面向推荐系统的多目标决策优化算法

李松[1]，王冠群[1]，郝晓红[1]，郝忠孝[1,2]
（1. 哈尔滨理工大学计算机科学与技术学院，150080，哈尔滨；
2. 哈尔滨工业大学计算机科学与技术学院，150001，哈尔滨）

摘要：针对推荐系统利用多目标决策技术进行位置信息的查询与推荐时，由于查询者位置的移动和空间障碍物的位置变化导致传统多目标决策技术的查询效率较低的问题，提出了一种基于范围的障碍空间连续 Skyline 查询算法。首先，根据静态 Skyline 点的特征对由空间数据对象信息组成的初始数据集进行约减；然后，根据障碍空间中查询者的位置移动的特点构建距离相交模型，利用距离相交模型和数据对象的属性提出了剪枝策略，再根据剪枝策略过滤掉当查询者的位置移动时对查询结果无影响的数据对象，从而精减了冗余数据，得到过滤后的候选数据集；最后，根据数据对象的非空间属性和相互间的支配关系特征得出影响候选数据集的事件，利用影响候选数据集的事件再对候选数据集进行精炼计算，从而减少了冗余计算，查询出当前时刻的结果集。理论研究与实验结果表明：所提算法在查询者位置移动和空间障碍物位置变化时，能提升多目标决策技术的查询效率；相对传统对比算法，在数据集规模、障碍物数量、查询范围增大时，所提查询算法的平均效率提升约 13%；针对多维度数据信息的查询，所提查询算法的平均效率提高了约 11%。

关键词：推荐系统；多目标决策；位置服务；移动查询；空间 Skyline 查询
中图分类号：TP311 **文献标志码**：A
DOI：10.7652/xjtuxb202208011 **文章编号**：0253-987X(2022)08-0104-09

OSID 码

A Multi-Objective Decision Optimization Algorithm for Recommendation System

LI Song[1], WANG Guanqun[1], HAO Xiaohong[1], HAO Zhongxiao[1,2]
(1. School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China;
2. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: This paper proposes a continuous range Skyline query algorithm in obstacle space to solve the problem of low query efficiency with traditional multi-objective decision technology caused by the movement of searcher's position and the change of spatial obstacles in the recommendation system. Firstly, the initial data set composed of object information in the spatial data is reduced according to the characteristics of the static skyline points; then, the distance intersection model is constructed according to the characteristics of the position movement of the querier in the obstacle space, and the pruning strategy is proposed by using the distance intersection model and the attributes of the data objects. According to the pruning strategy, the data objects that have no impact on the query results when the position of the querier moves are filtered out, so as to reduce the redundant data and obtain the filtered candidate data set; finally,

收稿日期：2022-01-16。 作者简介：李松(1977—)，男，博士，教授。 基金项目：国家重点研发计划资助项目(2020YFB1710200)；国家自然科学基金资助项目(61872105)；黑龙江省留学归国人员科学基金资助项目(LC2018030)。
网络出版时间：2022-05-05 网络出版地址：http://kns.cnki.net/kcms/detail/61.1069.T.20220505.0943.002.html

according to the non-spatial attributes of the data object and the characteristics of the dominant relationship between them, the events affecting the candidate data set are obtained and used to refine the candidate data set, so as to reduce redundant calculation and query the result set at the current time. Theoretical research and experiments show that the proposed algorithm can improve the query efficiency of multi-objective decision technology when the searcher moves and the position of spatial obstacles changes. Compared with the traditional comparison algorithm, the average efficiency of the query algorithm is improved by about 13% when the size of the data set, the number of obstacles and the query range are increased; for the query of multi-dimensional data information, the average efficiency of the proposed query algorithm is improved by about 11%.

Keywords: recommendation system; multi-objective decision; location services; mobile query; spatial skyline query

如今，海量的数据信息因其复杂的特性极大增加了获取真正有价值的信息难度。为了解决信息过载问题，推荐系统应运而生。随着应用需求日益复杂多样，多目标决策技术在推荐系统中的作用越来越大。多目标决策方法不仅能够基于多个相互矛盾乃至冲突的度量指标进行方案评估，而且还可以很好反映决策者对评价指标的主观偏好。多目标决策方法在数据挖掘、复杂查询、推荐系统、基于位置的服务等领域具有很大的价值。作为多目标决策的一个重要技术，Skyline 查询最早由文献[1]提出，是从多种维度属性的元组集合中返回最具有优势的元组。例如，在就餐时间，推荐系统利用 Skyline 查询方法可向游客智能推荐距离尽可能近，价格尽可能低廉，环境尽可能好，质量尽可能高，更贴切用户口味的餐馆进行就餐。利用 Skyline 查询技术可为用户推荐出与用户当前位置和搜索关键词相关的兴趣点(point of interest，POI)，且每个兴趣点都具有文本描述信息及位置坐标属性。

