

Answering Constrained k -NN Queries in Unstructured P2P Systems

Bin Wang¹, Xiaochun Yang^{1,*}, Guoren Wang^{1,**}, Lei Chen², Sean X. Wang³,
Xuemin Lin⁴, and Ge Yu¹

¹ School of Information Sciences and Engineering, Northeastern University, China
{binwang,yangxc,wanggr,yuge}@mail.neu.edu.cn

² Department of Computer Science, Hong Kong University of Science and Technology
leichen@cs.ust.hk

³ Department of Computer Science, University of Vermont, USA
xywang@cs.uvm.edu

⁴ School of Computer Sci. and Eng., The University of New South Wales, Australia
lxue@cse.unsw.edu.au

Abstract. The processing of k -NN queries has been studied extensively both in a centralized computing environment and in a structured P2P environment. However, the problem over an unstructured P2P system is not well studied despite of their popularity. Communication-efficient processing of k -NN queries in such an environment is a unique challenge due to the distribution, dynamics and large scale of the system. In this paper, we investigate the problem of efficiently computing k -NN queries over unstructured P2P systems. We first propose a location-based domination model to determine a search space. We then present two types of probing strategies, radius-convergence and radius-expanding. A comprehensive performance study demonstrates that our techniques are efficient and scalable.

1 Introduction

Due to their importance in many applications in a variety of domains, k -NN queries have been extensively studied [8,13]. An often used mechanism that provides much needed retrieval efficiency is a centralized index. However, for k -NN queries in a distributed environment, especially unstructured P2P environments, centralized indexing is not a practical solution. The following example shows the reason that new techniques are needed and hence motivate the work of this paper.

Consider a tsunami alarm system for a certain area, e.g., the bay area of Indonesia. Detection of a tsunami in many cases need data from areas that go across multiple nations. Assume the nations establish a logical cooperative

* Supported by National Natural Science Foundation of China No. 60503036 and Fok YingTong Education Foundation No. 104027.

** Supported by National Natural Science Foundation of China No. 60473074.

network, where for any two nations, they have either direct cooperative relationship, represented as logical neighbors in the network, or indirect cooperative relationship if their logical neighbors can cooperate. Each nation autonomously maintains a set of sensors to monitor her own sea area and can request data from her cooperative nations. Note that it is not necessary that two logical neighbors are geographically bordered. Therefore, those autonomous nations need to cooperate to answer a distributed k -NN query efficiently. In this application, global indexing is not available for the k -NN query, either, since data values may change dynamically. Furthermore, it is often not possible for all the nations to follow some organization rules (hashing functions) to arrange or store the data as required by a structured P2P system.

Compared to the distributed k -NN problems on P2P [6,7], a constrained k -NN search with value predicates in unstructured P2P systems has posed the following unique challenges: (i) Topology mismatch between the P2P logical overlay network and physical underlying network; and (ii) Data in each peer is maintained autonomously in unstructured P2P systems.

We address the above problems and make the following contributions. (i) We propose a new framework for processing k -NN queries in unstructured P2P systems, (ii) we give a novel filtering mechanism to reduce the communication cost and effectively terminate our search, and (iii) we also give detailed complexity analysis of our algorithms.

The rest of the paper is organized as follows. The formal definition of constraint k -NN queries and a proposed filter model, called domination model, to efficiently prune peers are given in Section 2. Section 3 provides our techniques searching k -NN queries with least communication cost. Experimental results and performance studies are discussed in Section 4. In Section 5, we discuss related work. Finally, we conclude in Section 6.

2 Problem Definition

We assume a set of logically connected, cooperative peers, each covering a spatial region that does not overlap with the spatial regions covered by all other peers. We use a non-directed graph $G = (P, E)$ to model the logical connections, where P is a set of vertices representing the peers and E a set of edges expressing the logical connections between the peers. For a peer $p \in P$, we use $R(p)$ to denote the spatial region it covers and $D(p)$ to denote data set maintained by p . We also assume that each peer has pre-knowledge of spatial regions covered by all other peers. This assumption is reasonable since it is easy for a peer to collect data from different peers and derive this knowledge in an incremental manner.

