Evaluating k Nearest Neighbor Query on Road Networks with no Information Leakage

Lu Wang, Ruxia Ma, and Xiaofeng Meng(✉)

Information School, Renmin University of China, Beijing, China
{luwang,maruxia,xfmeng}@ruc.edu.cn

Abstract. The development of positioning technologies and pervasiveness of mobile devices make an upsurge of interest in location based services (LBS). The k nearest neighbor(kNN) query in road networks is an important query type in LBS and has many real life applications, such as map service. However, such query requires the client to disclose sensitive location information to the LBS. The only existing method for privacy-preserving kNN query adopts the cloaking-region paradigm, which blurs the location into a spatial region. However, the LBS can still deduce some information (albeit not exact) about the location. In this paper, we aim at strong privacy wherein the LBS learns nothing about the query location. To this end, we employ private information retrivial (PIR) technique, which accesses data pages anonymously from a database. Based on PIR, we propose a secure query processing framework together with flexible query plan for arbitrary kNN query. To the best of our knowledge, this is the first research that preserves strong location privacy for network kNN query. Extensive experiments under real world and synthetic datasets demonstrate the practicality of our approach.

Keywords: Location privacy · Private information retrieval · kNN query · Spatial networks

1 Introduction

With the popularity of mobile devices and development of the positioning technologies, location based service(LBS) is becoming more and more popular. To provide users with location based service, LBS system (e.g., Map Quest and Google Maps for mobile users) has been widely deployed by mobile users. The nearest neighbor queries in LBS occupy an extremely important position. For example, client traveling on the road may want to get the nearest gas station, or tourist may hope to learn the nearest restaurant from his current location. LBS

This research was partially supported by the grants from the Natural Science Foundation of China (No. 61379050, 91224008); the National 863 High-tech Program (No. 2013AA013204); Specialized Research Fund for the Doctoral Program of Higher Education(No. 20130004130001); the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University(No. 11XNL010).

J. Wang et al. (Eds.): WISE 2015, Part I, LNCS 9418, pp. 508–521, 2015.
DOI: 10.1007/978-3-319-26190-4_34

gives us more convenience, however, also causes sensitive privacy problems. Once the client requests a query, he must submit his location to LBS, which leads to the leakage of location privacy, even personal information such as health status, economic conditions, shopping habits, etc. [25].

Therefore, there exist lots of approaches for privacy-aware kNN query [1–4]. However, these works only consider Eculidean distance rather than road network distance. The only existing work of privacy-aware kNN query under road network follows the location obfuscation approach [5] which blurs client's exact location into a cloaked region and computes network k nearest neighbors by network voronoi diagram [22]. However, the method reveals certain location information of client to the LBS.

To guarantee strong location privacy, a promising cryptography tool is private information retrieval (PIR) [11]. PIR allows a data item (e.g., a disk page) to be retrieved from a server without leaving any clue of the item being retrieved. PIR was considered to be resource-intensive, but thanks to the recent progress in cryptography, practical software or hardware PIR solutions have been proposed [14]. Since then it has been successfully applied to spatial queries, such as kNN, BRNN and shortest path search [7,9,13].

In this paper, our goal is to investigate privacy-preserving kNN query on road network without the LBS inferring any information about the query. To this end, we adopt practical PIR techniques that retrieve a single data page as the building block. The challenges of a PIR-based kNN solution lie in the following aspects: (1) although PIR guarantees secure access of a single page from the server, the variation of the number of page accesses from different queries may reveal information about the query point. Further, when user desires to propose queries with varied k, our processing must be safe for arbitrary k, which makes the problem more challenging. (2) as the database contains voluminous points, directly applying PIR for the kNN query on road network is inefficient, thus calling for an integration with spatial index. To address these challenges, we propose a PIR-based kNN query processing framework that guarantees strong privacy. Concretely, we design index structure and deduce query plans for arbitrary k, which means adversary cannot deduce any information from arbitrary query. To summarize, we have three main contributions as follows:

(1) To the best knowledge, this is the first research evaluating k nearest neighbor query on road network with no information leakage.
(2) We deduce the fixed query plans for arbitrary fixed k and thus guarantees the strong privacy for arbitrary k nearest neighbor query on road network.
(3) We conduct extensive experiments under real-world and synthetic datasets, which shows our proposed approach is practical.