近些年，国内外对 Skyline 查询技术进行了一些重要研究。文献[2]基于位图方法将每个数据点编码在位图中，提出了一种基于位运算的改进方法。文献[3]提出空间 Skyline 查询的概念和方法，利用 R-树和 Voronoi 图的特性进行剪枝，根据静态点提出了两个算法。文献[4]提出了针对大数据的动态 Skyline 查询方法，有效地解决了海量数据上动态查询的问题。文献[5]给出了度量空间中 TOP-k 反向 Skyline 查询算法。为了解决 Skyline 查询隐私保护的问题，文献[6]将索引和公钥加密技术相结合，实现了高效的数据隐私保护。

在现实生活中，查询者的位置是时刻变化的，连续 Skyline 查询也得到了更多的重视。文献[3]根据动态查询点提出的算法充分利用查询的变化模式，避免对 Skyline 查询的冗余计算，从而提高执行效率。文献[7]提出欧氏空间中的连续 Skyline 查询方法，该方法首先计算初始 Skyline 集，接着计算出可能使结果集发生变化的事件并存储到队列中，再根据每个事件更新结果集，但是在速度和方向上存在不合理的约束。考虑到查询点移动具有随意性，文献[8]研究了以增量运动模型对查询点进行建模，接着依据该模型过滤数据集，最后提出事件处理机制更新结果集。文献[9-10]基于安全区域提出了 Skyline 查询的方法。文献[11]考虑到查询点位置的不确定性，提出连续概率 Skyline 查询方法，首先定义查询点移动对象间的支配概率，接着提出微元概率计算方法，最后更新结果集。文献[12]提出了 Brute-forth 算法和 Delta-scanning 算法，引入了有效范围的概念用于 Skyline 查询，当下一个查询点仍在有效范围内时避免了重新计算。文献[13]提出了并行连续 Skyline 查询算法，该算法主要利用曼哈顿距离对数据点集进行排序和剪枝。文献[14]采用网格数据结构索引数据集，该网格分为大小相等的子网格，当影响区域内的数据点发生变化时会影响 Skyline 集的变化，而自由区域内数据点的变化将不会产生影响。

近年来，道路网 Skyline 查询越来越受到人们的关注。文献[15-16]解决了多成本运输网络的 Skyline 查询问题，道路网络的每条边都考虑了多种偏好，例如距离、驾驶时间、交通灯数量和油耗。文献[17]提出一种基于将道路边缘划分为子区间的基本算法，并通过检查子区间的端点来找到最佳位置。文献[18]提出了带有范围限制的连续 Skyline 查询和 k-NN 连续 Skyline 查询这两种改进的算法。文

献[19]通过对城市道路网进行多尺度区域划分,找到优化的查询规模和区域,其首先基于 Voronoi 图建立道路网络中每个交叉节点的主导区域,然后将兴趣点划分为每个相交节点的主导区域。文献[20]提出了基于关键字的 top-k Skyline 查询方法,该方法通过检索和查询相关的数据对象来有效减少搜索空间。文献[21]研究了基于网络 Voronoi 图的道路网环境下 K-支配空间 Skyline 查询方法,该方法通过网络 Voronoi 图的性质与查询区域的位置关系对数据集约减,然后对候选集的非空间属性进行 K-支配,检查得到道路网精炼集合,最后对精炼集合进行支配检查得到最终的空间 Skyline 集合。文献[22]采用网格索引来处理道路网连续 Skyline 查询。文献[23]研究了道路网连续 Skyline 查询方法,该方法首先为每个数据集构建道路网安全区域,当查询点位于安全区域内时,该数据集中的数据点为 Skyline 点,但是在道路网空间中提前构造安全区域将会增大计算代价。

现实中,查询者在位置移动的时候往往会受到地理条件的限制(例如湖泊、河流和建筑物等),因此距离的计算经常需要考虑障碍物的影响。文献[24]研究障碍空间中反 k 最近邻查询算法,根据 Voronoi 图和障碍距离的特性减少了数据点的处理个数。文献[25]基于 R-树提出了剪枝算法,有效优化障碍空间中反向最近邻查询。文献[26]提出了障碍环境中基于 Voronoi 图的查询算法,该算法利用 Voronoi 图进行约剪数据集,最终得到 Skyline 集合。

已有的障碍空间中范围 Skyline 查询方法只能处理查询点是静止的情况,无法有效解决障碍空间中连续范围 Skyline 查询问题,为了弥补已有方法的不足,本文对基于范围的障碍空间连续 Skyline 查询方法进行了系统研究。本文定义了移动环境下查询点和数据点间的距离,提出了一种基于范围的障碍空间连续 Skyline 查询算法。该算法首先计算查询点静止时的候选结果集,然后通过约剪数据集和事件剪枝策略有效避免了当查询点移动的时候所导致的大量重复计算。

1 基本定义

定义 1 非空间支配[1] 给定任意两个数据点 p_1、p_2,在所有的非空间属性中 p_1 都不比 p_2 差,称 p_1 非空间支配 p_2,记为 $p_1 \prec p_2$。

本文中,将非空间支配 p 的数据点的集合表示为 $T_{\text{sta}}(p)$。

定义 2 障碍空间支配 在障碍空间中,给定一个查询点 q 和任意两个数据点 p_1、p_2,$d_{\text{obs}}(q,p_1)$ 表示数据点 p_1、查询点 q 的距离。当满足以下两个条件:①$p_1 \prec p_2$;②$d_{\text{obs}}(q,p_1) < d_{\text{obs}}(q,p_2)$时,称 p_1 障碍空间支配 p_2,记为 $p_1 \rhd p_2$。