Each item in $D(p)$ maintained by p has two kinds of attributes, namely location attributes and non-location attributes. Hence, we denote each data item d in $D(p)$ as a pair $\langle \mathbf{v}_l(d), \mathbf{v}_n(d) \rangle$, where $\mathbf{v}_l(d)$ ($\mathbf{v}_n(d)$, resp.) is a value vector of the location (non-location resp.) attributes of d . For each d in $D(p)$,

$\mathbf{v}_l(d)$ must be contained in the region $R(p)$. A k -NN query q is composed of two parts, location value and non-location value predicate, denoted as $\langle \mathbf{v}_l(q), q_c \rangle$, respectively. A value predicate returns, when applied to a non-location value vector, true or false. We define the distance between q and data d as following:

Definition 1. *Given a query q and data d , let the distance between q and d be*

$$dist(q, d) = \begin{cases} dist(\mathbf{v}_l(q), \mathbf{v}_l(d)) & \text{if } q_c(\mathbf{v}_n(d)) = true \\ \infty & \text{otherwise} \end{cases}$$

where $dist$ can be any of the L_p -norm.

Given a query q and a peer p , we use $minDist(q, p)$ ($maxDist(q, p)$, resp.) to express the minimum (maximum, resp.) distance between q and all the points in $R(p)$.

Formal Problem Statement: Assume we have a set of logically connected peers p_1, \dots, p_n , each of which manages a set of data items, denoted $D(p_i)$, with locations of each data item in $D(p_i)$ being in the spatial region $R(p_i)$. Assume further that $R(p_i) \cap R(p_j) = \emptyset$ for all $i \neq j$. Given a continuous query q issued by a peer p_i , continually search k data items $\{d_1, \dots, d_k\}$ among the data items in $D(p_1) \cup \dots \cup D(p_n)$ such that there does not exist any data item d satisfying the condition $dist(q, d) \leq dist(q, d_h)$ for some $1 \leq h \leq k$. We aim to minimize total communication costs.

The intuition of the domination model comes from the following simple facts: If we know an upperbound r^u of the distance from the query point q to the k^{th} data item in the query results, then a peer p' does not need further probing (for possible answers) if $r^u \leq minDist(q, p')$. Similarly, if we know that a peer p provides the k^{th} item in the answer set, then a peer p' does not need further probing if $maxDist(q, p) \leq minDist(q, p')$. These facts form the basis for our optimized search algorithms, which are described in Section 4.

Definition 2. (*Dominate relationship*) *Given a query q , and two peers p_1 and p_2 , if $maxDist(q, p_1) \leq minDist(q, p_2)$, we then say that p_1 dominates p_2 , denoted $p_1 \prec_q p_2$, or $p_1 \prec p_2$ when q is understood. For two groups of peers P_1 and P_2 , if for each peer $p \in P_1$ and each peer $p' \in P_2$, $p \prec p'$, we say P_1 dominates P_2 , denoted $P_1 \prec P_2$.*

3 Pruning Candidate Peers

We propose two probing strategies, namely the *radius-convergence strategy* and the *radius-expanding strategy*. The radius-convergence strategy shrinks the probe radius gradually until all peers in the probe circle have been probed, while the radius-expanding strategy gradually expands the probe radius.

3.1 The Radius-Convergence Approach

The first strategy that we introduce for direct probing is radius-convergence strategy, which keeps shrinking the proving radius when more peers are probed and closer data are found. For each probe, it keeps the new k nearest data and let the new probe radius r_k be the Euclidean distance between q and the new k^{th} data. This procedure repeats until all peers overlapping the circle with the (shrinking) r_k have been probed.

Ideally, we prefer to find a “better” peer that can get a close probe radius in the first a few times probe. Seeking “better” peers to start with can be considered as a rank aggregation (RA) problem [5], which searching top candidates from a set of sorted rank list. We propose an algorithm to choose best- k peers in the probe circle and put emphasis on the minimum distance.

Our *RA-based approach* is shown in Algorithm 1. Line 1, in Algorithm 1, normalizes the attribute values $\langle min, max, plen \rangle$ for minimum distance, maximum distance, and the logical paths of each peer to the range of $[0,1]$, so that they have the same weight for estimating a peer. Line 2 sorts all peers according to minimum distance attribute in the ascending order. Lines 3-12 choose the top- k peers with the smallest grades. Compare it with the random-based approach, RA-based approach can decrease probe radius quickly, that is, probes peers with higher probabilities to provide closer answers.

