DOI: 10.11992/tis.201808011
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20190524.0952.004.html

基于位置−文本关系的空间对象 top-*k* 查询与排序方法

孟祥福, 张霄雁, 赵路路, 李盼, 毕崇春
(辽宁工程技术大学 电子与信息工程学院, 辽宁 葫芦岛 125105)

摘 要: 针对普通的空间关键字查询通常会导致多查询结果的问题。本文提出了一种基于空间对象位置-文本相关度的 top- *k* 查询与排序方法, 用于获取与给定空间关键字查询在文本上相关且位置上相近的典型空间对象。该方法分为离线处理和在线查询处理 2 个阶段。在离线阶段, 根据空间对象之间的位置相近性和文本相似性, 度量任意一对空间对象之间的位置-文本关系紧密度。在此基础上, 提出了基于概率密度的代表性空间对象选取算法, 根据空间对象之间的位置-文本关系为每个代表性空间对象构建相应的空间对象序列。在线查询处理阶段, 对于一个给定的空间关键字查询, 利用 Cosine 相似度评估方法计算查询条件与代表性空间对象之间的相关度, 然后使用阈值算法(threshold algorithm, TA)在预先创建的空间对象序列上快速选出 top- *k* 个满足查询需求的典型空间对象。实验结果表明: 提出的空间对象 top- *k* 查询与排序方法能够有效地满足用户查询需求, 并且具有较高的准确性、典型性和执行效率。

关键词: 空间数据库; 空间关键字查询; 位置-文本关系; 概率密度; 代表性对象选取; top-*k* 查询与排序
中图分类号: TP311.1 **文献标志码**: A **文章编号**: 1673−4785(2020)02−0235−08

中文引用格式: 孟祥福, 张霄雁, 赵路路, 等. 基于位置-文本关系的空间对象 top-*k* 查询与排序方法 [J]. 智能系统学报, 2020, 15(2): 235–242.
英文引用格式: MENG Xiangfu, ZHANG Xiaoyan, ZHAO Lulu, et al. A location-text correlation-based top-*k* query and ranking approach for spatial objects[J]. CAAI transactions on intelligent systems, 2020, 15(2): 235–242.

A location-text correlation-based top-*k* query and ranking approach for spatial objects

MENG Xiangfu, ZHANG Xiaoyan, ZHAO Lulu, LI Pan, BI Chongcun
(School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China)

Abstract: Due to the large size of spatial databases, a common spatial keyword query often leads to the problem of too many answers. To deal with this problem, this paper proposes a location-text correlation-based top- k query and ranking approach for spatial objects, which aims to find the typical spatial objects with high text relevancy and location proximity. This approach consists of offline processing and online query steps. The offline step scores the relationship between any pair of spatial objects by considering their location proximity and text similarity. Then, by using a probabilistic density-based representative spatial object selection method, a set of representatives over the spatial objects is selected to build a corresponding spatial object sequence. In the online query period, when a user issues a spatial keyword query, the location-text correlation between the query and representative objects is evaluated, and then, the top- k typical relevant objects can be expeditiously picked using the threshold algorithm (TA) algorithm over the sequences corresponding to representative spatial objects. The experiments demonstrate that the proposed top- k query and ranking approach can closely meet users' needs, with high precision, typicality, and good performance.

Keywords: spatial database; spatial keyword query; location-text correlation; probability density; representative object selection; top-*k* query and ranking

收稿日期: 2018−08−12. **网络出版日期**: 2019−05−27.
基金项目: 国家自然科学基金面上项目 (61772249); 辽宁省自然科学基金项目 (20170540418); 辽宁省教育厅科学研究项目 (LJYL018).
通信作者: 孟祥福. E-mail: marxi@126.com.

随着 Internet 的普遍应用, 对于兴趣点 (point of interests, POIs, 如餐厅、电影院、旅馆、旅游景点等) 的空间关键字查询已成为当前空间数据库

的研究热点[1]。空间关键字查询条件的形式为：

q: {位置, < 关键字 1, 关键字2,…, 关键字 m > }

其中“位置”代表空间位置查询条件，可以是一个空间点 (通常用经纬度表示)，也可以是一个空间范围；“关键字”代表文本信息查询条件，通常用若干关键字表示。空间数据库中的每个空间对象 (也称为兴趣点) 都包含 2 类信息：空间信息 (spatial information, 如经纬度) 和文本信息 (text information, 如 POI 的名称、设施、评论等文本描述信息)。

由于空间数据库通常蕴含海量数据，因此一个普通空间关键字查询很可能会导致多查询结果问题。例如，在 Yelp 网站上搜索“San Francisco”地理范围内的“Chinese Restaurants”，会返回 1 108 条查询结果。这种情况下，绝大多数用户期望空间数据库查询系统能够快速返回前 k 个与当前查询在位置和文本上最为相关且具有代表性、典型性的结果。