本文中,将障碍空间支配 p 的数据点的集合表示为 $T_{\text{all}}(p)$。

定义 3 基于点的障碍空间 Skyline 查询 在障碍空间中,给定一个查询点 q、数据点集 P,基于点的障碍空间 Skyline 查询返回数据点集 P 的一个子集,其中的每个数据点都不能被 P 中的任何其他点所障碍空间支配,记为 $\text{POS}(q,P)$。

定义 4 基于范围的障碍空间连续 Skyline 查询 在障碍空间中,给定一个范围 R、数据点集 P,当查询点 q 在范围 R 内移动时,基于范围的障碍空间连续 Skyline 查询返回范围 R 内的 Skyline 结果集,记为 $\text{CROS}(R,P)$。

数据点集的非空间属性如表 1 所示,障碍空间中数据点的分布及连续查询示例如图 1 所示,在障碍空间中有范围 R_1、R_2,宾馆集合为$\{p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8\}$,障碍线段集合为$\{l_1,l_2,l_3\}$。查询者由位置 q_1 移动到位置 q_2,其查询范围也在不断地变化,假设查询者想查找既价格便宜又服务质量高的宾馆。若只考虑宾馆的非空间属性,因为 p_4、p_5 和 p_8 不被任何其他数据点非空间支配,所以 p_4、p_5 和 p_8 为 Skyline 点。表 1 中数据点 p_5 非空间支配 p_6,数据点 p_4、p_8 分别支配剩下的数据点。若查询者想查找范围内价格既低、服务质量又高的宾馆,当查询者位于 q_1 的位置时,在查询范围内的数据点只有 p_4、p_5 和 p_6,因为数据点 p_5 距离查询点 q_1 比 p_6 更近,所以此时的结果集为$\{p_4,p_5\}$。当查询者移动到 q_2 位置时,范围内的数据点有 p_2、p_3、p_5 和 p_6,因为

表 1 数据点集的非空间属性

Table 1 Non spatial attributes of a data point set

宾馆	价格/元	服务级别
p_1	400	12
p_2	350	10
p_3	300	11
p_4	150	2
p_5	190	7
p_6	200	5
p_7	220	8
p_8	210	12

数据点 p_6 距离查询点 q_2 比数据点 p_5 更近，p_5 不能障碍空间支配 p_6，所以此时的结果集为{p_2，p_3，p_5，p_6}。

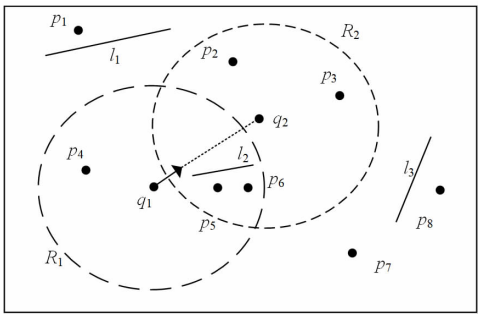

图 1　障碍空间中数据点的分布及连续查询示例

Fig. 1　Distribution of data points in obstacle space and examples of continuous query

2　障碍空间中连续范围 Skyline 查询

2.1　数据点和移动查询点间的距离表示

在进行连续 Skyline 查询的过程中，需要不断计算查询点和数据点间的距离。当查询点 q 和数据点 p 之间没有障碍物时，此时的距离函数[7]可表达为 $D(q,p,t)=\sqrt{at^2+bt+c}$，其中 a、b 和 c 为系数，是由它们的起始位置和速度决定的。若 q 的起始位置为(x_q,y_q)、速度为(v_{qx},v_{qy})，p 的起始位置为(x_p,y_p)、速度为(v_{px},v_{py})，则 $a=(v_{px}-v_{qx})^2+(v_{py}-v_{qy})^2$，$b=2[(x_p-x_q)(v_{px}-v_{qx})+(y_p-y_q)(v_{py}-v_{qy})]$，$c=(x_p-x_q)^2+(y_p-y_q)^2$。当数据点 p 静止时，a、b、c 仍然由此公式决定。当查询点 q 和数据点 p 之间有障碍物（本文用线段表示障碍物）时，它们的距离函数则可以表达为

$$D_{\mathrm{obs}}(q,p,t)=D(q,e_1,t)+\min\left\{\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}D(e_i,e_j,t)+D(e_n,p,t)\right\}$$

式中：n 为可见点总数；$i\neq j$；e_i 表示障碍线段的端点，障碍线段之间不相交。

查询点在移动的过程中，其距离是不断变化的，这样会导致数据点间支配关系发生改变，根据障碍距离的公式，可构造相应的距离曲线相交实例，如图 2 所示。假设数据点 p_1 支配数据点 p_2，在 t_1 时刻之前，$D_{\mathrm{obs}}(q,p_1)<D_{\mathrm{obs}}(q,p_2)$，$p_1$ 完全支配 p_2，此时的初始结果集不发生变化；在 t_1 时刻，p_1 和 p_2 在距离上相交；在 t_1 时刻之后，$D_{\mathrm{obs}}(q,p_1)>D_{\mathrm{obs}}(q,p_2)$，$p_1$ 不再完全支配 p_2，此时结果集为{p_1，p_2}。数据点的相交会造成查询结果集的变化，同样的，数据点 p_1、p_3 在 t_2 相交时刻前后，其支配关系也会发生变化。

