https://crad.ict.ac.cn
62(3): 808−819, 2025

基于时空约束和成本感知的集合空间关键字查询

李　松　曹文琪　郝晓红　张丽平　郝忠孝
（哈尔滨理工大学计算机科学与技术学院　哈尔滨　150080）
（lisongbeifen@163.com）

Collective Spatial Keyword Query Based on Time-Distance Constrained and Cost Aware

Li Song, Cao Wenqi, Hao Xiaohong, Zhang Liping, and Hao Zhongxiao
（*School of Computer Science and Technology*, *Harbin University of Science and Technology*, *Harbin* 150080）

Abstract　Collective spatial keyword queries play an important role in the fields such as spatial databases, location services, intelligent recommendations, and group intelligence perception. The existing collective spatial keyword query methods do not consider the problem of requiring time-distance constrained and cost aware, and cannot meet the query needs of most users under time-distance constrained. Existing research results have significant limitations. To make up for the shortcomings of existing methods, collective spatial keyword query based on time-distance constrained and cost aware (called TDCCA-CoSKQ) is proposed. To address the issue of not being able to include both keyword information and time information in existing indexes, the TDCIR-Tree index is proposed, which combines inverted files and time attribute label files. TDCIR-Tree can reduce the cost of query calculation. TDCCA_PP algorithm is proposed to address the issue of subsequent screening of collections that meet query criteria for TDCCA-CoSKQ, including *TDCCAPruning*1, *TDCCAPermutation*, and *TDCCAPruning*2, and it can improve the efficiency of keyword queries. The TDC cost function and its corresponding sorting algorithm are proposed. The TDC cost function is composed of distance cost and time cost, which includes independent variable coefficients representing user preference α and β, and it can increase users' freedom of choice. The problem of existing cost functions not meeting the collective spatial keyword query based on time-distance constrained and cost aware is effectively solved. Theoretical research and experiments have shown that the proposed method has good efficiency and accuracy.

Key words　collective spatial keyword query; time-distance constrained; cost aware; inverted files; time attribute label files

摘　要　集合空间关键字查询在空间数据库、位置服务、智能推荐和群智感知等领域具有重要的作用.现有的集合空间关键字查询方法没有考虑要求同时带有时空约束和成本感知的问题,不能满足大部分用户在时空约束条件下的查询需求问题,已有研究成果具有较大的局限性.为了弥补已有方法的不足,提出一种基于时空约束和成本感知的集合空间关键字查询 TDCCA-CoSKQ.为了解决现有索引中无法同时包含关键字信息和时间信息的问题,提出了一种 TDCIR-Tree 索引,该索引融合了倒排文件和时间属性标签文件,可以减小查询计算的开销;为了有效地筛选出符合查询条件的集合,提出了一种 TDCCA_PP 算法,其

收稿日期: 2023-10-11; **修回日期**: 2024-07-02
基金项目: 国家自然科学基金项目(62072136); 黑龙江省重点研发计划项目(JD2023SJ20); 国家重点研发计划项目(2020YFB1710200)
The work was supported by the National Natural Science Foundation of China (62072136), the Key Research and Development Project of Heilongjiang Province (JD2023SJ20), and the National Key Research and Development Program of China (2020YFB1710200).
通信作者: 张丽平(zhangliping0730@163.com)

中包括第 1 层剪枝算法、组间有序排列和第 2 层剪枝算法，可以提高关键字的查询效率；进一步提出了一种基于 TDC 成本函数的排序算法，TDC 成本函数是由距离成本和时间成本组成的，其中包含代表用户偏好度的自变量系数 α 和 β，可以增加用户的选择自由度，有效解决了现有的成本函数无法满足时空约束和成本感知的集合空间关键字查询的问题. 理论研究与实验表明，所提出的方法具有较好的效率与准确性.

关键词 集合空间关键字查询；时空约束；成本感知；倒排文件；时间属性标签文件

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202330815 **CSTR:** 32373.14.issn1000-1239.202330815

随着互联网信息技术和空间定位技术的快速发展，产生了海量的具有空间位置信息的文本对象，空间数据和文本数据的结合形成了一类新的具有海量、异构、多维等特点的数据，即空间文本数据. 位置服务数据、群智感知数据、社交网络中的社交数据和移动互联网中的交易数据等都是空间文本数据的重要来源.

以空间文本数据为背景的空间关键字查询(也称空间文本查询)是空间查询和关键字搜索的结合，给定一个查询关键字和一个查询位置，也即用户的查询不仅包含关键字而且包含地理位置，查询返回的结果则是满足空间和文本约束并与查询相关的对象，即返回距离查询位置相对接近且与查询关键字相关性较高的空间对象. 空间关键字查询在实际中的应用越来越广泛，研究发现单个对象不容易满足用户需求，但是对象可以组合成集合以满足用户需求，等同于集合中的对象共同满足用户需求，即集合空间关键字查询(collective spatial keyword query, CoSKQ).

尽管人们在集合空间关键字查询领域进行了一些重要的研究，但专门针对基于时空约束和成本感知的集合空间关键字查询处理的工作较为缺乏. 本文主要研究的是基于时空约束和成本感知的集合空间关键字查询问题，即找到一个可行解，满足用户预先设定的约束条件，并且它的成本尽可能的小. 对象的属性方面被解释为对象的成本，成本集的成本对应于成本的聚合形式. 使用成本有利于让用户更灵活地表达其偏好，施加约束条件并优化成本有利于捕获更多适合用户的用例，成本会随着约束条件的变化而变化.

本文的主要贡献包括 3 个方面：

1) 针对集合空间关键字查询的现有查询算法和索引结构已经不能满足大部分用户在时空约束条件下的需求问题，提出了一种基于时空约束和成本感知的集合空间关键字查询 TDCCA-CoSKQ. TDCCA-CoSKQ 需要处理大量的复杂时空数据对象，会产生很大的计算开销，为了提升查询效率，本文新提出了一种 TDCIR-Tree 索引结构，该索引融合了倒排文件和时间属性标签文件、非叶子结点连接关键字倒排文件、叶子结点连接时间属性标签文件，有效解决了现有索引中无法同时包含关键字信息和时间信息的问题，减小了查询计算开销.

2)为了有效筛选出符合查询条件的集合，提出了 TDCCA_PP 算法，TDCCA_PP 算法包含第 1 层剪枝、组间有序排列和第 2 层剪枝，从第 1 层剪枝、组间有序排列和第 2 层剪枝后分别得到可行集合、组合集合和有效集合，有效提高了查询效率.

3)针对现有的成本函数无法有效满足 TDCCA-CoSKQ 查询的问题，提出了一种基于 TDC 成本函数的排序算法，TDC 成本函数是由距离成本和时间成本组成的，其中包含代表用户偏好度的自变量系数 α 和 β，有效增加了用户的选择自由度，排序算法将有效集合放入 TDC 成本函数计算并得到以成本为升序的有序集合.

1 相关工作

空间关键字查询在实际中的应用越来越广泛，许多空间关键字查询变体也被研究，包括布尔空间关键字查询[1]、空间关键字近邻查询[2]、连续空间关键字查询[3]、支持语义的空间关键字查询[4]、面向空间兴趣区域的路线查询[5]、top-k 地理-社会的关键字查询[6]和反向 top-k 地理-社会的关键字查询[7]、基于关联规则挖掘的空间关键字查询索引[8]、基于隐私保护的空间关键字查询[9]、基于时间感知的空间关键字查询[10]、路网环境下的空间关键字查询[11]等.

