POI locations; (2) anchor locations; (3) processing algorithm on client side; (4) query processing algorithm on server-side. The attacker can get information about the vertex and angle of the sector area, as well as the fact that the inquirer located on the angle bisector of the vertex. In order to ensure the location privacy of the query, the range of feedback angle θ is extended to both sides with scale α and β, that is, the range is changed to $[(1-\alpha)\theta_1, (1+\beta)\theta_2]$ or $[0, \theta_1(1+\alpha)] \cup [\theta_2(1-\beta), 2\pi]$.

The calculating process of θ is as follows: (1) make tangent line along the anchor nodes to the outermost circle of the demand space, and get tangent points A and B. (2) calculate the angles θ_1 and θ_2 between q′A, q′B and qM, and get range of θ (3) The specified range of θ is extended to both sides.

4 Feedback Angel Based Location Privacy Preserving Nearest Neighbor Query Method FINN

4.1 Client-Side Processing

The client is mainly responsible for judging the end of the query and calculating the feedback angle.

Definition 2 Query request Q. Queries initiated by queriers can be expressed as Q = {l, t, con}. Among them, l = {x, y} represents the location of the anchor point, x denotes the longitude of the anchor, y is the latitude of the anchor; t is the query time; con is query content.

The main process of FINN algorithm in client side include four steps: (1) sending query request Q to LBS server; (2) calculating the distance dist (p, q) between querier q and p for each POI point p sent by server, and putting <dist (p, q), p> into w_k until the cumulative number reaches k. At this time, it corresponds that the current known k nearest neighbor has been found, and the real k nearest neighbor of the querier just locates in the current demand space; (3) judging whether the anchor is outside the demand space, if dist (p, q) < γ, then calculating θ and send it to the server; (4) when the critical condition is satisfied, that is, when the supply space contains the demand space, end the query process. During the whole process, the client continues updating the demand space and supply space and updates w_k. The specific implementation process is shown in Algorithm 1.

Algorithm1. FINN_Client
Input：q，k，**α**，β
Output: k nearest neighbors of the querier

```
$\tau \leftarrow 0$;
$\gamma \leftarrow \infty$;
$W_{k\leftarrow}$ k pairs of <NULL, $\infty$>;
querier qsubmit the query request to LBS server;
while $\gamma$+dist(p,q)> $\tau$;
  receive message S from the server;
  for each p in S
    while $|\mathbf{w_k}|$<k
      $\mathbf{w_k} \leftarrow$ < dist(p,q)，p>;
    end while;
     while dist(q',q)<$\gamma$
      if dist(p,q)< $\boldsymbol{\gamma}$
        $\tau \leftarrow$dist($\mathbf{p_i}$,q);
        heap tuple$\leftarrow$<p, dist(p,q)>;
      end if;
     end while;
    Generate_$\theta$;
    send $\theta$ to the server;
     if dist(p,q)< $\gamma$
       $\tau \leftarrow$dist($p_i$,q);
       heap tuple$\leftarrow$<p, dist(p,q)>;
     end if;
    end for;
  end while
  Send end query request to server
  return $W_k$;
```

4.2 Server-Side Processing

The server-side processing consists of four steps: (1) receiving anchor node q′; (2) sending back the query results before receiving θ; (2) after receiving θ, the server calculates the ∠P q′M shown in Fig. 4 and sends p to the querier if ∠P q′M ∈ range of θ; (4) receiving the request of the querier to ending the query. The specific implementation process is shown in Algorithm 2.

Algorithm2. FINN_Server
Input：the anchor q'，feedback angel θ
Output：query result p

```
receive query request;
while Not receiving request of ending the query
  while Not receiving feedback information from the querier
    send query result p to the querier
  end while; //receive the feedback information
  calculate $\angle \mathbf{pq'M}$;
  if $\angle \mathbf{pq'M} \in$ the range of $\theta$
   send query result p to the querier
end while.
```

5 Performance Analysis

This section analyses the privacy security, accuracy and query efficiency of FINN method. The query efficiency mainly considers the communication overhead and query client computing overhead.

5.1 Privacy Security Analysis

Given the execution process of FINN method, the attacker can filter the querier meeting the specific conditions. Without additional information, the number of the query can indicate the security of privacy.

