位置不确定移动对象的连续概率反 Skyline 查询

汤志俊 樊明锁 何贤芒 陈华辉 董一鸿
(宁波大学信息科学与工程学院 宁波 315211)

摘 要 反 Skyline 查询现在已经被广泛用于商业规划、环境监测及其它领域。当前的研究多集中在目标对象静止条件下的反 Skyline 计算。对移动环境下的目标对象不断运动变化并且位置不确定情况下的连续概率反 Skyline 计算进行了研究。在对移动对象间反支配关系进行细致分析的基础上,定义了反支配概率和反 Skyline 概率以及导致反 Skyline 集变化的 process,提出了一种基于过程事件的算法来处理位置不确定移动对象的连续概率反 Skyline 查询算法,以通过计算、跟踪、更新这些 process 来快速找出 *p-RSky* 集合,并提出两个剪枝规则进行优化,避免了大量的无效计算。大量的实验证实了该算法的有效性。

关键词 不确定数据,移动对象,反 Skyline 查询,反 Skyline 概率
中图法分类号 TP311 **文献标识码** A

Continuous Probabilistic Reverse Skyline Query on Moving Objects with Uncertainty

TANG Zhi-jun FAN Ming-suo HE Xian-mang CHEN Hua-hui DONG Yi-hong
(College of Information Science and Engineering, Ningbo University, Ningbo 315211, China)

Abstract Reverse Skyline Queries have been proved usefully in business planning, environmental monitoring and other applications. Existing researches focus on static reverse skyline. This paper considered reverse skyline queries processing on moving target objects with uncertainty. On the basis of a detailed analysis of the reverse dominant relationship between moving objects, by defining reverse dominant probability, reverse skyline probability and the process which lead to reverse skyline set change, a new process-based algorithm was proposed to handle continuous probabilistic reverse skyline query on moving object with uncertainty. Types of processes were defined that will affect the *p-RSky* set, and by tracting and calculating those processes, the *p-RSky* of any time can be found out speedly. Two pruning rules were proposed to avoid a large number of invalid calculating. Extensive experiments show that our algorithm is efficient and effective.

Keywords Uncertain data, Moving object, Reverse skyline queries, Reverse skyline probability

1 引言

作为一项基于位置服务的重要操作,Skyline[8]计算在很多领域有非常重要的应用,目前,Skyline 操作都已经得到了广泛深入的研究。近几年,Skyline 的变种操作——反 Skyline 也引起了广泛的关注。一个查询点的反 Skyline 点集,是那些动态 Skyline 点集中包含了查询点的对象的集合[1]。这意味着相对于其他点,这些反 Skyline 点集是对查询点感兴趣的最好的对象集。例如,每位顾客对于酒店的喜好作为记录存入数据库中。如图 1 描述了在 2 维空间中(价格和到海滨的距离)7 位顾客打算去入住的酒店作为点集($a-g$)。顾客要预定酒店,希望该酒店价格尽可能便宜,距离景点的距离尽可能小。但是,往往距离景点较近的酒店价格都较高,同样,价格低的酒店距离景点往往很远。Skyline 操作即返回所有方面都不比其他酒店差的酒店,供顾客决策参考。而反 Skyline 从酒店方面来考虑顾客可能的选择,一名酒店经理如果想在市场上宣传他的酒店,可以将酒店作为查询点 q,通过查找酒店的反 Skyline 点集,对于这些反 Skyline 点集,查询点所代表的酒店能比附近的酒店更好地符合顾客要求。事实上,查询结果为$\{a,c,e\}$,表明 a,c,e 点的顾客更愿意去住在酒店 q。这样酒店经理可以更加精准有效地向这 3 位顾客投放广告。而顾客可能会得到一个有趣的提议而不是垃圾。反 Skyline 查询不仅已经运用在上述市场决策领域,也运用在环境监测、数量经济研究等方面。

在现实生活中,要查询的对象往往不是全部由静态数据组成的,而是伴随着某些动态属性的数据对象,例如顾客的移动造成距离的变化,而顾客对酒店的其它方面的指标却没有变化。特别是随着位置感知技术和无线传感技术的深入研究、移动设备特别是智能手机等的普及,基于位置的服务(LBS)越来越流行。而这类服务的目标往往是移动的对象,

到稿日期:2012-09-19 返修日期:2013-01-29 本文受国家自然科学基金项目(60973047,61202007)资助。
汤志俊(1987—),男,硕士生,主要研究方向为移动数据库,E-mail:zjtangzhijun@163.com;樊明锁(1988—),男,硕士生,主要研究方向为移动数据库;何贤芒(1982—),男,博士生,讲师,主要研究方向为数据库、数据挖掘;陈华辉(1969—),男,博士,副教授,主要研究方向为数据库、数据挖掘、数据流处理;董一鸿(1969—),男,博士,教授,主要研究方向为移动数据库、数据挖掘、人工智能。

