计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(8):176−186 [doi: 10.15888/j.cnki.csa.009599]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

CoSTUR: 面向用户评级的空间文本竞争选址①

李晨伟, 默梓鹏, 赵梦霏

(西安工程大学 计算机科学学院, 西安 710048)
通信作者: 李晨伟, E-mail: strivelcw@163.com

摘　要: 随着 GPS 定位技术和移动互联网的发展, 各类 LBS (location-based service) 应用积累了大量带有位置和文本标记的空间文本数据, 这些数据广泛应用于市场营销、城市规划等设施选址决策中. 空间文本选址的目标是从候选位置集合中挖掘最佳地点新建设施, 以期影响最多空间文本对象, 如用户或车辆等, 其中空间距离越接近且文本越相似则影响力越大. 现有方案未考虑现实普遍存在的同行竞争, 也忽略了用户对设施的评价因素. 为更合理地在同行竞争环境结合用户评级进行选址决策, 本文提出新的空间文本竞争选址问题 CoSTUR. 通过引入权衡影响的确定性和数量的阈值, 解决传统模型中对象只能被单一设施影响的局限, 建模了用户可能同时受多个设施影响的真实情况. 借鉴经典的竞争均分模型, 实现了不同评级设施间竞争量化. 为降低大规模数据导致的高昂计算代价, 构建了新型空间文本索引结构 TaR-tree, 并结合阈值设计基于影响范围的两个剪枝策略, 实现基于分支定界思想的空间连接和范围查询两种方案. 在真实和合成数据集上的实验结果显示, 相比基线算法计算效率能够提升近一个量级, 说明提出方法的有效性.

关键词: 空间文本数据; 选址问题; 空间文本索引; 竞争影响; 多设施影响; 用户评级

引用格式: 李晨伟,默梓鹏,赵梦霏.CoSTUR: 面向用户评级的空间文本竞争选址.计算机系统应用,2024,33(8):176–186. http://www.c-s-a.org.cn/1003-3254/9599.html

CoSTUR: Competitive Spatio-textual Location Selection Based on User Rating

LI Chen-Wei, MO Zi-Peng, ZHAO Meng-Fei

(School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: With the development of GPS positioning technology and mobile Internet, various location-based services (LBS) applications have accumulated a large amount of spatio-textual data with location and text markup. These data are widely used in location selection decision-making scenarios such as marketing and urban planning. The goal of spatio-textual location selection is to mine the optimal locations from a given candidate set to build new facilities to influence the largest number of spatio-textual objects, such as people or vehicles, where the closer the spatial location and the more similar the text, the greater the influence. However, existing solutions not only fail to consider prevalent peer competition in real life but also ignore user evaluation factors for facilities. To make more reasonable location selection decisions in a peer competition environment combined with user ratings, this study proposes a more rational spatio-textual location selection problem, CoSTUR. To solve the limitation in traditional models where objects can only be influenced by a single facility, a threshold that makes a trade-off between the certainty and quantity of facility influence on objects is introduced, which also models the real-world situation in which multiple facilities could simultaneously influence a specific user. Based on the classical competitive equalization model, quantification of competition among facilities with different ratings is achieved. To reduce the high computational cost for large volumes of data, a novel spatio-textual index

① 基金项目: 陕西省自然科学基础研究计划 (2023-JC-YB-558)
收稿时间: 2024-02-17; 修改时间: 2024-03-19; 采用时间: 2024-04-10; csa 在线出版时间: 2024-07-03
CNKI 网络首发时间: 2024-07-08

structure, TaR-tree, is constructed and two pruning strategies based on influence range are designed with a combination of thresholds to achieve two branch-and-bound solutions for spatial connectivity and range queries. Experimental results on real and synthetic datasets demonstrate that the computational efficiency can be improved by nearly one order of magnitude compared to baseline algorithms, verifying the effectiveness of the proposed method.

Key words: spatio-textual data; location selection; spatio-textual index; competitive influence; simultaneous influence by multiple facilities; user rating

空间大数据是带有位置信息的大规模数据, 对其进行分析和利用能够为企业、政府和个人提供非常有价值的信息服务, 辅助做出更明智的空间决策. 推动空间大数据技术的发展, 对我国建设智慧城市、发展智能交通与优化城市资源配置具有重大意义. 空间选址是空间大数据领域的基本应用, 旨在根据用户与设施的空间数据从给定的候选位置集合中选择最优的一个或一组位置, 使得决策最优化或用户收益最大化. 随着 GPS 定位技术、移动网络技术的兴起, 以及智能设备的普及, 基于位置的服务 (location-based service, LBS) 迅速发展并得到广泛的应用, 网络上积累了大量带有位置和文本信息的空间文本数据. 近年来, 空间关键字查询[1]在时空数据领域得到了广泛的关注. 具体地, 给定空间位置和一系列关键字, 空间关键字查询将返回满足特定查询条件的一个或多个空间文本对象. 例如, 检索国家图书馆 1 km 内具有关键字“咖啡”“烤鸡”“意大利面”的所有餐厅 (即空间文本对象).

Choudhury 等人[2]进一步借鉴反向最近邻思想提出面向空间文本对象的选址问题 MaxBRSTkNN, 即从候选位置中选出最优的位置, 能够影响最多空间文本对象. 其中影响是由空间邻近程度和文本相似程度共同决定, 愈接近且越相似影响力越大. 例如打算从候选位置集合里选最优位置开一家意大利餐厅 (“意大利”和“餐厅”为文本), 上述选址方法可以挖掘出附近喜欢意大利菜顾客最多的候选位置.

空间文本选址问题应用领域十分广泛. 针对个人用户推荐服务设施, 比如为推荐最称心的餐厅就餐或最舒适的酒店入住等; 针对企业和组织提供合理的决策, 在商业领域, 能够为超市选择具有最多潜在客户的店址, 为物流企业选择交通便利的最佳仓储位置等; 在交通领域, 可以帮助组织规划更加合理的公交站或地铁站位置等; 在城市规划领域, 能够帮助城市管理者优化城市布局, 提高土地利用率等.

空间文本选址问题应用领域十分广泛. 针对个人用户推荐服务设施[3], 比如为推荐最称心的餐厅就餐或最舒适的酒店入住等; 针对企业和组织提供合理的决策, 在商业领域[4], 能够为超市选择具有最多潜在客户的店址, 为物流企业选择交通便利的最佳仓储位置等; 在交通领域[5], 可以帮助组织规划更加合理的公交站或地铁站位置等; 在城市规划领域[6], 能够帮助城市管理者优化城市布局, 提高土地利用率等.

