

## 基于 Voronoi-R<sup>\*</sup> 的隐私保护路网 $k$ 近邻查询方法<sup>\*</sup>

倪巍伟<sup>1,2</sup>, 李灵奇<sup>1,2</sup>, 刘家强<sup>1,2</sup>

<sup>1</sup>(东南大学 计算机科学与工程学院, 江苏 南京 211189)

<sup>2</sup>(计算机网络和信息集成教育部重点实验室(东南大学), 江苏 南京 211189)

通讯作者: 倪巍伟, E-mail: wni@seu.edu.cn



**摘要:** 针对已有的保护位置隐私路网  $k$  近邻查询依赖可信匿名服务器造成的安全隐患, 以及服务器端全局路网索引利用效率低的缺陷, 提出基于路网局部索引机制的保护位置隐私路网近邻查询方法. 查询客户端通过与 LBS 服务器的一轮通信获取局部路网信息, 生成查询位置所在路段满足  $l$ -路段多样性的匿名查询序列, 并将匿名查询序列提交 LBS 服务器, 从而避免保护位置隐私查询对可信第三方服务器的依赖. 在 LBS 服务器端, 提出基于路网基本单元划分的分段式近邻查询处理策略, 对频繁查询请求路网基本单元, 构建基于路网泰森多边形和 R<sup>\*</sup> 树的局部 Vor-R<sup>\*</sup> 索引结构, 实现基于索引的快速查找. 对非频繁请求路网基本单元, 采用常规路网扩张查询处理. 有效降低索引存储规模和基于全局索引进行无差异近邻查询的访问代价, 在保证查询结果正确的同时, 提高了 LBS 服务器端  $k$  近邻查询处理效率. 理论分析和实验结果表明, 所提方法在兼顾查询准确性的同时, 有效地提高了查询处理效率.

**关键词:** 路网; 位置隐私保护;  $k$  近邻查询; Voronoi-R<sup>\*</sup> 索引

**中图法分类号:** TP309

中文引用格式: 倪巍伟, 李灵奇, 刘家强. 基于 Voronoi-R<sup>\*</sup> 的隐私保护路网  $k$  近邻查询方法. 软件学报, 2019, 30(12): 3782-3797. <http://www.jos.org.cn/1000-9825/5583.htm>

英文引用格式: Ni WW, Li LQ, Liu JQ. Voronoi-R<sup>\*</sup>-based privacy-preserving  $k$  nearest neighbor query over road networks. Ruan Jian Xue Bao/Journal of Software, 2019, 30(12): 3782-3797 (in Chinese). <http://www.jos.org.cn/1000-9825/5583.htm>

## Voronoi-R<sup>\*</sup>-based Privacy-preserving $k$ Nearest Neighbor Query over Road Networks

NI Wei-Wei, LI Ling-Qi, LIU Jia-Qiang

<sup>1</sup>(School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

<sup>2</sup>(Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 211189, China)

**Abstract:** With the thriving of location based service, privacy-preserving nearest neighbor querying over road networks receives widespread attention. Most of existing methods highly depend on the trusted anonymous server and auxiliary server-side global index structure of the whole road networks. The trusted anonymous server is inclined to be the bottleneck of the querying system and the global index structure is commonly with low utilization. Concerning these problems, a new privacy-preserving  $k$  nearest neighbor querying schema is proposed leveraging local index schema at server side. Two rounds of handshakes are initiated between query user and the server side to avoid the dependence on any trusted anonymous server. At the first round, the query user generates anonymous query sequence satisfying  $l$ -diversity of road nodes at client side via initiating road sections request that locate within a special cloaking area. Thereafter, the anonymous query sequence and the number of target object  $k$  are submitted to the server to achieve the candidate query results. At the server, the whole road network is partitioned into a series of basic units. A hitting frequency guided segmented query processing strategy is proposed, which differentiates basic units with high hitting frequency from those ones with low hitting frequency. For those units with high hitting frequency, a local Voronoi-R<sup>\*</sup> index is designed and constructed. The  $k$  nearest neighbors of each location

\* 基金项目: 国家自然科学基金(61370077, 61772131)

Foundation item: National Natural Science Foundation of China (61370077, 61772131)

收稿时间: 2017-11-19; 修改时间: 2018-02-06; 采用时间: 2018-04-05

in anonymous query sequence can be easily achieved leveraging the new index, which promises higher querying processing efficiency. In contrast, the traditional server-side query strategy is applied to those units with low hitting frequency, which grasps nearest neighbors leveraging increment network expansion querying. This strategy can avoid redundant query process and index storage cost originated from global server side index structure, in parallel with no species difference query pattern based on the global index. Theoretical and empirical analysis demonstrates the effectiveness and efficiency of the proposed solution.

**Key words:** road network; location privacy protection;  $k$  nearest neighbor query; Voronoi-R\* index

近年来,随着位置服务(location based service,简称 LBS)的普及和人们对个体隐私的日益重视,保护位置隐私近邻查询得到了研究者的持续关注<sup>[1-7]</sup>.路网环境保护位置隐私近邻查询以其贴近真实生活和路网约束的复杂,成为保护位置隐私查询研究的重要方向.基本思想是:通过隐匿机制,将隐藏后查询者位置及查询请求提交 LBS 服务器,服务器返回候选 POI(point of interest)集合,供查询者从中甄选目标查询结果.从隐藏技术角度,这些方法可分为 4 类:位置干扰(location obstruction)、空间变换(space transformation)、空间混淆(spatial cloaking)和 PIR(private information retrieval)技术.

面向路网的保护位置隐私近邻查询主要采用空间混淆技术将查询者位置泛化为混淆区域(公路子网)提交服务器处理,通过扩大搜索空间返回候选解集,实现保护查询者位置隐私的近邻查询.例如:文献[8,9]利用可信第三方服务器对查询者的真实位置进行匿名,并将匿名区域及查询请求提交服务器,服务器扩展边界节点的邻接区域查找匿名区域的  $k$  近邻,这种方法具有保护强度可调节、服务器端处理可调控的优点;文献[10]提出利用可信匿名服务器构建路网泰森图,根据路网泰森单元(network voronoi polygon,简称 NVP)包含的路段数,采取不同匿名策略对用户真实位置进行匿名保护的方法;文献[11]采用 PIR 技术,提出了基于全局路网泰森图和 KD-Tree 的保护位置隐私近邻查询方法.基于 PIR 技术可以提供较高的隐私保护强度,但服务器端全局泰森图和 KD-Tree 索引构建过程复杂,存储消耗大,且  $k$  近邻查询时客户端与服务器端需要多轮迭代.服务器端主要采用增量式路网扩张法(incremental network expansion,简称 INE)和基于泰森图的近邻查找法实现路网近邻查询.增量式路网扩张法对查询者位置所在路网区域进行广度优先搜索,直到区域中近邻 POI 的数目满足查询需求;基于泰森图的近邻查询法利用路网泰森图查找最近邻,根据泰森多边形(Voronoi polygon)的邻接多边形查找其  $k$  近邻.搜索空间的扩大和复杂路网结构限制,加剧了 LBS 服务器端查询开销.已有研究大都通过在服务器端构建 POI 集合关于全局路网的索引结构,提高服务器端处理效率.

已有的基于空间混淆的保护位置隐私路网近邻查询方法主要存在以下问题.

- (1) 可信第三方服务器容易成为系统性能和隐私安全的瓶颈.
- (2) 路网规模庞大,使得服务器端全局路网索引规模较大,隐私保护需求导致服务器端近邻搜索范围扩大,查找路网索引的代价激增.
- (3) 查询目标 POI 在路网的位置分布不均衡,服务器端对全局路网索引的无差异使用,导致低频访问区域对应的索引结构利用率低,这部分索引结构增加了全局索引的规模,降低了路网高频访问区域查询的性能.

针对上述问题,提出了基于 Voronoi-R\* 索引的保护位置隐私路网  $k$  近邻查询方法 VRS-PNN(Voronoi-R\* based privacy-preserving  $k$  nearest neighbor query over road networks),主要贡献如下.