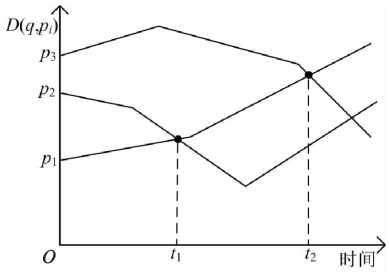

图 2　距离曲线相交示例

Fig. 2　Example of intersection of distance curves

2.2　约剪数据集

基于范围的障碍空间连续 Skyline 查询返回的数据集是由不被其他任意数据点障碍空间支配的点组成，此时的数据点不仅涉及到静态属性，还涉及到动态空间属性（即查询点和数据点之间的障碍距离）。数据点只要满足不被任何其他数据点非空间支配，那么必为所查询的 Skyline 点。任取两个数据点 p_1、p_2，假设 $T_{\mathrm{sta}}(p_1)=\phi$（即数据点 p_1 不被 p_2 非空间支配），无论数据点 p_2 是否比 p_1 距离查询点更近，p_2 都无法支配 p_1，由此可得出规则 1：若数据点 p 在非空间属性上没有任何数据点支配，即 $T_{\mathrm{sta}}(p)=\phi$，那么数据点 p 必为 Skyline 点。

例如，在障碍空间中，存在数据点 p_1、p_2、p_3、p_4 和查询点 q，当 $T_{all}(p_1)=\phi$，即不存在任何数据点非空间支配 p_1，无论查询点 q 在任何位置，非空间属性上都不可能存在数据点支配数据点 p_1，则数据点 p_1 必为 Skyline 点。

进一步，本文提出约剪初始数据集的剪枝策略 1，设 $D(pq)$ 表示查询点到数据点的不经过障碍物的最短距离，$D_{\mathrm{obs}}(pq)$ 表示查询点到数据点的最短障碍距离，圆 $C(pd)$ 表示以 p 为圆心、以 d 为半径的圆。剪枝策略示例如图 3 所示。

剪枝策略 1　在障碍空间中存在数据点 p，当 $T_{\mathrm{sta}}(p)=\phi$ 时，数据点 p、查询点 q 是可视的，若圆 $C(q,D(pq))$ 外存在其他数据点，那么该圆外数据点都被剪枝；当数据点 p 与查询点 q 不可视时，若圆 $C(q,D_{\mathrm{obs}}(pq))$ 外存在其他数据点，那么圆外数据点都被剪枝。

证明：由图 3 可知，在障碍空间中有任意一个查询点 q_1 和数据点集，可分两种情况。①当查询点和数据点可视时（查询点和数据点之间没有障碍物），假设数据点 p_3 非空间支配其他数据点，圆 $C(q_1,D(q_1,p_3))$ 外存在这样的数据点，其到查询点的距

离大于 $D(q_1,p_3)$，那么数据点 p_3 支配圆外数据点，数据点 p_1、p_2、p_6、p_7 和 p_8 都被剪枝；②当查询点和数据点不可视时（查询点和数据点之间有障碍物），假设数据点 p_5 非空间支配其他数据点，则圆 $C(q_1,D_{obs}(q_1,p_1))$外存在这样的数据点，其到查询点的障碍距离大于 $D(q_1,p_3)$，那么数据点 p_3 支配圆外数据点，数据点 p_1、p_6 和 p_8 都被删除。

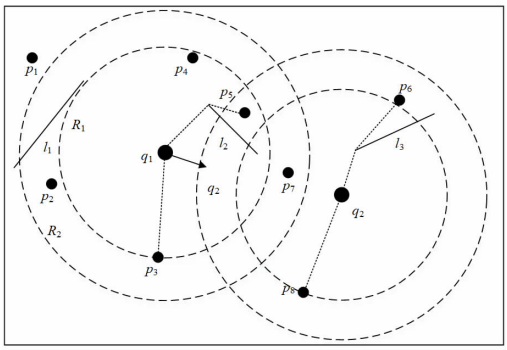

图 3 剪枝策略示例

Fig. 3 Example of pruning strategy

当空间查询点移动到 q_2 的位置时，同样可利用剪枝策略 1 进行数据集的过滤。基于以上讨论，进一步给出数据集的约剪过滤算法如算法 1 所示。算法 1 中，“←”表示赋值。

算法 1 CROS_Filter (P,q)