并且因设备精确度、时间延迟等问题会造成数据的不确定。位置的不确定性造成了反 Skyline 关系的不确定性，移动对象之间的反 Skyline 关系只能用概率的形式来表示。例如，由于定位设备的不精确，顾客的位置可以用一个概率圆来表示，而顾客的走动会造成距离上的变化，进而影响对酒店的选择，顾客与酒店间的反 Skyline 关系就会产生变化，即酒店的广告对象的选择就会发生变化。

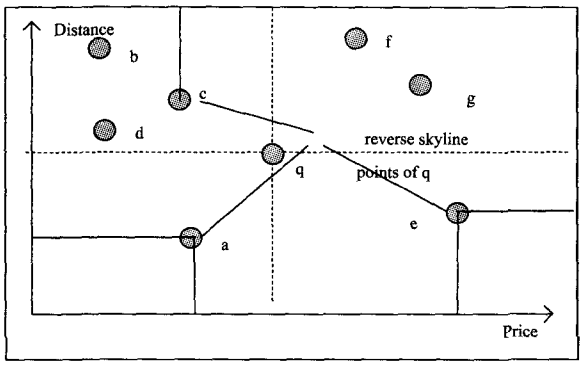

图 1　反 Skyline 查询实例

当前，反 Skyline 计算的研究主要集中在静态对象以及数据流的计算中，根据已有的知识，尚未发现移动环境下对象运动时的反 Skyline 查询的连续跟踪计算。

本文主要贡献如下：

1. 首次研究了移动环境下目标对象运动情况下的概率反 Skyline 计算，针对对象的运动特性，建立移动对象的不确定模型，对反 Skyline 查询进行动态更新计算。

2. 充分分析了移动对象位置不确定环境下对象间的反支配关系，并定义了对象间的反支配概率以及反 Skyline 概率的表示方式。根据影响概率反 Skyline 集合的 process 进行跟踪计算，并进行剪枝优化，提出了基于 process 事件的不确定移动对象连续概率反 Skyline 查询算法 U-PCRS(processes triggered continuous probabilistic Reverse Skyline Query on moving object with uncertainty)。

3. 在人工合成数据和实际数据下进行了大量的实验来验证 U-PCRS 算法的性能，并与其他算法进行了对比实验。实验结果显示了 U-PCRS 算法的有效性。

2　相关工作

Dellis 和 Seeger 等人[1]首先提出反 Skyline 的概念，并在静态的确定数据集环境下提出了 BBRS 和 RSSA 算法。首先计算全局 Skyline 的点集，并将其作为候选点，然后经过剪枝得到反 Skyline 集。RSSA 算法是 BBRS 算法的优化。面对不同数据集之间的查询，Lian 和 Chen[2] 提出了 MPRS 和 BPRS 两种概率算法来处理静态的不确定数据集，通过预计算的方法，用有效的剪枝方法来减小查询过程中的搜索范围。WU[6]在其基础之上探讨了确定数据集上的反 Skyline 查询，并提出几种重要的启发式方法来优化访问 R-tree，以此来减少 I/O 代价。Ah Han 等人[10]提出了一种用于在静态数据集中处理任意空间对象的算法 RSASO。RSASO 并不进行预计算，它通过对候选集用相应的方法进行剪枝得到最终的结果集。

在数据流方面，Zhu 等人[3]提出了一种基于 DC-tree 索引的数据流上的反 Skyline 查询算法，并提出了 2-层 Skyline 概念，完成了数据流上的反 Skyline 查询。白梅等人[5]研究不确定数据流上的概率反 Skyline 查询处理，提出了一种基于 R-tree 的不确定数据流上的概率反 Skyline 查询算法 RT2RS，运用了高效的剪枝策略，避免了大量的无效运算。

Wang 等人[4]引入半支配和 Full Skyline 的概念，并以 skyband 概念为基础，提出了传感器网络上的反 Skyline 查询算法。Prasad M. Deshpande 等人[7]在非标准的空间上，首先提出基于块的反 Skyline 查询方法(BRS)，并将其优化为排序反 Skyline 算法(SRS)，最后以树的反 Skyline 方法(TRS)减少了 I/O 代价。张彬、蒋涛等人[9]将 Skyline 与反 Skyline 相结合，提出了相互 Skyline 的概念。