- (1) 提出了查询客户端与 LBS 服务器间两次交互的路网隐私保护近邻查询机制,实现不依赖可信第三方的路网位置隐私保护,避免依赖可信第三方导致的隐私泄露威胁.
- (2) 提出了基于路网区域查询请求频度的分段式查询处理策略,为频繁请求区域设计了 Vor-R\* 树局部路网索引结构,并定制了查询策略.在降低服务器端索引存储规模的同时,提升了索引利用率,提高了查询处理效率.
- (3) 实现了所提保护位置隐私路网近邻查询机制,并设计了实验验证算法的有效性和性能.

本文第 1 节描述问题并介绍相关概念.第 2 节介绍 VRS-PNN 方法的架构和流程.第 3 节介绍 VRS-PNN 方法的查询客户端处理方法.第 4 节介绍 VRS-PNN 方法服务器端查询处理过程.第 5 节对 VRS-PNN 方法的性能

进行分析.第6节进行实验分析.最后总结全文,并展望下一步工作.

## 1 问题描述及相关概念

### 1.1 问题描述

路网环境保护位置隐私近邻查询的关键问题表现在两个方面:查询者/发起者位置隐私保护效果以及隐私保护近邻查询处理的效率.

目前,已有研究主要采用空间混淆技术,通过可信匿名服务器对查询用户的位置进行匿名,并将匿名区域发送给 LBS 服务器,可信匿名服务器接收 LBS 服务器返回的候选集,并将结果反馈查询者.对可信匿名服务器的依赖,源于查询客户端不掌握路网信息,即便客户端获取 LBS 服务器反馈的包含真实查询目标的候选 POI 集合,也无法采用非路网环境直接计算查询者位置到各候选 POI 直线距离的方法甄别目标查询结果.然而可信匿名服务器难以寻找,且匿名服务器容易成为查询性能和隐私安全瓶颈.

在近邻查询处理方面,路网结构的复杂性和位置隐私保护需求使得 LBS 服务器端处理代价较大,在服务器端构造索引完成近邻查询是常用的方法,已有研究主要采用构造全局索引的方法,LBS 服务器端预先计算并存储整个路网区域 POI 的索引,LBS 服务器接收到查询请求时,通过查找索引完成查询.然而整个路网区域 POI 索引规模较大,且索引的构造未考虑查询目标在路网的实际分布,索引利用率低,导致 LBS 服务器端近邻查询效率方面的缺陷.

### 1.2 算法思想

针对前述路网环境保护位置隐私保护与查询处理依赖可信第三方存在的缺陷,考虑设计空间混淆以及查询处理与反馈机制,实现无需可信第三方介入的位置隐私保护和路网  $k$  近邻查询.针对 LBS 服务器端对全局索引结构无差异利用导致的查询处理效率低的问题,将服务器端全局路网划分为一组基本单元区域,设计区域局部索引结构,实现 LBS 服务器端兼顾区域查询频度的分段式查询处理,提高查询处理效率.

基本思路:查询客户端向 LBS 服务器发送区域路网查询请求,根据 LBS 服务器返回结果对查询者所在路段的节点进行空间混淆,以保护查询者位置隐私.具体方法是将查询者所在路段的节点作为目标对象进行  $l$ -路段多样性匿名<sup>[12]</sup>,生成包含查询者所在路段两端节点的一组路段节点序列.

LBS 服务器根据查询客户端提交的路段节点序列中节点所在区域选择不同的查询策略:对频繁请求的路网区域,设计 Vor-R\*索引结构,实现基于 Vor-R\*局部索引的快速近邻查找;对非频繁请求区域,采用常规 INE 方法查找近邻,解决全局索引规模大以及索引利用率低带来的查询效率低的缺陷.

### 1.3 相关概念

本文采用超图路网划分法<sup>[13]</sup>将 LBS 服务器端的路网区域划分为若干个基本单元.每个基本单元设置命中率,命中率随查询者对基本单元的请求次数更新.叙述方便起见,引入下述定义.

**定义 1(基本单元).** 服务器端采用超图路网划分法将路网划分为  $n$  个单元区域,即  $n$  个基本单元,单元区域内的所有 POI 构成该基本单元的 POI 集合.

**定义 2(基本单元命中次数).** 若查询者提交 LBS 服务器的匿名路网区域落在某基本单元区域内,该基本单元的命中次数加 1.

**定义 3(基本单元命中率).** 给定时间内,基本单元的命中率为该段时间基本单元的命中次数与 LBS 服务器接受查询请求总次数的比值.

**定义 4(路网泰森单元)<sup>[14]</sup>.** 某基本单元的 POI 集合为  $\{P_1, P_2, \dots, P_m\}$ ,  $P_i$  的路网 Voronoi 单元的定义如下:  $NVP(P_i) = \{q | \text{dist}(q, P_i) < \text{dist}(q, P_j), (j=1, 2, \dots, i-1, i+1, \dots, m), q \text{ 为基本单元所包含路段上的点}\}$ , 其中,  $\text{dist}(q, P_i)$  为  $q$  到  $P_i$  的最短路网距离.

**定义 5(基本单元的路网泰森单元集(network voronoi diagram,简称 NVD))<sup>[14]</sup>.** 某基本单元的 POI 集合为

$\{P_1, P_2, \dots, P_m\}$ , 该基本单元的路网泰森单元集  $NCD = \bigcup_{i=1}^m \{NVP(P_i)\}$ .

定义 6(邻接  $NVP^{[14]}$ ). 对  $P_1, P_2$  两个 POI, 若  $NVP(P_1)$  与  $NVP(P_2)$  存在相同的边界点, 则称  $NVP(P_1)$  与  $NVP(P_2)$  邻接,  $P_1$  与  $P_2$  互为 1 近邻.

### 2 VRS-PNN 算法处理流程

VRS-PNN 算法查询处理流程如图 1 所示, 步骤如下.

- (1) 查询者向 LBS 服务器发送包含其位置的匿名区域及参数  $m$  (LBS 需返回的最小路段数);
- (2) LBS 服务器查找匿名区域所在的基本单元(若匿名区域与多个基本单元存在交集, 选取与匿名区域重叠面积最大的基本单元), 将该基本单元的命中次数加 1 (匿名区域通常远小于基本单元, 即便匿名区域覆盖超过一个基本单元, 也可以进行类似处理, 更新相关基本单元命中数);
- (3) LBS 服务器查找匿名区域包含的所有路段, 生成并返回路段集合  $Segs$  (如果路段数小于  $m$ , 沿路网边界节点扩展, 直到路段数达到  $m$ );
- (4) 查询者从  $Segs$  中选取  $l$  个路段的节点  $P_i (i=1, 2, \dots, l)$ , 使得查询者所在路段满足  $l$ -路段多样性<sup>[12]</sup> (保证查询者所在路段被推断的可能性低于  $1/l$ ), 将查询请求序列  $Q(\{P_1, P_2, \dots, P_l\}, k)$  提交 LBS 服务器;
- (5) LBS 服务器接收查询序列  $Q$ , 分析  $Q$  中路段节点所在的基本单元, 对频繁请求单元, 利用预先构建的 Vor-R\* 索引, 查询计算路段节点的近邻 POI 集合; 否则, 使用路网增量扩张查找方法查询相应路段节点的近邻 POI 集合;
- (6) 将  $Q$  的近邻查询结果返回查询客户端;
- (7) 客户端根据查询者真实位置筛选准确路网  $k$  近邻.

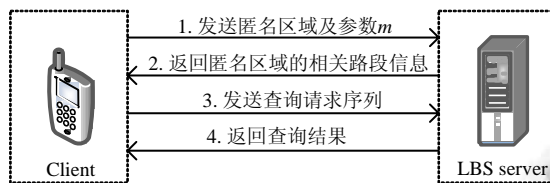


Fig.1 Querying process

图 1 处理流程

### 3 VRS-PNN 算法客户端处理方法

不依赖可信匿名服务器进行查询者位置隐私保护的难点在于查询客户端不掌握查询者所在区域的路网分布信息, 而路网信息存储在 LBS 服务器端, 因此查询者需要从服务器端获取局部路网信息, 以便完成自身位置的保护.

采取查询客户端向 LBS 服务器提交两次查询请求的方式完成保护位置隐私近邻查询.