Definition 3 Privacy Area Ψ [8]. The set of all possible users that an attacker can infer.

For a query, the angle information sent to the LBS server is recorded as θ. Assuming that the number of messages sent by the server is m and the capacity of the message is c, the range of Ψ can be limited by the following formula:

$$\begin{cases} dist(q',p) + \underset{1 \leq i \leq (m-1)c}{\overset{k}{\min}} dist(q,p_i) > dist(q',p_{(m-1)c}) \\ dist(q',p) + \underset{1 \leq i \leq mc}{\overset{k}{\min}} dist(q,p_i) \leq dist(q',pmc) \\ \forall p \in \Psi, \angle \mathrm{pq}'\ \mathrm{M} \in \theta \end{cases} \tag{1}$$

Property 1. FINN method has the ability of regulating privacy security it can provide.

Proof. The angle the user sends to the server is marked as θ. The attacker infers that the real location of the user is located in the light gray sector S with the anchor as the origin and theta as the angle and the radius of the distance between the final query result of the server and the anchor as the radius, as shown in Fig. 5. Note that the area of the light grey ring is L. Assume that the user initiates two queries, and the angles are θ_1 and θ_2, respectively, and $\theta_1 > \theta_2$, hence $S_1 = L_1 \cdot \theta_1$, $S_2 = L_2 \cdot \theta_2$. Considering the arbitrariness of node distribution and the communication mechanism between querier and LBS server, $\mathrm{P}(L_1 > L_2) = 1/2$, and L and θ are independent of each other. It can be deduced that:

$$\mathrm{P}(S_1 > S_2) > P(L_1 > L_2 \wedge \theta_1 > \theta_2) = \mathrm{P}(L_1 > L_2) \cdot P(\theta_1 > \theta_2) = \mathrm{P}(L_1 > L_2) = 1/2$$

It demonstrates that the larger the feedback angle, the larger the privacy area and the stronger the privacy security. That is to say, FINN has a regulatory effect on privacy security.

Property 2. If POI and users in the query space are uniformly distributed, then the attacker guesses that the probability of the user's real location is $\frac{\mathrm{N}}{\mathrm{M}\left(\mathrm{c}-\sqrt{\frac{\mathrm{kc}_\theta}{2\pi \mathrm{m}}}\right)}$. Among them, N is the number of POI, M denotes the number of users, k is the number of nearest neighbor POI that users need to get.

Proof. For $\frac{\pi\gamma^2}{S} = \frac{k}{N}$, radius γ of final demand space is $\sqrt{\frac{Sk}{\pi N}}$. Similarly, when the query node receives (m–1) and m messages, the corresponding supply space radius $\tau_{m-1} = \sqrt{\frac{(m-1)cS}{\pi N}\cdot\frac{2\pi}{\theta}}$, $\tau_m = \sqrt{\frac{mcS}{\pi N}\cdot\frac{2\pi}{\theta}}$. The area of the area is denoted as S. Number of users in the privacy area is: $\pi\left[(\tau_m - \gamma)^2 - (\tau_{m-1} - \gamma)^2\right]\cdot \frac{M}{S}\cdot\frac{\theta}{2\pi} =$

$$\left[c+\theta\left(\sqrt{\frac{2(m-1)ck}{\pi\theta}}-\sqrt{\frac{2mck}{\pi\theta}}\right)\right]\cdot\frac{M}{N} = \frac{M}{N}\left(\beta - \frac{2ck}{\pi(\sqrt{\frac{2(m-1)ck}{\pi\theta}}+\sqrt{\frac{2mck}{\pi\theta}})}\right) \approx$$

$\frac{M}{N}\left(\beta-\sqrt{\frac{kc\theta}{2\pi m}}\right)$, the probability that the real location can be inferred is $\frac{N}{M\left(\beta-\sqrt{\frac{kc\theta}{2\pi m}}\right)}$.

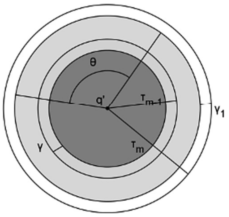


Fig. 5. Privacy area of the querier

5.2 Accuracy and Query Cost Analysis

FINN algorithm makes tangent line along the requirement space, and the effective query area of LBS server (the sector area with anchor q′ as the vertex and θ acting as circle angle) contains the requirement space, so the LBS server returns to the user POI containing the real k nearest neighbor of the querier, so FINN algorithm obtains accurate results.