The rest of the paper is arranged as follows. Related works are surveyed in Sect. 2. In Sect. 3, we define our security model and prove its security. We then present our solutions for the PIR-based kNN query processing in Sect. 4. The solutions are evaluated by experiments in Sect. 5. Section 6 concludes this paper.

2 Related Work

In this section, we review related works in the following two areas: (1) priavcy-aware k nearest neighbor query on road network and (2) the application of PIR based approaches on spatial query.

2.1 Privacy-Preservation for *k*NN on Road Network

There are several existing network nearest neighbor query processing methods in the literature, such as the network expansion based methods [19,20], solution based methods [22–24] and hierarchical road networks based methods [21]. The classic method without privacy is the network voronoi nearest neighbor based solution which is proposed by Kolahdouzan [22]. It utilizes network voronoi diagram to partition the network into cells to reduce computation cost and communication cost. The only existing work of private network NN query based on spatial network follows the location obfuscation approach [5] which blurs client's exact location into a cloaked region and computes network k nearest neighbors by network voronoi diagram. However, the method reveals certain location information of client to the LBS.

So far, PIR technology is the only tool to guarantee strong privacy which means server cannot deduce any information about the query. There has been no works on applying PIR-based method to network nearest neighbor query to provide strong privacy guarantee. As the hardware PIR based method requires different queries execute the same query plan which implies that every query incur the same processing cost, the existing methods above cannot apply directly to our PIR-based private network nearest neighbor query.

2.2 Application of PIR

PIR is a type of technology that can request a data item on a database and does not let the database know which item is requested [11]. To make oblivious data item access in malicious server, various Private Information Retrieval (PIR) technology have been widely adopted since its first proposal [6]. Then, there are three streams of relevant research: (1) information based PIR theoretic [10,11]; (2) computational PIR [6,12] and (3) secure hardware based PIR [14]. In this paper, we adopt the secure hardware as its implementation. The secure hardware relies on a temper-resistant CPU which is positioned at the server and is trusted by the clients. It is considered as an interface that supports oblivious data page access. The overhead of one PIR access involves two parts: (1) the online cost which represents the overhead of retrieving, re-encrypting and storing the data page (2) the offline cost which is taken to reorganize data pages in the data structure. All the online and offline cost grow sub-linearly to the space size. This fact explains why we mainly focus on reducing the number of PIR accesses rather than saving the storage space in later design.

For spatial NN query, to prevent location information leakage, [8] presents a novel LBS privacy preserving approach based on computational PIR for

NN query. Later, based on some computing impractical problem(e.g.,Quadratic Residuosity Assumption, QRA), nontrivial implementation for PIR is proposed. Then some works utilize Oblivious Transfer [15] or Paillier encryption scheme [16] integrating with computational PIR technology to protect location privacy of the spatial NN query [17,18]. The only existing spatial query with hardware based PIR methods is PIR-based kNN query [7] and PIR-based BRNN query [13]. To guarantee equal number of PIR access for query proposed by any location, all these methods figure out a maximal number of PIR access after pre-computation over the dataset.

3 Problem Definition

In this section we review the preliminaries of network nearest neighbor query, and then describe our system model and security model.

3.1 System Model

The kNN query on road network has received much attention in research community since its seminal work [26]. A road network is modeled as a graph $G(V, E)$, where a vertex $v \in V$ denotes a road junction or point of interest(POI) and an edge $e \in E$ denotes the path between two vertices; and the weight of the edge denotes the network distance of the two points. A k nearest neighbor query issued at q on road network returns k POIs that are the closest to q in terms of network distance. Without considering the privacy protection, the client poses network kNN query to LBS, and LBS reports the results back to the client based on G.

To guarantee strong privacy, a naive solution is transferring the whole dataset to the client when a query is processing so that the server cannot get extra information about the query except just a query occurring. However, this way is not practical due to heavy communication cost. Thanks to the private information retrieval(PIR) technology, we can design index structure and query plan to combine with it to reduce both the communication and computation cost. In this paper, we adopt the secure co-processor($SCOP$) [9] which is installed at LBS to execute PIR functionality. It offers a PIR interface that can allow clients to retrieve data pages from the database of LBS. The interface can be trusted by the clients as it support complete tamper detection. Figure 1 shows our system model. There exists two parts, the client and the LBS which deploy $SCOP$. Both plaintext of road network G and the encrypted PIR-based index are hosted by LBS. And the indexing information is encrypted by $SCOP$ after being organized in equal-sized data pages. When clients issues a query, he need to follow the query plan to retrieve multi-rounds data pages from the encrypted PIR-based index by $SCOP$.