- 第 1 阶段: 客户端向 LBS 服务器提交匿名区域  $\{AR, m\}$ , 获取局部路网信息, 其中,  $AR$  为客户端生成的查询者所在匿名区域,  $m$  为路段数, LBS 服务器查询  $AR$  覆盖的路段. 若匿名区域  $AR$  内路段数不小于  $m$ , 将  $AR$  内的路段信息返回客户端; 否则, 从匿名区域的边界节点进行路网扩张查询直到获取路段数大于  $m$ , 将扩充后的路段信息返回查询客户端.
- 第 2 阶段: 客户端根据 LBS 服务器反馈的路段生成包含查询者所在路段节点, 且满足  $l$ -路段多样性的匿名查询序列, 并将查询序列及近邻数  $k$  提交 LBS 服务器.

第 1 阶段: 随机生成的匿名区域如图 2 所示,  $P_u$  为用户真实位置, 通过两个随机数  $r_1, r_2$  确定正方形区域的左下端点  $P_1$  和右上端点  $P_2$ , 正方形匿名区域的大小需满足查询者对位置匿名强度的要求 (通过参数  $R$  设置).

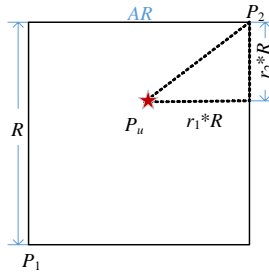


Fig.2 Anonymized area at client side

图2 客户端匿名区域

第2阶段:客户端收到LBS服务器返回的路段 $\{seg_1, seg_2, \dots, seg_m\}$ ,将其真实位置所在路段的两个端点加入待查询序列  $Query$  中,再随机选取  $l-2$  个路段节点加入  $Query$  中,生成查询者所在路段满足  $l$ -路段多样性的匿名查询序列,具体见算法 1.

**算法 1.** 用户位置隐匿算法.

输入:路段信息 $\{seg_1, seg_2, \dots, seg_m\}, l$ ;

输出:查询序列  $Query\{P_1, P_2, \dots, P_l\}$ .

- 1: add start node and ending node of  $seg_i$  where user locates in into  $Query$
- 2: **for**  $j=1$  to  $m$
- 3: generate unique random integer (1 to  $2*m$ ) for each endpoint of  $seg_j$  ( $i \neq j$ )  
/\*为每个节点生成一个唯一的随机数,各节点的随机数不同\*/
- 4: sort these random integers from minimum to maximum using quick sort algorithm
- 5: choose the first  $l-2$  nodes as elements of  $Query$
- 6: **return**  $Query$

图3为查询客户端根据服务器返回的区域路网信息生成的查询序列示意,矩形  $ABCD$  为客户端向服务器端发送的匿名区域,  $P_u$  为查询者真实位置,  $\{n_i | i=1, \dots, 7\}$  为客户端生成的查询序列,  $n_1$  和  $n_2$  为  $P_u$  所在路段的两个端点,  $n_i (3 \leq i \leq 7)$  为客户端在区域路网中随机选取的节点.

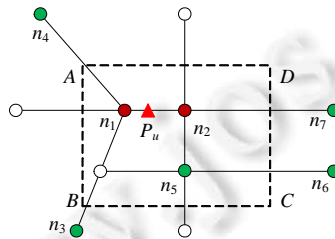


Fig.3 Node sequence illustration with  $l$ -diversity ( $l=7$ )

图3  $l$ -路段多样性节点序列生成示意图( $l=7$ )

**性质 1.** 查询者位置  $P_u$  所在路段节点  $n_1$  和  $n_2$  包含于向 LBS 服务器发送的匿名查询序列中,则 LBS 服务器返回的候选结果集  $S$  一定包含  $P_u$  的路网  $k$  近邻 POI.

证明:设  $n_1$  的路网  $k$  近邻 POI 集为  $M_1, n_2$  的路网  $k$  近邻 POI 集为  $M_2$ ; 将  $P_u$  的路网  $k$  近邻 POI 集合分为两个子集  $S_1$  和  $S_2, S_1$  为途经节点  $n_1$  的路网近邻 POI 集,  $S_2$  为途经节点  $n_2$  的路网近邻 POI 集. 假设存在  $Q_k, Q_k$  为  $P_u$  的第  $k$  近邻, 满足  $Q_k \notin M_1$  且  $Q_k \notin M_2$ . 若  $Q_k \in S_1$ , 设  $Q'_k$  为  $M_1 \cup M_2$  中离  $P_u$  第  $k$  近的 POI, 则有  $Q'_k \neq Q_k$ .

(1) 若  $Q'_k \in M_1$ , 存在以下两种情况:

- a)  $n_1$  到  $Q'_k$  的路径经过  $n_2$ , 则有  $dist(P_u, n_1) + dist(n_1, Q_k) < dist(n_1, Q'_k) - dist(P_u, n_1)$ , 推导得  $dist(n_1, Q_k) <$

$dist(n_1, Q'_k)$ , 则  $Q_k$  是  $n_1$  的路网  $k$  近邻, 这与  $Q'_k \in M_1, Q_k \notin M_1$  矛盾, 所以  $Q_k$  不存在;

b)  $n_1$  到  $Q'_k$  的路径不经过  $n_2$ , 则有  $dist(P_u, n_1) + dist(n_1, Q_k) < dist(n_1, Q'_k) + dist(P_u, n_1)$ , 推导得  $dist(n_1, Q_k) < dist(n_1, Q'_k)$ . 同上述证明过程, 与  $Q'_k \in M_1, Q_k \notin M_1$  矛盾, 故  $Q_k$  不存在.

(2) 同理可证  $Q'_k \in M_2$  时, 也不存在  $Q_k$ , 满足  $Q_k \notin M_1$  且  $Q_k \notin M_2$ , 且  $Q_k$  是  $P_u$  的第  $k$  个最近邻.

$Q_k \in S_2$  时, 与上述证明过程类似, 无论  $Q'_k \in M_1$  还是  $Q'_k \in M_2$ , 都不存在上述定义的  $Q_k$ .

由此可证,  $P_u$  的  $k$  近邻一定存在于  $M_1 \cup M_2$  中. □

查询客户端获取 LBS 服务器返回的候选结果集后, 只需将其所在路段节点的  $k$  近邻集合作为候选结果集进行筛选计算. 由于客户端缺乏路网信息, 无法计算其真实位置到各候选 POI 的路网距离, 因此 LBS 服务器返回的查询序列  $Q$  中添加每个查询位置到其各个近邻 POI 路径上的第一个路段节点信息, 以便客户端计算查询者位置距离候选 POI 的路网距离 (具体见第 4.3 节), 客户端根据 LBS 服务器返回的信息计算查询者的真实  $k$  近邻 POI. 客户端筛选查询结果的流程见算法 2.

**算法 2.**  $k$  近邻结果集筛选算法.

输入: 用户位置  $P_u$  所在路段的两个节点  $n_1$  和  $n_2$  的 kNN 结果集  $M_1$  和  $M_2$ ;

输出: 用户真实位置的  $k$  近邻结果集  $RS$ .

```

1:  $RS \leftarrow \emptyset$ 
2:  $map(p, dist(P_u, p)) \leftarrow \emptyset$ 
3: For each element  $poi$  in  $M_1$ 
4:   if  $poi.firstNode$  is  $n_2$  /* $n_1$  到  $poi$  最短路径经过的第 1 个路段节点*/
5:     add  $(poi, (poi.distance - dist(P_u, n_1)))$  to  $map$ 
6:   else add  $(poi, (poi.distance + dist(P_u, n_1)))$  to  $map$ 
7: End for
8: For each element  $poi$  in  $M_2$ 
9:   if  $poi$  is not in  $map$ 
10:    if  $poi.firstNode$  is  $n_1$  /* $n_2$  到  $poi$  最短路径经过的第 1 个路段节点*/
11:      add  $(poi, (poi.distance - dist(P_u, n_2)))$  to  $map$ 
12:    else add  $(poi, (poi.distance + dist(P_u, n_2)))$  to  $map$ 
13:   End If
14: End for
15: sort all the distance in  $map(p, dist(P_u, p))$  by quick sort algorithm
16:  $RS \leftarrow$  the topfirst  $k$  elements of  $map(p, dist(P_u, p))$ 
17: return  $RS$ 

```

## 4 VRS-PNN 服务器端处理方法

保护位置隐私路网近邻查询中, LBS 服务器处理代价大的重要原因是保护查询者位置隐私导致查询范围扩大. 考虑根据各基本单元的命中率, 对不同的基本单元采取不同的查询策略: 对命中率较高的基本单元, 构建基本单元泰森图, 并基于  $R^*$  树<sup>[15]</sup> 构建基本单元内 POI 关于泰森图的索引结构 Vor- $R^*$ . 由于  $R^*$  树在插入节点时采取组合策略, 使用多个参数作为选择标准, 而  $R$  树在插入节点时仅考虑面积参数; 在分割节点的过程中,  $R^*$  树采取拓扑分割策略基于周长选择分割轴, 最小化空间重叠大小; 因此,  $R^*$  树的空间重叠度相比于  $R$  树更小, 查询时效率优于  $R$  树, 可以有效提升频繁查询区域的处理效率. 对命中率较低的基本单元, 采用 INE 方法获取查询序列的  $k$  近邻.