The LBS server filters the query results according to the feedback angle and returns them to the user. So the communication cost of FINN is less than that of SpaceTwist, and the smaller the feedback angle is, the smaller the communication cost is.

Queriers usually acquire LBS services from mobile terminals such as smart phones. Although the performance of mobile terminals has been greatly improved, it is still the bottleneck of system computing performance compared with servers. Therefore, this paper mainly analyses the computational overhead of queriers. Compared with SpaceTwist method, FINN needs to calculate the location of anchor and requirement space in the early stage of query, but its computation amount is less than that of filtering POI in the later stage of query, so the computation cost of FINN is less than that of

SpaceTwist, and the smaller the feedback angle is, the smaller the computation cost of querier is.

6 Experimental Analysis

The experiment mainly investigates two aspects: one is to compare and analyze the differences between FINN method and SpaceTwist method in terms of query time and communication cost from the number of neighbors and POI nodes that queriers need to find; the other is to analyze the regulation of feedback angle theta on the privacy security and query performance of FINN. Response time refers to the time when the querier initiates the query to the LBS server to obtain the final query result. Traffic refers to the number of messages sent by the LBS server to the query client.

6.1 Experimental Environment

The algorithm is implemented in Java language. The experimental hardware environment is 2.9 GHz processor, 4G memory, and the operating system is Windows 7. The data set is generated by Thomas Brinkhoff Road Network Generator [15], which is widely recognized by the industry. It is based on the traffic network data of Aldenburg City, Germany. The user can define the data set attributes by himself. The area of the data set is 23.57 km × 26.92 km. The default experimental parameters are shown in Table 2.

Table 2. Default parameters of experiments

Parameters	Default value
Number of queriers	10000
K(Number of target POIs)	10
Number or POIs	100000
Feedback angle enlarging range	10%

The maximum transmission unit of network between user and server is 576 bytes, the header of message is 40 bytes, the length of each communication message is 8 bytes, and the size of message capacity is (576–40)8 = 67.

6.2 Experimental Result

6.2.1 Comparison of FINN and SpaceTwist Algorithms

Figures 6 and 7 demonstrates variation trend of communication traffic and response time. It can be seen the communication overhead and response time increase with increasing K. This is because the demand space expands with the increase of query demand, and the server needs to query more POIs to make the supply space cover the demand space. In addition, the communication overhead and response time of FINN method are better than those of SpaceTwist method. The larger the number of nearest neighbors, the more obvious the advantage of FINN method shows. The reason lies in that only those POIs located within specific angle range need sending from server to query client in FINN, which reduces the communication overhead and the computing overhead at client side.

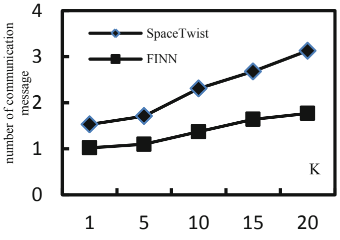


Fig. 6. Communication cost vs. varying K

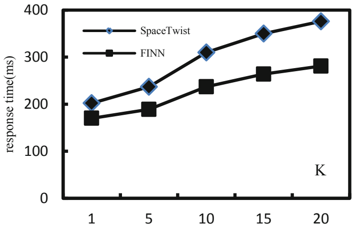


Fig. 7. Response time vs. varying K

Figures 8 and 9 show communication cost and response time varying trends with increasing K. It is obviously that the communication overhead and response time increase with increasing K. The reason is with querier's demand space not changed, higher POI density leads to more POIs in the supply space. It is noted that the communication overhead and response time of FINN are better than that of SpaceTwist, and the advantage increases sharply with larger K.

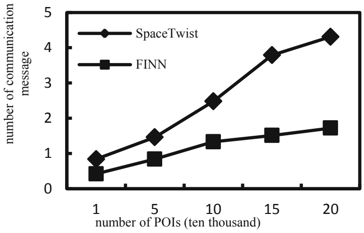


Fig. 8. Communication cost vs. number of POIs

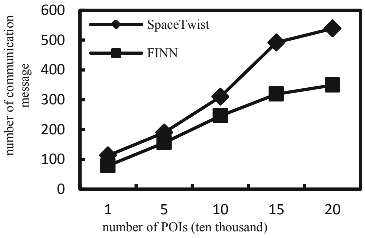


Fig. 9. Response time vs. number of POIs

The experimental results show that FINN is suitable for different POI densities and different K, and has good scalability. In addition, compared with SpaceTwist, FINN significantly improves query performance, and the extent of improvement increases with K.