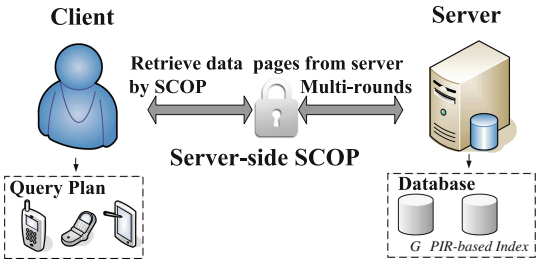


Fig. 1. System Model

3.2 Security Model

Without loss of generality, we assume that LBS is the adversary as it may know the client's identity (e.g.,via user log-in) or may infer it. Also, we assume the adversary is curious, but not malicious, that is, it executes page access routines correctly with no falsified data and wishes to gain extra information about the client's query. The adversary is also aware of the processing protocol in use and its computational power is polynomially bounded.

Our objective is to create PIR protocol for processing network nearest neighbor queries at the LBS without the latter deducing any information about the queries. We assert that every network nearest neighbor query follows the same query plan which is necessary to achieve our privacy goal. Specifically, the query plan needs to ensure each query (i) executes in the same number of rounds, (ii) in each round it accesses the same index in the same order, and (iii) from each index accessed in a specific round, it retrieves the same number of pages. In our paper, we name PIR-based index as *database*. And commonly, we need to design more than one *database* to improve the query performance. For example, if the protocol confirms that 3 pages are fetched from *database* DB_1 and 10 from DB_2(in this order), each query must fetch 3 pages from DB_1 and 10 pages form DB_2. If some query may need fewer than the determined number of pages, the protocol will pad its requests with dummy page accesses in order to conform to the query plan. The following theorem proves that our methodology achieves the security objective.

Theorem 1. *The network nearest neighbor query processing methodology that combines PIR technology with common query plan can achieve strong privacy. Equivalently, it leaks no information to the adversary about query location.*

Proof. In our methodology, each data page requested from database via PIR protocol. Therefore, the adversary is oblivious of which page of the database is being read. What is only visible to the LBS is the number of data pages being accessed in the database. Since all queries follow the same query plan, the number of pages retrievals in the database is identical for all queries. Consequently, adversary cannot tell any two of them apart.

4 PIR-based kNN Processing Framework

In this section, we describe PIR-based kNN Processing Framework to provide strong privacy. Recall our security model, any query processing must follow the same query plan. And, a group of moderate indices stored at LBS are needed to accelerate the query process. In our paper, we split the whole dataset into i databases DB_1, DB_2, ..., DB_i which can help reduce the update cost and the communication cost. So the query plan [cnt_1, cnt_2, ..., cnt_i] represents the maximal number of PIR based page accesses for each *database*.

4.1 Preliminaries

The Network Voronoi Diagram (NVD) has shown to be successful to solve spatial queries such as kNN on road networks. As Fig. 2 illustrates, v_1–v_{10} are vertices of the road networks, wherein v_1–v_4 are POIs. Each cell of the Voronoi Diagram is centered by one POI and contains the locations that are closest to this POI than any other POIs. In road networks, neighboring voronoi cells are separated by border points, such as b_1 to b_6. For example, voronoi cells centerd at v_1 and v_3 are separated by border points b_1, b_3, and b_5. For each voronoi cell, its border points construct a region. The distance between the voronoi cell center and any query point of the cell can be computed by given all edges in such region.