### 4.1 Vor- $R^*$ 索引结构

以基本单元内的所有 POI 为对象生成基本单元的路网 Voronoi 图, 并对基本单元边界处的 NVP 予以标记,

以图 4 的路网基本单元为例,  $\{P_1, P_2, P_3, P_4, P_5\}$  为该单元的 POI 集合. 生成的 NVD 如图 5 所示, 与计算欧氏距离产生的 Voronoi 图不同, 路网中生成的 NVP 边界图形存在凹多边形.

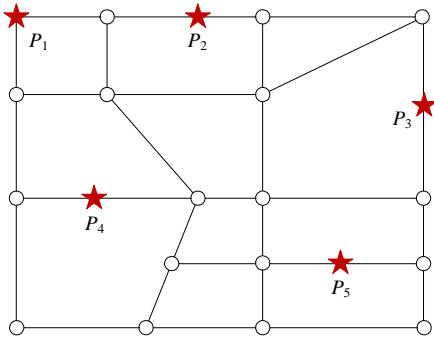


Fig.4 Road network  
图 4 路网图

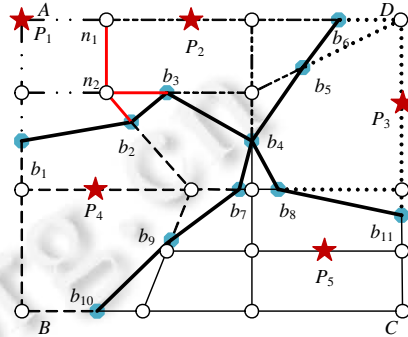


Fig.5 Illustration of NVD  
图 5 路网泰森单元集示意

NVP 由边界点所围成的区域构成, 表 1 为  $NVP(P_i)$  的边界点及邻接 POI 信息. 相邻的两个 NVP 可能存在重复区域, 如图 5 中的  $P_1$  和  $P_2$ :  $P_1$  的 NVP 边界为  $\{n_1, n_2, b_3, b_2, b_1, A\}$ ,  $P_2$  的 NVP 的边界为  $\{n_1, n_2, b_2, b_3, b_4, b_5, b_6\}$ , 两者存在交集区域  $\triangle b_2 n_2 b_3$ , 当查询序列  $Q$  中的节点位于  $n_2 b_2$  或者  $n_2 b_3$  时, 存在两个最近邻位置  $P_1$  和  $P_2$ .

Table 1 POI information

表 1 POI 信息

NVP(POI)	边界点列表	邻接 POI
$NVP(P_1)$	$\{n_1, b_3, b_2, b_1, A\}$	$P_2, P_4$
$NVP(P_2)$	$\{n_1, n_2, b_2, b_3, b_4, b_5, b_6\}$	$P_1, P_5, P_4, P_3$
$NVP(P_3)$	$\{b_6, b_5, b_4, b_8, b_{11}, D\}$	$P_2, P_5$
$NVP(P_4)$	$\{b_1, b_2, b_3, b_7, b_9, b_{10}, B\}$	$P_1, P_2, P_5$
$NVP(P_5)$	$\{b_4, b_7, b_9, b_{10}, b_{11}, b_8, C\}$	$P_4, P_2, P_3$

LBS 服务器对各个基本单元内的 POI 进行编号, 将编号及其位置信息存储到 POI 表中(见表 2). 在构建基本单元 NVD 的过程中, 将所有 NVP 信息保存到 NVD 表中(见表 3), 表中存储 POI 编号、 $NVP(P_i)$  边界点、 $NVP(P_i)$  内部路段节点、邻接 POI 信息, 每个 POI 的 NVP 边界点按顺时针顺序存储.

Table 2 POI table

表 2 POI 表

POI Id	位置坐标
1	$P(x_1, y_1)$
2	$P(x_2, y_2)$
...	...
N	$P(x_n, y_n)$

Table 3 NVD table

表 3 NVD 表

POI 编号(Id)	Border points	Inner nodes	Neighbors
1	$\langle b_{11}, b_{12}, \dots, b_{1m_1} \rangle$	$\langle n_{11}, n_{12}, \dots, n_{1s_1} \rangle$	$\langle Id_{11}, Id_{12}, \dots, Id_{1k_1} \rangle$
2	$\langle b_{21}, b_{22}, \dots, b_{2m_2} \rangle$	$\langle n_{21}, n_{22}, \dots, n_{2s_2} \rangle$	$\langle Id_{21}, Id_{22}, \dots, Id_{2k_2} \rangle$
...	...	...	...
N	$\langle b_{n1}, b_{n2}, \dots, b_{nm_n} \rangle$	$\langle n_{n1}, n_{n2}, \dots, n_{ns_n} \rangle$	$\langle Id_{n1}, Id_{n2}, \dots, Id_{nk_n} \rangle$

以基本单元内 POI 的 NVP 的最小外接矩形为叶子节点构建  $R^*$  树. Vor- $R^*$  树的构造过程如下.

- (1) 构建基本单元内所有 POI 的路网泰森图;

- (2) 创建空  $R^*$  树;
- (3) 对所有的 NVP, 执行下列操作.
  - ① 在  $R^*$  树中查找待插入对象 NVP<sub>i</sub> 的合适的叶子节点 Leaf;
  - ② 如果叶子节点 Leaf 存在足够空间, 将 NVP<sub>i</sub> 添加至节点 Leaf, 转步骤③; 否则分裂此节点, 并对 Leaf 和分裂后的新节点执行操作步骤③、步骤④;
  - ③ 根据节点向上调整树的结构;
  - ④ 若根节点分裂, 插入新的根节点, 分裂后的两个节点分别为树的孩子节点.

对图 6 所示区域划分, Vor-R\* 索引树结构如图 7 所示, 中间节点存储所有子节点矩形的轮廓最小外接矩形, 叶子节点存储最小外接矩形内部 NVP 对应的 POI 编号, 并指向 NVD 表 POI 编号所在位置. 由于  $R^*$  树的节点之间区域重叠区域较小, 服务器端查询路段节点最近邻时可以避免搜索多余的子树, 能够快速获得路段节点所在 NVP 的边界节点信息, 根据这些信息, 可以计算判定目标位置的准确所在 NVP.

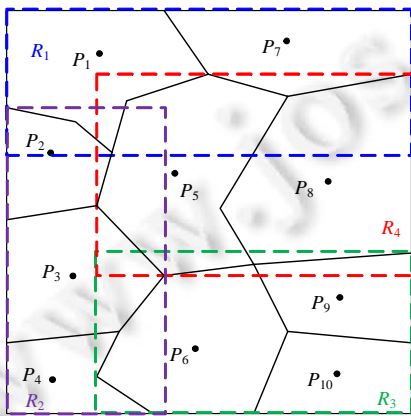


Fig.6 Region partition illustration of  $R^*$  tree  
图 6 路网泰森图  $R^*$  区域划分示意图

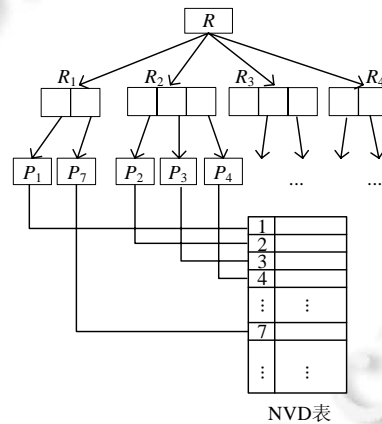


Fig.7 Structure illustration of index Vor-R\*  
图 7 Vor-R\* 索引结构图

#### 4.2 服务器端局部 Vor-R\* 索引更新

VRS-PNN 算法采用 LBS 服务器, 根据查询请求所在基本单元命中率进行个性化查找的分段处理策略, 因此需要动态维护各基本单元的命中率, 以便阶段性进行基本单元局部 Vor-R\* 索引的更新.