尽管现有方法一定程度解决了空间文本选址问题, 但仍存在以下 3 方面局限. 首先, 在线消费平台的迅速发展积累了大量用户评价数据 (如 Yelp、大众点评等). 用户通过这些平台选择服务设施时往往会根据过往用户评价等级 (通常表示为 1–5 星) 进行筛选, 比如通过 Yelp 优先选择用户评级高的酒店住宿 (或餐厅就餐). 显然, 空间文本选址问题应当考虑服务设施的用户评级. 其次, 传统空间文本选址中用户只会被对其影响力最大的服务设施影响, 当设施间影响力相差不大时, 该模型与真实世界中用户可能同时受到多个设施影响[7]的情况并不一致. 最后, 位置影响力的计算未考虑周边其他同类服务设施的竞争. 比如当设施 A 和设施 B 都能服务顾客时, 二者必然产生同行竞争[8]. 传统方法非此即彼的影响模式并不完全符合现实情况.

为更合理地在同行竞争环境中结合用户评级因素做出选址决策, 本文提出了新的空间文本竞争选址问题 CoSTUR (competitive spatio-textual location selection based on user rating), 首次同时将用户评级和同行竞争因素纳入空间文本选址的考虑范围, 用于从候选位置集合中选择最优的前 k 个, 使它们相比剩余候选, 综合文本相关性、距离远近、评级高低等因素, 在设施竞争环境中能够影响更多的用户. 其中, 现有设施和候选位置均具备位置、文本和用户评级属性, 每个用户具有位置和文本信息. CoSTUR 问题所选择的最优 k 个位置可用于企业和组织进一步决策.

为了解决 CoSTUR 问题: 首先, 借鉴 Liu 等人[8]对同类竞争设施间均分影响力的思想, 在此基础上结合用户评级完善竞争影响力的定义. 其次, 利用可权衡的影响力阈值摆脱了顾客只能受单个设施影响的假设局限. 阈值越大时, 设施对潜在顾客的影响估计就越准确; 反之, 设施可能影响的潜在顾客数量则越大. 最后, 构建了基于 aR-tree[9]的索引结构 TaR-tree, 其通过文本相似性将树节点的信息向父节点聚合, 进而实现空间性与文本的共同检索. 利用阈值设计了一个基于用户受影响范围的新剪枝策略, 结合 TaR-tree 结构建立基于范围查询和空间联合思想的两种解决方案. 相比基线算法, 所设计的方法能够提升近一个量级的计算效率.

本文的贡献总结如下.

• 提出了一个新的空间文本选址问题 CoSTUR, 首次将设施的同行竞争和用户对设施的评级因素融入到空间文本选址问题当中.

• 设计了一个基于用户受影响的空间范围剪枝策略, 结合文本信息构建新的类似 aR-tree 的索引结构实现范围查询和空间联合两种解决方案.

• 在真实和合成数据集上的实验结果表明, 所提出的优化算法能够显著提高计算效率.

1 问题定义

1.1 基本概念

空间文本数据由空间位置$loc=(lon,lat)$和文本描述des组成, 其中lon和lat分别为地理经度和纬度, des为描述数据的多个词语组合, 如标签、特征、偏好等. 对于特定的用户集合$U=\{u_1,u_2,\cdots,u_m\}$, 其中$u_i=(des,loc)$ $(u_i\in U\wedge i\in[1,m])$, $u_i.des$和$u_i.loc$分别为用户的文本描述和空间位置. 现有设施集表示为$F=\{f_1,f_2,\cdots,f_n\}$, 其中$f_j=(des,loc,rat)\left(f_j\in F\wedge j\in[1,n]\right)$, 其中$f_j.des$描述设施的服务内容、特色或职能等, $f_j.loc$和$f_j.rat$分别为设施的空间位置和用户评级. 候选位置的集合表示为$C=\{c_1,c_2,\cdots,c_r\}$, 其中$c_k.loc$ $(c_k\in C\wedge k\in[1,r])$为候选位置的经纬度. 由于将要在候选位置建设的设施具有明确的目标, 即文本描述和预期评级, 因此, 可以认为$c_k.des$和$c_k.rat$对于所有候选位置均相同. 考虑到设施f_j和将要建设设施的候选位置c_k在竞争环境中均可能影响用户, 下文用对象o表示抽象的设施或候选位置.

定义 1 (空间相关性$S(u,o)$). 给定一个用户u和一个对象o, 二者之间的空间相关性定义如式 (1):

$$S(u,o)=1-\frac{d(u.loc,o.loc)}{D_{\max}} \tag{1}$$

其中, $d(u.loc,o.loc)$表示用户u和对象o之间的欧氏距离, $D_{\max}$表示整个问题域空间内各设施/候选位置与用户之间的最大距离.

定义 2 (文本相关性$T(u,o)$). 给定一个用户u和一个对象o, 二者之间的文本相关性定义如式 (2). 本文使用 Jaccard 相关度, 其他度量方式也适用.

$$T(u,o)=\frac{|u.des\cap o.des|}{|u.des\cup o.des|} \tag{2}$$

定义 3 (空间文本相关性$P(u,o)$). 给定用户u和对象o, 二者之间的空间文本相关性定义如式 (3):

$$P(u,o)=\alpha S(u,o)+(1-\alpha)T(u,o) \tag{3}$$

其中, α是一个空间-文本调节参数, 取值$\alpha\in[0,1]$, 用于平衡空间相关性和文本相关性二者在影响力中的重要程度, 显然$P(u,o)\in[0,1]$.

1.2 综合竞争影响模型

真实世界中用户可能同时受多个设施影响, 因此, 为摆脱传统空间文本选址问题中顾客只受单个设施影响假设的局限, 本文引入影响阈值τ, 以适应用户可以同时被多个设施影响的情况. 由于定义 3 中空间文本相关性取值范围是$[0,1]$, 因此影响阈值的取值范围也与之相一致.

定义 4 (对象影响用户). 给定一个用户u和对象o, 以及一个期望影响阈值$\tau\in[0,1]$, 当$P(u,o)>\tau$时, 认定对象o会影响用户u.

值得一提的是, 阈值τ的设置不仅用于评价影响与否, 还能够用作权衡对象o影响用户的程度: τ越大, o对潜在用户的影响估计就越准确; 反之, 对象o可能影响的潜在顾客数量就越大.

定义 5 (对象影响力$inf(o)$). 给定一个对象o、影响阈值τ和特定用户集合$U=\{u_1,u_2,\cdots,u_m\}$, 则对象o影响的用户构成的集合为$S_{inf}(o)=\{u_i|P(u_i,o)>\tau\wedge u_i\in U\}$, 将对象$o$影响的用户数量定义为其影响力, 即$inf(o)=|S_{inf}(o)|$.

基于定义 4 可定义对象间是否存在竞争关系.

定义 6 (对象间竞争关系). 两个对象o_a和o_b, 对于用户u, 若$P(u,o_a)>\tau\wedge P(u,o_b)>\tau$, 则$o_a$和$o_b$都独立地影响用户$u$, 意味着对象$o_a$和$o_b$之间存在竞争.