Algorithm 1: RA-based algorithm

```

Input:  $P = \{p_1, \dots, p_h\}$ , each  $p_i = \langle id, min, max, plen \rangle$ ,  $k(\leq h)$ ,
          geographical weight  $w_g$ , logical weight  $w_l$ 
Output: a set of top- $k$  peers
1: normalize  $P$  to  $P'$ ; //  $p.min, p.max, p.plen \in [0, 1]$ 
2:  $l_1 = \text{sort } P'$  according to  $min$ ; List  $result = \emptyset$ ;  $count = 0$ ;  $\lambda = 0$ ;
3: for  $i = 1; i \leq |P|; i++$  do
4:  $\lambda = w_g \times ((l_1+i).min + (l_1+i).max) + w_l \times (l_1+i).plen$ ;
5:  $r =$  last element in  $result$ ;
6: if  $\lambda \leq w_g \times (r.min + r.max) + w_l \times r.plen$  then
7: insert  $\{(l_1+i).id\}$  to  $result$ ;  $count++$ ;
8: if  $count == k$  then return  $result$ ;
9: else insert  $(l_1+i)$  in  $candidate$  in ascending order;
10: end if
11: end if
12: end for
13: return  $result$ ;

```

Note that, even some peers can provide k closest data, we are not able to prune out all the rest peers whose minimal distances to q are less than the new query radius r_k . We then choose another top- k peers, using the same procedure, in the remaining peers, and probing these peers to get another set of k' -closest satisfying data ($k' \leq k$). We choose the first k data among these two sets of returned satisfying data and shrink r_k . The above procedure repeats until all peers in the shrunk probe circle have been probed.

3.2 The Radius-Expanding Approach

We propose a *partition-based approach* (PA in short) to gradually grow the probing radius. Ideally, we hope to find a small set of peers P_1 , such that if P_1 can answer the k closest answers, then all other peers can be safely pruned out. For peers in each group, we build up a steiner tree using the local filtering mechanism. Since messages are forwarded through edges of the constructed tree, we hope to find and merge groups (such as P_1 and P_2) such that the summation of edges of corresponding steiner trees T_1 and T_2 is minimal. In fact, this property has been proved and stated in the following Theorem.

Theorem 1. *Given an undirected graph $G(P, E)$ and two set of target nodes $P_1 \subseteq P$ and $P_2 \subseteq P$, the summation of edges of the two trees $T(P_1)$ and $T(P_2)$ is not less than the edge of $T(P_1 + P_2)$, i.e. we have the following property*

$$|T(P_1)| + |T(P_2)| \geq |T(P_1 + P_2)|.$$

The basic task for the PA is the following. Given an undirected graph $G = (P, E)$, a set of terminal nodes $P' \subseteq P$, and subsets $P_1 \prec \dots \prec P_h$, $P_i \subseteq P'$ ($1 \leq i \leq h$), find an optimal partition of $P' = \{C_1, C_2\}$, such that neither C_1 nor C_2 is empty, each P_i can only belong to C_1 or C_2 (but not both), and the sum of number of edges in steiner trees for $\{C_1, C_2\}$ is minimal. PA combines geographical dominate relationship and logical steiner tree together. We use Algorithm 2 to greedily approximately build two steiner trees.

Algorithm 2: PA

Input: k , undirected graph G , source node s , and terminal nodes $P' = \{P_1, \dots, P_h\}$
Output: two steiner trees T_1 and T_2

- 1: **for** each P_i between P_1 and P_k **do**
- 2: $ST_i = \text{greedyST}(G, \{s\} \cup P_i)$;
- 3: **end for**
- 4: find two minimal trees, assuming they are for P_u and P_v ;
- 5: $T_1 = ST_u$; $RT_1 = P_u$; $T_2 = ST_v$; $RT_2 = P_v$;
- 6: **for** each peer P_i in $P' - RT_1 - RT_2$ **do**
- 7: pick the a minimal tree ST_i for P_i ;
- 8: $T'_1 = \text{greedyST}(T_1 \cup T_i, \{s\} \cup RT_1 \cup P_i)$; $T'_2 = \text{greedyST}(T_2 \cup T_i, \{s\} \cup RT_2 \cup P_i)$;
- 9: **if** $\text{edge}(T'_1) \leq \text{edge}(T'_2)$ **then** $T_1 = T'_1$; $RT_1 = RT_1 \cup P_i$;
- 10: **else** $T_2 = T'_2$; $RT_2 = RT_2 \cup P_i$;
- 11: **end if**
- 12: **end for**

PA classifies peers into two groups C_1 and C_2 corresponding to the two steiner trees T_1 and T_2 , respectively. Peers in C_1 can be probed together. Note that, groups in $C_2 = C - C_1$ may contain a group P_j dominates groups in C_1 . If the first round probe gets k' ($\leq k$) closest answers, then using the locations of these k' answers, we can know k_1 ($\leq k'$) out of k' answers are returned by P_1 . Then it classifies $P' - C_1$ into two classes and request $k - k_1$ to C_1 . PA iterative repeats the above procedures until k closest answers are returned.