假设某时间段内, LBS 服务器接收查询请求总数为  $Amt$ , 启动索引更新的查询请求数阈值为  $A_0$ , 频繁请求基本单元的命中率阈值为  $\beta$ . 若基本单元的命中率不小于  $\beta$ , 该基本单元被标注为频繁请求单元; 否则, 该基本单元为非频繁请求单元, 为每个基本单元设置信号量  $flag$ , 标注基本单元是否已建立 Vor-R\* 索引.

LBS 服务器端索引构建策略如下.

- (1) LBS 服务器接收客户端发送的匿名区域  $AR$ , 并查找  $AR$  所在基本单元  $Reg_0$  (或查找与  $AR$  重叠面积最大的基本单元), 将  $Reg_0$  的命中次数加 1, 同时,  $Amt = Amt + 1$ ;
- (2) 若  $Amt < A_0$ , 返回步骤(1)继续接受客户端的查询请求; 否则, 进行基本单元局部 Vor-R\* 索引更新. 若基本单元的命中率大于等于  $\beta$ , 且  $flag$  为 false, 构建该基本单元的 Vor-R\* 索引, 并将该基本单元的  $flag$  设置为 true; 若基本单元的命中率小于  $\beta$ , 且  $flag$  为 true, 则删除该基本单元的 Vor-R\* 索引 (对已创建索引的基本单元, 可以通过设置合理的索引删除策略 (例如若连续多个更新周期, 该基本单元均评估为非频繁请求单元, 则删除该基本单元已创建局部索引), 缓解可能出现的部分基本单元反复删除、创建局部索引现象), 将  $flag$  设置为 false;
- (3) 将  $Amt$  和所有基本单元的命中次数均设为 0, 返回步骤(1).



LBS 服务器阶段性评估最近阶段客户端查询请求涉及基本单元的分布频度,创建或删除相应基本单元的 Vor-R\*索引,实现路网区域访问频度分布与查询处理策略的动态匹配.同时,通过合理的设置基本单元的 Vor-R\*索引结构删除策略,提高服务器端索引利用率,降低索引存储与维护的代价.

### 4.3 基于查询序列的近邻查询

服务器端接收到客户端的查询序列后,分别查询  $l$  个路段节点的最近邻.对每个路段节点,若其位于非频繁请求基本单元,使用 INE 方法查询其近邻;否则,基于 Vor-R\*树查询其近邻.

本节主要讨论落在频繁请求基本单元中的路段节点的近邻查询过程.查询 Vor-R\*树可以得到路段节点所在的最小外接矩形,由于 NVP 的最小外接矩形存在重叠区域,所以搜索树结构可能会得到多个最近邻 POI,而每个 POI 的 NVP 的边界点构成(凹/凸)多边形,将 NVP 内部的路段封装起来,若路段节点在多边形内部,则此多边形内部的 POI 为其最近邻.主要流程见算法 3.

**算法 3.** 最近邻查询算法.

输入:路段节点位置  $P_u(x,y)$ , Vor-R\*树;

输出:最近邻 POI 结果集  $R$ .

- 1: Search Vor-R\* and get all leaf nodes that  $P_u$  locates in
- 2:  $NN\_Candidates \leftarrow$  the corresponding pois of leaves
- 3:  $R \leftarrow \emptyset$
- 4: **For**  $poi$  in  $NN\_Candidates$
- 5:   **if**  $P_u$  locates in the  $NVP(poi)$  /\*计算  $P_u$  是否在  $NVP(poi)$  的边界点确定的多边形内\*/
- 6:     add  $poi$  to  $R$
- 7:   **End if**
- 8: **End for**
- 9: **Return**  $R$

利用一个点的  $k$  近邻一定存在于其  $k-1$  近邻的邻接 POI 集合的性质<sup>[16]</sup>,计算路段节点的  $k$  近邻.LBS 服务器查找 Vor-R\*树获得查询序列中路段节点最近邻后,根据上述性质计算路段节点的候选  $k$  近邻集合.考虑路段节点的  $k$  近邻可能位于不同基本单元.为了计算结果的准确性,在计算  $k$  近邻候选集时增加对 NVP 是否位于基本单元边界的判定.计算路段节点的  $k$  近邻候选集流程如下.

- (1) 利用 Vor-R\*查询路段节点  $P$  的最近邻集合  $S^i(i=1)$ ,将  $S^i(i=1)$ 添加至候选集  $temp\_list$  中;
- (2) 对  $S^i$ 中所有的 POI,执行步骤(3)、步骤(4);
- (3) 若 POI 所在 NVP 已被标记为边界 NVP,使用 INE 方法向另一个基本单元扩张查找此 POI 的  $(k-i)$ 近邻集合  $S'$ ,并保存  $P$  到其近邻 POI 的最短路径距离,将  $S'$ 添加至候选集  $temp\_list$ ;
- (4) 若 POI 所在的 NVP 未被标记为边界 NVP,则将此 POI 的邻接 POI(已存在于  $temp\_list$  的 POI 不计入内)分别添加至  $temp\_list$  和  $S^{i+1}$  中;
- (5) 如果  $i < k$ ,则  $i=i+1$ ,执行步骤(2);否则,结束候选集计算过程.

在筛选候选集、计算路段节点准确  $k$  近邻的过程中,需要计算路段节点到各个候选 POI 的最短路径距离,然后选取最近的  $k$  个 POI 作为路段节点的准确  $k$  近邻.在计算路段节点到候选 POI 的距离时,LBS 服务器预先计算 NVP 边界节点间的距离,以降低路段节点到 POI 的距离计算量.

在查询最近邻 POI 的基础上,进一步查找某个路段节点  $k$  近邻 POI 的主要流程见算法 4.

**算法 4.** KNN 查询算法.

输入:路段节点  $P_u(x,y)$ ;

输出: $k$  近邻 POI 结果集  $RS$ .

- 1: Compute the nearest POI  $poi_1$  of  $P_u$
- 2: Add  $poi_1$  to the list of knn poi candidates  $C$

```

3: temp←poi1, ithNNs←poi1
4: For i from 1 to k-1
5:   For ele in ithNNs
6:     If ele not exists in C
7:       If NVP(ele) is not boundary of current region unit
8:         add the neighbors of POI ele which not exist in C to temp
9:       Else
10:        otherReg←compute the (k-i) nearest neighbors of POI ele and distance using INE
11:        add the elements of otherReg which not exist in C to C
12:      End If
13:      add elements of temp which not exist in C to C
14:      ithNNs←temp
15: temp←∅
16:   End If
17: End For
18: End For
19: For each e in C
20:   compute the nearest distance from Pu to e, and add this distance to distList
21: End For
22: sort distList from minimum to maximum via Quick Sort algorithm /*快速排序*/
23: RS←Get k pois corresponding to the first k distance of distList
24: Return RS
    
```

服务器向客户端返回  $l \times k$  个 POI 位置信息,而客户端提交的  $l$  个位置在同一匿名区域,它们的  $k$  近邻可能存在交集.为了避免重复的近邻 POI 信息传输,使用第 4.1 节所定义的 POI 表结构.服务器端返回结果时,返回各目标位置近邻的 Id 序列以及 Id 序列对应的 POI 信息;同时,服务器端在计算路段节点的  $k$  近邻 POI 时,保存节点到各近邻 POI 的最短路径距离以及到近邻 POI 最短路径所经过的第 1 个路段节点(便于客户端计算准确近邻),与  $k$  近邻 POI 查询结果一起反馈给用户.图 8 为某基本单元的部分路网结构, $n_0$  为客户端所提交的某个路段节点, $A$  为  $n_0$  的一个近邻 POI, $n_0$  到  $A$  的最短路径为  $n_0n_1n_2A$ ,则  $n_1$  为  $n_0$  到路网近邻点  $A$  路径的第 1 个节点.