近年来，有关空间关键字查询的研究工作在一些主流期刊和会议上涌现出来[2-10]。根据文献[2-3]，这些查询方法可分为 3 类：1) 布尔 k 近邻查询[4-6]：该类方法用于检索文本描述信息中包含所有查询关键字且距离查询位置最近的前 k 个空间对象，查询结果按其对查询的位置相近性进行排序。2)top-k 范围查询[7-8]：用于检索位于查询区域内且与查询关键字具有较高文本相似度的前 k 个空间对象，查询结果按其对查询关键字的文本相似度进行排序。3)top-k k 近邻查询[3, 9-11]：该类方法根据空间对象的文本相似性和位置相近性进行 top-k 检索和排序，查询结果按其对查询的位置相近性和文本相似度进行综合排序。上述 3 类方法中，第 3 类是当前空间关键字查询最为常用的方法。为了获取 top-k 查询结果，上述 3 类方法都需要先从目标数据集中获取与查询位置相近且文本与查询关键字全部或部分匹配的候选结果，然后再根据评分函数从候选结果中选取 top-k 个结果。为了快速获取候选结果集合，上述方法通常采用空间索引与文本索引相结合的混合索引结构。空间搜索的索引技术主要有 R-tree[12]、R*-tree[13] 和 Quad-tree(四分树)[14] 等，其中 R-tree 是最基本的空间索引结构。文本搜索的索引技术主要有倒排文件 (inverted file)、签名文件 (signature file) 和位图索引 (bitmap) 等。空间与文本索引相结合的空间-文本数据混合索引结构可归为以下几类：1) 两阶段索引[15]：该类索引先利用 R-tree 或 Quad-tree 获取与查询位置相近的候选结果，然后再用倒排文件 (inverted file) 从候选结果中获取包含查询关键字的最终查询结果。该类方法的缺点是无法确定第 1 阶段产生的候选结果个数，当候选结果个数过少时，很可能会导致最终结果不能满足查询关键字的匹配需求，而候选结果过多时，又可能会导致计算资源的浪费。2)IR-tree 索引[9-10, 16]：该类索引将倒排文件集成到 R-tree 节点中，当查询到来时，判断 IR-tree 节点与查询的位置距离以及该节点对应的倒排文件是否包含查询关键字。该类方法同时判断节点与查询的位置关系和文本匹配度，在很大程度上提升了查询效率。该类方法的缺点是，对于满足查询位置要求和 (全部或部分) 包含查询关键字的树节点都要进行遍历，从而得到候选结果集合，然后再按评分函数从中选取前 k 个综合分数最高的结果。3)Quad-tree 索引[14, 17]：该类索引将倒排文件与 Quad-tree 相结合，与 IR-tree 类似，在查询过程中同时判断节点与查询的位置范围重叠程度以及节点对应的倒排文件是否包含查询关键字。Quad-tree 索引的优点是区域搜索效率高，缺点是存储代价高，树结构不平衡，top-k 结果选取的耗时较长。综上可见，给定一个空间关键字查询，现有方法利用混合索引结构从空间对象集合中获取候选结果，由于产生的候选结果数量通常多于最终的 top-k 结果数量，并且在 top-k 结果选取中需要根据评分函数计算所有候选结果对象与当前查询的位置与文本综合相关度才能确定最终结果，因此 top-k 选取效率仍有提升空间。此外，现有方法按空间对象与查询的综合相关度获取 top-k 结果，结果之间往往比较相似，实际上用户希望看到既与查询相关且彼此之间又具有一定差异的代表性查询结果。

针对现有空间关键字查询研究工作存在的问题，本文提出一种基于位置-文本关系的空间对象 top-k 查询与排序方法，该方法可分为 2 个处理阶段：在离线阶段，计算空间对象集合中任意一对空间对象之间的位置相近度和文本相似度，二者结合构成空间对象的关系紧密度，然后根据空间对象之间的关系紧密度从空间对象集合中获取少数代表性对象，并为每个代表性对象创建一个序列，序列中的元素是数据集中除该空间对象之外的所有空间对象，序列中的对象按其对代表性对象的关系紧密度降序排列。在线查询处理阶段，对于一个给定的空间关键字查询，首先计算该查询与每个代表性空间对象之间的关系紧密度，然后使用阈值算法 (threshold algorithm, TA) 在预创建的序列上快速选出前 k 个与当前查询最为接近的空间对象，其中查询条件与代表性空间对

象之间的关系紧密度作为评分函数的权重, 即当前查询条件与某个代表性空间对象在位置和文本上越接近, 那么该代表性空间对象对应的序列对结果影响越大。该算法的优势在于: 1) 在线查询处理阶段不需要检查所有与查询匹配的对象就能够快速识别出 top-k 结果。2) 选取的 top-k 结果对象在全局中具有代表性, 也就是典型化程度高。

1　空间对象的位置-文本关系度量

本节分别讨论空间对象的位置相近度和文本相似度的度量方法, 二者的线性叠加构成了空间对象之间的位置-文本关系紧密度。

表 1 给出了一个空间数据库实例, 每行代表一个空间对象。下文将以表 1 为例, 阐述空间对象的位置相近度和本文相似度计算方法。

表 1　空间数据库实例

Table 1　An instance of spatial database

对象	纬度	经度	文本描述
$\boldsymbol{o}_1$	116.36	39.91	泳池、wifi、早餐
$\boldsymbol{o}_2$	116.20	39.99	wifi、早餐
$\boldsymbol{o}_3$	110.58	35.74	早餐、泳池、地铁
$\boldsymbol{o}_4$	119.65	33.32	会议室、Internet、泳池
$\boldsymbol{o}_5$	121.16	42.58	Internet、接送机场、宠物

1.1　空间对象的位置相近度

空间对象的位置信息通常由经纬度表示, 现有方法大多采用两个空间对象之间的欧氏距离来计算二者之间的位置相近度。给定一对空间对象 $\boldsymbol{o}_i$ 和 $\boldsymbol{o}_j$, 它们之间的欧氏距离定义如下:

$$D(\boldsymbol{o}_i,\boldsymbol{o}_j)=\sum_{k=1}^{n} d(\boldsymbol{o}_i^{(k)},\boldsymbol{o}_j^{(k)}) \tag{1}$$

其中 n 代表空间对象的空间维度。基于式 (1), 空间对象 $\boldsymbol{o}_i$ 和 $\boldsymbol{o}_j$ 在位置上的相近度定义为:

$$\mathrm{Sim}_{\mathrm{Loc}}(\boldsymbol{o}_i,\boldsymbol{o}_j)=1-D(\boldsymbol{o}_i,\boldsymbol{o}_j)/\mathrm{Max}D \tag{2}$$