随着人们需求的增加，单个对象已无法满足用户需求，文献 [12] 提出的集合空间关键字查询是指检索一组空间 Web 对象，每组对象都与一组关键字相关联，使得该组包含查询的关键字，并且通过它们

到查询点的距离以及组中对象之间距离的成本最低. 针对处理大规模数据集中存在不足的问题，文献[13]提出了可扩展的集合空间关键字查询，并提出了一个遵循 Spark 编程范式的分布式解决方案. 文献[14]研究了 top-k 集合空间关键字查询(TkCoSKQ)，提出了启发式算法. 文献[15]提出了一种贪心算法和一种具有可证明近似界的近似算法，为了提高查询性能，提出了新的剪枝策略和启发式规则，有效解决了现有集合空间关键字查询算法在生成可行候选集时忽略不同目标之间距离的问题.

基于成本感知的集合空间关键字查询作为集合空间关键字查询的一个重要分支，在查询过程中使用成本有利于让用户更灵活地表达其偏好，施加约束条件并优化成本有利于捕获更多适合用户的用例. 文献[16]研究了固有成本感知的集合空间关键字查询，提出了一种代价成本函数，捕获对象之间的距离和对象的固有成本，并提出了精确算法和近似算法. 文献[17]提出了一种层次感知的集合空间关键字查询，该查询返回一组对象通过阈值约束，共同覆盖查询关键字并最小化代价成本函数，并提出了精确算法和具有可证明近似约束的近似算法. 为了系统地解决集合空间关键字查询问题，文献[18]提出了统一成本函数和统一方法. 文献[19]研究了知识库上的集合关键字查询(CoSKQ-KB)，为了找到一组语义位置来共同覆盖查询关键字，并最小化排名代价成本函数，提出了一种可扩展方法. 为了在距离约束下找到成本最小的对象集，文献[20]研究了成本感知和距离约束的集合空间关键字查询(CD-CoSKQ)，提出了一种精确算法和一种近似算法，为用户提供了一个更细粒度的接口来表达他们在地理文本对象的地理空间方面和属性方面的偏好. 随着互联网技术的快速发展，用户的查询需求越来越多元化和个性化，基于时空约束的集合空间关键字查询在实际应用中越来越广泛. 为了满足对象和查询之间的位置相关性、文本相关性和时间相关性，文献[21]提出了时间感知的集合空间关键字查询(TCoSKQ)，并且提出了 2 个评估函数及对应算法来满足用户的不同需求. 针对现有的工作大多只关注精确关键字匹配的集合关键字查询问题，文献[22]研究了交通网络环境下时间感知的近似集合关键字查询(TACoSKQ)，提出了一种新的混合索引 TDAG-Tree 和 2 种具有证明的近似界的近似算法，有效处理了在实际应用中常见的拼写错误和传统拼写差异的问题.

针对已有的集合空间关键字查询方法没有考虑要求同时处理带有时空约束和成本感知的问题，不能满足大部分用户在时空约束条件下更为复杂的查询需求问题，实际应用中具有很大的局限性，为了弥补已有方法的不足，本文提出了一种基于时空约束和成本感知的集合空间关键字查询 TDCCA-CoSKQ.

2 主要定义

定义 1. 时间区间和有效时间. 查询关键字 key_{n_i} 的时间区间的符号表示为 $key_{n_i}.t=(st,pt)$，时间戳是由 2 个整数组成的，由于用户查询点的时间域与时空数据对象的时间域结点不一定为整点，我们提出时间区间的概念，将时间域最小单位设为 30 min. 有效时间是时间区间中计划结束时间 pt 与计划开始时间 st 的差值，有效时间的符号表示为 $\left|key_{n_i}.t\right|=pt-st$.

定义 2. 初始集合. 在 TDCIR-Tree 索引中查询到的满足关键字 key_{n_i} 及其时间域的集合称为初始集合，用 G 表示，第 i 个初始集合用 G_{n_i} 表示.

定义 3. 可行集合. 在所提出的 TDCCA_PP 算法的第 1 层剪枝中，基于初始集合 G_{n_i} 得到不仅满足关键字 key_{n_i} 及其时间域，G_{n_1} 和 G_{n_n} 也满足用户查询的距离域 $q.b$ 的集合称为可行集合，用 G' 表示，第 i 个可行集合用 G'_{n_i} 表示，G'_{n_i} 中有 m_i 个时空数据对象.

定义 4. 组合集合. 在所提出的 TDCCA_PP 算法的组间有序排列中，基于可行集合 G'_{n_i} 得到的集合称为组合集合，用 M 表示，第 i 个组合集合用 M_i 表示，组合集合 M 的总数记作 f，$f=\prod_{i=1}^{n}m_i$.

定义 5. 有效集合. 在所提出的 TDCCA_PP 算法的第 2 层剪枝中，基于组合集合 M_i 得到不仅满足关键字 key_{n_i} 及其时间域而且同样满足用户查询的距离阈值 $q.b$ 的集合称为有效集合，用 M' 表示，第 i 个有效集合用 M'_i 表示.

定义 6. 有序集合. 在所提出的 TDC 成本函数对应的排序算法中，基于有效集合 M'_i 得到的以 TDC 成本函数计算结果为升序的集合称为有序集合，用 V 表示，第 i 个有序集合用 V_i 表示，有序集合 V 的总数记作 h，有序集合 V 为基于时空约束和成本感知的集合空间关键字查询的最终返回结果集.

定义 7. TDC 成本函数. TDC 成本函数是由距离成本和时间成本组成的，自变量系数 α 和 β 称为用户偏好度，可以由用户自行调整，并且规定 $\alpha+\beta=1$. 基于统一成本函数[18]，计算有效集合 M'_i 的 TDC 成本函数表示为

$$cost(M_i')=\alpha\times(dist(q,o_{key_{n_1}})+dist(q,o_{key_{nn}})+\sum_{i=2}^{n-1}dist(o_{key_{n_i}},o_{key_{n_{i+1}}}))+\beta\times(1-\frac{1}{n}\sum_{i=1}^{n}\frac{|key_{n_i}.t\cap o_{key_{n_i}}.t|}{|key_{n_i}.t|}),\quad(1)$$

$\alpha\times(dist(q,o_{key_{n_1}})+dist(q,o_{key_{nn}})+\sum_{i=2}^{n-1}dist(o_{key_{n_i}},o_{key_{n_{i+1}}}))$ 表示的是 TDC 成本函数中的距离成本，$\beta\times(1-\frac{1}{n}\sum_{i=1}^{n}\frac{|key_{n_i}.t\cap o_{key_{n_i}}.t|}{|key_{n_i}.t|})$表示的是 TDC 成本函数中的时间成本. $\sum_{i=2}^{n-1}dist(o_{key_{n_i}},o_{key_{n_{i+1}}})$表示的是 M_i' 中相邻时空数据对象 $o_{key_{n_i}}$ 和 $o_{key_{n_{i+1}}}$ 的欧氏距离之和，α对应的乘数表示的是从查询点q开始顺序经过 M_i' 中所有时空数据对象再返回查询点q的欧氏距离之和，$|key_{n_i}.t\cap o_{key_{n_i}}.t|$表示的是 $o_{key_{n_i}}$ 在关键字 key_{n_i} 的有效时间，$\frac{|key_{n_i}.t\cap o_{key_{n_i}}.t|}{|key_{n_i}.t|}$表示的是 $o_{key_{n_i}}$ 在关键字 key_{n_i} 的有效时间与 key_{n_i} 有效时间的比值，称为 $o_{key_{n_i}}$ 的有效时间比，$\sum_{i=1}^{n}\frac{|key_{n_i}.t\cap o_{key_{n_i}}.t|}{|key_{n_i}.t|}$表示的是 M_i' 中 n 个时空数据对象有效时间比的和，称为 M_i' 的有效时间比.