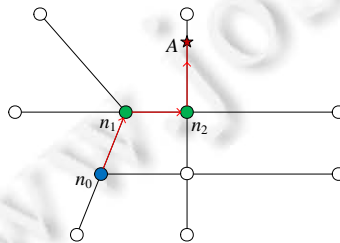


Fig.8 Road distance illustration between query location and its nearer POI

图 8 查询点与近邻 POI 的路网最短路径

表 4 和表 5 为服务器反馈查询客户端的查询结果.表 4 中,每个路段节点的路网  $k$  近邻序列由  $k$  个三元组组成,元素分别为近邻 POI、MinDist 为路段节点到此 POI 的最短路径距离、FirstNode 为路段节点到此 POI 的最短路径中经过的第 1 个路段节点.表 5 为表 4 中各 POI 和路段节点与实际位置坐标的映射.

**Table 4**  $k$  nearest neighbors of  $l$  road nodes in  $Q$  ( $l=4, k=3$ )**表 4**  $Q$  中  $l$  个路段节点的路网  $k$  近邻集( $l=4, k=3$ )

路段节点	$k$ 近邻序列 $\{(POI, MinDist, FirstNode)\}$
$P_1$	$\{(9,5,1), (8,7,2), (4,8,3)\}$
$P_2$	$\{(13,4,4), (12,6,2), (11,7,2)\}$
$P_3$	$\{(10,3,5), (9,7,1), (4,9,1)\}$
$P_4$	$\{(4,6,6), (8,8,5), (9,10,7)\}$

**Table 5** Location of POIs and road nodes ( $l=4, k=3$ )**表 5** POI 和路段节点的坐标映射表( $l=4, k=3$ )

POI	位置坐标	$FirstNode$ 编号	位置坐标
4	(19,6)	1	(10,20)
8	(30,19)	2	(20,15)
9	(40,40)	3	(40,25)
10	(6,3,69)	4	(35,50)
11	(70,45)	5	(55,10)
12	(58,6)	6	(60,60)
13	(57,57)	7	(57,80)

## 5 算法性能分析

本节从 LBS 服务器端和查询客户端计算复杂度的角度,分析 VRS-PNN 方法的效率.对 LBS 服务器端以单个路段节点为查询对象,分别分析频繁请求基本单元和非频繁请求基本单元的处理效率.

对于频繁请求的基本单元,因其索引 Vor-R\* 是预构建好的,每次查询只需查找索引,基于 Vor-R\* 查询路网最近邻的时间复杂度为  $O(\log n)$ ,  $n$  为基本单元内的 POI 数目.查询路网  $k$  近邻的计算量主要消耗在计算路段端点到候选 POI 的最短路径距离,如第 4.3 节所述,通过在 LBS 服务器预计算 NVP 各边界节点之间的距离,计算路段节点到其候选 POI 的最短距离时只需进行简单的求和、比较计算.考虑路网实际情况,每个 NVP 的边界节点个数为常量,所以计算路段节点到候选 POI 最短路径距离的时间复杂度为  $O(C)$ .文献[16]提出,每个 NVP 的邻接 NVP 的数目平均不超过 6.根据此性质,路段端点的  $k$  近邻候选结果集平均大小为  $6k$ .本文采用快速排序计算准确路网  $k$  近邻,时间复杂度为  $O(6k \log k)$ .综上,计算每个路段端点路网  $k$  近邻的复杂度为  $O(\log n + kC + 6k \log k)$ ,即  $O(\log n + k \log k)$ .上述过程为路段节点的真实  $k$  近邻全都位于路段节点所在基本单元的情形;若路段节点的部分真实  $k$  近邻位于其他基本单元,最坏情况为路段节点所在的 NVP 为边界 NVP,此时的计算量包括查询一次 Vor-R\* 树和利用 INE 方法查找单个路段节点  $k$  近邻.

对于非频繁请求基本单元,采用 INE 方法查询  $k$  近邻的时间受 POI 分布密集程度影响<sup>[17]</sup>(用  $|N|/|S|$  描述分布密度,  $|N|$  为基本单元内 POI 的个数,  $|S|$  对应基本单元内路段数).文献[18]验证了 POI 分布越密集, INE 策略越高效:若  $|N|/|S| \geq 1$ , 查询复杂度为  $O(k)$ ; 否则, INE 方法效率较低, 查询复杂度为  $O(k * |S|/|N|)$ .

上述复杂度分析为单个路段节点的近邻查询复杂度, VRS-PNN 服务器端需要对查询序列进行处理, 总体时间复杂度受各个路段节点所处基本单元影响. 假设频繁请求基本单元的路段节点近邻查询复杂度为  $O_1$ , 非频繁请求单元的路段节点近邻查询复杂度为  $O_2$ , 处理查询序列的最坏时间复杂度为  $l * \max(O_1, O_2)$ .

客户端计算主要集中在从候选集筛选准确路网  $k$  近邻的过程, 由于客户端接收查询序列  $Q$  的  $k$  近邻候选 POI 集的同时, 也获取  $Q$  中每个路段节点到其所有  $k$  近邻 POI 的最短路径经过的第一个路段节点, 根据算法 2, 计算时间主要消耗在快速排序过程, 时间复杂度为  $O(k \log k)$ . 整个查询过程, 查询客户端和 LBS 服务器间发生两轮通信. 首先, 客户端向 LBS 服务器发送区域路网信息请求过程中, 通信代价为  $O(1)$ , LBS 服务器向查询客户端反馈路网信息的通信消耗为  $O(m)$  ( $m$  为路段数量参数); 随后, 查询客户端向 LBS 服务器发送查询序列的通信代价为  $O(l)$  ( $l$  为查询序列大小), LBS 服务器将候选结果返回给客户端的通信代价为  $O(lk)$ . 综上, VRS-PNN 方法中查询客户端和 LBS 服务器间的总通信代价为  $O(lk+m)$ .

### 6 实验分析

本节对 VRS-PNN 的效率进行实验分析,实验数据采用美国加利福尼亚的路网数据(路段节点 21 048 个,覆盖 21 689 条道路,http://www.cs.utah.edu/~lifeifei/SpatialDataset.htm),选取包括 2 024 个路段节点、5 266 条道路的路网子区域(内含 1 000 个 POI).实验环境配置如下:Windows7 系统,3.4GHz Intel Core i7 处理器,内存容量 8GB.

首先,设计实验对本文提出的 Vor-R\* 树索引结构与已有基于路网泰森图的 R 树索引的查询性能进行对比分析,构建基于基本单元的 Vor-R\* 树索引与 Vor-R 树索引,分别基于两种索引查询目标 POI 的最近邻,对其进行性能比较.由于 R\* 树相比于 R 树的空间覆盖和重叠度均比较小,其查询效率更高.如图 9 所示,伴随着基本单元 POI 数目的变化,Vor-R\* 树索引查询性能始终优于 Vor-R 树索引.

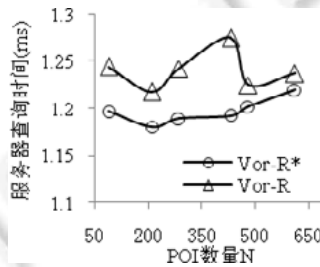


Fig.9 Index effectiveness comparison between Vor-R\* and Vor-R

图 9 Vor-R\* 与 Vor-R 索引性能比较图

进一步分析主要参数对 VRS-PNN 方法性能的影响.图 10 为 VRS-PNN 方法中 LBS 服务器处理效率随参数 l、近邻参数 k 和 POI 数目 N 的变化趋势.由图可见,服务器端处理代价随参数 l 的增长几近呈线性增长,与 k 也呈线性关系.