我们已经对移动环境下位置不确定移动对象的概率 Skyline 查询展开了研究，文献[11]分析了目标点静止、查询点移动且位置不确定时的连续概率 Skyline 查询，文中提出了针对不确定移动对象的连续概率 Skyline 查询算法 U-CPSC。文献[12]分析了查询点静止、目标点移动且不确定时的连续概率 Skyline 查询，提出了移动环境下的 Skyline 集进行连续查询和更新的算法 U-ECPS，以实现概率 Skyline 计算的连续跟踪和动态更新。与已有研究不同的是，本文是针对 Skyline 的变种操作——反 Skyline 操作，研究的是在查询点静止、被查询的目标对象移动且位置不确定情况下基于 process 事件的连续概率反 Skyline 查询。

3　预备知识

3.1　相关概念

假设一个对象包括 k 维动态属性$\langle s_1,\cdots,s_j,\cdots,s_k\rangle$和 m 维静态属性$\langle p_1,\cdots,p_j,\cdots,p_m\rangle$。为简单起见，本文给定 n 点的空间数据集$\langle x_i,y_i,v_i,\theta_i,p_{i1},\cdots,p_{ij},\cdots,p_{im}\rangle$，$i=\{1,\cdots,n\}$，$x_i,y_i$ 是初始时刻对象 i 在空间中的位置坐标，v_i 是初始时刻的速度，$p_{i1},\cdots,p_{im}$ 是对象 i 的静态属性。如上述实例中，旅馆的价格、环境、服务质量等是静态属性，而距离是动态属性，不确定移动对象的移动导致数据集中的对象位置不断变化，继而造成与各旅馆的距离的改变。本文假定移动对象位置在不确定区域圆内以一定的概率密度函数 $f(x,y)$ 分布。移动对象速度$\vec{v}_A(t)=(v,\theta)$，其中 v 是移动对象的速度大小，θ 是其速度与 x 轴的夹角。表 1 为相关定义。

表 1　相关定义

Symbol	meaning
u≺v	u 支配 v
u⪯v	u 静态支配 v
Sky(P)	P 的 Skyline 集
$u\prec_q v$	u 关于 q 动态支配 v
DSky(P,q)	P 关于 q 的动态 Skyline 集
$u\vec{\prec}_q v$	u 关于 q 全局支配 v
GSL(P,q)	P 关于 q 的全局 Skyline 集
RSky(P,q)	P 关于 q 的反 Skyline 集

定义 1(支配[8])　对两个数据点 u 和 v，u 支配 v(记作 $u\prec v$)满足条件(本文以小值为优进行处理)：

1) $\forall i\in\{1,2,\cdots,d\}, u.i\leqslant v.i$；

2) $\exists j\in\{1,2,\cdots,d\}, u.j<v.j$。

定义 2(静态支配[8])　对两个数据点 u 和 v，如果在静态维属性上 u 都比 v 小，则 u 静态支配 v(记作 $u\preceq v$)。

定义 3(Skyline 集[8])　给定数据集 P，P 的 Skyline 集是所有不被其它点支配的点的集合。其中，每个点称为 Skyline 点。$Sky(P)=\{p\mid\neg\exists h\in P,h\prec p\}$。

定义 4(动态支配[1]) 给定查询点 q,对两个数据点 u 和 v,u 关于 q 动态支配 v(记作 $u \prec_q v$),满足条件:

1) $\forall i \in \{1,2,\cdots,d\}, |u.i-q.i| \leqslant |v.i-q.i|$;

2) $\exists j \in \{1,2,\cdots,d\}, |u.j-q.j| < |v.j-q.j|$。

定义 5(动态 Skyline 集[1]) 给出数据集 P 和查询点 q,P 关于 q 的动态 Skyline 就是所有不被其它点关于 q 动态支配的点的集合,即 $DSky(P,q)=\{p \mid \neg \exists h \in P, h \prec_q p\}$。

定义 6(全局支配[1]) 给定查询点 q,对两个数据点 u 和 v,u 关于 q 全局支配 v(记作 $u \overrightarrow{\prec}_q v$),满足条件:

1) $\forall i \in \{1,2,\cdots,d\}, |u.i-q.i| \leqslant |v.i-q.i|, (u.i-q.i)(v.i-q.i) \geqslant 0$;

2) $\exists j \in \{1,2,\cdots,d\}, |u.i-q.i| < |v.i-q.i|$。

定义 7(全局 Skyline 集[1]) 给定数据集 P 和查询点 q,P 关于 q 的全局 Skyline 就是相对于 q 点,不被其它点全局支配的点。$GSL(P,q)=\{p \mid \neg \exists h \in P, h \overrightarrow{\prec}_q p\}$。