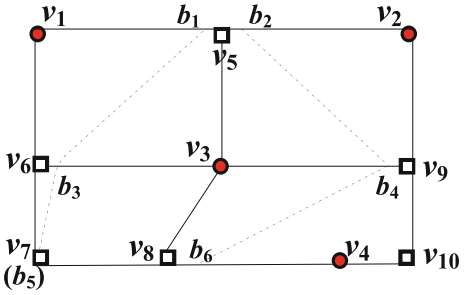


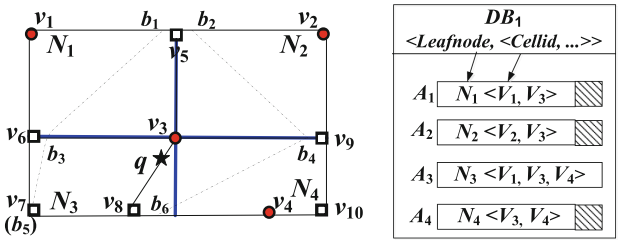
Fig. 2. Example for voronoi diagram in road network

Given properties of the NVD that are described in [22], we can easily compute the kNN query in road network:

(1) The 1NN of query point q is the center of the voronoi cell that q locates in. For example, we assume the query q located at v_8, and v_8 is in the voronoi cell of v_3, then q's nearest neighbor object is v_3.
(2) The k_{th} NN lies in the neighbor of previously found voronoi cells of the (k-1) NN results. That means, if q's nearest neighbor object is v_3, then q's second nearest neighbor object must be the center of neighboring voronoi cells v_1, v_2 and v_4. We further develop efficient method to determine which voronoi cells to fetch without actually obtaining their network information.

4.2 Three Databases

To enhance PIR performance, we first construct a spatial index to partition the whole road networks so that the candidate voronoi cells of q can be located efficiently. To achieve high space utilization, we use the widely adopted KD-tree to partition the whole map. The KD-tree leaf node splits when the voronoi cell overlapping with it occupies more space than a data page. Note that if only one voronoi cell takes more than one page, the leaf node will not split and this corner case is handled by augmenting the leaf node with linked overflow data pages. As Fig. 3(a) illustrates, the four dotted rectangle N_1, N_2, N_3 and N_4 represent the leaf nodes of KD-tree. Each node contains 3 to 5 edges. Correspondingly, in Fig. 3(b), DB_1 occupies 4 data pages referring to the four leaf nodes in KD Tree. Each page records the ID of the voronoi cell residing in the leaf node. We assume the client issues a query at q(the black star). According to the location of q, client can access the leaf node N_3's record A_3, and then client can get the candidate voronoi cell q located in: V_1, V_3 and V_4. Once we obtain the distance between the query point q with these candidate voronoi cell centers, we can know q's 1NN. As such, we design the second structure DB_2.



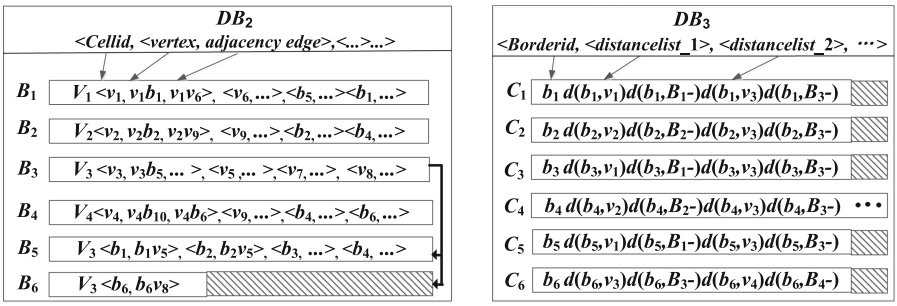
(a) KD-tree Index (b) DB_1

Fig. 3. Example for spatial network partitioning

As illustrated in Fig. 4(a), DB_2 stores the network information of each network voronoi cell including the vertices, edges and border points. In the example above, the client can access the records B_1, B_3 and B_4 from DB_2. With these network information of V_1, V_3 and V_4, the client can compute which voronoi cell q located in by employing distance computation algorithms under road networks, such as Dijkstra algorithm [19].

According to the property (2) of NVD, the next nearest neighbor of q resides in a number of candidate voronoi cells. Note that we have obtained their border points because they are neighboring to our obtained voronoi cells. Since we have obtained the distance between q to all border points, if we know the distance between each border point to the voronoi cell center on the other side of the edge (voronoi cell on this side of the edge has been obtained), we can determine which voronoi cell center is the next nearest neighbor without fetching all their network information.