其中 Max$D$ 代表所有空间对象之间的最大距离。式 (2) 得到的是一个归一化结果, 即 $\mathrm{Sim}_{\mathrm{Loc}}()$ 的值介于 0 和 1 之间。表 2 给出了表 1 中所有空间对象之间的位置相近度。

表 2　空间对象的位置相近度

Table 2　The location proximity between spatial objects

	$\boldsymbol{o}_1$	$\boldsymbol{o}_2$	$\boldsymbol{o}_3$	$\boldsymbol{o}_4$	$\boldsymbol{o}_5$
$\boldsymbol{o}_1$	1.000 0	0.985 8	0.434 3	0.415 4	0.564 0
$\boldsymbol{o}_2$	0.985 8	1.000 0	0.440 7	0.403 9	0.555 9
$\boldsymbol{o}_3$	0.434 3	0.440 7	1.000 0	0.254 9	0.000 0
$\boldsymbol{o}_4$	0.415 4	0.403 9	0.254 9	1.000 0	0.255 3
$\boldsymbol{o}_5$	0.564 0	0.555 9	0.000 0	0.255 3	1.000 0

1.2　空间对象的文本相似度

给定一个空间对象 $\boldsymbol{o}$, 其文本信息 $\boldsymbol{o}$.doc 可由对象 $\boldsymbol{o}$ 的名字、设施描述、用户评论所构成, 本文利用 jieba、wikipedia 等工具先对空间对象的文本信息进行关键字提取, 进而 $\boldsymbol{o}$.doc 将由一组关键字集合表示, 其中每个关键字都有一个权重 w ($t|\boldsymbol{o}$.doc), 权重的计算方法采用传统的 TF-IDF 计算方法:

$$w(t|\boldsymbol{o}.\mathrm{doc})=\mathrm{tf}(t|\boldsymbol{o}.\mathrm{doc})*\mathrm{idf}(t,\boldsymbol{O}) \tag{3}$$

式中: 词频为 $\mathrm{tf}(t\boldsymbol{o}.\mathrm{doc})=\dfrac{f(t,\boldsymbol{o}.\mathrm{doc})}{\mathrm{MaxFrequency}}$; $f(t|\boldsymbol{o}.\mathrm{doc})$ 表示 t 在 $\boldsymbol{o}$.doc 中出现的次数; MaxFrequency 表示在 $\boldsymbol{o}$.doc 中关键字的最多出现次数; $\mathrm{idf}(t,\boldsymbol{O})=\log\dfrac{|\boldsymbol{O}|}{f(t|\boldsymbol{O})+1}$, 其中 $f(t|\boldsymbol{O})$ 表示空间对象集合 $\boldsymbol{O}$ 中包含关键字 t 的对象数量; $|\boldsymbol{O}|$表示空间对象集合中的对象总数。

在此基础上, 给定一对空间对象 ($\boldsymbol{o}_1$, $\boldsymbol{o}_2$), 其中每个对象的文本信息可转换成相应的向量表示, 向量的维度是空间数据库中所有空间对象文本信息中包含的所有不同关键字个数。然后, 利用 Cosine 相似度评估方法得到一对空间对象的文本相似度, 计算方法如下:

$$\mathrm{Sim}_{\mathrm{Doc}}(\boldsymbol{o}_1,\boldsymbol{o}_2)=\frac{\sum_{i=1}^{n}\boldsymbol{o}_1[i]\boldsymbol{o}_2[i]}{\sqrt{\sum_{i=1}^{n}\boldsymbol{o}_1[i]^2}*\sqrt{\sum_{i=1}^{n}\boldsymbol{o}_2[i]^2}} \tag{4}$$

式中: $\vec{\boldsymbol{o}}_1$、$\vec{\boldsymbol{o}}_2$ 分别是对象 $\boldsymbol{o}_1$、$\boldsymbol{o}_2$ 中关键字的向量表示。根据式 (4), 数据集中任意一对空间对象之间的文本相似度都可以在离线阶段计算得到。表 3 给出了表 1 中所有对象的文本相似度。

表 3　空间对象的文本相似度

Table 3　The text similarity between spatial objects

	$\boldsymbol{o}_1$	$\boldsymbol{o}_2$	$\boldsymbol{o}_3$	$\boldsymbol{o}_4$	$\boldsymbol{o}_5$
$\boldsymbol{o}_1$	1.000 0	0.928 4	0.171 3	0.077 4	0.000 0
$\boldsymbol{o}_2$	0.928 4	1.000 0	0.092 3	0.000 0	0.000 0
$\boldsymbol{o}_3$	0.171 3	0.092 3	1.000 0	0.048 0	0.000 0
$\boldsymbol{o}_4$	0.077 4	0.000 0	0.048 0	1.000 0	0.175 1
$\boldsymbol{o}_5$	0.000 0	0.000 0	0.000 0	0.175 1	1.000 0

1.3　空间对象的位置-文本关系紧密度

空间对象 $\boldsymbol{o}_i$、$\boldsymbol{o}_j$ 在位置上的相近度和在文本上的相似度通过线性叠加构成了它们之间的关系紧密度, 用 $\mathrm{Sim}(\boldsymbol{o}_i,\boldsymbol{o}_j)$ 表示:

$$\mathrm{Sim}(\boldsymbol{o}_i,\boldsymbol{o}_j)=\alpha\mathrm{Sim}_{\mathrm{Loc}}(\boldsymbol{o}_i,\boldsymbol{o}_j)+(1-\alpha)\mathrm{Sim}_{\mathrm{Doc}}(\boldsymbol{o}_i,\boldsymbol{o}_j) \tag{5}$$