4 Experimental Study

In this section, we built a peer-to-peer simulator to evaluate the performance of our proposed system over large-scale networks. To evaluate the cost of query processing, we tested the network with different number of peers N from 200 to 1800. Each peer contains a set of data from 1K to 200K with two dimensional location values and one non-location value. We generated two datasets. Dataset 1 conforms to uniform distribution and dataset 2 conforms to normal distribution. All approaches were implemented in C++ and run on Intel XEON(TM) 3.2GHz dual-CPU with 2G RAM on Windows 2003 Server. For each setting, we tested an algorithms by running it 10 times to compute the average result.

Comparison of different k -NN searches. We compared the communication cost (represented as # of messages) of getting k -NN results using different search strategies. We ran 10 different k -NN queries whose locations and predicate values are randomly specified. Besides flooding search, we tested two radius-convergence approaches and two radius-expanding approaches. Radius-convergence approaches include random search and RA-based search, whereas radius-expanding approaches include exhausted search every dominate groups (EX in short) and Partition-based search (PA in short). We used two datasets with uniform and normal distributions, respectively. In addition, we also compared three straightforward approaches according to the ranking of the minimal and the maximal geographical distances and the shortest path in logical graph, respectively.

Fig. 1 shows the conveying messages of these algorithms using two datasets conforming to uniform and normal distributions. We let the number of peers vary between 500 and 2500, and ran 10 different 100-NN queries. Figs. 1(a) and (b) compare five approaches using dataset 1. It shows that flooding search costs the most conveying messages, whereas PA uses the least conveying messages. Our proposed two approaches RA and PA are all independent on the number of peers. RA is the second best approach. Fig. 1(c) shows the similar result using dataset 2. Figs. 1(b) and (d) show the number of conveying messages when fixed the number of peers to 1000 and varied k from 20 to 100. They all show that both RA and PA outperform than the other approaches, and PA always use least messages to get k -NN nearest results.

Comparison of filtering capabilities. We tested the capabilities of filtering peers using PA and RA approaches on dataset 1. We first changed the number of k values from 20 to 70. Fig. 1(e) shows the filtering capability decreased when increasing k values. RA has less filtering capability than PA. The reason is RA is a radius-convergence approach that gradually shrink query radius. It keeps probing at least k peers for each iteration until all peers in the shrinking query radius have been probed. Whereas, PA is a radius-expanding approach that when k closest data is met, it can stop. Fig. 1(f) shows the filter capabilities when varying the number of peer groups. It shows that PA has better filter capability than RA. When the peer group number increased to 16, the capability of PA climbs to a peak value 52%, whereas RA has no peak value. It proved that the

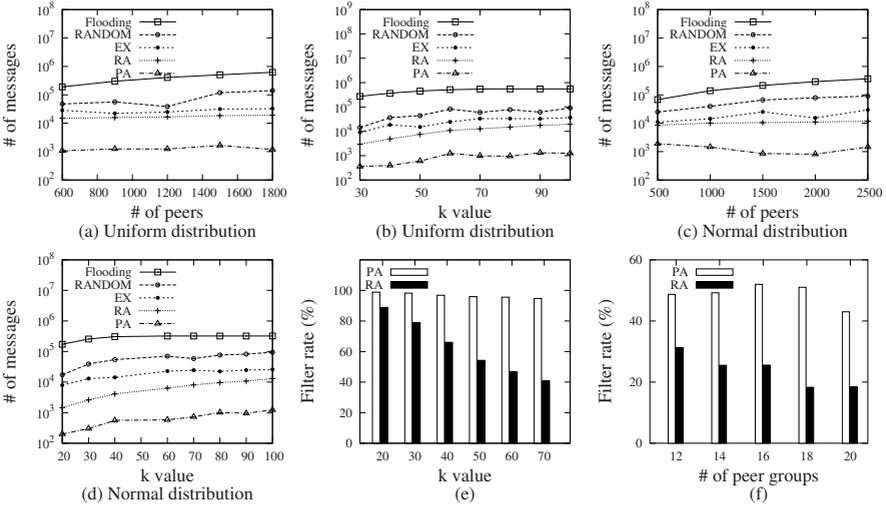


Fig. 1. Comparison of different k -NN searches

filter capability of PA depends on the chosen number of groups in all dominate group set, whereas RA does not since RA only considers the first k ranked peers using geographical distances and logical graphs.