As Fig. 4(b) illustrates, we assume each border point b belongs to m_b voronoi cells, so DB_3 stores m_b distance lists for each border point b. Each distance list contains two parts: one is the distance between the border point b_i and the center of voronoi cell V_j it belongs to; The other is the distance between each border point $b_i \in B_{j-}$ falling on the same voronoi cell (B_{j-} represents all the border points fall on the same voronoi cell V_j). Therefore DB_3 can help to compute the minimum distance between q and candidate voronoi cell centers. In the example above, the 2NN of q is the voronoi cell center v_4 adjacent to the q's 1NN via border point b_6 with minimal overall distance $dist(q, b_6) + dist(b_6, v_4)$.



(a) DB_2 (b) DB_3

Fig. 4. Example for DB_2 and DB_3

4.3 Query Plan

To achieve the security goal, we determine the query plan $[cnt_1, cnt_2, cnt_3]$ which represents the maximal number of PIR based data page accesses for each *database*. For ease of description, we use n_i to represent the maximum number of data pages for a single record in DB_i. Take Figs. 3 and 4 as an example, $n_1 = n_3 = 1$, $n_2 = 3$. According to the rational of PIR, each query must follow query plan to retrieval data pages from DB_1, DB_2 and DB_3 respectively. The implementation of the algorithm is as follows:

(1) For each query q, we use DB_1 and DB_2 to compute which network voronoi cell q locates in, and the center of this network voronoi cell is the 1NN of q. Then $cnt_1 = n_1$. For cnt_2, we assume that the maximal number of voronoi cells in each record in DB_1 is c_2, so $cnt_2 = n_2 \times c_2$.
(2) according to the DB_3, the client computes q's next nearest neighbor as its $2NN$ which takes the minimal distance between q and $1NN$'s neighbor centers. Recursively, we can repeat the step (2) to compute the $3-k$ NN as the query result. In this process, the client needs to maintain the distance of q to every border point of the voronoi cells obtained before.

For cnt_3, we assume that each record in DB_2 refers to maximal c_3 border points. Then, each query result needs $cnt_3 = n_3 \times c_3$ data page accesses. To get the complete result set, there needs $k - 1$ iterations.

Overall, the deduced retrieval plan for kNN query on road network requires n_1 PIR based data page accesses for $DB_1, n_2 \times c_2$ page accesses for DB_2 to obtain the 1NN and (k-1) rounds to obtain the rest nearest neighbors. And each round takes $n_3 \times c_3$ page accesses to obtain all distance related to these border points and the new voronoi centers for the next round. In this way the trivial query plan requires $n_1 + n_2 \times c_2 + (k-1) \times n_3 \times c_3$ PIR accesses for arbitrary kNN query on road network.

4.4 Algorithm

In this following, we present our PIR-kNN algorithm. According to the three databases and query plan, we design our algorithm as follows:

Algorithm 1. PIR-kNN algorithm

Input: Query point q, query parameter k
Output: network k nearest neighboring object points, R

1: $R = \emptyset$
2: $C = \emptyset$
3: Fetch entries corresponding to voronoi cells from DB_1, denoted by E^{DB_1}, by locating the leaf node in KD Tree that contains q via cnt_1 PIR page accesses
4: Fetch detailed contents of such voronoi cells E^{DB_1}, denoted by E^{DB_2} from DB_2 by cnt_2 PIR page accesses
5: **for** each record $e \in E^{DB_2}$ **do**
6: $distance_e = dist(e, q)$
7: $C.push(e, distance_e)$
8: **end for**
9: $cc = \{C.top()\}$
10: **if** $k == 1$ **then**
11: $R = cc$
12: return R
13: **end if**
14: **for** $i = 2$ to k **do**
15: **for** each border point b of cc **do**
16: Fetch all pre-computed distance of border b via cnt_3 PIR page accesses
17: **for** each b's relevant voronoi center vc **do**
18: $distance_e = mindist(q, vc)$
19: $C.push(vc, distance_e)$
20: **end for**
21: $cc = C.top()$
22: $R = R \cup \{cc\}$
23: **end for**
24: **end for**
25: return R

As Algorithm 1 illustrates, the first step is to get the 1NN of q (Line 3–13). There needs n_1 PIR data page accesses for DB_1 and $n_2 \times c_2$ page accesses for DB_2. Then, we fetch all the border points and their associated distance to neighboring voronoi cells from DB_3 via $(k-1) \times n_3 \times c_3$ PIR accesses (Line 14–22). Note that in this step, we only need to obtain the detailed distance information of the new border points obtained in the last iteration. Obtaining all the distance, we can determine the next nearest neighbor (Line 18–19). Until we obtain k nearest neighbor, the algorithm terminates (Line 25).