其中 α 是一个调节参数 ($\alpha\in[0,1]$), α 值越大, 空间对象的位置相近度对总体紧密度的影响就越大;

反之，空间对象在文本上的相似度对总体紧密度的影响越大。需要指出的是，利用式 (5) 得到的空间对象之间的位置−文本关系紧密度在 [0, 1] 范围内。表 4 给出了表 1 中所有空间对象的位置-文本关系紧密度 (其中 α=0.5)。

表 4　空间对象的位置-文本关系紧密度

Table 4　The location-text closeness between spatial objects

	o_1	o_2	o_3	o_4	o_5
o_1	1.000 0	0.957 1	0.302 8	0.246 4	0.282 0
o_2	0.957 1	1.000 0	0.266 5	0.202 0	0.278 0
o_3	0.302 8	0.266 5	1.000 0	0.151 5	0.000 0
o_4	0.246 4	0.202 0	0.151 5	1.000 0	0.215 2
o_5	0.282 0	0.278 0	0.000 0	0.215 2	1.000 0

所有空间对象之间的位置−文本关系紧密度可用一个 n 阶矩阵 $\boldsymbol{R}$ 表示 (假设共有 n 个空间对象)，其中元素 r_{ij} 代表了空间对象 $\boldsymbol{o}_i$ 和 $\boldsymbol{o}_j$ 之间的位置−文本关系紧密度。由于矩阵 $\boldsymbol{R}$ 是一个对称矩阵，因此仅存储上半矩阵即可。

2　top-k 结果选取与排序方法

令 q 代表一个空间关键字查询，$\boldsymbol{O}$ 是空间对象集合，查询结果的 top-k 选取与排序问题定义如下：

$$\Gamma_k = \arg\max_{\Gamma} \sum_{i=1}^{k(k\ll n)} \mathrm{Sim}(q, \boldsymbol{o}_i) \tag{6}$$

式中：Γ_k 是包含 k 个结果对象的有序列表；n 表示数据集中空间对象的总数。top-k 结果选取的目的是从数据集中快速获取与给定查询在位置和文本上相关且具有代表性的前 k 个空间对象。

本文方法分为 3 个步骤：1) 从数据集中选取少量代表性空间对象。2) 为代表性空间对象创建序列，该序列是由数据集中其他对象与代表性对象之间的位置−文本关系紧密度的降序排列。3) 在线计算当前查询条件与所有代表性对象之间的紧密度，然后在预先创建的对象序列上使用 TA 算法快速获取 top-k 个结果对象。

2.1　选取代表性空间对象

选取代表性对象的基本思想：根据第 2 节的空间对象之间的位置−文本关系评估方法，使用代表性对象选取算法获取 $l(l\ll n)$ 个代表性空间对象，代表性空间对象集合用 W_l 表示，代表性对象用 $\bar{\boldsymbol{o}}_i$ 表示，即 $W_l = \{\bar{\boldsymbol{o}}_i | i \in l\}$。

本文提出了一种基于概率密度优先选取代表性对象的方法。该方法的基本思想是利用高斯核函数评估某个空间对象在整个对象集合中的概率密度，某个对象的概率密度越大，表明与其关系紧密的对象越多，则该对象也就越有代表性，即典型程度越高[18]。本文采用概率密度估计算法从空间对象集合中选取代表性对象。

对于一个空间对象集合 $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_n\}$，对象 $\boldsymbol{o}$ 的概率密度 $f(\boldsymbol{o}_i)$ 可以表示为：

$$f(\boldsymbol{o}_i) = \frac{1}{n}\sum_{j=1}^{n} G_h(\boldsymbol{o}_i, \boldsymbol{o}_j) = \frac{1}{n\sqrt{2\pi}}\sum_{j=1}^{n} \mathrm{e}^{-\frac{\mathrm{d}(\boldsymbol{o}_i,\boldsymbol{o}_j)^2}{2h^2}} \tag{7}$$

式中：$d(\boldsymbol{o}_i, \boldsymbol{o}_j)^2$ 是 $\boldsymbol{o}_i$ 和 $\boldsymbol{o}_j$ 的位置−文本距离 (用 1 减去空间对象之间的位置−文本关系紧密度)；$G_h(\boldsymbol{o}_i, \boldsymbol{o}_j) = \frac{1}{n\sqrt{2\pi}}\sum_{j=1}^{n} \mathrm{e}^{-\frac{\mathrm{d}(\boldsymbol{o}_i,\boldsymbol{o}_j)^2}{2h^2}}$ 是高斯核函数。

根据空间对象的位置-文本距离矩阵 $\boldsymbol{M}$(可从对象之间的位置-文本关系紧密度矩阵 $\boldsymbol{R}$ 转化而来) 和概率密度估计方法，下面采用淘汰思想选取代表性空间对象。该方法的基本过程如下：

1) 将空间对象集合 $\boldsymbol{O}$ 随机划分成若干小组，每个小组都包含 u 个空间对象，即将集合 $\boldsymbol{O}$ 划分成了 n/u 个小组，接着在每个小组中利用式 (7) 计算所有空间对象的概率密度，选取每个小组中概率密度最高的对象构成一个集合，然后从 $\boldsymbol{O}$ 中将其他对象去除。

2) 对于新得到的集合，重复上述过程，直到空间对象集合 $\boldsymbol{O}$ 中只剩下一个对象为止，并将该对象放入代表性对象候选集合中 (上述过程记为一次选取过程)。