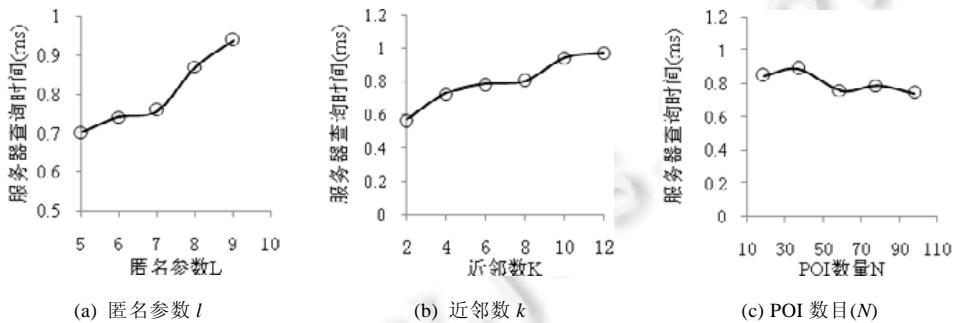


Fig.10 Server-side querying cost with increasing l, k and number of POI

图 10 参数 l, k, POI 数目对服务器端处理效率的影响

图 11 所示为客户端与服务器间的通信代价受参数 l 和 k 影响的变化曲线.由图 11 可见,其通信代价随 l 和 k 缓慢增长.

将 VRS-PNN 与文献[19]基于路网泰森图的保护位置隐私近邻查询方法 VDNN 进行性能对比,文献[19]同样采用区域混淆方法,将匿名区域与路网交点的近邻查询结果返回给客户端供其筛选最近邻结果.对两种方法分别顺次串行发起相同的 10 组查询,取处理时间的均值进行对比,图 12 为 VRS-PNN 方法与 VDNN 方法通信量对比分析图,参数 l 在 VDNN 方法中等同于匿名区域与路网中路段的交点数.不同于 VDNN 直接返回所有交点的 k 近邻,VRS-PNN 方法生成候选 POI 集合过程合并查询序列中路段节点的近邻 POI,有效规避了重复 POI 传输,VRS-PNN 方法的通信代价低于 VDNN.

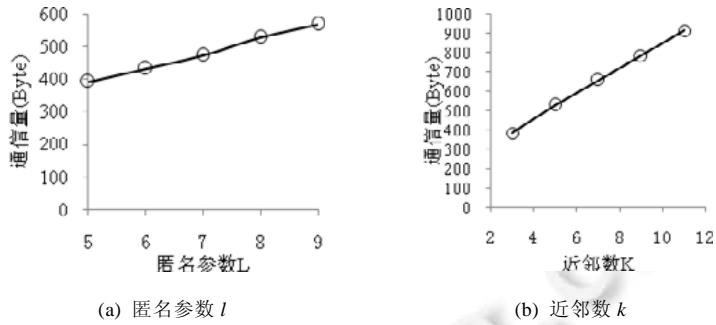


Fig.11 Communication cost with increasing  $l$  and  $k$

图 11 参数  $l, k$  对通信代价的影响

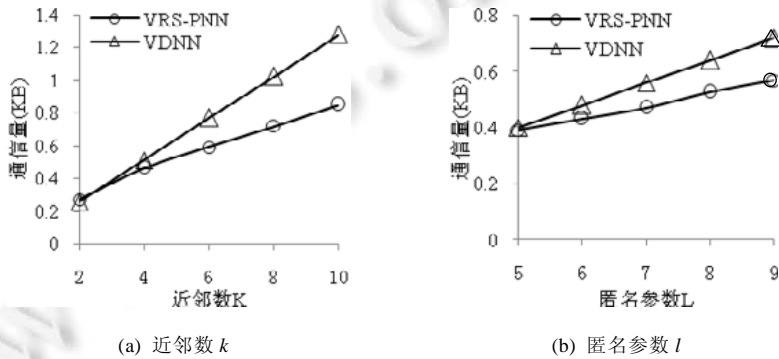


Fig.12 Commucation cost comparing with VDNN algorithm

图 12 VRS-PNN 与 VDNN 方法的通信量比较

图 13 为 VRS-PNN 与 VDNN 方法服务器端  $k$  近邻查找过程性能对比.由于 Vor-R\* 树查询性能优于 Vor-R 树,且 VRS-PNN 方法仅对频繁查询区域构建局部 Vor-R\* 索引,而 VDNN 方法在服务器端构建全局路网 R 树索引,因而 VRS-PNN 方法相较 VDNN 方法能够充分降低频繁检索区域的路网近邻 POI 查询消耗.由图易见, VRS-PNN 在服务器端查询性能优于 VDNN 方法.

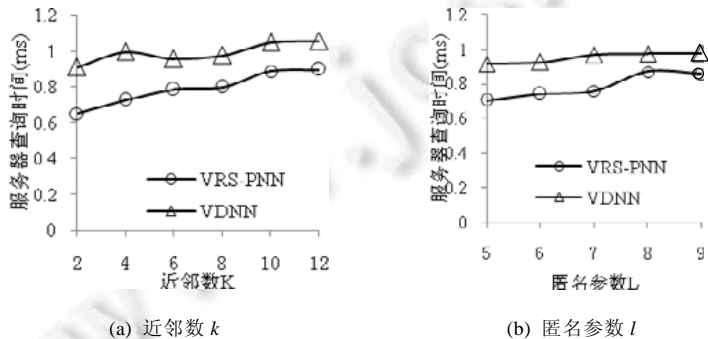


Fig.13 Querying workload comparing with VDNN algorithm

图 13 VRS-PNN 与 VDNN 方法的服务器查询性能比较

对 LBS 服务器端分段式查询处理策略的效果进行分析,采用客户端向服务器端同时随机发送 100 次不同位置的近邻查询的方式,对比 VRS-PNN 方法和 VDNN 方法的服务器端处理效率.VRS-PNN 方法的频繁请求阈值设为 0.6,统计得出这些查询共涉及服务器端 20 个基本单元,实验结果如图 14 所示,VRS-PNN 方法的服务器

端查询处理效率显著优于 VDDN 方法.原因在于:LBS 服务器接收到大量查询时,不同于 VDDN 对所有查询请求基于全局路网索引的无差异检索处理,VRS-PNN 方法区分频繁路网区域和非频繁路网区域,并动态维持各个频繁基本单元的 Vor-R\*索引,对涉及频繁基本单元的大量查询,利用各基本单元的局部 Vor-R\*索引进行查找,避免了对大量查询无差别采用基于全局索引的近邻查找导致的查询代价激增.

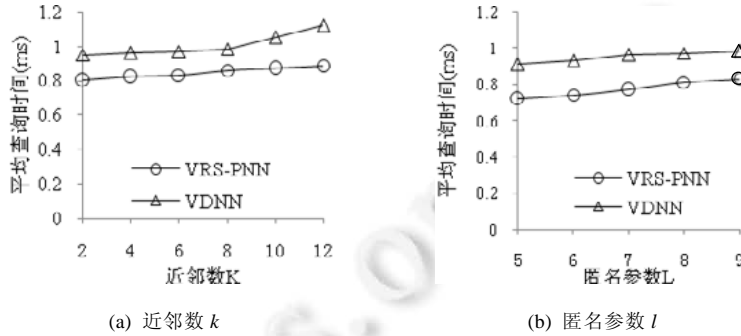


Fig.14 Server side batch query performance comparison between VRS-PNN and VDDN

图 14 VRS-PNN 方法与 VDDN 方法服务器端批量查询效率对比

最后,对 VRS-PNN 方法的客户端处理性能进行实验分析.VRS-PNN 方法客户端主要开销是对候选结果集的筛选,图 15 为客户端处理效率随参数  $k$ 、POI 数目  $N$  的变化趋势.由图可知:客户端处理时间随近邻数  $k$  的增长缓慢上升,维持在数十微秒级别,且处理时间受 POI 数目  $N$  的影响不显著,所提方法适用于目前手机、导航仪等客户端设备.