定义 8(反 Skyline 集[1]) 给出数据集 P 和查询点 q,P 关于 q 的反 Skyline 点就是那些动态 Skyline 点中包含点 q 的点组成的集合,即 $RSky(P,q)=\{p \mid \forall p \in P, q \in DSky(p)\}$。

3.2 距离函数

移动对象 i 初始时刻位置为 (x_i,y_i),其以速度 $\vec{v}_i(t)=(v_i,\theta_i)$ 移动,查询点 q 位于 (x_q,y_q) 处,速度为 $\vec{v}_q(t)=(v_q,\theta_q)$。则在 t 时刻移动对象 i 与查询点 q 之间的欧式距离用函数 $d(q,i,t)$ 表示,$d(q,i,t)=\sqrt{(x_{it}-x_q)^2+(y_{it}-y_q)^2}$。令 $d(q,i,t)=\sqrt{at^2+bt+c}$,则 $a=(v_i\cos\theta_i-v_q\cos\theta_q)^2+(v_i\sin\theta_q-v_q\sin\theta_q)^2$,$b=2[(v_i\cos\theta_i-v_q\cos\theta_q)(x_i-x_q)+(v_i\sin\theta_i-v_q\sin\theta_q)(y_i-y_q)]$,$c=(x_i-x_q)^2+(y_i-y_q)^2$。本文假设查询点静止且位于原点,即:$x_q=0, y_q=0, v_q=0, \theta_q=0$。

最小距离函数:t 时刻不确定对象 i 到查询点 q 的最小距离函数用符号 $d_{\min}(q,i,t)$ 表示。为了简化计算,本文中移动对象的不确定区域假设为一个圆,移动对象 i 到查询点 q 的最小距离等价于一个点到一个圆的最小距离。当点在圆内(上)时,点到圆的最小距离为零;点在圆外时,点到圆的最小距离为点到圆心的距离减去半径长度。即:

$$d_{\min}(q,i,t)=\begin{cases} d(q,i,t)-r, & d(q,i,t)>r \\ 0, & d(q,i,t)\leqslant r \end{cases} = \begin{cases} \sqrt{at^2+bt+c}-r, & d(q,i,t)>r \\ 0, & d(q,i,t)\leqslant r \end{cases}$$

最大距离函数:t 时刻不确定对象 i 到查询点 q 的最大距离函数用符号 $d_{\max}(q,i,t)$ 表示。无论点在圆上或者圆外时,一点过到一个圆的最大距离都为点到圆心的距离加上半径长度。即:$d_{\max}(q,i,t)=\sqrt{at^2+bt+c}+r$。

4 位置不确定移动对象连续概率反 Skyline 查询

这一节提出了 U-PCRS 算法来处理位置不确定移动对象的连续概率反 Skyline 查询。首先通过静态维上的反 Skyline 点得到候选点集,其次对候选点集的动态维进行计算,只有同时在静态属性和动态属性上满足反 Skyline 定义的点才是最终的结果集。

4.1 概率反 Skyline 的计算

本文将数据集分为静态维属性 $p_{i1},\cdots,p_{im}$ 和动态维属性 x_i,y_i,用静态维上的 GSL 集来有效缩小反 Skyline 的候选集。

定理 1 如果 q 为查询点,$GSL(q)$ 为静态维上 q 的全局 Skyline 点,那么 $RSL(q) \subseteq GSL(q)$。

证明:假设对象 x 的静态维属性 $p_{xi} \notin GSL(q)$,那么必存在一点 p_y 全局支配 p_x。对于 $i \in \{1,2,\cdots,m\}$:$|q_i-p_{xi}| \geqslant |q_i-p_{yi}|$ 并且至少有一维 $j \in \{1,2,\cdots,m\}$:$|q_i-p_{xj}| > |q_i-p_{yj}|$,由反 Skyline 定义知 $x \notin RSL(q)$。

通过定理 1 保证了反 Skyline 点集的上界,减小了搜索空间。

定理 2 给定查询点 q 的静态维上的一点 p_x,且 $p_x \in GSL(q)$,如果 $\exists p_y \in P$,对于所有 $|q-p_y| < |p_x-p_y|$,那么点不是 p_x 的反 Skyline 点。

证明:假设如果 $\exists p_y \in P$,那么 $p_{yi} \prec_q p_{xi}$,由反 Skyline 的定义知,px 不属于 $RSL(q)$。