3) 为了保证选取对象的准确性，需要将上述选取过程重复执行 m 次 (记为一轮)，这样代表性对象候选集合中最多会包含 m 个空间对象，接着在最初的空间对象集合 $\boldsymbol{O}$ 上计算这 m 个对象的概率密度，最后将具有最高概率密度的对象作为当前轮次的选取结果，并从 $\boldsymbol{O}$ 中将该空间对象去除。上述整个过程重复 l 轮，这样就能得到 l 个近似于准确解的代表性空间对象。

该算法的时间复杂度为 $\boldsymbol{O}$(lmun)。在本文实验部分，将重点比较在所提 2 种算法得到的代表性对象序列上进行 top-k 结果选取的效果和性能。

2.2　创建空间对象序列

对于每个代表性空间对象 $\bar{\boldsymbol{o}}_i$，为其构建一个序列 τ_i，该序列中的元素是数据集中除该空间对象外的所有空间对象，这些对象按其对代表性空间对象的紧密度降序排列。如图 1 是一个包含 n 个空间对象和 3 个代表性对象的空间对象序列示意图。权重是通过计算查询条件与代表性对象

之间的紧密度获取的, 每个序列左侧表示数据集中除代表性对象之外的空间对象 ID, 右侧表示空间对象在该序列中对应的分数, 该分数由对象在序列中的位置决定, 计算方法为

$$s(\boldsymbol{o}_i|\tau) = n - p(\boldsymbol{o}_i) + 1 \tag{8}$$

其中 $p(\boldsymbol{o}_i)$ 表示对象 $\boldsymbol{o}_i$ 在序列中的位置。

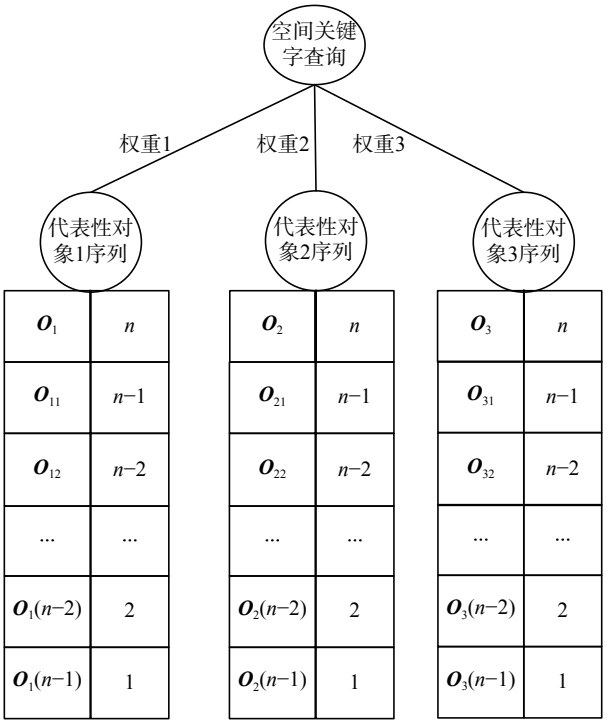


图 1　代表性空间对象序列示意

Fig. 1　Orders corresponding to representative objects

由于选取了 l 个代表性对象, 因此将生成 l 个代表性对象序列, 这些序列用于使用 TA 算法的查询结果 top-k 选取。

2.3　top-k 结果选取与排序

在代表性对象序列上, TA 算法通过顺序访问方式发现每个序列中的某个空间对象的排序分数, 然后利用随机访问方式从其他序列中发现该对象的排序分数, 这些分数之和作为该对象在所有序列中的总分数[19](注意, 总分数是以查询条件与该对象对应的代表性对象之间的位置−文本关系紧密度作为权重系数, 进行加权求和得到), 定义如下:

$$\text{score}(\boldsymbol{q}, \boldsymbol{o}_j) = \sum_{i=1}^{l} \text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i) s(\boldsymbol{o}_j|\boldsymbol{\tau}_i) \tag{9}$$

式中: τ_i 是对应代表性对象 $\bar{\boldsymbol{o}}_i$ 的序列; $\text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i)$ 表示代表性对象 $\bar{\boldsymbol{o}}_i$ 与查询 $\boldsymbol{q}$ 之间的位置-文本关系紧密度。需要注意的是, 在计算 $\bar{\boldsymbol{o}}_i$ 与 $\boldsymbol{q}$ 之间的文本相似度时, 转换成的向量的维度为查询 $\boldsymbol{q}$ 和所有代表性对象中包含的不同关键字总数。

基于上述思想, top-k 结果选取与排序算法处理步骤如下:

算法 1　top-k 结果选取与排序算法

输入　代表性序列集合 L, 空间关键字查询 $\boldsymbol{q}$, top-k 中的 k 值

输出　top-k 结果对象

1) 令 B={ }是一个缓存

2) 令 L 是一个大小为 l 的数组, 存储每个序列中最近一次检索得到的分数 (score)

3) repeat

4) for each $i \in \{1,2,\cdots,l\}$ do

5) 从序列 $\boldsymbol{\tau}_i$ 中检索下一个对象 $\boldsymbol{o}_j$, 计算 $\boldsymbol{o}_j$ 在 $\boldsymbol{\tau}_i$ 中的分数: $s(\boldsymbol{o}_j, q) = \text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i) s(\boldsymbol{q}_j|\boldsymbol{\tau}_i)$

6) 用对象 $\boldsymbol{o}_j$ 在 $\boldsymbol{\tau}_i$ 中的分数更新 L 中对应于对象 $\boldsymbol{o}_j$ 的分数

7) 利用随机访问方式获取 $\boldsymbol{o}_j$ 在其他序列中的分数, 将所有检索到的分数加权求和, 得到 score $(\boldsymbol{o}_j, \boldsymbol{q})$

8) 按照降序方式, 将<$\boldsymbol{o}_j$, score$(\boldsymbol{o}_j, \boldsymbol{q})$>插入到 B 中的正确位置