定义 8. 集合空间关键字查询[12]. 令 S 为一组关键字，令 D 为由 d 个空间网络对象 a 组成的数据库，D 中的每个对象 a 都与一个位置 $a.\lambda$ 和一组描述该对象的关键字 $a.\psi$ 相关联且 $a.\psi\subset S$，考虑一个集合空间关键字查询 q，$q=(q.\lambda,q.\psi)$，其中 $q.\lambda$ 表示一个位置，$q.\psi$ 表示一组关键字. 集合空间关键字查询的目的是在 D 中找到一组对象 $\chi(\chi\subseteq D)$，使得 $\bigcup_{r\in\chi}r.\psi\supseteq q.\psi$，并且使得成本最小化.

定义 9. 基于时空约束和成本感知的集合空间关键字查询. 给定一个时空数据对象集合 O 和一个查询点 q. 其中集合 O 是由具有位置信息、文本描述及其对应的时间信息的时空数据对象 o 组成的集合，查询点 q 是具有查询位置信息、查询文本描述及其对应的时间信息、查询距离阈值的查询点. 基于时空约束和成本感知的集合空间关键字查询的目的是在 O 中查询并返回以有效集合 M' 的成本为升序的有序集合 V，其中集合 V 不仅满足查询关键字及其对应时间域，而且满足查询距离阈值. 时空数据对象 o 的表达式为 $o=(o.id,o.loc,o.\varphi,o.\varphi.t(bt,et))$，其中 $o.id$ 表示的是 o 的唯一标识，$o.loc$ 表示的是 o 的位置信息，$o.\varphi$ 表示的是 o 的关键字集合，$o.\varphi.t(bt,et)$ 表示的是 o 的关键字对应的时间区间，bt 和 et 表示的是 o 的开始时间和结束时间；查询点 q 的表达式为 $q=(q.loc,q.\varphi,q.\varphi.t(st,pt),q.b)$，其中 $q.loc$ 表示的是 q 的位置信息，$q.\varphi$ 表示的是 q 的查询关键字集合，$q.\varphi.t(st,pt)$ 表示的是 q 的关键字对应的时间区间，st 表示的是 q 计划开始的时间，pt 表示的是 q 计划结束的时间，$q.b$ 表示的是 q 的距离范围，也称为距离阈值.

3 基于时空约束和成本感知的集合空间关键字查询过程

本文所提出的基于时空约束和成本感知的集合空间关键字查询主要分 3 个部分：TDCIR-Tree 索引的构建与查询、TDCCA_PP 算法的构建与查询、TDC 成本函数的计算和排序.

3.1 TDCIR-Tree 索引的构建与查询

我们将查询点及 s 个时空数据对象的查询场景采用最小边界矩阵 MBR 来进行划分并构建索引，从叶子结点开始用 MBR 将空间框起来，结点越往上越靠近根结点，框住的空间就越大，从而实现对空间的分割. TDCIR-Tree 索引的空间划分图示例如图 1 所示. TDCIR-Tree 索引图示例如图 2 所示.

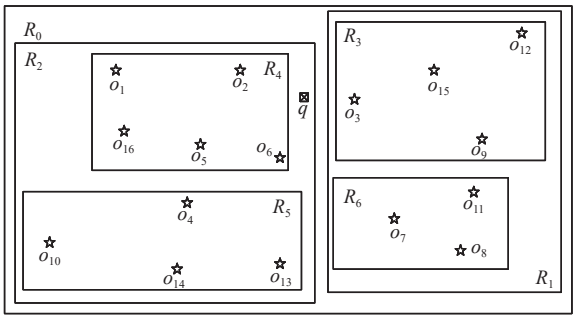


Fig. 1 Partition illustration example of TDCIR-Tree

图 1 TDCIR-Tree 划分示意图示例

TDCIR-Tree 以 R-Tree 的形式进行构建，由空树开始，不断插入划分的 MBR，直到生成一棵完整的 R-Tree，为了提高查询效率，在 TDCIR-Tree 索引中，每个非叶子结点连接关键字倒排文件列表，每个叶子结点连接时间属性标签文件. 根据关键字信息表、查询场景划分图和 o_1，o_2，o_3，o_4，o_5 的时间域表，每个非叶子结点用三元组 $(ra,re,ra.ki)$ 表示. 其中 ra 存储的是子结点地址；re 表示覆盖所有子结点的 MBR 最小矩阵；$ra.ki$ 表示关键字描述标识符，连接关键字倒排文件列表，其中第 1 列为关键字信息，第 2 列为包含对应关键字信息的子结点. 每个叶子结点用三元组 $(o,o.re,o.ti)$ 表示，其中 o 是空间数据对象，$o.re$ 是 o 的

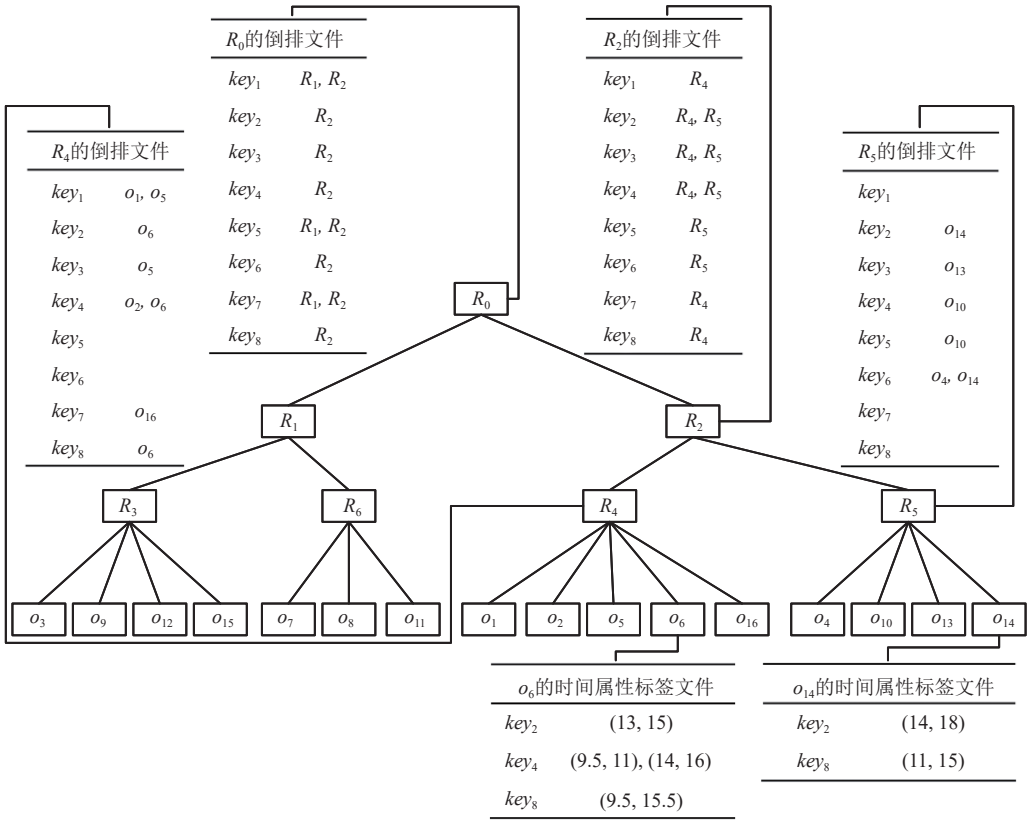


Fig. 2 Illustration example of TDCIR-Tree

图 2 TDCIR-Tree 示意图示例

边界矩阵, *o.ti* 是时间区间描述标识符, 连接时间属性标签文件, 其中第 1 列为 *o* 的关键字信息, 第 2 列为关键字对应的时间区间.