5 Experimental Evaluation

In this section, we conduct experiments under real world and synthetic datasets to demonstrate the effectiveness of our PIR-based kNN approach. We also compare the performance with a weaker location privacy preservation approach — the cloaking region-based kNN method on road networks(CR-kNN) [5] and show our algorithm is of great practical value.

5.1 Experiment Settings

Datasets. We conduct our experiments on two public real-world networks, namely California map (CA) and New York map (NY). Both datasets are collected from Open Street Map[1]. Both datasets have relatively uniform distribution, while the junctions and roads are more denser in NY than in CA. We summarise the statistics of our datasets in Table 1.

Table 1. Statistics of our datasets.

Dataset	# Edges	# Junctions	# Point of Interests
CA	47, 185	20, 997	84, 328
NY	56, 263	14, 890	60, 327

As for the synthetic dataset, we scatter 10^6 point of interests on aforementioned CA map to simulate different data distribution. To emulate a skewed distribution, a portion $f \in (0, 1]$ of these points are distributed on edges in a skewed way, while the rest $1 - f$ portion of points are uniformly generated on edges.

All algorithms are implemented in C# and run on a machine with an Intel Core2 Quad CPU 2.53 Ghz and 4 GByte of RAM. As with previous hardware-based PIR methods, we assume the IBM 4764 PCI-X Cryptographc Coprocessor as the SCOP and strictly simulate its performance. The client communicates with the LBS using a link with round trip time of 700ms and bandwidth 384 Kbit/s, which emulates a moving client connected via a 3G network.

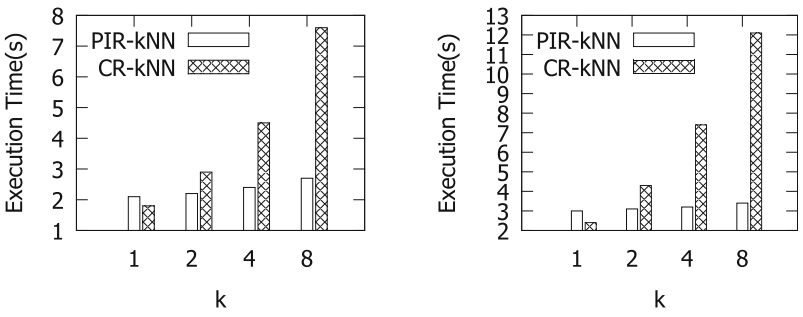
[1] www.openstreetmap.org.

5.2 Performance Comparison

In this section, we compare the performance of our PIR-kNN method with the CR-kNN method under both real world datasets. The latter method fetches all POIs that overlap with the client-issued cloaking region and the candidate voronoi cells, and then returns all these POIs to the client. Note that the performance of CR-kNN is plotted only for reference, as it still discloses a cloaking region to the LBS.

Figure 5 illustrates that when $k = 1$, PIR-kNN approach takes more time to return the nearest point of interests from the road network than cloaking region based approach. This is because PIR-kNN first requires to locate the voronoi cell in road network, and when multiple voronoi cell overlaps in a rectangle in the map, all map contents in these cells must be fetched. Thus, this routine consumes much running time. Interestingly, when $k > 1$, PIR-kNN approach gradually outperforms CR-kNN approach and the performance gap enlarges when k gets larger. This is because after the voronoi cell in the road network is located for the query point, each increment of k only incurs one extra fetch for pre-computed distance information via PIR interface. While, for CR-kNN, as k increases, POIs locating in larger map area must be fetched.

We can also see that in NY dataset, where the junctions and roads are denser, it takes more time to return the query result. This is because each POI has more neighbor POIs. Note that the CR-kNN approach also takes more time in NY dataset. This is because the cloaking region with the same size now contains more point of interests.