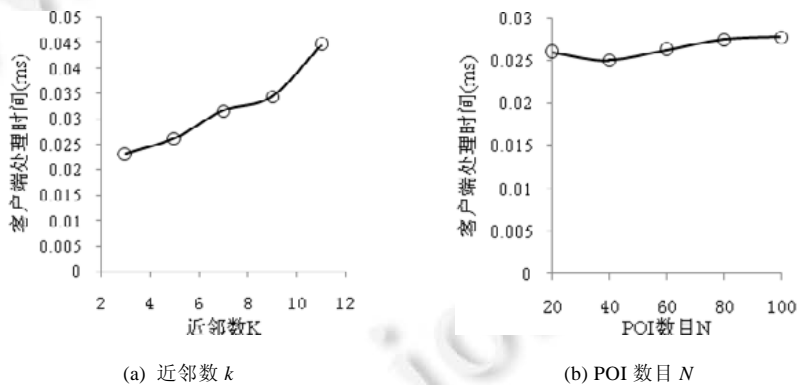


Fig.15 Client-site cost with increasing  $k$  and the number of POIs

图 15 参数  $k$ 、POI 数目对客户端处理效率的影响

## 7 结 论

本文提出了基于局部索引机制的保护位置隐私路网  $k$  近邻查询方法 VRS-PNN,查询客户端通过与 LBS 服务器的一轮通信,获取局部路网信息,生成查询位置所在路段满足  $l$ -路段多样性的匿名查询序列,并将匿名查询序列提交 LBS 服务器,从而避免保护位置隐私查询对可信第三方服务器的依赖.在 LBS 服务器端,提出基于路网基本单元划分的分段式近邻查询处理策略,对频繁查询请求路网基本单元,构建基于路网泰森多边形和 R\*树的局部 Vor-R\*索引结构,实现基于索引的快速查找.对非频繁请求路网基本单元,采用常规路网扩张查询处理.有效降低了索引存储规模和基于全局索引进行无差异近邻查询的访问代价,在保证查询结果正确的同时,提高了 LBS 服务器端  $k$  近邻查询处理效率.下一步,将对所提索引策略及查询处理方法在保护位置隐私连续路网近邻

查询中的应用进行研究.

## References:

- [1] Yi X, Paulet R, Bertino E, Varadharajan V. Practical approximate  $k$  nearest neighbor queries with location and query privacy. *IEEE Trans. on Knowledge & Data Engineering*, 2016,28(6):1546–1559. [doi: 10.1109/TKDE.2016.2520473]
- [2] Utsunomiya Y, Toyoda K, Sasase I. LPCQP: Lightweight private circular query protocol for privacy-preserving  $k$ -NN search. In: Uluagac AS, ed. *Proc. of the IEEE Consumer Communications and NETWORKING Conf. Las Vegas: IEEE*, 2015. 65–70. [doi: 10.1109/CCNC.2015.7157947]
- [3] Tian S, Cai Y, Zheng Q. A hybrid approach for privacy-preserving processing of  $k$ NN queries in mobile database systems. In: He Q, Iyengar A, eds. *Proc. of the ACM Int'l Conf. on Information & Knowledge Management*. San Francisco: ACM Press, 2013. 1161–1164. [doi: 10.1145/2505515.2507814]
- [4] Ni WW, Chen X. Users' privacy preferences supporting in location privacy-preserving nearest neighbor query. *Ruan Jian Xue Bao/ Journal of Software*, 2016,27(7):1805–1821 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5053.htm> [doi: 10.13328/j.cnki.jos.005053]
- [5] Zhang XJ, Gui XL, Wu ZD. Privacy preservation for location-based services: Asurvey. *Ruan Jian Xue Bao/Journal of Software*, 2015,26(9):2373–2395 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4857.htm> [doi: 10.13328/j.cnki.jos.004857]
- [6] Gao Z, Zhu H, Liu Y, *et al.* Location privacy in database-driven cognitive radio networks: Attacks and countermeasures. *Proceedings—IEEE INFOCOM*, 2013,12(11):2751–2759.
- [7] Li LQ. Research on location privacy preserving query techniques in road network [MS. Thesis]. Nanjing: Southeast University, 2018 (in Chinese with English abstract).
- [8] Kim HI, Chang JW.  $k$ -nearest neighbor query processing algorithms for a query region in road networks. *Journal of Computer Science and Technology*, 2013,28(4):585–596. [doi: 10.1007/s11390-013-1359-8]
- [9] Ni WW, Chen X, Ma ZX. Location privacy preserving  $k$  nearest neighbor query method on road network in presence of user's preference. *Chinese Journal of Computers*, 2015,38(4):884–896 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2015.00884]
- [10] Pan X, Wu L, Hu CJ. A privacy protection algorithm based on network Voronoi graph over road networks. *Journal of Computer Research and Development*, 2015,52(12):2750–2763 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2015.20140602]
- [11] Wang L, Ma R, Meng X. Evaluating  $k$  nearest neighbor query on road networks with no information leakage. In: Wang J, ed. *Proc. of the Web Information Systems Engineering (WISE 2015)*. Miami: Springer-Verlag, 2015. 508–521. [doi: 10.1007/978-3-319-26190-4\_34]
- [12] Wang T, Liu L. Privacy-Aware mobile services over road networks. *Proc. of the VLDB Endowment*, 2010,2(1):1042–1053. [doi: 10.14778/1687627.1687745]
- [13] Xu Y, Tan G. An offline road network partitioning solution in distributed transportation simulation. In: Boukerche A, Cahill V, eds. *Proc. of the Int'l Symp. on Distributed Simulation and Real Time Applications*. Dublin: IEEE, 2012. 210–217. [doi: 10.1109/DS-RT.2012.38]
- [14] Kolahdouzan M, Shahabi C. Voronoi-Based  $K$  nearest neighbor search for spatial network databases. In: Nascimento MA, ed. *Proc. of the 30th Int'l Conf. on Very Large Data Bases*. Toronto: VLDB Endowment, 2004. 840–851. [doi: 10.1016/b978-012088469-8/50074-7]
- [15] Beckmann N, Kriegel HP, Schneider R, Seeger B. The  $R^*$ -tree: An efficient and robust access method for points and rectangles. In: Jagadish HV, ed. *Proc. of the ACM SIGMOD Conf.* New York: ACM Press, 1990. 322–331. [doi: 10.1145/93605.98741]
- [16] Jing Y, Hu L, Ku WS, Shahabi C. Authentication of  $k$  nearest neighbor query on road networks. *IEEE Trans. on Knowledge & Data Engineering*, 2014,26(6):1494–1506. [doi: 10.1109/TKDE.2013.174]

- [17] Papadias D, Zhang J, Mamoulis N, *et al.* Query processing in spatial network databases. In: Freytag JC, ed. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB 2003). Berlin: VLDB Endowment, 2003. 802–813. [doi: 10.1007/springerreference\_63121]
- [18] Abeywickrama T, Cheema MA, Taniar D.  $k$ -nearest neighbors on road networks: A journey in experimentation and in-memory implementation. Proc. of the VLDB Endowment, 2016,9(6):492–503. [doi: 10.14778/2904121.2904125]
- [19] Um JH, Jang M, Chang JW. A privacy protected kNN query processing algorithm based on network Voronoi diagram in spatial networks. IEICE Trans. on Information & Systems, 2014,E97.D(7):1735–1745. [doi: 10.1587/transinf.E97.D.1735]

#### 附中文参考文献:

- [4] 倪巍伟,陈萧.保护位置隐私近邻查询中隐私偏好问题研究.软件学报,2016,27(7):1805–1821. <http://www.jos.org.cn/1000-9825/5053.htm> [doi: 10.13328/j.cnki.jos.005053]
- [5] 张学军,桂小林,伍忠东.位置服务隐私保护研究综述.软件学报,2015,26(9):2373–2395. <http://www.jos.org.cn/1000-9825/4857.htm> [doi: 10.13328/j.cnki.jos.004857]
- [7] 李灵奇.路网环境隐私保护近邻查询技术研究[硕士学位论文].南京:东南大学,2018.
- [9] 倪巍伟,陈萧,马中希.支持偏好调控的路网隐私保护  $k$  近邻查询方法.计算机学报,2015,38(4):884–896. [doi: 10.3724/SP.J.1016.2015.00884]
- [10] 潘晓,吴雷,胡朝君.道路网络上基于网络 Voronoi 图的隐私保护算法.计算机研究与发展,2015,52(12):2750–2763. [doi: 10.7544/issn1000-1239.2015.20140602]



倪巍伟(1979—),男,江苏淮安人,博士,教授,博士生导师,CCF 专业会员,主要研究领域为数据隐私安全保护,复杂数据管理.



刘家强(1992—),男,硕士生,主要研究领域为隐私保护位置服务.



李灵奇(1994—),女,硕士生,主要研究领域为隐私保护位置服务.