定义 7 (用户竞争集$S_{comp}(u,o)$). 给定一个用户u和由设施或候选位置构成的对象集合$O=\{o_1,o_2,\cdots,o_m\}$, 则影响用户u的竞争对象集表示为$S_{comp}(u,o)=\{o_i|P(u,o_i)>\tau\wedge o_i\in O\}$.

基于对象间竞争关系, 借鉴文献对同类竞争设施间均分影响力的思想, 定义影响平分模型.

定义 8 (平分影响力$\overline{score}(o)$). 若多个对象影响同一用户u, 且这些对象属于集合O, 则这些对象将平分用户u的影响力, 则对于$o\in O\wedge P(u,o)>\tau$, 对象o平分影响力$\overline{score}(o)=\sum_{u\in S_{inf}(o)}\frac{1}{|S_{comp}(u,O)|}$, 其中分子 1 表示每个用户对影响值的贡献为 1, 由于$u\in S_{inf}(o)$, 则至少存在o影响u, 故分母$S_{comp}(u,O)$必然不为 0.

此时想建立新的设施获利, 就需要考虑与现有设施集F的竞争. 若候选位置c与用户u的空间文本相关性大于阈值τ, 则c会影响用户u, 如果存在现有设施同样也影响u, 则候选c将与影响用户u的已有设施竞争, 则计算c的影响力就需要计入平分影响力, 根据定义 8 可得$\overline{score}(o)=\sum_{u\in S_{inf}(c)}\frac{1}{|S_{comp}(u,F)|+1}$, 其中$|S_{comp}(u,F)|+1$表示$c$带来的影响.

进一步考虑在竞争的基础上增加用户评级给影响力计算带来的变化. 由于每个用户对影响值的贡献为 1, 换句话说, 一个用户受到的总影响值表征为 1, 考虑到用户评级通常以 1–5 星进行从劣到优的评价, 故将对象的用户评级$o.rat$设置为一组离散值 (也可采用更细粒度的量化标准, 不会影响本文结论), 如 1–5 星分别对应 0.2, 0.4, 0.6, 0.8 和 1. 直觉上, 设施的用户评级越高, 其在竞争中越有利. 因此, 由于用户评级的存在, 影响显然无法用平分来表示. 考虑到评级能够反映整个用户群体对设施的平均偏好程度, 借鉴时空偏好查询的线性累加质量原则[10], 本文采用归一化加权平分影响力度量竞争.

定义 9 (竞争尺度$comp(u,O)$). 给定用户u和其竞争集$S_{comp}(u,O)$, 若$o\in S_{comp}(u,O)$, 则o针对u的归一化加权竞争值可由$\frac{o.rat}{comp(u,O)}$计算得出, 其中分母$comp(u,O)=\sum_{o_i\in S_{comp}(u,O)}o_i.rat=o_1.rat+\cdots+o_i.rat+\cdots+o_m.rat\left(o_i\in S_{comp}(u,O)\right)$, 称为集合$S_{comp}(u,O)$在$u$上的竞争尺度, 简称$u$上的竞争尺度.

此时, 可以定义结合设施间竞争和设施用户评级的综合竞争影响值.

定义 10 (综合竞争影响值$score(o,O)$). 给定对象集合O和特定用户集合U, 则对象$o\in O$在考虑O中其他对象竞争和各自用户评级的共同作用下, 针对所有用户的综合竞争影响值定义如式 (4):

$$score(o,O)=\sum_{u\in S_{inf}(o)}\frac{o.rat}{comp(u,O)} \tag{4}$$

其中, $u\in S_{inf}(o)$表示对象o影响U中用户的集合, $o.rat$表示对象o的用户评级. 根据定义 10, 当给定任意候选c, 则其综合竞争影响值可计算为:

$$score(c,O\cup\{c\})=\sum_{u\in S_{inf}(c)}\frac{c.rat}{comp(u,O\cup\{c\})} \tag{5}$$

其中, $comp(u,O\cup\{c\})=comp(u,O)+c.rat$.

定义 11 (CoSTUR). 给定具有空间位置、文本描述和评级的已有设施集合F, 具有空间位置和文本描述的用户集合U, 空间-文本调节参数α和影响力阈值τ. 根据输入的预期建设设施文本描述des和评级rat, 基于用户评级的空间文本竞争选址问题 CoSTUR (competitive spatio-textual location selection based on user rating) 将从候选位置集合C中选择输出k个最优的候选位置构成子集$C_{opt}\subseteq C\wedge|C_{opt}|=k$, 使得对于$\forall c_k\in C_{opt}\wedge\forall c'\in c\backslash C_{opt}$有$score(c_k)>score(c')$. 具体来说, 给定已有设施集合作为己方企业的竞争对手, 根据初步选定的建址候选位置集合, 以在竞争环境中吸引更多用户为优化目标, 通过计算各候选综合竞争影响值, 从中选择最优的k个候选位置, 以供己方企业进一步决策.

2 策略设计

本研究通过影响阈值的设置, 不仅解决了现有方案中用户只能被唯一设施影响的局限, 也可以帮助选址决策者在更高质量的用户影响估计和更多可能的潜在用户间做出权衡. 针对同类设施间竞争影响力的公式化定义, 完善了融入用户评级对竞争的影响. 接下来, 本文将着力解决在给定用户集合和候选位置集合基数较大时计算代价升高的问题. 本文实验中的用户、已有设施和候选位置等数据集取自 New York 真实数据集和基于 London 的合成数据集.

2.1 基线方法

根据前文中问题的公式化定义, 解决 CoSTUR 最直接的方法是检索所有的用户和现有设施, 找到影响每个用户的用户竞争集, 然后再同样扫描并利用式 (5) 判断每个候选位置的综合竞争影响值. 过程中需要根

据α计算设施或候选位置与用户间的空间文本相关性, 若其值大于阈值τ, 则表明影响该用户. 最终, 根据候选位置集合的综合竞争影响值排序, 取出最优的k个供决策使用, 称为基线算法 Baseline, 具体流程如算法 1 所示.

基线算法使用 HashMap *comp*记录用户u上的竞争尺度, 其中键为u的 id, 值为u竞争尺度. 使用大顶堆*canScore*存放按照的综合竞争影响值排序的候选位置, 根据候选位置c_k的 id 索引. 算法首先初始化*comp*和*canScore* (第 1 行). 然后遍历所有用户U和现有设施F, 找到每个用户的竞争集 (第 2–9 行). 遍历用户集U和候选C, 过程中根据影响用户与否来决定是否采用式 (5) 计算某个候选位置的综合竞争影响值 (第 10–17 行), 其中*canScore*通过迭代不断累加每个用户对竞争影响值的贡献 (第 14 行). 最后返回 top-k个最优候选位置.