在 TDCIR-Tree 索引中, 以自上而下、自左而右的层序的方式按照查询关键字进行查询, 从第 1 层根结点R_0开始, 在R_0的倒排文件中找到查询关键字对应的子结点. 例如在图 2 中, 查询点q的查询关键字为key_2, 在 TDCIR-Tree 索引中第 1 步在R_0的倒排文件中找到查询关键字key_2对应的子结点为R_2, 接下来查询第 2 层的R_2, 按照相同的方式在R_2找出的子结点为R_4和R_5, 以此类推, 直到找出的子结点为叶子结点为止. 此时需要在叶子结点的时间属性标签文件中找到查询关键字对应的时间区间, 判断是否含有查询关键字对应的时间区间, 也就是找到的叶子结点在查询关键字对应的时间区间与查询关键字对应的时间区间交集不为空, 满足条件的叶子结点作为时空数据对象放入查询关键字对应的初始集合.

3.2 TDCCA_PP 算法的构建与查询

TDCCA_PP 算法分 3 个部分: 基于 TDCIR-Tree 首先输入初始集合G进行第 1 层剪枝, 得到可行集合G'; 其次对可行集合G'进行组间有序排列, 得到组合集合M; 然后对组合集合M进行第 2 层剪枝, 得到有效集合M'.

第 1 层剪枝的主要思想是在G_{n_1}和G_{n_n}中筛选出满足用户查询点距离阈值的时空数据对象, 得到可行集合, 算法 1 输入关键字个数 n、初始集合G_{n_i}、距离阈值$q.b$, 输出的是可行集合G'_{n_j}. 首先将输出的关键字key_{n_i}对应的可行集合G'_{n_j}进行初始化, 最外层循环从全部 n 个初始集合先分离出第一个和最后一个初始集合G_{n_1}和G_{n_n}, 对G_{n_1}和G_{n_n}中的所有时空数据对象进行筛选, 将时空数据对象到查询点距离满足查询点距离阈值$q.b$的时空数据对象放入对应可行集合中, 其余初始集合G_{n_2}到$G_{n_{n-1}}$不变, 直接转入到可行集合G'_{n_2}到$G'_{n_{n-1}}$中. 从算法 1 第 1 层剪枝算法中得到可行集合G'_{n_j}.

算法 1. 第 1 层剪枝算法(*TDCCAPruning*1(n,G_{n_i}, $q.b$)).

输入: 关键字个数 n, 初始集合G_{n_i}, 距离阈值$q.b$;

输出: 可行集合G'_{n_j}.

① $G'_{n_j}=\varnothing$; /*将输出的关键字key_{n_j}对应的可行集合G'_{n_j}进行初始化*/

② for i in *range*(1, n+1) do /*其中函数 *range* 意为

变量 i 从 1 开始到 n+1 结束但不包括 n+1，共循环 n 次，面向关键字 key_{n_i} 对应的初始集合 G_{n_1} 到 G_{n_n}*/

③ if i==1 or i==n then /*第 1 层剪枝算法只面向初始集合 G_{n_1} 和 G_{n_n}*/

④ for each $o_{i_x} \in G_{n_i}$ do /*o_{i_x} 为 G_{n_i} 中的第 x 个时空数据对象*/

⑤ if $dist(q, o_{i_x}) \leqslant q.b$ then /*判断查询点 q 到 o_{i_x} 的距离是否满足查询点的距离阈值 $q.b$*/

⑥ 将 o_{i_x} 放入 G'_{n_j} 中；/*G_{n_i} 中满足条件的 o_{i_x} 放入 G'_{n_j} 中*/

⑦ else

⑧ 处理 G_{n_i} 中剩下的时空数据对象；

⑨ end if

⑩ end for

⑪ else

⑫ $G'_{n_j} = G_{n_i}$；/*剩下的初始集合 G_{n_2} 到 $G_{n_{n-1}}$ 不变，直接转入 G'_{n_2} 到 $G'_{n_{n-1}}$*/

⑬ end if

⑭ end for

⑮ return G'_{n_j}.

为了得到满足用户全部关键字及其时间域的集合，其中存在有同一时空数据对象满足 2 个及以上查询关键字的可能，但在传统集合的定义中不允许有 2 个相同元素出现在同一集合中，所以我们提出了组合集合的概念和一种组间有序排列的方式.

组间有序排列的主要思想是将可行集合进行可重复元素的组合排列，得到组合集合，算法 2 输入关键字个数 n、组合集合数 f、可行集合 G'_{n_i}，输出组合集合 M_j. 首先将输出的组合集合 M_j 进行初始化，依次将 n 个可行集合中时空数据对象进行组间有序组合，先将 G'_{n_1} 中的第 1 个、G'_{n_2} 中的第 1 个、G'_{n_3} 中的第 1 个，一直到 G'_{n_n} 中的第 1 个时空数据对象按照顺序结合成组合集合 M_1，再将 G'_{n_1} 中的第 1 个、G'_{n_2} 中的第 1 个、G'_{n_3} 中的第 1 个，一直到 G'_{n_n} 中的第 2 个时空数据对象按照顺序结合成组合集合 M_2，以此类推，由于可行集合 G'_{n_i} 中有 m_i 个时空数据对象，我们能够得出有 $f = \prod_{i=1}^{n} m_i$ 个组合集合. 从算法 2 组间有序排列算法中得到组合集合 M_j.

算法 2. 组间有序排列算法（*TDCCAPermutation*(n, f, G'_{n_i})).

输入：关键字个数 n，组合集合数 f，可行集合 G'_{n_i}；

输出：组合集合 M_j.

① $M_j = \varnothing$；/*将输出的组合集合 M_j 进行初始化*/

② for j in *range*(1, f+1) do /*其中函数 *range* 意为变量 j 从 1 开始到 f+1 结束但不包括 f+1，共循环 f 次*/

③ 依次将 n 个可行集合中时空数据对象进行组间有序组合，并放入 M_j 中；

④ end for

⑤ return M_j.

第 2 层剪枝的主要思想是在组合集合中筛选出满足用户查询点距离阈值的有效集合，算法 3 输入关键字个数 n、组合集合数 f、组合集合 M_i、距离阈值 $q.b$，输出有效集合 M'_j. 首先将输出的有效集合 M'_j 进行初始化，最外层循环是面向 f 个组合集合的，其中内层循环是判断每个组合集合中的时空数据对象到查询点距离是否满足查询点的距离阈值 $q.b$，当 M_i 中的全部空间数据对象都满足查询点的距离阈值的情况下，就将 M_i 直接转入有效集合 M'_j 中，不满足条件就直接跳出循环，进行下一个组合集合直到最后一个组合集合. 从算法 3 第 2 层剪枝算法中得到有效集合 M'_j.

算法 3. 第 2 层剪枝算法（*TDCCAPruning*2(n, f, M_i, $q.b$)).

输入：关键字个数 n，组合集合数 f，组合集合 M_i，距离阈值 $q.b$；

输出：有效集合 M'_j.

① $M'_j = \varnothing$；/*将输出的有效集合 M'_j 进行初始化*/

② for i in *range*(1, f+1) do /*其中函数 *range* 意为变量 i 从 1 开始到 f+1 结束但不包括 f+1，共循环 f 次，面向组合集合 M_1 到 M_f*/

③ for y in *range*(1, n) do /*其中函数 *range* 意为变量 y 从 1 开始到 n 结束但不包括 n，共循环 n−1 次，o_{i_y} 为 M_i 中的第 y 个时空数据对象*/

④ if $dist(o_{i_y}, o_{i_{y+1}}) > q.b$ then /*判断 o_{i_y} 到 $o_{i_{y+1}}$ 的距离是否满足查询点的距离阈值 $q.b$*/

⑤ break; /*若不满足则直接跳出循环，面向下一个组合集合*/

⑥ else

⑦ 处理 M_i 中剩下的时空数据对象；

⑧ end if

⑨ $M'_j = M_i$；/*当 M_i 中的全部数据对象都满足查询点的距离阈值的情况下，就将 M_i 直接转入 M'_j*/

⑩ end for

⑪ end for

⑫ return M'_j.