(a) CA dataset (b) NY dataset

Fig. 5. Performance comparison under real world datasets.

We validate this argument by the more detailed measurement in Fig. 6. In Fig. 6, we can clearly see that the network overhead for our PIR-kNN approach is much less than that of CR-kNN approach. This demonstrates that there are significant unnecessary POIs are transferred from LBS to the client. In contrast, our PIR-kNN approach seldom conveys unnecessary data and the major overhead comes from the online and offline processing routine in SCOP to implement oblivious fetch.

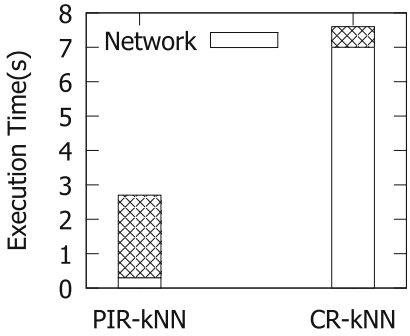


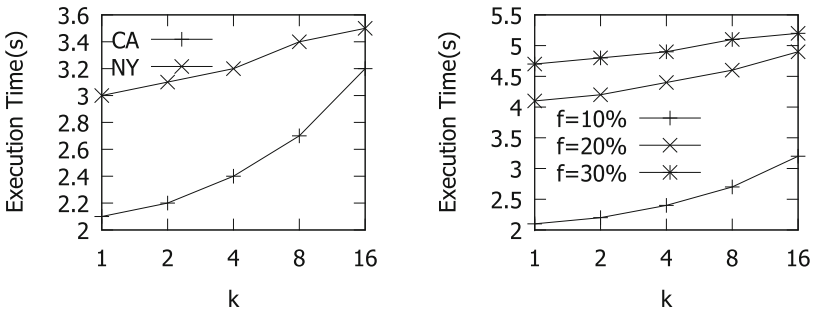
Fig. 6. Execution time proportion over network communication on the CA dataset.

5.3 Detailed Analysis of Our Method

In the following, we give a detailed analysis of our method. In specific, the performance of our PIR-kNN approach under different data distribution and its scalability are evaluated.

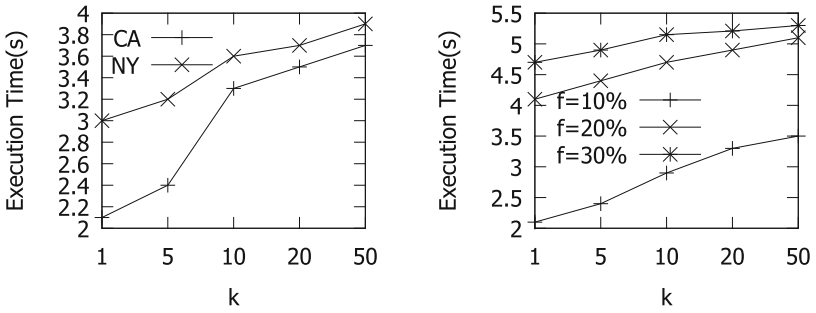
Effect of Data Distribution. First, we compare the performance under CA and NY datasets. It should not be surprising that in Fig. 7(a), when k is less than 8, the execution time for our PIR-kNN approach is significantly shorter under CA dataset than under NY dataset. As we have mentioned, this is because the initial locating for the query point tends to fetch more neighboring voronoi cells in NY dataset. So when k is larger, the latter needs to fetch more extra distance information than the former to determine the kNN query processing.

Under synthetic dataset, we can see from Fig. 7(b) that when f increases, the maximal number of PIR accesses for one voronoi cell increases, the overall execution time increases as well. This is intuitive because our query plan should cover the worse case in terms of the PIR accesses.



(a) Real World dataset (b) Synthetic dataset under CA map

Fig. 7. Effect of data distribution.



(a) Real World dataset (b) Synthetic dataset under CA map

Fig. 8. Effect of scalability.

Evaluation of Scalability. Finally, we evaluate the scalability of our approach. Regardless of whether real world data or synthetic dataset, we can see from Fig. 8 that the execution time increases linearly to the query parameter k. Further, we can see that the increasing rate of the execution time gets slower as k gets larger than 10. This is because the voronoi cells that contain much more POIs or edges than normal voronoi cells have been considered by a smaller threshold k and when k goes beyond this threshold, the number of PIR accesses required for increased k is much less.