算法 1. Baseline

输入: F,U,C,τ,α,k.
输出: top-k c.

```
1 Initialize comp, canScore
2 for u∈U do
3   for f∈F do
4     Calculate P(u,f)
5     if P(u,f)>τ then
6       comp[u]+=f.rat
7     end if
8   end for
9 end for
10 for u∈U do
11   for c_k∈C do
12     Calculate P(u,c_k)
13     if P(u,c_k)>τ then
14       canScore[c_k]+= c_k.rat / (comp[u]+c_k.rat)
15     end if
16   end for
17 end for
18 return top-k c in canScore
```

2.2 优化方法

为降低基线算法线性扫描的复杂度, 本节分别提出基于范围查询 (range query) 和空间连接 (spatial join) 思想的两个优化算法, 结合索引结构和剪枝策略提升计算效率.

空间范围聚集索引结构 aR-tree 能将节点空间和附带属性进行聚合, 以提升联合检索效率, 该优点适于 CoSTUR 问题. 图 1 显示了基于 aR-tree 构建的索引结构 TaR-tree, 它融合了空间和文本属性. 其中, TaR-tree 每个节点由聚合空间信息的 MBR (minimum bounding rectangle) 及聚合文本信息的并集 (便于计算 Jaccard 相似性) 组成. 叶子节点的父节点将所包含子节点的位置构造成 MBR; 非叶子节点的父节点则将所包含子节点的多个 MBR 囊括为更大的 MBR. TaR-tree 将子节点的文本属性逐级向父节点并集聚合, 直至根节点. 遍历 TaR-tree 的过程中, 可以同时处理空间和文本信息.

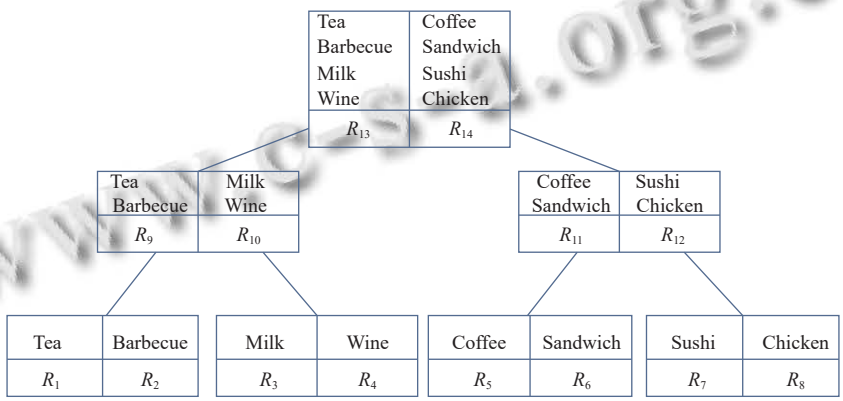


图 1 TaR-tree 示例

2.2.1 基于 range query 方法

影响阈值τ使得用户可以同时被多个对象影响, 而随着阈值τ的改变, 用户受到对象的影响范围也随之改变. 根据定义 1–定义 4, 容易反推用户受影响的范围半径, 如式 (6) 所示:

$$d = D_{\max}\left[1 - \frac{\tau-(1-\alpha)T}{\alpha}\right] \tag{6}$$

式 (6) 的具体推导如下 (为便于理解, 此处用户与设施之间的距离、空间和文本相关性分别简写为d、S和T: 当$\alpha S+(1-\alpha)T>\tau$时, $S>\dfrac{\tau-(1-\alpha)T}{\alpha}$, 带入定

义 1 可得 $1-\frac{d}{D_{\max}}>\frac{\tau-(1-\alpha)T}{\alpha}$, 即用户和设施间距离 $d<D_{\max}\left[1-\frac{\tau-(1-\alpha)T}{\alpha}\right]$. 考虑 $T\in[0,1]$, 当 T 取最大值1时, 距离 d 将取其最大值 $d_{\max}$, 若设施与用户间距离超过 $d_{\max}$, 在给定 τ 和 α, 该设施必然不会影响用户; 反之, 当 T 取最小值0时, 距离 d 将取其最小值 $d_{\min}$, 若设施位于用户 $d_{\min}$ 范围内, 则必然影响用户. 而对于与用户间距离在 $(d_{\min},d_{\max})$ 范围的设施, 需要根据定义 4 进一步判断是否影响用户. 用户受与不受影响的范围如图 2 所示.

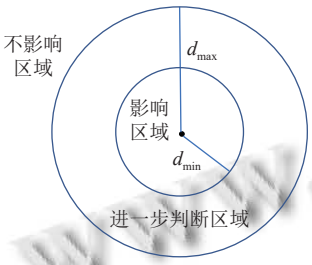


图 2 用户受影响范围示意

根据上述用户必然受影响和必然不受影响区域, 可以结合 TaR-tree 索引结构进行范围查询. 在用户必然受影响区域内的设施将参与竞争, 而在必然不影响范围内的设施可以直接剪枝. 在上述两个范围间 (圆环区域) 的设施, 仍需通过定义 4 进行计算, 若大于阈值 τ 则参与竞争, 反之则舍弃.

鉴于此, 提出基于范围查询 (range query) 的解决方案. 由于现有设施 F 事先已知, 因此可以用 TaR-tree 索引. 当输入特定用户集、候选集和参数 τ 和 α 后, 利用索引范围查询的性能优势提升效率. 算法 2 的 RangeQuery 展示了方法的流程细节.

算法 2. RangeQuery

输入: U,F,C,τ,α,k, TaR-tree FT of F.
输出: top-k c.

```
1 Initialize comp, canScore
2 for u∈U do
3   Calculate d_max and d_min
4   Build u.rect_out and u.rect_in
5   S_comp←rq(u.rect_out,FT.mbr)
6   for f∈S_comp do
7     if f intersects u.rect_in or P(u,f)>τ
8       comp[u]+=f.rat
9     end if
10  end for
11 end for
12 for u∈U do
13   Build TaR-tree CT of C
14   S_comp←rq(u.rect_out,CT.mbr)
15   for c∈S_comp do
16     if c intersects u.rect_in or P(u,c)>τ
17       canScore[c]+=c.rat/(comp[u]+c.rat)
18     end if
19   end for
20 end for
21 return top-k c in canScore
```