输入：数据集合 P，查询点 q；

输出：过滤后数据集 P'。

1. 计算初始 Skyline 点并加入到链表 L 中，$L\leftarrow$ ComputeSK$_{sta}(P)$；
2. 将链表 L 中的 Skyline 点按照与查询点 q 的距离从小到大排序，Sort(L)；
3. 获得距查询点 q 最远的 Skyline 点 p_s，p_s = getmaxdist(L)；
4. If q 与 p_s 之间无障碍物 then
5. 以 q 为圆心，$D(q,p_s)$为半径生成圆 $C(q,D(q,p_s))$；
6. If p 在圆 $C(q,D(q,p_s))$内 then
7. $P'\leftarrow P'\cup p$；
8. Else
9. 以 q 为圆心，$D_{obs}(q,p_s)$为半径生成圆 $C(q,D_{obs}(q,p_s))$；
10. If p 在圆 $C(q,D_{obs}(q,p_s))$内 then
11. $P'=P'\cup p$；
12. Return P'；

CROS_Filter(P,q)算法主要解决当查询点移动时对 Skyline 结果集无影响的数据进行有效过滤的问题。算法首先计算初始 Skyline 集并赋值给链表 L。算法第 4～7 行首先判断查询点 q、p_s 是可视的，若无障碍物，然后以 q 为圆心、$D(q,p_s)$为半径作圆，筛选掉当前时刻对 Skyline 集合无影响的数据，并把圆内的数据点集存储在集合 P'中。算法 9～11 行判断查询点 q、p_s 是不可视的，然后以 q 为圆心、$D_{obs}(q,p_s)$为半径作圆，处于圆外的数据点都被剪枝，圆内的数据点存储在集合 P'中。

2.3 查询点移动时连续 Skyline 处理

当查询点移动时，有些数据集始终在 Skyline 结果集中，而其他数据集会随着距离的变化动态进入或离开结果集。若要跟踪计算查询点和每个数据点的距离交叉时刻，则会产生大量的计算开销。本文根据数据点的非空间属性和支配关系有效避免大量的冗余计算。例如，在数据集中任取数据点 p_1、p_2，且 $T_{sta}(p_1)=\phi$、$T_{sta}(p_2)=\phi$，即数据点 p_1、p_2 互相不支配，数据点和查询点的障碍空间距离并不会影响 Skyline 结果集的变化。因此，本节提出了 2 个剪枝策略，当查询点和数据点的距离曲线发生相交时，Skyline 结果集不会发生变化。

数据点相交事件表示为$\langle p_1,p_2,t\rangle$（在相交时刻 t 之前，$D_{obs}(q,p_1)<D_{obs}(q,p_2)$），倘若满足相应的剪枝策略，则距离曲线发生相交时候不会影响 Skyline 结果集的变化。

剪枝策略 2 若 $p_1\notin T_{sta}(p_2)$，$p_2\notin T_{sta}(p_1)$，则不会对结果集产生影响。

证明：在静态属性上，p_1 与 p_2 互不支配，故不存在非空间支配。当查询点与 p_1、p_2 的距离曲线发生相交时，不会影响 Skyline 集的变化。

剪枝策略 3 若 $T_{sta}(p_1)=p_2$、$T_{sta}(p_1)=p_3$，则不会对结果集产生影响。

证明：在静态属性上，数据点 p_2 和 p_3 都支配 p_1，若查询点至数据点 p_1 的距离大于查询点至 p_2 的距离，即 $D_{obs}(q,p_1)>D_{obs}(q,p_2)$，那么 p_2 空间支配 p_1。当查询点至 p_1 的距离小于查询点至 p_3 的距离，则 p_1 不会成为 Skyline 结果集的数据点。

本文使用一个双向链表 L 存储当前全部 Skyline 点，$P(f_{lag},d_{ist},t_{val},t_{in})$表示 Skyline 点的信息，其中 f_{lag}表示 Skyline 点是否为静态 Skyine 点，d_{ist}表示查询点和数据点之间的障碍距离，t_{val}表示该 Skyline 点的有效时间，t_{in}表示 Skyline 与后继节点发生相交的时刻。Q_{event}用来存储影响 Skyline 集合的事件，事件表示为 $E(s_{elf},p_{eer}\ T_t,)$，其中 s_{elf} 为 Skyline 点，p_{eer}为与 Skyline 相关的数据点，T_t 为事

件的发生时间。本文提出的算法 2 即用于计算影响 Skyline 结果集的事件。

算法 2　CROS_CreateEvent(q,L)

输入：查询点 q，链表 L，数据集 P'

输出：Q_{event}

1. 将 Q_{event} 初始化为空集，$Q_{event}\leftarrow\phi$；
2. 遍历链表 L；
3. 遍历数据集 P'；
4. If L 和 P'中的数据点 l_i 和 p_j 满足剪枝策略 1 或策略 2；
5. 　继续处理下一组数据点；
6 Else；
7. 　计算 l_i 和数据点 p_j 距离的交换时间 T_t；
8. 　计算数据点 p 成为 Skyline 点的持续时间 t_{val}；
9. 　　更新 Point 节点 $P(f_{lag},d_{ist},t_{val},t_{in})$；
10. 　　生成事件 $E(s_{elf},p_{eer},T_t)$；
11. 　　将生成的事件 $E(s_{elf},p_{eer},T_t)$加入 Q_{event}；
12. Return Q_{event}；

CROS_CreateEvent(q,L)算法用于计算影响 Skyline 结果集的事件。算法第 4 行用来判断 l_i 和 p_j 是否满足剪枝策略 1 或剪枝策略 2，倘若满足则不进行处理，算法第 7 行计算 l_i 和数据点 p_j 距离交换时间，算法第 8 行计算数据点 p 成为 Skyline 点的持续时间，算法 9～11 行生成事件并插入队列中。