6 Conclusion

In this paper we introduce the novel problem of PIR-based kNN query on road networks with strong privacy guarantee, where an adversary cannot distinguish a kNN query from any other query in the network space. This is the first work that applies PIR to network kNN query. Further, we design the data structure to fetch only necessary data and deduce a query plan for arbitrary query parameter k. Finally, we evaluate our method on real world dataset and synthetic dataset. Extensive experiments demonstrate the practicality of our method.

References

1. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: VLDB, pp. 763–774 (2006)
2. Yiu, M.L., Jensen, C., Huang, X., Lu, H.: Spacetwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile systems. In: ICDE, pp. 366–375 (2008)
3. Wong, W.K., Cheung, D.Q., Kao, B., Manoulis, N.: Secure kNN Computation on encrypted databases. In: SIGMOD, pp. 139–152 (2009)
4. Khoshgozaran, A., Shahabi, C., Shirani-Mehr, H.: Location privacy: going beyond k-anonymity, cloaking and anonymizers. KAIS **26**(1), 435–465 (2011)

5. Jung-Ho, U., Yong-Ki, K., Hyun-Jo, L., Miyoung, J., Jae-Woo, C.: K nearest neighbor query processing algorithm for cloaking regions towards user privacy protection in location-based services. J. Syst. Archit. EUROMICRO J. **58**(9), 354–371 (2012)
6. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: FOCS, pp. 364–373 (1997)
7. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest neighbor search with strong location privacy. Proc. VLDB **3**, 619–629 (2010)
8. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private queries in location based services: anonymizers are not necessary. In: SIGMOD (2008)
9. Mouratidis, K., Yiu, M.L.: Shortest path computation with no information leakage. PVLDB **5**(1), 692–703 (2012)
10. Beimel, A., Ishai, Y., Kushilevitz, E., Raymond, J.-F.: Breaking the O(n1/(2k–1)) barrier for information-theoretic private information retrieval. In: Proceedings of FOCS, pp. 261–270 (2002)
11. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. JACM **45**(6), 965–981 (1998)
12. Gentry, C., Ramzan, Z.: Single-database private information retrieval with constant communication rate. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 803–815. Springer, Heidelberg (2005)
13. Wang, L., Meng, X., Hu, H., Xu, J.: Bichromatic reverse nearest neighbor query without information leakage. In: Renz, M., Shahabi, C., Zhou, X., Cheema, M.A. (eds.) DASFAA 2015. LNCS, vol. 9049, pp. 609–624. Springer, Heidelberg (2015)
14. Williams, P., Sion, R.: Usable PIR. In: NDSS (2008)
15. Naor, M., Pinkas, B.: Oblivious transfer with adaptive queries. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 573–590. Springer, Heidelberg (1999)
16. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
17. Paulet, R., Kaosar, Md.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. In: ICDE, pp. 44–53 (2012)
18. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical k nearest neighbor queries with location privacy. In: ICDE, pp. 640–651 (2014)
19. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
20. Goldberg, A.V., Harrelson, C.: Computing the shortest path: A* search meets graph theory. In: SODA, pp. 156–165 (2005)
21. Lee, K.C.K., Lee, W.-C., Zheng, B.: Fast object search on road networks. In: EDBT, pp. 1018–1029 (2009)
22. Kolahdouzan, M., Shahabi, C.: Voronoi-based K nearest neighbor search for spatial network databases. In: VLDB, pp. 840–851 (2004)
23. Hu, H., Lee, D.-L., Xu, J.: Fast nearest neighbor search on road networks. In: Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 186–203. Springer, Heidelberg (2006)
24. Safar, M.: K nearest neighbor search in navigation systems. Mob. Inf. Syst. **1**(3), 207–224 (2005)
25. Narayanan, A., Shmatikov, V.: Robust De-anonymization of large sparse datasets. In: IEEE Symposium on Security and Privacy, pp 111–125 (2008)
26. Jensen, C., Kolarvr, J., Pedersen, T.B., Timko, I.: Nearest neighbor queries in road networks. In: GIS, pp. 1–8 (2003)