由定理 2 知,可以通过一个窗口查询来验证 $GSL(q)$ 是否是 $RSL(q)$,以此在静态维上进行剪枝得到候选点集。

定理 3 给定查询点 q 动态维属性 s_q(q 为查询点 $s_q=0$),在动态维上如果 $2|s_j-s_q| \leqslant |s_k-s_q|$,那么 j 是动态维属性上的一个反 Skyline 点的候选点。

证明:假设点 j 为动态维上的反 Skyline 点,由反 Skyline 定义知,不存在一点 k 使得对于点 j,在动态属性上满足 $|s_k-s_j| < |s_j-s_q|$,即 $|s_j-s_q| \leqslant |s_k-s_j|$。经变化得 $2|s_j-s_q| \leqslant |s_k-s_j|+|s_j-s_q|$,而 $|s_k-s_q| \leqslant |s_k-s_j|+|s_j-s_q|$,所以即当满足 $2|s_j-s_q| \leqslant |s_k-s_q|$ 时,j 在动态维上是 q 的反 Skyline 点的候选点。

定义 9(反支配) 给定查询点 q,对两个对象点 u 和 v,在动态维上 u 关于 q 反支配 v(记作 $u \underline{\preceq}_q v$),满足条件:

1) $\forall i \in \{1,2,\cdots,k\}, 2|s_j-s_q| \leqslant |s_k-s_q|$;

2) $\exists j \in \{1,2,\cdots,k\}, 2|s_j-s_q| < |s_k-s_q|$。

定义 10(反支配概率) 对于查询点 q,若 o_j 在动态维度上反支配 o_i,则对 o_j 内任一点 $\vec{x}_j$,圆弧内部阴影部分即为 o_j 所能反支配 o_i 的区域,如图 2 所示。令:

$$fun(\vec{x}_j,\vec{x}_i)=\begin{cases} 1, & 2|s_{\vec{x}_j}-s_q| < |s_{\vec{x}_i}-s_q| \\ 0, & \text{otherwise} \end{cases}$$

式中,$\vec{x}_j \in Ur_{O_j}$,$\vec{x}_i \in Ur_{O_i}$,Ur_{O_i} 表示 o_i 的不确定区域,则:$p(o_j \underline{\preceq}_q o_i) = \int_{Ur_{o_i}} \int_{Ur_{o_j}} f_{o_i}(\vec{x}_i) f_{o_j}(\vec{x}_j) fun(\vec{x}_j,\vec{x}_i) d_{o_i} d_{o_j}$,其 f_{o_i},f_{o_j} 分别为对象 o_i,o_j 的不确定区域的概率密度函数,若 o_i 在 k 维空间中,则 $\int_{Ur_{o_i}} f_{o_i}(\vec{x}_i) d_{o_i} = \int_{a_1}^{b_1} \cdots \int_{a_k}^{b_k} f_{o_i}(x_1,x_2,\cdots,x_k) d_{x_1} \cdots d_{x_k}$。

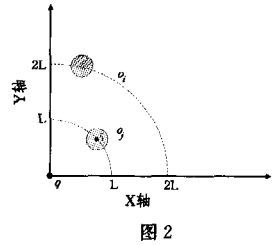

图 2

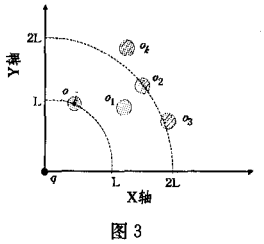

图 3

定义 11(概率反 Skyline) 对于查询点 p,对象 O 内任意一点 $\vec{x}$,O 不被其他任何对象所反支配的概率记为 $p(o)$。如图 3 所示,各对象的圆弧内非阴影部分是不被 $\vec{x}$ 所反支配的,即

$$p(o)=p(\wedge(o_i \not\underline{\preceq} o))$$

$$=\int_{Ur_o} f_o(\vec{x})\prod(1-\int_{Ur_{o_i}} f_{o_i}(\vec{x_i})\,fun(\vec{x_i},\vec{x})d_{o_i})d_o$$

式中，f_o，f_{o_i}分别为对象 o，o_i 的概率密度函数，$\vec{x}\in Ur_O$，$\vec{x_i}\in Ur_{O_i}$。

p-RSky 集合，给定阈值 $p\in[0,1]$，集合 D 中所有反 Skyline 概率不小于 p 的对象构成的集合即为 *p-RSky* 集。即：$RSky(p)=\{x\mid p(x)\geqslant p, x\in D\}$。