9) end for

10) until $B[k]\,\text{score} \geqslant \sum_{i=1}^{l} L[i]$

11) return B

算法 1 的处理过程由以下 3 步组成:

1) 循环访问每个代表性序列。在每次循环过程中, 当一个空间对象 $\boldsymbol{o}_j$ 在某个序列 $\boldsymbol{\tau}_i$ 中被发现时, 计算它在该代表性序列中的分数, 计算公式为

$$s(\boldsymbol{o}_j, \boldsymbol{q}) = \text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i) s(\boldsymbol{o}_j|\boldsymbol{\tau}_i) \tag{10}$$

该分数由 2 部分构成: 一部分是 $\text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i)$, 表示查询条件 $\boldsymbol{q}$ 与序列 $\boldsymbol{\tau}_i$ 对应的代表性空间对象 $\bar{\boldsymbol{o}}_i$ 之间的位置−文本关系紧密度; 另一部分是 $s(\boldsymbol{o}_j|\boldsymbol{\tau}_i)$, 表示对象 $\boldsymbol{o}_j$ 在序列 $\boldsymbol{\tau}_i$ 中的排序分值。之后, 通过随机访问方式获取 $\boldsymbol{o}_j$ 在其他序列中的分数, 由式 (9) 可计算出对象 $\boldsymbol{o}_j$ 对于查询 $\boldsymbol{q}$ 的综合分数。

2) 令 $s(\boldsymbol{o}_m|\boldsymbol{\tau}_i)$ 为第 m 次循环结束后, 在每个序列 $\boldsymbol{\tau}_i$ 中最后被访问对象的分数。TA 算法阈值 threshold 为第 j 次循环结束后数组 L 中的分数之和。

$$\text{threshold} = \sum_{i=1}^{l} \text{Sim}(\boldsymbol{q}, \bar{\boldsymbol{o}}_i) s(\boldsymbol{o}_m|\boldsymbol{\tau}_i) \tag{11}$$

阈值 threshold 根据式 (11) 由算法自动计算得到, 每当进行新一轮循环 (算法 1 中的 3)~10)) 时, 算法 1 会重新自动计算一次阈值, 当缓存 B 中存在 k 个总体排序分数都不小于当前阈值的对象

时，算法终止。通过使用阈值 threshold，使得第 1 步无需遍历每个代表性序列中的所有对象就能够提前结束算法执行，因此提高了执行效率。

3) 在所有被发现的空间对象中，输出前 k 个具有最高总体排序分数的对象。

注意，当第 2) 步完成后，对于任意一个没有在循环访问中被发现的对象 $\boldsymbol{o}_n$，它的总体排序分数都将小于设定的阈值，即 $s(\boldsymbol{o}_n, \boldsymbol{q})<$threshold。

3　性能实验分析

3.1　实验环境

所有实验在 Windows 10 操作系统、Intel i7 3.1-GHz CPU 和 12 GB 内存的电脑上运行，使用下列真实数据来评估所提算法的性能和效果。

数据集：测试数据使用 Yelp 数据集（该数据集来自 Yelp 数据集官网：http://www.yelp.com/dataset），Yelp 是美国最大的点评网站，包含了用户签到信息、POI 地点信息、用户评论信息等数据。实验截取经度在−115.0 到−110.9 之间，纬度在 32.3 至 35.6 之间的 53 516 个兴趣点作为分析数据。每个兴趣点都包含位置信息和文本信息，位置信息通过经度和纬度表示，文本信息由 POI 的 name、city、categories、postal_code、stars 等属性对应的信息构成。数据存放在文本文件中。测试数据集的特点如表 5 所示。

表 5　测试数据集特点

Table 5　The properties of Yelp dataset

特征	Yelp
POI总数	53 516
数据集中所有不同的关键字个数	94 035
每个POI平均拥有的关键字个数	7
数据集中所有的关键字个数	658 246

3.2　top-k 结果的准确性和典型性测试

从数据集中获取前 k 个与给定查询最为相关的空间对象，准确方法是计算给定查询与数据集中所有空间对象之间的位置-文本关系紧密度，然后再按紧密度大小选取出 top-k 个空间对象。因此，本实验需要验证准确获取 top-k 个空间对象与利用 TA 算法在代表性对象序列上选出的 top-k 个空间对象之间的重叠程度，即本文 top-k 查询方法的准确性。这里用 R(Rep, k) 表示在代表性对象序列上选取的 top-k 个对象；R(All, k) 表示通过计算数据集中所有对象与给定查询之间的紧密度选取的 top-k 个对象。这两个集合的重叠度可用 Jaccard 系数进行评估：

$$\mathrm{J}(R(\mathrm{Rep},k),R(\mathrm{All},k))=\frac{\left|R(\mathrm{Rep},k)\cap R(\mathrm{All},k)\right|}{\left|R(\mathrm{Rep},k)\cup R(\mathrm{All},k)\right|} \tag{12}$$

Jaccard 系数的值在 [0,1] 之间，值越高表明 2 个集合的重叠度越高，即 top-k 结果的准确性也就越高。

从数据集中随机抽取 10 个空间对象，再从每个空间对象的文本描述信息中随机抽取 2−4 个关键字，最后将每个空间对象的经纬度信息和抽取的关键字组合在一起形成空间关键字查询。此外，用参数 l 表示代表性对象序列的个数 (选取与当前查询紧密度最高的前 l 个代表性对象对应的序列)，k 表示返回的结果个数。在空间对象之间以及代表性对象与当前查询的位置-文本关系紧密度计算中，式 (5) 的调节参数 α 设为 0.5。图 2 给出了在 Yelp 数据集上分别使用本文方法 (当 l={2, 3, 4, 5, 6}时) 和 IR-tree 索引结构在不同 k 值下得到的 top-k 结果的准确性对比。