基于以上讨论，进一步给出计算当前时刻 Skyline 结果集的查询算法 CROS(q,P)。

算法 3　CROS(q,P)

输入：查询点 q，数据点集 P；

输出：当前时刻 Skyline 结果集 R_{sk}。

1. 调用算法 1 对数据进行剪枝，CROS_FILTER(SK_{sta},q)；
2. 计算出全局的 Skyline 点并加入到链表 L 中，Compute(L,P)；
3. 调用算法 2 计算影响 Skyline 结果集的事件，并返回 Q_{event}；
4. 遍历 Q_{event}中的每一个事件；
5. 　　更新 Skyline 结果集 R_{sk}；
6. Return R_{sk}；

CROS(q,P)算法的第 1 行首先调用算法 1，其依据剪枝策略 1 过滤掉距离查询点较远的数据点。算法第 3 行调用算法 2 计算出引起 Skyline 结果集发生变化的事件，其依据剪枝策略 2、3 避免了无效事件的生成，并存储在 Q_{event}中。算法 4～5 行处理每个事件并更新 Skyline 结果集。

3　实验分析

本节主要对所提算法进行实验比较分析。本文提出的 CROS 算法主要由 CROS_Filter 算法和 CROS_CreateEvent 算法组成。CROS_Filter 算法首先求出静态 Skyline 点，并由此过滤数据集，所以要对每个数据集进行比较，比较次数为 $N(N-1)/2$，每次比较的时间复杂度为 $O(k_1)$，故总时间复杂度为 $O(k_1N(N-1)/2)$。本文对静态 Skyline 点到查询点的距离进行排序，根据障碍空间中查询点和数据点间的距离函数，需要计算查询点和数据点的最短距离，时间复杂度为 $O(N(N-1)/2)$。CROS_CreateEvent 算法要为静态 Skyline 点和其支配点构建生成事件，生成事件的时间复杂度为$O(k_2)$。在 CROS 算法中，设 l 表示 Q_{event}队列长度，处理事件的时间复杂度为 $O(k_3)$，故其时间复杂度为 $O(lk_3)$。综上，算法的总时间复杂度为 $O((k_1+1)N(N-1)/2+k_2+lk_3)$。

本实验中对比算法为文献[7]的 CSQ 算法和文献[26]的 STA_OSSQ 算法。CSQ 算法主要处理欧氏空间中连续 Skyline 查询问题，STA_OSSQ 算法解决了障碍空间中查询点静止的 Skyline 查询问题。障碍空间中连续范围 Skyline 查询问题是一个新问题，已有的这两个算法无法直接处理障碍空间中连续范围 Skyline 查询问题，因此主要采用对算法中的核心计算进行反复调用和进行多次障碍距离计算的方式，对这两个算法进行了适当改进。

本文的实验环境为：3.4 GHz CPU，4 GB 内存，Windows10 操作系统，使用的数据集是由数据发生器随机生成的，同时随机生成线段障碍物。实验测试的指标是 CPU 运行时间，并取 100 次查询的平均值作为结果。

图 4 分析了数据集大小对 CPU 执行时间的影响。数据集的维度是 3 维，障碍物数量为 1 000，查询点随机生成。图 4 中 3 种算法的 CPU 运行时间都随着数据集的增大而上升。STA_OSSQ 算法执行时间最长，因为当查询点移动的时候，该算法需要反复利用 Voronoi 图的邻接特性在剪枝阶段和精炼阶段过滤大量非候选点，造成了大量重复的计算。在障碍空间中，因为过滤掉大量无效的数据点，CROS 算法处理生成 Event 数量要比 CSQ 算法少的多，所以执行时间优于 CSQ 算法。

图 5 分析了数据集不同维度对 CPU 执行时间的影响。实验设定数据点数为 1 000，障碍物数量为

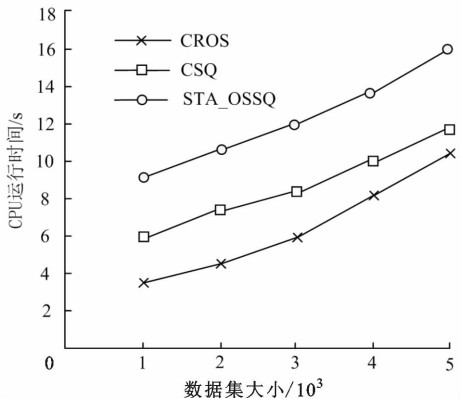

图 4 数据集大小对 CPU 运行时间的影响

Fig. 4 Effect of data set size on CPU running time

1 000,查询点随机生成。当数据集维度增大时,一个数据点可以非空间支配其他数据点或者被支配的可能性会降低,减小了 Event 的数量,因此 CROS、CSQ 算法的查询时间增长缓慢。