由于移动对象不断地移动，每个移动对象与查询点之间的距离随时间变化。设有两个移动对象 o_i，o_j，且 $o_i\prec^2 o_j$ 则 $p(o_i\prec^2 o_j)=$

$$\begin{cases}1, 2d\max(q,o_i,t)\leqslant d\min(q,o_j,t)\\ 0, 2d\min(q,o_i,t)\geqslant d\max(q,o_j,t)\\ \int_{Ur_{o_j}}\int_{Ur_{o_i}} f_{o_j}(\vec{x_j})f_{o_i}(\vec{x_i})\,fun(\vec{x_i},\vec{x_j})d_{o_j}d_{o_i}, \text{otherwise}\end{cases}$$

式中，$\vec{x_j}\in Ur_{O_j}$，$\vec{x_i}\in Ur_{O_i}$，f_{o_i}，f_{o_j}分别为对象 o_i，o_j 的概率密度函数。

4.2 影响 *p-RSky* 的 process

移动对象移动过程中，其与查询点间的距离不断变化，造成数据点间反支配关系的变化，如图 4 所示，$f(P)$，$f(Q)$分别是数据点 P，Q 与查询点之间建立的距离曲线函数。Max_P，Max_Q 分别是数据点 P，Q 与查询点之间建立的最大距离折线函数。Min_P，Min_Q 分别是数据点 P，Q 与查询点之间建立的最小距离折线函数。T_{begin1} 是折线 $Min_P-Max_Q=Max_Q$ 的相交时刻，T_{end1} 是折线 $Max_P-Min_Q=Min_Q$ 的相交时刻。

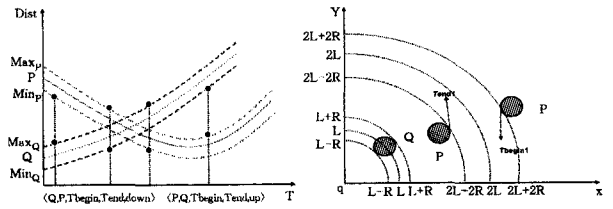

图 4

综合分析移动对象的移动对反 Skyline 关系的影响，本文定义了如下两种可能影响 *p-RSky* 集合的 process 类型。

定义 12(down process) 对于集合 S 中作为静态属性上反 Skyline 的候选点的数据点 P、Q，如果 $dist(q,Q,t)$ 与 $dist(q,P,t)$对任意两时刻 t_1，$t_2\in[t_{begin},t_{end}]$，假设 $t_1<t_2$，若 $P(Q\prec^2 P)\mid t_1>P(Q\prec^2 P)\mid t_2$ 成立，则 P，Q 产生一个 down process，表示为$\langle Q,P,t_{begin},t_{end},\text{down}\rangle$。其中 t_{begin} 是 $Mindist(q,P,t)-Maxdist(q,Q,t)=Maxdist(q,Q,t)$的时刻，$t_{end}$ 是 $Maxdist(q,P,t)-Mindist(q,Q,t)=Mindist(q,Q,t)$ 的时刻。

定义 13(up process) 对于集合 S 中满足的数据点 P、Q，如果 $dist(q,Q,t)$ 与 $dist(q,P,t)$对任意两时刻 t_1，$t_2\in[t_{begin},t_{end}]$，假设 $t_1<t_2$，若 $P(Q\prec^2 P)\mid t_1<P(Q\prec^2 P)\mid t_2$ 成立，则 P，Q 产生一个 up process，表示为$\langle Q,P,t_{begin},t_{end},\text{up}\rangle$。其中 t_{begin} 是 $Mindist(q,P,t)-Maxdist(q,Q,t)=Maxdist(q,Q,t)$的时刻，$t_{end}$ 是 $Maxdist(q,P,t)-Mindist(q,Q,t)=Mindist(q,Q,t)$的时刻。

4.3 剪枝策略

由 process 定义可知，由于每一个数据点都要与对其存在反支配可能性的数据点计算可能产生的 process，对于大规模数据集来讲，计算产生的 process 的数量是很庞大的。通过对 process 的研究发现，通过相关的剪枝策略可以减少计算量。

剪枝规则 1 数据点 P，Q 产生$\langle P,Q,t_{begin},t_{end},\text{down}\rangle$，在$\langle P,Q,t_{begin},t_{end},\text{down}\rangle$. t_{end} 之后如果存在一点 K，P、K 产生$\langle P,K,t_{begin},t_{end},\text{down}\rangle$，且$\langle P,K,t_{begin},t_{end},\text{down}\rangle$. $t_{end}\leqslant\langle P,Q,t_{begin},t_{end},\text{down}\rangle$. t_{end}，那么在$\langle P,K,t_{begin},t_{end},\text{down}\rangle$. t_{end}和$\langle P,Q,t_{begin},t_{end},\text{down}>$. t_{end} 之间的$\langle K,Q,t_{begin},t_{end},\text{down}\rangle$属于无效 process。