3.3 TDC 成本函数的计算和排序

针对现有的成本函数无法满足 TDCCA-CoSKQ 的问题, 提出了一种基于 TDC 成本函数的排序算法. 基于定义 7 中提及到的 TDC 成本函数, TDC 成本函数由距离成本和时间成本组成, α和 β是自变量系数也称为用户偏好度, 可由用户自行调整, 并且规定 $\alpha+\beta=1$.

将有效集合M'_i放入基于 TDC 成本函数的排序算法, 计算并得到以成本为升序的有序集合V_j, 返回用户结果集, 具体过程如算法 4 所示.

算法 4. 基于 TDC 成本函数的排序算法 (*TDCCostSorting*($n, f, M'_i, q.b$)).

输入: 有效集合M'_i, 有序集合数 h, TDC 成本函数;

输出: 有序集合V_j.

① $V_j=\varnothing$; /*将输出的有序集合V_j进行初始化*/

② 将有效集合M'_i代入 TDC 成本函数中进行计算, 得出对应的$cost(M'_i)$;

③ 将 h 个$cost(M'_i)$进行升序排列; /*其中第 j 个$cost(M'_i)$对应M'_i用$M'_i(j)$表示*/

④ for j in *range*(1, h+1) do /*其中函数 *range* 意为变量 j 从 1 开始到 h+1 结束但不包括 h+1, 共循环 h 次*/

⑤ $V_j=M'_i(j)$; /*按照升序位置, 将$cost(M'_i)$对应的有效集合M'_i放入有序集合V_j中*/

⑥ end for

⑦ return V_j.

本文所提出的 TDCCA-CoSKQ 方法首先利用算法 1 进行第 1 层剪枝得到可行集合G', 再用算法 2 进行组间有序排列得到组合集合M_i, 接着利用算法 3 进行第 2 层剪枝得到有效集合M'_j, 最后利用算法 4 得到最终的查询结果有序集合V_j而返回用户结果集.

3.4 实例分析

假设查询场景划分图如图 3 所示, TDCIR-Tree 索引图如图 4 所示. 图 3 中关键字信息为key_1代表中式饭馆、key_2代表西式餐厅、key_3代表购物商城、key_4代表书店、key_5代表花店、key_6代表水果店、key_7代表博物馆, q的距离阈值$q.b$为 1 000, 查询点q的关键字为key_1, key_3, key_6, 其对应的时间域分别为 11:00—12:30, 12:30—15:00, 15:00—15:30. 图 3 中有 5 个时空数据对象o_1, o_2, o_3, o_4, o_5, 其关键字信息分别为 {key_1,key_2,key_5}, {key_4,key_5}, {key_1,key_2,key_3}, {key_6, key_7}, {key_1,key_3,key_6}, 时空数据对象到q的欧氏距离分别是 430, 950, 780, 650, 975, o_1代表的关键字为key_1, key_2, key_5, 分别对应的时间域为 10:00—13:00 和 17:30—20:30, 12:00—17:00, 8:00—13:30, o_2代表的关键字为key_4, key_5, 分别对应的时间域为 10:00—15:00, 9:00—15:30, o_3代表的关键字为key_1, key_2, key_3, 分别对应的时间域为 10:00—15:00 和 18:00—20:00, 13:30—19:00, 9:30—19:30, o_4代表的关键字为key_6, key_7, 分别对应的时间域为 8:00—17:30, 10:00—16:00, o_5代表的关键字为key_1, key_3, key_6, 分别对应的时间域为 9:00—13:30 和 17:30—21:00, 10:00—19:30, 10:00—16:00, 时空数据对象之间的距离信息为$dist(o_1,o_2)$=780, $dist(o_1, o_3)$=1 000, $dist(o_1, o_4)$=800, $dist(o_1, o_5)$=1 330, $dist(o_2, o_3)$=1 725, $dist(o_2, o_4)$=580, $dist(o_2, o_5)$=1 600, $dist(o_3, o_4)$=1 530, $dist(o_3, o_5)$=800, $dist(o_4, o_5)$=1 120.

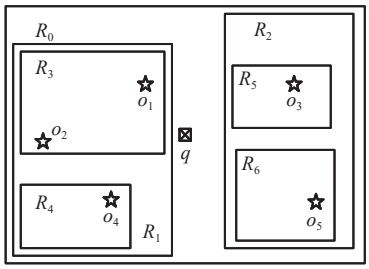


Fig. 3 Scene partition illustration of the case

图 3 实例场景划分示意图

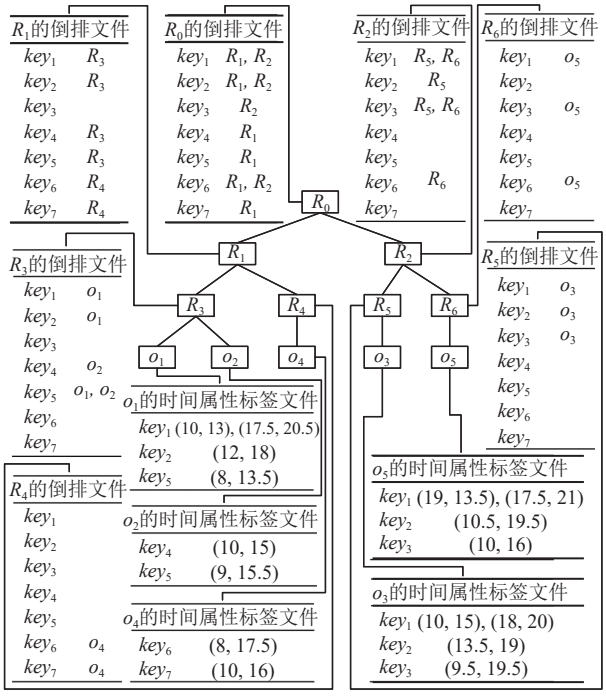


Fig. 4 TDCIR-Tree of the case

图 4 实例 TDCIR-Tree

基于所提的 TDCCA-CoSKQ 方法，首先在 TDCIR-Tree 索引中查询到满足查询关键字及其对应时间域的时空数据对象，得到关键字 key_{n_i} 对应的初始集合 G_{n_i}. 在 TDCIR-Tree 索引中，以自上而下、自左而右层序的方式按照 key_1, key_3, key_6 这 3 个查询关键字的主题查询. 首先查询关键字 key_1，从第 1 层根结点 R_0 开始，在 R_0 的倒排文件中找到关键字 key_1 对应的子结点为 R_1 和 R_2；那么接下来依次查询第 2 层的 R_1 和 R_2，在 R_1 的倒排文件中找到关键字 key_1 对应的子结点为 R_3，按照相同的方式在 R_2 找出的子结点为 R_5 和 R_6，依次查询第 3 层的 R_3, R_5, R_6，在 R_3 的倒排文件中找到关键字 key_1 对应的叶子结点为 o_1，按照相同的方式依次在 R_5 和 R_6 找出的叶子结点为 o_3 和 o_5；最后从第 4 层叶子结点层依次对 o_1, o_3, o_5 进行查询. 在 o_1 的时间属性标签文件中找到关键字 key_1 对应的时间区间为 (10, 13), (17.5, 20.5)，而查询点 q 的查询关键字 key_1 对应的时间区间为 (11, 12.5)，由于 $o_1.key_1.t \supset q.key_{n_1}.t$，可以得出 o_1 是满足查询关键字 key_1 及其对应时间区间的时空数据对象，将 o_1 放入关键字 key_1 对应的初始集合 G_{n_1} 中，o_3 和 o_5 依次按照相同的方式进行查询和比较，得出 o_3 和 o_5 是满足查询关键字 key_1 及其对应时间区间的时空数据对象，将 o_3 和 o_5 放入关键字 key_1 对应的初始集合 G_{n_1} 中，最后得到 $G_{n_1}=\{o_1,o_3,o_5\}$. 按照相同的方式依次查询关键字 key_3 和 key_6，得到 $G_{n_2}=\{o_3,o_5\}$, $G_{n_3}=\{o_5\}$.