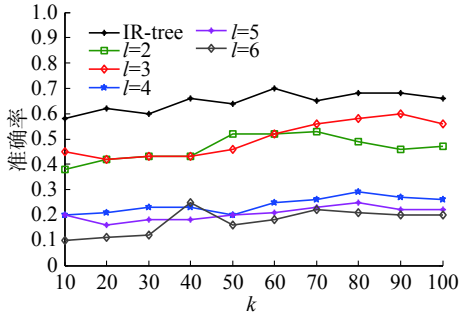


图 2　利用本文方法和 IR-tree 索引得到的 top-k 结果准确性对比

Fig. 2　Comparison for the precision of top-k results returned by using our method and IR-tree index

从图 2 可以看出：1)IR-tree 方法得到的 top-k 结果的准确性一直优于本文方法，top-10, 20, …, 100 的平均准确性达到 66%。2) 当 l={2, 3}时，使用本文方法得到的 top-k 结果的准确性较高，当 l=3 时准确性最高，top-10, 20, …, 100 的平均准确性为 51%。但当代表性序列超过 3 个时，准确性就会降低，原因是参与 top-k 结果选取的序列数越多，序列对应的代表性对象与当前查询的紧密度越低，因此参与序列与真实结果序列的相关度就越低，进而导致 top-k 结果准确性就越低。

需要指出的是，虽然 IR-tree 方法的准确性一直优于本文方法，但本文目的是在确保 top-k 结果具有一定准确性的前提下，同时具有更高的多样性和典型性。因此，还需测试本文方法得到的 top-k 结果与利用 IR-tree 索引得到的 top-k 结果，二者在典型程度上的差别。查询结果典型程度的

评价标准为:

$$\text{Typicality}(T)=\frac{\sum_{i=1}^{k} f(o_i)}{k} \tag{13}$$

式中: T 代表 top-k 结果集合; k 表示结果对象个数; $f(o_i)$ 根据式 (7) 计算。该实验中令 k=10。top-k 结果典型程度的计算范围按如下方法确定: 取本文方法和 IR-tree 方法得到的 top-k 结果中与给定查询紧密度最低的对象作为阈值, 高于该阈值的对象构成的集合就是计算 top-k 结果典型程度的对象范围。top-k 结果的典型程度越高, 说明结果对象越能体现整个相关结果集合的总体特征, 因此也就越能开阔用户视野和有助于用户对查询相关结果集合的总体认知。典型程度比较的基准是在对应计算范围内, 利用式 (7) 计算得到的典型程度最高的前 10 个对象的平均典型程度。图 3 给出了不同测试查询下本文方法和 IR-tree 索引得到的 top-k 结果典型程度在基准下的对比。

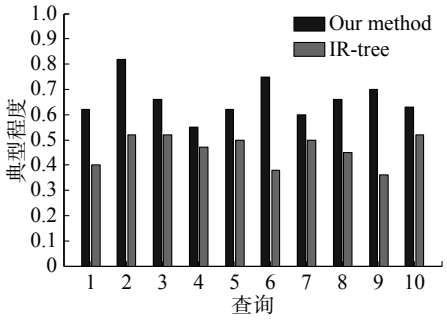


图 3　本文方法和 IR-tree 得到的 top-k 结果典型程度对比

Fig. 3　Comparison for the typicality of the top-k results returned by using our method and IR-tree index

图 3 显示了本文方法得到的 top-k 结果的典型程度一直明显高于 IR-tree, 原因是本文方法是从代表性对象对应的序列中获取结果, 这些结果对象与代表性对象的紧密度较高, 相应地典型程度也就较高, 而 IR-tree 方法得到的 top-k 结果之间通常比较相似, 多样性和典型程度相对较低。综合图 2 和图 3 还可以得出, 本文方法在准确性较高的前提下, 在很大程度上提高了查询结果的典型程度, 即结果对象更具代表性, 能够更好地满足用户对查询结果的相关性和典型性查询需求。

3.3　top-k 查询算法的性能测试

该实验的目的是测试本文提出的 top-k 查询算法的响应时间。在该实验中, 设置代表性序列个数 l 分别为 2、3、4 和 5, 然后测试在不同 k 值下, 分别使用本文方法和 IR-tree 索引获取 top-k 结果的响应时间 (如图 4 所示)。

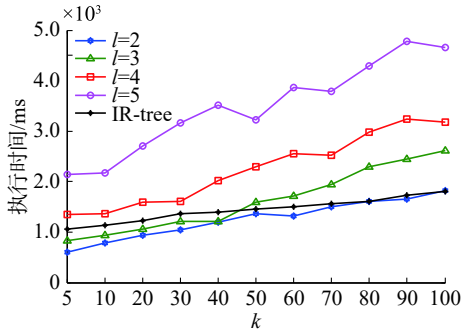


图 4　不同 l 和 k 值下利用本文方法和 IR-tree 的 top-k 查询响应时间

Fig. 4　Execution time of top-k query selection by using our method and IR-tree index under different values of l and k

从图 4 可以看出, 当 l={2, 3}时本文方法与 IR-tree 索引的响应时间相近, 并且当 k≤40 时, 响应时间要低于 IR-tree, 这也说明了 TA 算法在 k 值较小时具有优越的性能。此外, 本文方法的响应时间随着 k 和 l 值的增加而逐渐增长, 其原因是当 k 和 l 增加后, 算法需要处理的对象个数也会增多, 因此导致响应时间增加。

4　结束语

本文提出了一种基于位置-文本关系的空间对象 top-k 查询与排序方法, 目的是从空间数据库中快速获取与查询位置和文本相关且具有代表性的 top-k 空间对象。实验结果表明, 提出的 top-k 查询与排序方法同时具有较高的准确性、典型性和执行效率, 特别适用于大规模空间数据库的空间关键字 top-k 检索。