证明：数据点 P，Q 产生$\langle P,Q,t_{begin},t_{end},\text{down}\rangle$，在 t_{end} 时刻 $p(P\prec^2 Q)=0$，即 $Mindist(q,K,t)>2Maxdist(q,P,t)$。而$\langle P,K,t_{begin},t_{end},\text{down}\rangle$在 t_{end} 时刻 $p(P\prec^2 K)=0$，即 $Maxdist(q,K,t)\leqslant 2Mindist(q,P,t)$。而 $Mindist(q,K,t)>Mindist(x,P,t)$，所以 $Maxdist(q,Q,t)<2Mindist(q,K,t)$，即 $p(K\prec^2 Q)=0$，所以不必计算 K，Q 产生的$\langle K,Q,t_{begin},t_{end},\text{down}\rangle$的持续时间。

剪枝规则 2 数据点 P，Q 产生$\langle P,Q,t_{begin},t_{end},\text{up}\rangle$，$Q$ 与 K 存在 up process$\langle Q,K,t_{begin},t_{end},\text{up}\rangle$，那么在$\langle P,Q,t_{begin},t_{end},\text{up}\rangle$. t_{begin} 和$\langle Q,K,t_{begin},t_{end},\text{up}\rangle$. t_{end} 之间产生的$\langle P,K,t_{begin},t_{end},\text{up}\rangle$属于无效 process。

证明：数据点 P，Q 产生$\langle P,Q,t_{begin},t_{end},\text{up}\rangle$，在$\langle P,Q,t_{begin},t_{end},\text{up}\rangle$. t_{end} 时刻，$Mindist(q,Q,t)\geqslant 2Maxdist(q,P,t)$。同时在数据点 Q，K 产生 up process$\langle Q,K,t_{begin},t_{end},\text{up}\rangle$，$\langle Q,K,t_{begin},t_{end},\text{up}\rangle$. t_{end} 时刻，$Mindist(q,K,t)\geqslant 2Maxdist(q,Q,t)$。因为 $Maxdist(q,Q,t)>Maxdist(q,P,t)$，所以 $Mindist(q,K,t)>2Maxdist(q,P,t)$，即在$\langle P,Q,t_{begin},t_{end},\text{up}\rangle$. t_{begin} 到$\langle Q,K,t_{begin},t_{end},\text{up}\rangle$. t_{end} 之间产生的$\langle P,K,t_{begin},t_{end},\text{up}\rangle$的持续时间不必计算。

4.4 U-PCRS 算法实现

通过以上的分析，可以看到在任意时刻并不是所有对象的反 Skyline 概率都发生变化，只有当 P 最大(小)距离函数与 Q 中的对象的最小(大)距离函数相交时具有反支配关系时反 Skyline 概率才发生变化，即对象的 process 的开始时刻到结束时刻之间的时间反 Skyline 概率才发生变化。

process 是一个五元组($self$，$peer$，t_{begin}，t_{end}，$type$)，其中 $self$ 和 $peer$ 分别是与 process 相关的两个移动对象的 id 号；t_{begin} 和 t_{end} 分别是 process 的开始和结束时刻；$type$ 是指 process 对于移动对象 $self$ 是 down process 或是 up process。

U-PCRS 算法将通过预计算并保存距离函数具有反支配关系的相交时刻，并生成相应的 process 队列。process 按照开始时刻的时间顺序升序排列。对于 process 过程中的每一时刻 t，更新 t 时刻 process 所对应的位置不确定移动对象的反 Skyline 概率，即更新 *p-RSky* 集合。

Procedure: U-PCRS
Input: the set of moving objects D, query point q
a user-specified probability threshold p, time interval T.
Output: the p-RSky at any time instant t within T
1. Initialization();
2. Creat_process();
3. Handle process();

Algorithm 1 Initialization()
Input: the set of moving objects D, query point q
Output: the candidates of RSL(q)
1. for any point i in D
2. if($p_i\notin$GSL(q)) then discard i
3. else for any point p_i in GSL(q)

4. for each point p_j(j! =i) in D
5. compare(p_i,p_j);//比较 j 与 i 的静态属性
6. if $|q-p_i|<|p_i-p_j|$
7. flag(p_i,p_j)=1;
8. push j into the L_i;
9. Else flag(p_i,p_j)=0;
10. add i to the result set;
11. return the candidates of RSL(q)