RangeQuery 算法基线算法使用 HashMap $comp$ 记录用户 u 上的竞争尺度, 其中键为 u 的 id, 值为 u 竞争尺度. 使用大顶堆 $canScore$ 存放按照的综合竞争影响值排序的候选位置, 根据候选位置的 id 索引. 算法首先初始化 $comp$ 和 $canScore$ (第 1 行). 对于每一个用户 $u\in U$, 可以根据式 (6) 计算得到 $d_{\max}$ 和 $d_{\min}$. 以 u 为圆心, $d_{\max}$ 和 $d_{\min}$ 为半径, 构造两个 MBR 范围 $u.rect\_out$ 和 $u.rect\_in$ (第 2–4 行). 然后, 通过在 F 的 TaR-tree FT 上对 $u.rect\_out$ 进行范围查询 (算法中用函数 rq 表示), 剔除不参与 u 竞争的设施, 将影响的设施存储在集合 S_{comp} 中 (第 5 行). 然后, 遍历 S_{comp} 中的设施 f, 对于与 $u.rect\_in$ 相交的设施, 将其用户评级加入用户 u 上的竞争尺度 $comp[u]$ 中; 而对于在 $(d_{\min},d_{\max})$ 范围内的设施, 需要根据定义 4 进一步判断是否影响用户, 若影响则更新用户 u 上的竞争尺度 (第 6–10 行). 然后按照和处理现有设施 F 类似的方式计算 C 中每个候选的综合竞争影响值 (第 12–20 行). 唯一的两处区别在于: 1) 需要先构建 C 的 TaR-tree CT, 用于进行 $d_{\max}$ 和 $d_{\min}$ 的 MBR 范围查询 (第 13 行); 2) 利用式 (5) 迭代地计算候选综合竞争影响值 (第 17 行). 基于 CT 的范围查询结束后返回 top-k 个最优候选位置.

2.2.2 基于 spatial join 方法

空间连接 (spatial join)[11]的核心思想是在不同空间数据集间根据空间谓词进行“连接”操作, 即当两个空间对象间存在如相交 (intersect)、包围 (enclose) 等谓词关系时, 可以进行 join 匹配.

根据空间连接思想, 可以将设施集 (候选集) 和用户集视为要进行连接的双方, 参照范围查询思路, 可以使用定义 4 的阈值判定作为谓词进行匹配. 为更有效地进行数据集空间连接遍历, 用户集也构建为 TaR-tree, 利用树的剪枝过程优化计算性能. 该方法将从设施集 (候选集) 和用户集两棵 TaR-tree 的根结点开始同时执行深度优先遍历. 若两树中间节点的聚合空间文本相关性大于阈值 τ, 即节点间的空间 MBR 和文本并集满

足阈值τ, 递归地进入子一级节点; 反之, 则遍历 TaR-tree 树的兄弟节点. 此过程直到两树满足条件的节点全部访问结束为止.

按照空间连接阈值判定遍历的逻辑, 遍历结束后可以得到所有对象$o \in O$的$S_{inf}(o)$集合, 即对于候选集而言, 每个候选能影响的用户数量可知. 那么能否利用该影响用户数量减少综合竞争影响值的计算和存储结构的读写呢? 前述 Baseline 和 RangeQuery 算法均需通过迭代计算所有候选位置的真实综合竞争影响值进行排序, 下面提出一个新的数值上界简化计算过程.

引理 1 (综合竞争影响值上界). 对于$\forall u \in S_{inf}(c)$, 在用户评级 1–5 星离散归一化到对应值$0.2, 0.4, 0.6, 0.8$和1的情况下, c的综合竞争影响值上界为$\frac{5}{6}$倍的$|S_{inf}(c)|$, 用$score_{\max}(c, F \cup \{c\}) = \frac{5}{6}|S_{inf}(c)|$表示.

证明: 当$\exists f \in F \wedge u_i \in S_{inf}(f)$时, 若$S_{comp}(u_i, F) = \{F\}$, 即此时仅有$f$影响$u_i$时, u_i对$comp(f, F)$的贡献值为 1, 即f独享对u_i的影响. 此时若$u_i \in S_{inf}(c)$, 根据式 (5), 针对u_i有:

$$score(c, F \cup \{c\}) = \frac{c.rat}{f.rat + c.rat} \tag{7}$$

由于 $0.2 \leqslant c.rat = f.rat \leqslant 1$, 则必有$\frac{0.2}{0.2+1} \leqslant \frac{c.rat}{f.rat + c.rat} \leqslant \frac{1}{0.2+1}$, 即$\frac{1}{6} \leqslant \frac{c.rat}{f.rat + c.rat} \leqslant \frac{5}{6}$.

当$\exists f, f' \in F \wedge u_i \in S_{inf}(f) \wedge u_i \in S_{inf}(f')$时, 即有两个设施$f$和$f'$同时影响$u_i$. 若$u_i \in S_{inf}(c)$, 根据式 (7), 有$\frac{0.2}{0.2+1\times 2} \leqslant \frac{c.rat}{f.rat + f'.rat + c.rat} \leqslant \frac{1}{0.2\times 2+1}$, 即$\frac{0.2}{2.2} \leqslant \frac{c.rat}{f.rat + f'.rat + c.rat} \leqslant \frac{1}{1.4}$, 显然$\frac{1}{1.4} < \frac{5}{6}$.

以此类推, 当有更多设施影响u_i时, 必有$score(c, F \cup \{c\}) \leqslant \frac{1}{|S_{comp}(u_i, F)| \times 0.2 + 1}$. 考虑到$|S_{comp}(u_i, F)| \geqslant 1$, 因此$score(c, F \cup \{c\}) \leqslant \frac{5}{6}$. 即$\sum_{u_i \in S_{inf}(c)} score(c, F \cup \{c\}) \leqslant \frac{5}{6}|S_{inf}(c)|$. 证明完毕.

结合空间连接思想和引理 1 设计了 SpatialJoin 算法, 具体流程如算法 3 所示. 与前文两种算法类似, SpatialJoin 算法也使用 HashMap $comp$记录用户u_i上的竞争尺度, 使用大顶堆$canScore$根据竞争影响值存放排序后的候选位置, 并同样先初始化它们 (第 2 行). 不同点在于算法首先为空间连接构造用户集U和候选集C的两棵 TaR-tree (第 1 行), 并初始化一个根据候选综合竞争影响值存放的小顶堆HC, 且该小顶堆最多存放 k 个候选 (第 2 行). 算法开始时先利用阈值判定谓词的空间连接函数sj处理用户集和设施集两个 TaR-tree 树的根节点. 遍历将得到影响每个用户$u_i \in U$的已有设施, 并根据定义 9 计算u_i上的竞争尺度$comp$ (第 3 行).