由于从查询点到第 1 个和最后 1 个时空数据对象的欧氏距离 $dist(q,o_{n_1})$ 和 $dist(q,o_{n_n})$ 都应不大于查询点的距离阈值 $q.b$，所以初始集合 G_{n_1} 和 G_{n_3} 中的时空数据对象 o_i 应满足 $dist(q,o_{n_i})\leqslant 1\,000$. 为得到可行集合 G'_{n_i}，我们结合时空数据对象信息表中的时空信息数据 $dist(q,o_{n_i})$ 来进行剪枝. $G_{n_1}=\{o_1,o_3,o_5\}$, $dist(q,o_{n_1})=430<1\,000$，故 o_1 满足查询点的距离阈值 $q.b$. 将 o_1 放入初始集合 G_{n_1} 对应的可行集合 G'_{n_1} 中，按照相同的方式依次对 o_3 和 o_5 进行剪枝，得出 o_3 和 o_5 也同样满足查询点的距离阈值 $q.b$. 将 o_3 和 o_5 放入初始集合 G_{n_1} 对应的可行集合 G'_{n_1} 中，得到可行集合 $G'_{n_1}=\{o_1,o_3,o_5\}$. 按照相同的方式对 G_{n_3} 进行剪枝，得到可行集合 $G'_{n_3}=\{o_5\}$. 剩余的初始集合 G_{n_2} 中的时空数据对象被直接放入可行集合 G'_{n_2} 中，得到可行集合 $G'_{n_2}=\{o_3,o_5\}$.

在可行集合 G'_{n_i} 中有 m_i 个时空数据对象，而组间有序排列就是将从 G'_{n_1} 到 G'_{n_n} 这 n 个可行集合之间进行可重复的排列组合，得到 $f=\prod_{i=1}^{n} m_i$ 个组合集合. 在本实例中可行集合 $G'_{n_1}=\{o_1,o_3,o_5\}$, $G'_{n_2}=\{o_3,o_5\}$, $G'_{n_3}=\{o_5\}$，将可行集合 G'_{n_i} 进行组间有序排列，就可以得到 6 个组合集合 M_i，分别为 $M_1=\{o_1,o_3,o_5\}$, $M_2=\{o_1,o_5,o_5\}$, $M_3=\{o_3,o_3,o_5\}$, $M_4=\{o_3,o_5,o_5\}$, $M_5=\{o_5,o_3,o_5\}$, $M_6=\{o_5,o_5,o_5\}$.

在该实例中，由于相邻 2 个时空数据对象之间的欧氏距离 $dist(o_{n_i},o_{n_{i+1}})$ 应不大于查询点的距离阈值 $q.b$，所以组合集合 M_i 中相邻的空间数据对象 o_{n_i} 和 $o_{n_{i+1}}$ 之间的欧氏距离应满足 $dist(o_{n_i},o_{n_{i+1}})\leqslant 1\,000$，我们结合时空数据对象之间距离表中数据 $dist(o_i,o_j)$ 来进行剪枝. $M_1=\{o_1,o_3,o_5\}$, $dist(o_1,o_3)=1\,000$, $dist(o_3,o_5)=800$，故组合集合 M_1 中的时空数据对象满足查询点的距离阈值 $q.b$. 将 M_1 中的时空数据对象按顺序放入有效集合 M'_1 中，得到有效集合 $M'_1=\{o_1,o_3,o_5\}$，按照相同的方式依次对 M_2, M_3, M_4, M_5, M_6 进行剪枝，得到组合集合 M_2 不满足查询点的距离阈值 $q.b$，故 M_2 被剪枝，组合集合 M_3, M_4, M_5, M_6 满足查询点的距离阈值 $q.b$，故有效集合 $M'_3=\{o_3,o_3,o_5\}$, $M'_4=\{o_3,o_5,o_5\}$, $M'_5=\{o_5,o_3,o_5\}$, $M'_6=\{o_5,o_5,o_5\}$. 最后将有效集合放入基于 TDC 成本函数的排序算法中，计算有效集合 M'_i 的成本，得到以成本为升序的有序集合 V_i. 将有效集合 M'_1, M'_3, M'_4, M'_5, M'_6 依次放入所提出的 TDC 成本函数中计算成本，得到 $cost(M'_1)=\alpha\times 3\,205$, $cost(M'_3)=\alpha\times 2\,555$, $cost(M'_4)=\alpha\times 2\,555$, $cost(M'_5)=\alpha\times 3\,550$, $cost(M'_6)=\alpha\times 1\,950$.

由于有效集合 M'_i 成本函数中的时间成本都为 0，所以无论参数 α 和 β 设为何值，排列后顺序都不变，所以我们设定 $\alpha=1$, $\beta=0$，以成本为主要关键字进行升序排序，得到有序集合为 $V_1=\{o_5,o_5,o_5\}$, $V_2=\{o_3,o_3,o_5\}$, $V_3=\{o_3,o_5,o_5\}$, $M'_4=\{o_1,o_3,o_5\}$, $V_5=\{o_5,o_3,o_5\}$，其中 V_1 为最优结果集，表示仅 o_5 就能够满足用户的查询需求，其他有序集合可作为用户参考，为用户提供相较合适的其他出行选择，有序集合 V 为返回用户结果集.

4 实验分析

为了评估方法性能，本节设置了 3 个方面的对比实验，第 1 方面对比不同数据集对不同方法效率的影响，第 2 方面对比不同查询关键字数量对不同方法效率的影响，第 3 方面对比不同方法在不同数据集上执行结果的准确率.

实验采用的操作系统：Microsoft Windows10 的 64 位操作系统. 实验中计算机的硬件配置：16 GB RAM, 256 GB ROM, 处理器：Intel® Core™ i7 处理器（主频为 2.90 GHz），使用 Python 语言进行编程.

本文用来对比的方法为成本感知和距离约束的集合空间关键字查询方法 CD-CoSKQ[20], 集合空间关键字查询的高效索引独立方法 DCA[15] 和同时考虑到对象和查询之间的位置相关性、文本相关性和时间相关性的时间感知集合空间关键字查询方法 TCoSKQ[21]. 本节为所提的 TDCCA-CoSKQ 方法与以上 3 种对比方法随机分配了查询关键字信息和对应的查询时间区间信息, 并为已知时间的集合空间关键字查询从数据集中随机读取了查询位置. 实验采用真实数据集并且进行小幅度数据处理使其更符合实验要求. 实验使用了 3 个真实的数据集 Yelp①, San Francisco②, Hotel③, 使用 zipf[23] 分布为每个数据对象分配 1~5 个关键字, 并且随机生成 [0, 24) 的时间信息, 每个数据对象包含位置信息、文字描述及其对应的时间信息. 我们将准确度定义为方法中最优结果集与方法返回集平均 TDC 成本函数比值的相反数与 1 的和, 返回结果集的 TDC 成本函数的值越小表示方法执行结果的准确度越高. 在基于时空约束和成本感知的集合空间关键字查询中, 实验结果分别是随机生成的 50 组和 100 组查询结果取得的平均值. 表 1 为实验中所用数据集的信息表, 表 2 为实验参数的设置.

Table 1 Experiment Datasets

表 1 实验数据集

数据集	数据对象数量	关键词数量
Yelp	192 609	2 468
San Francisco	174 956	207
Hotel	20 790	602

Table 2 Parameters Setting

表 2 参数设置

参数	值	默认值
查询关键字数量	1, 2, 3, 4, 5	3
α	0.1, 0.3, 0.5, 0.7, 0.9	0.5
β	0.9, 0.7, 0.5, 0.3, 0.1	0.5

实验 1. 该部分实验旨在不同数据集下对比 TDCCA-CoSKQ 方法与 CD-CoSKQ 方法、DCA 方法和 TCoSKQ 方法的效率. 随着数据量的增加, 4 种方法的 CPU 执行时间变化如图 5~7 所示.