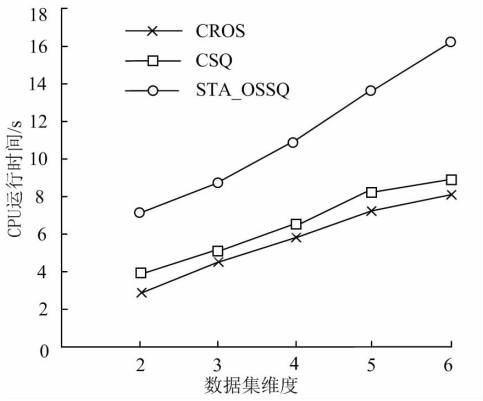

图 5 数据维度对 CPU 运行时间的影响

Fig. 5 Effect of data dimension on CPU running time

图 6 分析了障碍物数量对 CPU 执行时间的影响。实验设定数据点数为 1 000,数据集维度为 3 维,查询点随机生成。STA_OSSQ 算法的 CPU 运行时间优于 CROS、CSQ 算法的,因为在约减数据集的过程中对没有影响的数据集进行剪枝,而 CROS、CSQ 算法需要反复计算障碍距离,增加了开销。

图 7 分析了查询范围大小对算法 CPU 运行时间的影响,实验通过改变查询范围半径来改变查询范围的大小。实验设定的数据点数为 1 000,障碍物数量为 1 000,数据集的维度为 3 维,查询点随机生成。对于 STA_OSSQ 算法,随着范围的扩大需要反复约减数据集,查询时间增长迅速。CROS 算法查询时间增长缓慢,因为有剪枝策略,减少了事件的数量,所以要优于 CSQ 算法。

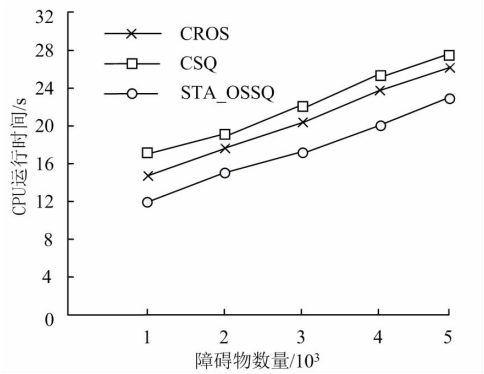

图 6 障碍物数量对 CPU 运行时间的影响

Fig. 6 Effect of number of obstacles on CPU running time

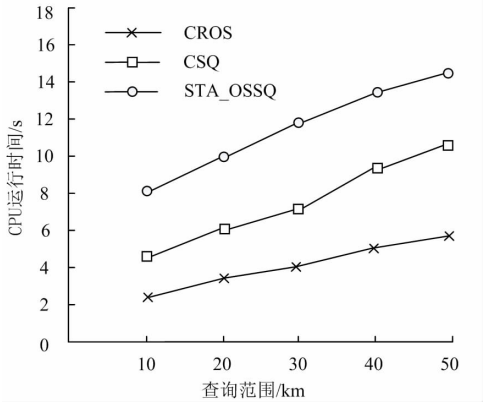

图 7 查询范围对 CPU 运行时间的影响

Fig. 7 Effect of query range on CPU running time

4 结 论

多目标决策优化在推荐系统中具有重要的作用,Skyline 查询是一类重要的多目标决策优化技术。随着推荐系统和位置服务技术的快速发展,空间连续 Skyline 查询具有更为重要的价值。已有 Skyline 查询方法无法直接处理基于范围的障碍空间连续 Skyline 查询问题,针对已有 Skyline 查询方法的不足,本文提出了基于范围的障碍空间连续 Skyline 查询算法。所提算法根据静态 Skyline 点的特征约减初始数据集,利用提出的剪枝策略有效减少数据点的数量,避免大量重复计算,提升了查询效率。本文的研究成果增强了推荐系统和基于位置服务的技术基础,能有效支持基于位置的兴趣点推荐功能。今后的研究重点主要集中在障碍空间中数据点移动情况下连续范围的 Skyline 查询等方面。

参考文献:

[1] BORZSONY S, KOSSMANN D, STOCKER K. The skyline operator [C]// Proceedings of 17th Internation-

al Conference on Data Engineering. Piscataway, NJ, USA: IEEE, 2001: 421-430.

[2] TAN K L, ENG P K, OOI B C. Efficient progressive skyline computation [C]// Proceedings of the 27th International Conference on Very Large Data Bases. New York, NY, USA: ACM, 2001: 301-310.

[3] SHARIFZADEH M, SHAHABI C. The spatial skyline queries [C]// Proceedings of the 32nd International Conference on Very Large Data Bases. New York, NY, USA: ACM, 2006: 751-762.

[4] HAN Xianan, WANG Bailing, LAI Guojun. Dynamic skyline computation on massive data [J]. Knowledge and Information Systems, 2019, 59(3): 571-599.

[5] 张彬，蒋涛，高云君，等. 度量空间中的 Top-k 反向 Skyline 查询算法 [J]. 计算机研究与发展，2014，51(3)：627-636.
ZHANG Bin, JIANG Tao, GAO Yunjun, et al. Top-k query processing of reverse skyline in metric space [J]. Journal of Computer Research and Development, 2014, 51(3): 627-636.

[6] ZHENG Yandong, LU Rongxing, LI Beibei, et al. Efficient privacy-preserving data merging and skyline computation over multi-source encrypted data [J]. Information Sciences, 2019, 498: 91-105.