算法 3. SpatialJoin

输入: U,F,C,τ,α,k,TaR-tree FT of F.
输出: HC.

```
1 Build TaR-trees UT of U, CT of C
2 Initialize comp, canScore, HC
3 comp←sj(UT.root,FT.root)
4 canScore←sj(UT.root,CT.root)
5 for pop c_top from canScore do
6   if score_max(c_top,F∪c_top)<HC_top
7     break
8   Calculate score(c_top,F∪{c_top})
9   if |HC|≥k
10    if score(c_top,F∪{c_top})>HC_top
11        pop HC
12      insert ⟨c_top,score(c_top,F∪{c_top})⟩ to HC
13    end if
14  else insert ⟨c_top,score(c_top,F∪{c_top})⟩ to HC
15  end if
16 return HC
```

接下来利用空间连接函数sj处理用户集和候选位置集两个 TaR-tree 树的根节点. 根据引理 1, 直接计算每个候选位置的综合竞争影响值上界$score_{\max}(c, F \cup \{c\})$并存储到大顶堆$canScore$当中. 此时$canScore$堆顶是所有候选位置综合竞争影响值上界的最大值 (第 4 行). 考虑到上界更大的候选, 其计算出的综合竞争影响值$score(c, F \cup \{c\})$也可能较大, 因此, 从$canScore$堆顶中逐一弹出候选, 并计算实际综合竞争影响值 (第 8 行). top-k候选位置的实际综合竞争影响值存放在小顶堆HC中. 当HC堆中数据量少于k时, 可以直接压入HC; 反之, 需要将候选c的$score(c, F \cup \{c\})$和HC堆顶的候选进行比较, 如果HC堆顶值较小, 则弹出堆顶并压入候选c的值, 否则处理下一个$canScore$堆顶元素 (第 9–14 行). 当发现HC堆顶候选的综合竞争影响值大于$canScore$堆顶中候选的上界时, 意味着此时$canScore$堆中不可能再有候选的实际综合竞争影响值大于当前HC堆中的 top-k候选位置, 此时针对$canScore$堆的遍历结束 (第 6, 7 行). HC堆中的 top-k候选位置即为结果 (第 16 行).

2.3 算法复杂度分析

本节对前述各解决方案的计算复杂度进行理论分

析, 其中定义 4 的计算作为原子操作.

基线方法 Baseline 采用线性扫描方法遍历用户集、候选位置集和已有设施集, 方法的时间复杂度为 $O(|F|\cdot|U|+|C|\cdot|U|)$.

在结合 TaR-tree、d_{max}和d_{min}的 RangeQuery 算法中, 假定 TaR-tree 每个节点有n个子节点. 对于用户集合与设施该 TaR-tree 的范围查找, 其时间复杂度为 $O(|U|\log_n|F|)$. 对于用户集与候选集间的计算, 首先需要构建候选 TaR-tree, 时间复杂度为$O(|C|\log_n|C|)$; 然后进行范围查找, 时间复杂度为$O(|U|\log_n|C|)$. 考虑到d_{max}和d_{min}的作用, 则整个算法的时间复杂度为$O(|U|\log_n|F'|+O(|C|\log_n|C|)+O(|U|\log_n|C'|))$, 其中$|F'|<|F|\wedge|C'|<|C|$.

SpatialJoin 算法利用 TaR-tree 之间的空间连接思想. 首先构建用户集与候选集 TaR-tree 需要$O(|U|\log_n|U|+O(|C|\log_n|C|))$复杂度. 空间连接遍历过程理论时间复杂度为$O(\log_n|F|\log_n|U|+\log_n|U|\log_n|C|)$. 最终根据引理 1 的上界, 遍历大顶堆的计算复杂度可以认为接近 $O(1)$. 算法最终时间复杂度为$O((|U|+\log_n|F|)\log_n|U|+(|C|+\log_n|U|)\log_n|C|)$.

容易看出 RangeQuery 算法和 SpatialJoin 算法的时间复杂度明显低于基线方法.

3 实验论证与分析

3.1 实验设置

3.1.1 实验数据集

本文实验中的用户、已有设施和候选位置等数据集取自 New York 真实数据集和基于 London 的合成数据集. 其中后者是由真实 POI 设施位置随机捕获一组 POI 文本数据的方式合成, 数据特征如表 1 所示. 所选数据集为不同国家区域的开源数据集, 有助于体现本文研究方法和实验结果的普适性.

表 1 数据集基本信息

数量	New York	London
总用户数量	20.4k	7667
已有设施数量	11k	3k
候选位置数量	200/400/600/800/1k	200/400/600/800/1k

经数据抽样, 用户和设施间空间相关性和文本相关性均值分别在 0.7 和 0.1 附近, 差距较大. 为使空间相关性和文本相关性的α权衡参数具有可比性, 利用高斯函数归一化方法将文本相关性归一化到均值 0.7, 则用户和设施间空间文本相关性均值接近 0.8. 实验若无明确说明, 将采用如下默认实验参数: 影响阈值$\tau=0.9$, $k=10$, 平衡参数$\alpha=0.6$, 候选位置数量$|C|=200$, 在 London 和 New York 数据集中用户数量$|U|$默认分别设为 5.5k 和 5k. 其中τ设为 0.9 是为了更准确地估计设施对潜在用户的影响; α设为 0.6 是使该参数更接近数据样本均值, 以得到更有意义的实验结果.

3.1.2 实验环境与评估算法

以下解决方案将在本节进行评估, 实验采用 Python 语言编码, 环境为 Windows 10 (64 位), Intel(R) i5-13600KF 3.5 GHz, 内存 16 GB.

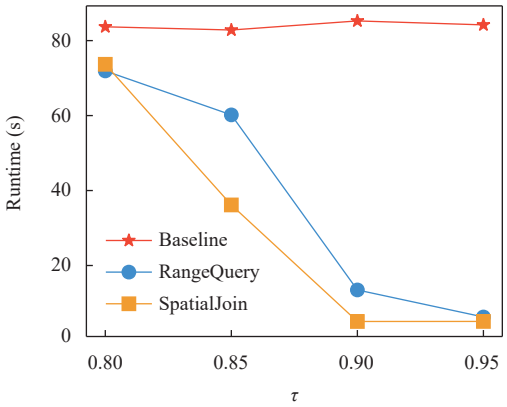
实验分别评估本文第 2.1 节 Baseline 算法, 第 2.2.1 RangeQuery 算法, 第 2.2.2 节 SpatialJoin 算法.

3.2 实验结果

3.2.1 阈值τ的影响

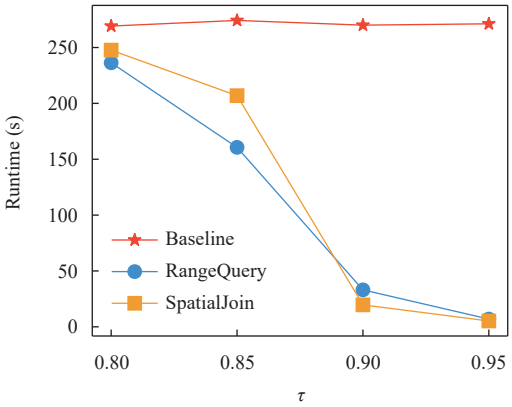
验证影响阈值τ变化对性能产生的影响, τ的取值分别为 0.8、0.85、0.9 和 0.95. 由图 3 展示的实验结果可知, 3 种算法中 Baseline 效率最差, 影响阈值τ的变化对其影响不大, 而 RangeQuery 和 SpatialJoin 性能随影响阈值τ的增大相比基线算法的优势愈发显著. 这意味着τ越大时, 可能影响的潜在顾客越准确, 用户数量也越少, 剪枝效果越明显; 反之, 可能影响的顾客越多, 剪枝收益随之下降.