图 5、图 6、图 7 分别表示数据集 Yelp, San Francisco, Hotel 对不同方法效率的影响. 从图 5~7 可以看出, 在数据集 Yelp, San Francisco, Hotel 下, 随着数据量的增加, 本文所提的 TDCCA-CoSKQ 方法的 CPU

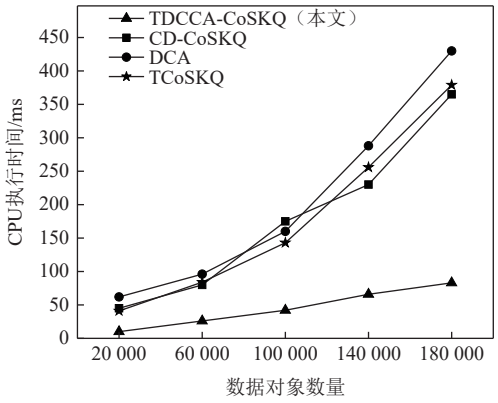


Fig. 5 Comparison of different methods on the efficiency under dataset Yelp

图 5 在数据集 Yelp 下不同方法的效率对比

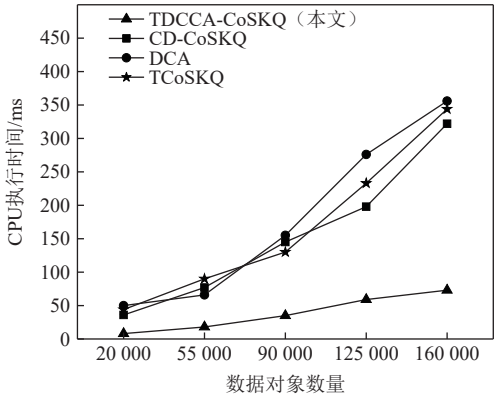


Fig. 6 Comparison of different methods on the efficiency under dataset San Francisco

图 6 在数据集 San Francisco 下不同方法的效率对比

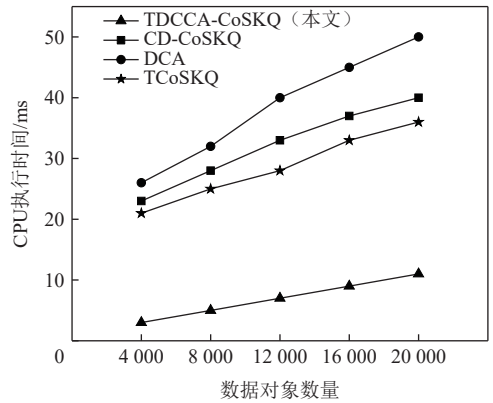


Fig. 7 Comparison of different methods on the efficiency under dataset Hotel

图 7 在数据集 Hotel 下不同方法的效率对比

① https://www.yelp.com/dataset/challenge

② https://www.cs.fsu.edu/lifeifei/SpatialDataset.htm

③ https://www.allstays.com

执行时间并不会陡增，对于数据集的可扩展性相对于其他对比方法表现较好，然而对比方法 CD-CoSKQ, DCA, TCoSKQ, CPU 执行时间上升较快，并且 TDCCA-CoSKQ 方法的 CPU 执行时间远小于其他对比方法.

实验 2. 该部分实验旨在对比查询关键字数量不同对不同方法效率的影响，具体地针对每一个数据集，在数据集 Yelp, San Francisco, Hotel 下，从数据对象的所有关键字信息中随机生成一定数量的关键字作为查询关键字，其数量变化区间为 1~5. 本文提出的 TDCCA-CoSKQ 方法与对比方法 CD-CoSKQ, DCA, TCoSKQ 的 CPU 执行时间变化如图 8~10 所示.

图 8~10 分别表示在数据集 Yelp, San Francisco, Hotel 下不同关键字数量对不同方法效率的影响. 可以看出，TDCCA-CoSKQ 方法随着查询关键字数量的增加，CPU 执行时间上升较缓，在数据集 Yelp, San Francisco, Hotel 下变化相较于对比方法并不明显，并且 CPU 执行时间小于其他对比方法，效率较高.

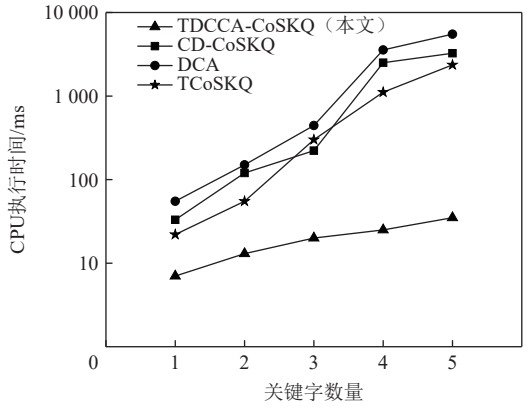


Fig. 8 Impact of different keyword numbers on the efficiency under dataset Yelp

图 8 在数据集 Yelp 下不同关键字数量对效率的影响

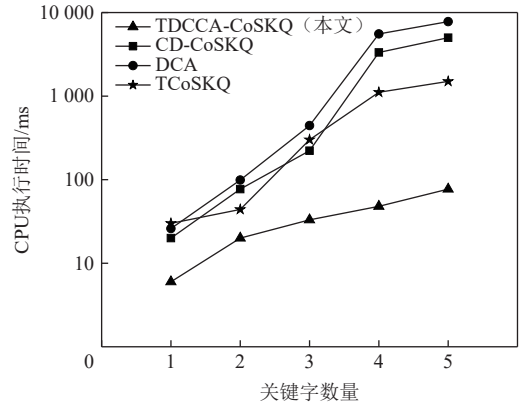


Fig. 9 Impact of different keyword numbers on the efficiency under dataset San Francisco

图 9 在数据集 San Francisco 下不同关键字数量对效率的影响

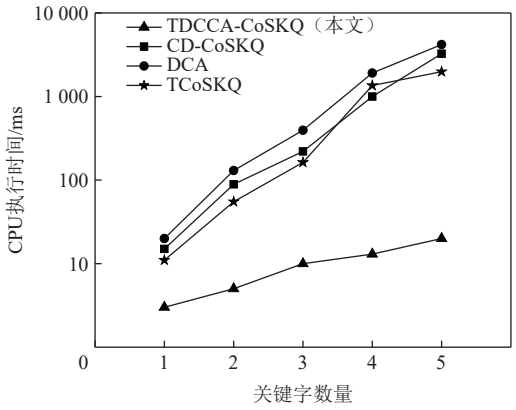


Fig. 10 Impact of different keyword numbers on the efficiency under dataset Hotel

图 10 在数据集 Hotel 下不同关键字数量对效率的影响

实验 3. 该部分实验旨在对比不同方法在不同数据集下执行结果的准确率，具体地针对每个数据集在应用方法时控制查询关键字个数及其他指标相同. 本文提出的 TDCCA-CoSKQ 方法和对比方法 CD-CoSKQ, DCA, TCoSKQ 执行准确率情况如图 11 所示.