5 Related Work

As far as we aware, there is no similar work has been on the proposed problem setting. The most closed work to ours fall in to two categories: continuous k -NN searches over spatial database and k -NN queries over structured P2P systems. We will review the work in this two directions and explain the difference between our work and theirs. There are many work have been proposed for continuous k -NN queries over moving objects in spatial database domain. Most of these works focus on reducing the number of updates to the indexes. In order to achieve this, the trajectories of moving objects are modeled by some linear functions, thus a R-tree can be built use time as a function [12,16]. Compared to these work, our problem setting is for distrusted environment and we do not assume existence of a centralized index. Moreover, we have the logical communication cost as a constraint to the k -NN queries.

k -NN queries over P2P systems can be classified into search over unstructured P2P and structured P2P. For structured P2P systems, data allocation strategies are important for k -NN search. Distrusted Hashing Table (DHT) is often used to allocate data, such as CAN [9], Chord [15], Pastry [11], and Tapestry [17], which use uniform hash functions and achieve good load balance. However, these hashing functions destroys data locality (data that are similar should be allocate near to each other in the space). Complicated queries such as k -NN have to rely

on multi-cast or additional indexes. Some locality-preserving data allocation approaches are also proposed, such systems include P-Grid [2], P-Ring [4], Baton [7], Vbi-tree [6], and Mercury [3]. The basic idea of these approaches is to keep data locality over the attribute as much as possible. For unstructured P2P systems, very few work have been done so far. Gnutella [1] using flooding techniques to do k -NN search. Compared to these work, our work focus on continuous constrained k -NN search over unstructured P2P system. Besides finding the k -NN, we have to guarantee that searched value satisfying the value predicate specified in the query. Furthermore, in contrary to traditional setting of P2P systems that each peer maintains static data, a peer in our system maintains dynamic data.

6 Conclusion

This paper has investigated the new problem of processing the constrained k -NN queries over unstructured P2P systems, and proposed two approaches to efficiently filter peers in the search space. The experimental results on the two synthetic datasets have shown that (i) the algorithms proposed outperform most the other heuristic algorithms, (ii) the novel adaptive histogram can save more communication cost, and (iii) our technique can efficiently process continuous k -NN queries in a distributed, and large scale environment.

References

1. Gnutella. <http://www.gnutella.com/>.
2. K. Aberer, P. C. Mauroux, and et al. P-grid: a self-organizing structured p2p system. *SIGMOD Rec.*, 32(3):29–33, 2003.
3. A. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In *Proc. of the 2004 ACM SIGCOMM*.
4. A. Crainiceanu, P. Linga, and et al. P-ring: an index structure for peer-to-peer systems. Technical report, TR2004-1946, Cornell University, 2004.
5. R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS*, pages 102–113, 2001.
6. H. V. Jagadish, B. C. Ooi, and et al. Vbi-tree: A peer-to-peer framework for supporting multi-dimensional indexing schemes. In *ICDE*, pages 34–45, 2006.
7. H. V. Jagadish, B. C. Ooi, and Q. H. Vu. Baton: a balanced tree structure for peer-to-peer networks. In *VLDB*, pages 661–672, 2005.
8. F. Korn, N. Sidiropoulos, and et al. Fast nearest neighbor search in medical image databases. In *VLDB*, pages 215–226, 1996.
9. S. Ratnasamy, P. Francis, and et al. A scalable content addressable network. In *In ACM SIGCOMM*, pages 161–172, 2001.
10. G. Robinsy and A. Zelikovskyz. Improved steiner tree approximation in graphs.
11. A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *In IFIP/ACM Distributed Systems Platforms*, pages 329–350, 2001.
12. S. Saltenis, C. S. Jensen, and et al. Indexing the positions of continuously moving objects. In *SIGMOD*, pages 331–342, 2000.

13. T. Seidl and H. Kriegel. Optimal multi-step k-nearest neighbor search. pages 154–165, 1998.
14. A. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang. A sampling-based approach to optimizing top-k queries in sensor networks. In *ICDE*, 2006.
15. I. Stoica, R. Morris, and et al. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM*, pages 149–160, 2001.
16. Y. Tao and D. Papadias. MV3R-tree: A spatio-temporal access method for timestamp and interval queries. In *VLDB*, pages 431–440, 2001.
17. B. Y. Zhao and J. Kubiatowicz. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, UC Berkeley, 2001.