本文方法与现存方法有 2 个方面不同: 1) 提出了基于概率密度的代表性对象选取算法来获取代表性空间对象, 进而构建代表性空间对象序列, 为快速选取 top-k 结果提供基础。2) 将阈值算法 threshold algorithm(TA) 算法应用到了对空间对象的 top-k 查询与排序中, 在确保较高查询准确性的前提下兼顾了查询结果的典型性, 并且具有较快的响应时间。下一步工作将研究如何将公路网络 (road network) 应用于空间对象之间的距离计算, 以及如何进行查询结果的局部典型化和多样典型化选取。

参考文献:

[1] QI Jianzhong, ZHANG Rui, JENSEN C S. Continuous spatial query processing: a survey of safe region based techniques[J]. ACM computing surveys, 2018, 51(3): 64.

[2] CONG Gao, JENSEN C S. Querying geo-textual data: spa-

tial keyword queries and beyond[C]//Proceedings of 2016 International Conference on Management of Data. San Francisco, California, USA, 2016: 2207−2212.

[3] ZHENG Kai, SU Han, ZHENG Bolong, et al. Interactive top-k spatial keyword queries[C]//Proceedings of the 31st International Conference on Data Engineering. Seoul, South Korea, 2015: 423–434.

[4] LU Ying, LU Jiaheng, CONG Gao, et al. Efficient algorithms and cost models for reverse spatial-keyword *k*-nearest neighbor search[J]. ACM transactions on database systems, 2014, 39(2): 13.

[5] DE FELIPE I, HRISTIDIS V, RISHE N. Keyword search on spatial databases[C]//Proceedings of the 24th IEEE International Conference on Data Engineering. Cancun, Mexico, 2008: 656–665.

[6] LI Guoliang, XU Jing, FENG Jianhua. Keyword-based k-nearest neighbor search in spatial databases[C]//Proceedings of the 21st ACM International Conference on Information and Knowledge Management. Maui, Hawaii, USA, 2012: 2144–2148.

[7] WANG Xiang, ZHANG Ying, ZHANG Wenjie, et al. Skype: top-k spatial-keyword publish/subscribe over sliding window[J]. Proceedings of the VLDB endowment, 2016, 9(7): 588–599.

[8] ZHANG Dongxiang, CHEE Y M, MONDAL A, et al. Keyword search in spatial databases: towards searching by document[C]//Proceedings of the 25th International Conference on Data Engineering. Shanghai, China, 2009: 688–699.

[9] CONG Gao, JENSEN C S, WU Dingming. Efficient retrieval of the top-k most relevant spatial web objects[J]. Proceedings of the VLDB endowment, 2009, 2(1): 337–348.

[10] CHEN Lei, LIN Xin, HU Haibo, et al. Answering why-not questions on spatial keyword top-k queries[C]//Proceedings of the 31st International Conference on Data Engineering. Seoul, South Korea, 2015: 279–290.

[11] KWON H Y, WANG Haixun, WHANG K Y. G-index model: a generic model of index schemes for top-k spatial-keyword queries[J]. World wide web, 2015, 18(4): 969–995.

[12] GUTTMAN A. R-trees: a dynamic index structure for spatial searching[C]//Proceedings of 1984 ACM SIGMOD International Conference on Management of Data. Boston, Massachusetts, 1984: 47–57.

[13] BECKMANN N, KRIEGEL H P, SCHNEIDE R, et al. The R*-tree: an efficient and robust access method for points and rectangles[C]//Proceedings of 1990 ACM SIGMOD International Conference on Management of Data. Atlantic City, New Jersey, USA, 1990: 322–331.

[14] ZHANG Chengyuan, ZHANG Ying, ZHANG Wenjie, et al. Inverted linear quadtree: efficient top k spatial keyword search[J]. IEEE transactions on knowledge and data engineering, 2016, 28(7): 1706–1721.

[15] ZHOU Yinghua, XIE Xing, WANG Chuang, et al. Hybrid index structures for location-based Web search[C]//Proceedings of the 14th ACM International Conference on Information and Knowledge Management. Bremen, Germany, 2005: 155–162.

[16] LI Zhisheng, LEE K C K, ZHENG Baihua, et al. IR-Tree: an efficient index for geographic document search[J]. IEEE transactions on knowledge and data engineering, 2011, 23(4): 585–599.

[17] HONG H J, CHIU G M, TSAI W Y. A single quadtree-based algorithm for top-k spatial keyword query[J]. Pervasive and mobile computing, 2017, 42: 93–107.

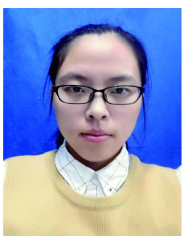
[18] HUA Ming, PEI Jian, FU A W C, et al. Top-*k* typicality queries and efficient query answering methods on large databases[J]. The VLDB journal, 2009, 18(3): 809–835.

[19] FAGIN R, LOTEM A, NAOR M. Optimal aggregation algorithms for middleware[J]. Journal of computer and system sciences, 2003, 66(4): 614–656.

作者简介：

孟祥福，教授，博士生导师，主要研究方向为空间关键字查询、大数据分析与可视化、机器学习算法。主持国家自然科学基金项目 2 项、辽宁省各类基金项目 3 项。发表学术论文 30 余篇。

张霄雁，工程师，主要研究方向为空间数据查询与分析、城市计算、机器学习算法。主持辽宁省教育厅科研项目 1 项。发表学术论文 10 余篇。



赵路路，硕士研究生，主要研究方向为空间数据查询与分析。