[7] HUANG Zhiyong, LU Hua, OOI B C, et al. Continuous skyline queries for moving objects [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(12): 1645-1658.

[8] ZHENG Jiping, CHEN Jialiang, WANG Haixiang. Efficient geometric pruning strategies for continuous skyline queries [J]. ISPRS International Journal of Geo-Information, 2017, 6(3): 91.

[9] CHEEMA M A, LIN Xuemin, ZHANG Wenjie, et al. A safe zone based approach for monitoring moving skyline queries [C]// Proceedings of the 16th International Conference on Extending Database Technology. New York, NY, USA: ACM, 2013: 275-286.

[10] HIDAYAT A, CHEEMA M A, LIN Xuemin, et al. Continuous monitoring of moving skyline and top-k queries [J]. The VLDB Journal, 2022, 31(3): 459-482.

[11] 付世昌，董一鸿，唐燕琳，等. 基于事件的位置不确定移动对象连续概率 Skyline 查询 [J]. 自动化学报，2011，37(7)：836-848.
FU Shichang, DONG Yihong, TANG Yanlin, et al. Continuous probabilistic Skyline queries for moving objects with uncertainty based on event [J]. Acta Automatica Sinica, 2011, 37(7): 836-848.

[12] ZHENG Baihua, LEE K C K, LEE W C. Location-dependent skyline query [C]// The Ninth International Conference on Mobile Data Management. Piscataway, NJ, USA: IEEE, 2008: 148-155.

[13] MONTAHAIE E, GHAFOURI M, RAHMANI S, et al. Efficient continuous skyline computation on multi-core processors based on Manhattan distance [C]// 2015 ACM/IEEE International Conference on Formal Methods and Models for Codesign. Piscataway, NJ, USA: IEEE, 2015: 56-59.

[14] 田李，邹鹏，李爱平，等. 基于网格索引的连续 Skyline 计算方法 [J]. 计算机学报，2008，31(6)：998-1012.
TIAN Li, ZOU Peng, LI Aiping, et al. Grid index based algorithm for continuous skyline computation [J]. Chinese Journal of Computers, 2008, 31(6): 998-1012.

[15] KRIEGEL H P, RENZ M, SCHUBERT M. Route skyline queries: a multi-preference path planning approach [C]// 2010 IEEE 26th International Conference on Data Engineering. Piscataway, NJ, USA: IEEE, 2010: 261-272.

[16] MOURATIDIS K, LIN Yimin, YIU M L. Preference queries in large multi-cost transportation networks [C]// 2010 IEEE 26th International Conference on Data Engineering. Piscataway, NJ, USA: IEEE, 2010: 533-544.

[17] BAO Jinling, LIU Xingshan, ZHOU Rui, et al. Keyword-aware optimal location query in road network [C]// Web-Age Information Management. Cham, Germany: Springer International Publishing, 2016: 164-177.

[18] HUANG Y K, CHANG C H, LEE C. Continuous distance-based skyline queries in road networks [J]. Information Systems, 2012, 37(7): 611-633.

[19] CAI Zhi, CUI Xuerui, SU Xing, et al. Continuous road Network-based skyline query for moving objects [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(12): 7383-7394.

[20] ATTIQUE M, AFZAL M, ALI F, et al. Geo-social top-k and skyline keyword queries on road networks [J]. Sensors, 2020, 20(3): 798.

[21] 李松，窦雅男，郝晓红，等. 道路网环境下 K-支配空间 Skyline 查询方法 [J]. 计算机研究与发展，2020，57(1)：227-239.
LI Song, DOU Yanan, HAO Xiaohong, et al. The method of the K-dominant space skyline query in road network [J]. Journal of Computer Research and De-

velopment, 2020, 57(1): 227-239.

[22] HUANG Y K, CHANG C H, LEE C. Continuous distance-based skyline queries in road networks [J]. Information Systems, 2012, 37(7): 611-633.

[23] CAI Zhi, CUI Xuerui, SU Xing, et al. Continuous road network-based skyline query for moving objects [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(12): 7383-7394.

[24] 于晓楠, 谷峪, 张天成, 等. 一种障碍空间中的反 k 最近邻查询方法 [J]. 计算机学报, 2011, 34(10): 1917-1925.
YU Xiaonan, GU Yu, ZHANG Tiancheng, et al. A method for reverse k-nearest-neighbor queries in obstructed spaces [J]. Chinese Journal of Computers, 2011, 34(10): 1917-1925.

[25] GAO Yunjun, LIU Qing, MIAO Xiaoye, et al. Reverse k-nearest neighbor search in the presence of obstacles [J]. Information Sciences, 2016, 330: 274-292.

[26] 李松, 窦雅男, 张丽平, 等. 障碍环境中空间 Skyline 查询方法 [J]. 计算机科学与探索, 2018, 12(12): 1882-1890.
LI Song, DOU Yanan, ZHANG Liping, et al. Method of spatial skyline query in obstacle environment [J]. Journal of Frontiers of Computer Science & Technology, 2018, 12(12): 1882-1890.

(编辑 赵炜 刘杨)