如图 3(a) 所示, SpatialJoin 性能在 London 数据集上提升更理想, 这很大程度上取决于其计算复杂度优势. 图 3(b) 中 New York 数据集上情况则相对复杂. 当影响阈值τ较小时 RangeQuery 效果更优, 反之 SpatialJoin 性能稍好. 其原因在于 New York 和 London 数据集空间范围相近 (分别是 59 km 和 56 km), 但前者竞争设施更密集, 当影响阈值τ变小时d_{max}和d_{min}对剪枝的贡献更明显.



(a) London

图 3 阈值τ改变的影响



(b) New York

图 3 阈值τ改变的影响 (续)

3.2.2 用户数量|U|的影响

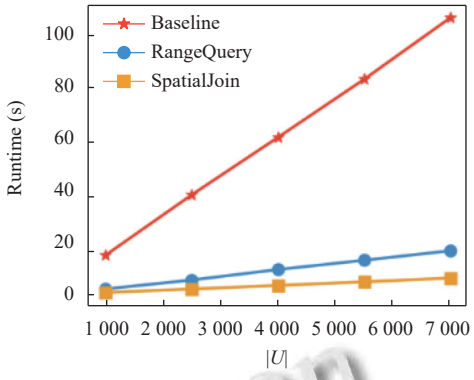
接下来针对输入用户数据集规模|U|, 验证不同方法影响的情况. 对于 London 和 New York 数据集, |U|的取值分别为{1 000, 2 500, 4 000, 5 500, 7 000}和{1 250, 2 500, 5 000, 10 000, 20 000}. 从图 4 可以观察到, Baseline 方法的时间开销随|U|呈线性增长, 这与两个数据集中|U|数量增长方式一致. 而 SpatialJoin 和 RangeQuery 算法时间开销同样随着|U|的增加而提高, 但相比 Baseline 方法均能提升一个量级的性能, 其中 SpatialJoin 提升更优. 这说明实验结果和理论分析是一致的, 即文中提出的索引结构结合剪枝及综合竞争影响值上界均起到了作用.

3.2.3 候选位置|C|的影响

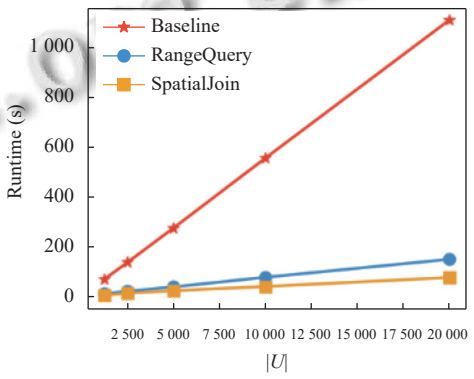
将候选位置|C|数量分别设置为 200、400、600、800 和 1 000 时, 分析 3 种算法对效率的影响. 如图 5 所示, 实验结果在数量上和|U|对结果的影响十分类似, 3 种算法中同样是 SpatialJoin 算法效率最优, RangeQuery 次之, 二者均能提供超过 Baseline 一个量级的性能提升. 对比图 5(a) 和图 5(b), 可以发现在 London 数据集上性能随|C|的增长下降的更快, 说明现有设施越多, 候选位置|C|由于竞争的原因, 对性能的影响越不明显, 算法的效果也相对更稳定.

3.2.4 k值变化

本部分的实验着眼于验证k变化对算法影响, 其中k取值为 10–50 的离散值. 如图 6(a) 和图 6(b) 所示, 3 种算法中 SpatialJoin 算法效率最优, RangeQuery 次之, 这两种方法均能提供超过 Baseline 近一个量级的性能提升. 注意到k值的变化对所有算法的效率几乎都不会产生影响, 这一点从第 2.3 节理论分析就可以看出, 即k值并不影响算法的时间复杂度.