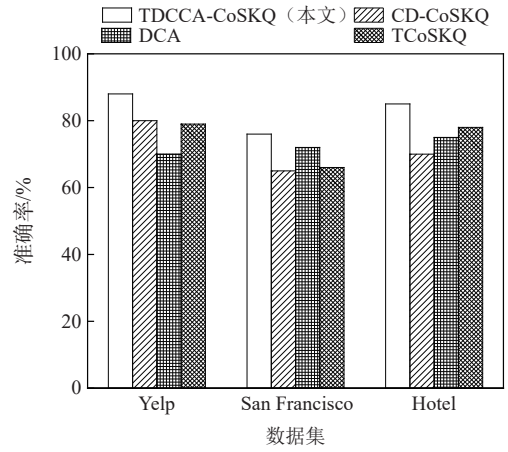


Fig. 11 Impact of different datasets on the accuracy

图 11 不同数据集对准确率的影响

从图 11 可以看出，TDCCA-CoSKQ 方法的准确率在不同数据集 Yelp, San Francisco, Hotel 下相较于其他对比方法 CD-CoSKQ, DCA, TCoSKQ 来说准确率较高. 本文所提的 TDCCA-CoSKQ 方法在数据集 Yelp 的准确率在 3 个数据集中相对较高，而对比方法 CD-CoSKQ, DCA, TCoSKQ 分别在数据集 Yelp, Hotel, Yelp 的准确率相对较高.

5 总　　结

针对现有的集合空间关键字查询研究没有考虑要求带有时空约束和成本感知的问题，本文提出一

种基于时空约束和成本感知的集合空间关键字查询TDCCA-CoSKQ.利用提出的TDCIR-Tree索引提高了对时空数据对象进行关键字属性和时间属性上查询的效率,解决了现有索引中无法同时包含关键字信息和时间信息的问题,利用TDCCA_PP算法可以将大部分不满足约束条件的集合高效过滤掉,利用TDC成本函数及其对应排序算法对有效集合进行计算并排序,从而得到以成本为升序的有序集合返回用户结果集.理论研究与实验结果表明,本文所提出的TDCCA-CoSKQ能有效解决基于时空约束和成本感知的集合空间关键字查询问题,返回的结果集可以满足用户多元需求.未来研究工作主要集中在隐私保护下的基于时空约束和成本感知的集合空间关键字查询方面.

作者贡献声明:李松提出了方法思路和技术方案,撰写部分论文;曹文琪和郝晓红负责优化方法技术、完成部分实验并撰写论文;张丽平完成部分实验;郝忠孝提出指导意见并修改论文关键部分.

参 考 文 献

[1] Chen Gang, Zhao Jingwen, Gao Yunjun, et al. Time-aware Boolean spatial keyword queries[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(11): 2601−2614

[2] Polychronis V, Michael V, Antonio C, et al. GPU-based algorithms for processing the K nearest-neighbor query on spatial data using partitioning and concurrent kernel execution[J]. International Journal of Parallel Programming, 2023, 51(6): 275−308

[3] Zhong Ying, Li Jianmin, Zhu Shunzhi. Continuous spatial keyword search with query result diversifications[J]. World Wide Web, 2023, 26(4): 1−14

[4] Pan Xiao, Yu Qidi, Ma Ang, et al. Efficient algorithm of Top-*k* spatial keyword search with OR semantics[J]. Journal of Software, 2020, 31(10): 3197−3215 (in Chinese)
(潘晓, 于启迪, 马昂, 等. 支持OR语义的高效受限Top-*k*空间关键字查询技术[J]. 软件学报, 2020, 31(10): 3197−3215)

[5] Liu Junling, Liu Baihe, Zou Xinyuan, et al. Spatial region of interests oriented route query[J]. Journal of Computer Research and Development, 2022, 59(11): 2569−2580 (in Chinese)
(刘俊岭, 刘柏何, 邹鑫源, 等. 面向空间兴趣区域的路线查询[J]. 计算机研究与发展, 2022, 59(11): 2569−2580)

[6] Zhu Huaijie, Liu Wei, Yin Jian, et al. Towards keyword-based geo-social group query services[J]. IEEE Transactions on Services Computing, 2023, 16(1): 670−683

[7] Chang Xueqin, Luo Chengyang, Yu Hanlin, et al. Answering non-answer questions on reverse Top-*k* geo-social keyword queries[J]. Journal of Computer Science and Technology, 2022, 37(6): 1320−1336

[8] Jia Lianyin, Tang Haotian, Zhao Bingxin, et al. An efficient association rule mining-based spatial keyword index[J]. International Journal of Data Warehousing and Mining, 2023, 19(2): 1−19

[9] Tong Qiuyun, Miao Yinbin, Li Hongwei, et al. Privacy-preserving ranked spatial keyword query in mobile cloud-assisted fog computing[J]. IEEE Transactions on Mobile Computing, 2023, 22(6): 3604−3618

[10] Zhang Liping, Li Jing, Li Song. Research on time-aware group query method with exclusion keywords[J]. ISPRS International Journal of Geo-Information, 2023, 12(10): 1−20

[11] Zhang Liping, Li Jing, Li Song. Research on approximate spatial keyword group queries based on differential privacy and exclusion preferences in road networks[J]. ISPRS International Journal of Geo-Information, 2023, 12(12): 480−503

[12] Cao Xin, Cong Gao, Christian S, et al. Collective spatial keyword querying[C]//Proc of the 11th ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2011: 373−384

[13] He Peijun, Xu Hao, Zhao Xiang, et al. Scalable collective spatial keyword query[C]//Proc of the 13th IEEE Int Conf on Data Engineering Workshops. New York: IEEE, 2015: 182−189

[14] Su Danni, Zhou Xu, Yang Zhibang, et al. Top-*k* collective spatial keyword queries[J]. IEEE Access, 2019, 7: 180779−180792

[15] Yang Zhibang, Zeng Yifu, Du Jiayi, et al. Efficient index-independent approaches for the collective spatial keyword queries[J]. Neurocomputing, 2021, 439: 96−105

[16] Chan H, Long Cheng, Wong R. Inherent-cost aware collective spatial keyword queries[C]//Proc of the 15th Int Symp on Spatial and Temporal Databases. Berlin: Springer, 2017: 357−375

[17] Zhang Pengfei, Lin Huaizhong, Yao Bin, et al. Level-aware collective spatial keyword queries[J]. Information Sciences, 2017, 378: 194−214

[18] Chan H, Long Cheng, Wong R. On generalizing collective spatial keyword queries[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1712−1726

[19] Jin Xiongnan, Shin S, Jo E, et al. Collective keyword query on a spatial knowledge base[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(11): 2051−2062

[20] Chan H, Liu Shengxin, Long Cheng, et al. Cost-aware and distance-constrained collective spatial keyword query[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1324−1336

[21] Chen Zijun, Zhao Tingting, Liu Wenyuan. Time-aware collective spatial keyword query[J]. Computer Science and Information Systems, 2021, 18(3): 1077−1100

[22] Feng Zhe, Jin Changlong, Kim H, et al. Time-aware approximate collective keyword search in traffic networks[J]. Knowledge-Based Systems, 2021, 229: 107367

[23] Wu D, Yiu M Y, Cong G, et al. Joint top-*k* spatial keyword query processing[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(10): 1889−1903



Li Song, born in 1977. PhD, professor. Member of CCF. His main research interests include spatial and spatio-temporal database, data mining, and information data security.

李　松, 1977 年生. 博士, 教授. CCF 会员. 主要研究方向为时空数据库、数据挖掘、数据信息安全.



Cao Wenqi, born in 1999. Master. Her main research interests include spatial database and data mining.

曹文琪, 1999 年生. 硕士. 主要研究方向为空间数据库、数据挖掘.



Hao Xiaohong, born in 1969. Master, senior experimentalist master. Her main research interest includes spatial database.

郝晓红, 1969 年生. 硕士, 高级实验师. 主要研究方向为空间数据库.



Zhang Liping, born in 1976. Master, associate professor. Her main research interests include spatial and spatio-temporal database, and information data security.

张丽平, 1976 年生. 硕士, 副教授. 主要研究方向为时空数据库、数据信息安全.



Hao Zhongxiao, born in 1940. PhD, professor. His main research interests include relational database, null database, acyclic database, active database, and spatial database.

郝忠孝, 1940 年生. 博士, 教授. 主要研究方向为关系数据库、空值数据库、非循环数据库、主动数据库、空间数据库.