6.2.2 Controllability of Feedback Angle to FINN

As shown in Figs. 10 and 11 communication cost, as well as response time, increase with enlarging feedback angel. This is because with feedback angle increases, the LBS server needs to find and process more POIs, and the communication and computing overhead will increase accordingly.

Figure 12 represents the impact of feedback angle on privacy security, the privacy preserving effect is measured by the number of users in the privacy area Ψ. The larger the feedback angle is, the more users locate inside the privacy area, and the smaller the probability that the attacker infers that the real query users are, the stronger the privacy security is.

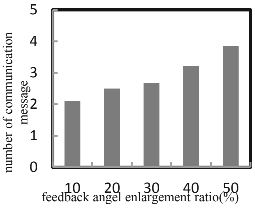


Fig. 10. Communication cost vs. feedback angel

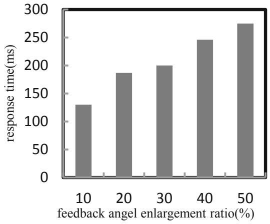


Fig. 11. Response time vs. feedback angel

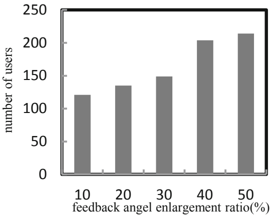


Fig. 12. Privacy protection vs. feedback angel

7 Conclusion

Aiming at the problem of high query cost, as well as uncontrollable query performance and privacy protection strength, in location perturbation-based privacy-preserving nearest neighbor queries, a feedback angle-based privacy-preserving nearest neighbor query method FINN is proposed. By sending feedback angle to the LBS server, the querier can provide guidance information for the subsequent query processing of the LBS server. It realizes the regulation of iteration rounds and location privacy protection intensity in query processing process, and improves query processing efficiency. The next step is to consider applying the proposed method to continuous location privacy preserving nearest neighbor query in road network environmental.

References

1. 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)
2. Um, J.-H., Kim, H.-D., Chang, J.-W.: An advanced cloaking algorithm using Hilbert curves for anonymous location based service. In: Proceedings of 2010 IEEE Second International Conference on Social Computing, pp. 1093–1098 (2010)
3. Hossain, A.-A., Hossain, A., Yoo, H.-K., Chang, J.-W.: H-star: Hilbert-order based star network expansion cloaking algorithm in road networks. In: Proceedings of IEEE 14th International Conference on Computational Science and Engineering (CSE), pp. 81–88, August 2011
4. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42 (2003)

5. Wu, J., Ni, W., Zhang, S.: Generalization based privacy-preserving provenance publishing. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 287–299. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_27
6. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 177–189 (2004)
7. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Pervasive Services, Proceedings of International Conference, pp. 88–97 (2005)
8. Yiu, M.L., Jensen, C.S., Huang, X.G., Lu, H.: SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: IEEE 24th International Conference on Data Engineering, pp. 366–375 (2008)
9. Huang, Y., Huo, Z., Meng, X.: CoPrivacy: a collaborative location privacy-preserving method without cloaking region. Chin. J. Comput. **34**(10), 1975–1985 (2001). (in Chinese)
10. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans. Mobile Comput. **7**(1), 1–18 (2008)
11. Chow, C.Y., Mokbel, M.F., Aref, W.G.: Casper*: query processing for location services without compromising privacy. ACM Trans. Database Syst. **34**(4), 1–45 (2009)
12. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 239–257. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73540-3_14
13. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest neighbor search with strong location privacy. Proc. VLDB Endow. **3**(1–2), 619–629 (2010)
14. Paulet, R., Kaosar, M.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. In: Kementsietsidis, A., Salles, M.A.V. (eds.) Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE 2012), pp. 44–53. IEEE Computer Society, Los Alamitos (2012)
15. Brinkhoff, T.: A framework for generating network based moving objects. GeoInformatica **6**(2), 153–180 (2000)