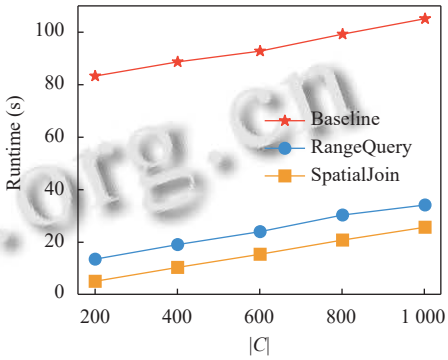


(a) London

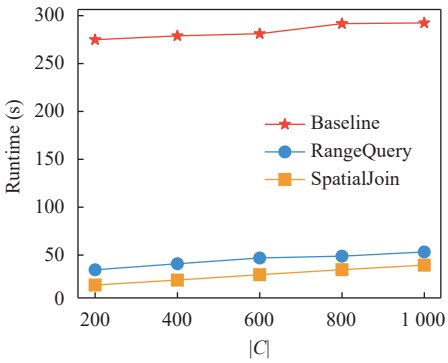


(b) New York

图 4 用户数量|U|改变的影响

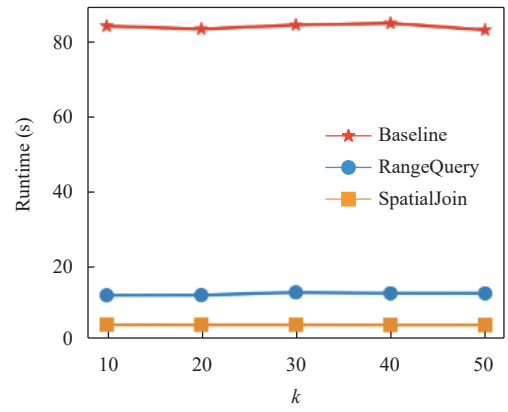


(a) London

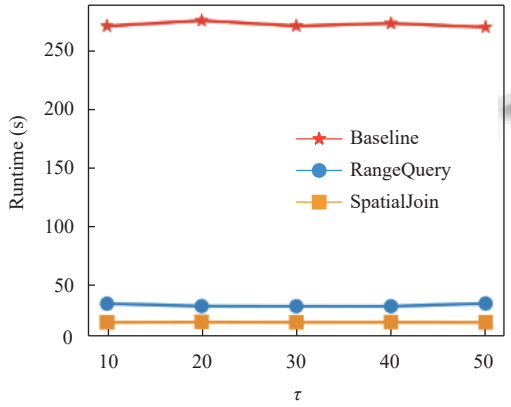


(b) New York

图 5 候选位置|C|改变的影响



(a) London



(b) New York

图 6 k改变的影响

4 相关研究

空间文本位置选择问题: Ahmed 等人[12]提出反向空间 top-k 关键字查询 (RSK), 并采用网格索引划分数据集, 其中每个网格存储关键字频率表, 本文的索引结构设计也借鉴了其思想. Choudhury 等人[2]提出了 MaxBRSTkNN 查询的经典空间文本选址模型, 为本研究的选址模型提供了理论支撑. 文献[13]是对 Choudhury 等人[2]研究的扩展, 提出了一种最大化双色反向空间文本最近邻查询 (MaxST), 并在有阻碍和无阻碍空间中, 为对象寻找最佳位置和一组关键字. Chen 等人[14]将研究扩展至路网环境, 其反向 top-k 关键字的位置查询 (RTkKL) 旨在找到最大空间区域, 以便将查询对象包含在任何 top-k 空间关键字查询的结果中. 该研究的路网场景与本文模型不相符, 但可考虑作为未来的研究方向. 上述研究解决了各自应用场景中的问题, 然而均未考虑设施之间的竞争因素, 难以直接用于应对 CoSTUR 问题的挑战.

设施竞争位置选择问题: 传统的竞争性选址问题[15]假设一家新公司将入驻存在现有公司运营的市场, 其最佳选址期望在竞争中争夺最大的市场份额. Revelle[16]引入新的竞争选址问题, 它在现有设施的基础上增加了新设施, 使新设施群的影响力最大化扩展. 其中若两个设施捕获同一对象, 则会平分对它的影响, 这与 CoSTUR 依据的经典竞争影响模型相同, 但该模型其未考虑设施评级. Huang 等人[17]基于最大影响力模型考虑现有设施对选址的影响, 影响关系的定义以最近邻为基础. Liu 等人[8]将竞争问题扩展到运动对象, 使用影响关系剪枝规则和影响值剪枝规则来加速查询过程. 现有竞争选址研究尽管有些也采用和本文相同的竞争影响模型, 但均未考虑设施和对象的文本因素, 也缺少对设施评级的应对, 方法不适用 CoSTUR 问题.

5 结束语

本文引入一个称为 CoSTUR 的新型空间文本位置选择问题, 并提供了两种有效且高效的解决方案. 相比于经典空间文本选址问题, CoSTUR 问题综合考虑了用户可以同时被多个设施影响、同类设施间存在竞争, 及用户评级对竞争的影响等因素. 该问题在现实世界中应用广泛, 如城市规划、市场营销、LBS 等. 提出的其中一种算法基于范围查询思想, 通过构建必然影响/不影响的两个新剪枝距离范围, 结合空间文本索引 TaR-tree, 能够有效解决问题. 另一种方法建立在空间连接思想上, 利用一个新的分值上界简化计算过程. 在真实及合成数据集上的大量实验证明了所提方法的有效性和高效性.

参考文献

1 Chan HKH, Liu SX, Long C, *et al*. Cost-aware and distance-constrained collective spatial keyword query. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1324–1336.

2 Choudhury FM, Culpepper JS, Sellis T, *et al*. Maximizing bichromatic reverse spatial and textual k nearest neighbor queries. Proceedings of the VLDB Endowment, 2016, 9(6): 456–467. [doi: 10.14778/2904121.2904122]

3 Lv Z, Shang KY, Huo HW, *et al*. RASK: Range spatial keyword queries on massive encrypted geo-textual data. IEEE Transactions on Services Computing, 2023, 16(5): 3621–3635. [doi: 10.1109/TSC.2023.3289654]

4 Wang L, Yu ZW, Yang DQ, *et al*. Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data. IEEE Transactions on Industrial Informatics, 2020, 16(2): 1058–1066. [doi: 10.1109/TII.2019.2891258]

5 Darwish TSJ, Bakar KA, Kaiwartya O, *et al*. TRADING: Traffic aware data offloading for big data enabled intelligent transportation system. IEEE Transactions on Vehicular Technology, 2020, 69(7): 6869–6879. [doi: 10.1109/TVT.2020.2991372]

6 Priyadarshini I, Kumar R, Alkhayyat A, *et al*. Survivability of industrial internet of things using machine learning and smart contracts. Computers and Electrical Engineering, 2023, 107: 108617. [doi: 10.1016/j.compeleceng.2023.108617]

7 Wang M, Li H, Cui JT, *et al*. PINOCCHIO: Probabilistic influence-based location selection over moving objects. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(11): 3068–3082. [doi: 10.1109/TKDE.2016.2580138]

8 Liu P, Wang M, Cui JT, *et al*. Top-*k* competitive location selection over moving objects. Data Science and Engineering, 2021, 6(4): 392–401. [doi: 10.1007/s41019-021-00157-1]

9 Papadias D, Kalnis P, Zhang J, *et al*. Efficient OLAP operations in spatial data warehouses. Proceedings of the 7th International Symposium on Spatial and Temporal Databases. Redondo Beach: Springer, 2001. 443–459.

10 Yiu ML, Dai XY, Mamoulis N, *et al*. Top-*k* spatial preference queries. Proceedings of the 23rd IEEE International Conference on Data Engineering. Istanbul: IEEE, 2007. 1076–1085.

11 Qiao BY, Hu B, Zhu JH, *et al*. A top-*k* spatial join querying processing algorithm based on spark. Information Systems, 2020, 87: 101419. [doi: 10.1016/j.is.2019.101419]

12 Ahmed P, Eldawy A, Hristidis V, *et al*. Reverse spatial top-*k* keyword queries. The VLDB Journal, 2023, 32(3): 501–524. [doi: 10.1007/s00778-022-00759-9]

13 Choudhury FM, Culpepper JS, Bao ZF, *et al*. Finding the optimal location and keywords in obstructed and unobstructed space. The VLDB Journal, 2018, 27(4): 445–470. [doi: 10.1007/s00778-018-0504-y]

14 Chen ZJ, Wang X, Liu WY. Reverse keyword-based location search on road networks. GeoInformatica, 2022, 26(1): 201–231. [doi: 10.1007/s10707-021-00440-3]

15 Plastria F. Static competitive facility location: An overview of optimisation approaches. European Journal of Operational Research, 2001, 129(3): 461–470. [doi: 10.1016/S0377-2217(00)00169-7]

16 Revelle C. The maximum capture or “sphere of influence” location problem: Hotelling revisited on a network. Journal of Regional Science, 1986, 26(2): 343–358. [doi: 10.1111/j.1467-9787.1986.tb00824.x]

17 Huang J, Wen ZY, Pathan M, *et al*. Ranking locations for facility selection based on potential influences. Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society. Melbourne: IEEE, 2011. 2411–2416.

(校对责编: 孙君艳)