U-PCRS 算法中的 Initialization()的功能主要是计算静态属性上的反 Skyline 集合作为候选集，再结合动态属性上的反 Skyline 集合从而得到每个时刻的数据集 D 的反 Skyline 集。

U-PCRS 算法中的 Creat_process()的功能是计算可能对 *p-RSky* 产生影响的 process 的开始、结束时刻，并保存 t_{begin}，t_{end}。Handle_process()的功能是处理 process 队列并通过剪枝规则更新相关位置不确定移动对象的反 Skyline 概率，从而进一步更新 *p-RSky*。

Algorithm 2 Creat_process()

Input: the set of moving objects D, query point q
Output: processes that may affect the p-skyline
1. for any point i in the candidate of RSL(q)
2. for each point j in L_i
3. computer a,b,c in term of q for i,j and the intersection time t_x
4. compute process time(i,j);//计算 process 的开始和结束时刻
5. if tend <=T//有效 process
6. judge the process type of (i,j);//判断该 process 的类型
7. push it into Q_i;//有效 process 加入 Q_i

Algorithm 3 Handle process()

Input: the set of the candidate RSL (q), query point q
Output: the p-RSky at any time t within T.
1. For any two point i,j in the candidate of RSL(q)
2. while (time++<=T)
3. while(Queue(Q_i)!=NULL)
4. p=Queue(Q_i),k=Queue(Q_i);
5. Update(Q_k);//according to pruning rule 1
5. while(Queue(Q_i)! =NULL&& Queue(Q_j)!=NULL)
6. p=Queue(Q_i),k=Queue(Q_j);
7. Update(Q_i);//according to pruning rule 2
8. Update(p-RSky)

4.5 算法性能与分析

本文将反 Skyline 概率的计算分成两步，首先比较数据点 u,v 间的静态维属性，只有在静态维属性上满足反 Skyline 定义的点才能成为候选点；其次考虑动态维属性，根据不确定移动对象的位置，由概率反 Skyline 公式计算反 Skyline 的概率，判断其是否大于阈值。针对不确定移动对象的连续移动产生了大量的 process，为减少计算代价，提出了两项剪枝策略。通过对剪枝后的 process 的跟踪计算就可以动态地更新 *p-RSky* 集合，而不必每一时刻去遍历整个数据集，减少了查询和更新的操作。

对于数据集中的任意两个数据点 u、v，Algorithm 1 中首先需要在静态维属性上进行比较，以得到候选点，比较次数随着维数的增多而变大，比较次数为 $n(n-1)$，时间复杂度为 $o(d \cdot n(n-1))$，其中 d 为静态属性的维度。Algorithm 2 中对于任意两数据点间存在可能性反支配的情况最多为 $n(n-1)/2$，同时移动对象到数据点间的距离曲线函数，t_{begin}，t_{end} 以及判断产生的 process 类型可以在常数时间内完成，时间复杂度为 $o(k \cdot n(n-1)/2)$。Algorithm 3 在最坏的情况下需要在每一时刻计算 $n(n-1)/2$ 个 process，时间复杂度为 $o(k \cdot n(n-1))$。根据对任意数据点 u 来说，其反 Skyline 概率是不存在被其它 $n-1$ 数据点反支配的有效区域的交集，其反 Skyline 概率的计算的时间复杂度为 $o(n-1)$。同时与阈值的比较需要 $o(n)$时间。综上所述，U-PCRS 算法的时间复杂度约为 $o(k \cdot n(n-1))$。

5 实验与分析

本文使用 Visual Studio C++ 2008 实现了一个仿真系统，以验证 U-PCRS 算法的有效性。实验运行环境为 Intel (R) Core(TM)2 Duo, 2.93 GHz CPU, 2GB RAM, Windows 7 的 PC 机。实验数据分别采用人工生成的具有多维属性的数据和真实数据集，其中动态维度为两维的数据空间，位置坐标记为(x,y)。人工数据集静态属性采用 3 种遵循不同分布的模拟数据：(1)随机数据；(2)正相关数据；(3)反相关数据。实验数据分布在 500×500 的数据空间上(闭区间[−250,250]上)。移动速度大小在闭区间[5,10]内，移动方向任意。为了方便运算，查询点的位置在原点。

本文的人工数据集上的实验主要就位置不确定移动对象的不确定区域半径 R、静态属性维度 d、移动对象数目 N 对算法的影响进行了分析，实验将算法与文献[10]中的 RSASO 算法进行比较，以验证 U-PCRS 算法的有效性。实验中不确定区域采用均匀分布的概率密度分布函数。

图 5(a)和(b)分别显示了当 $d=5$，$R=10$ 时移动对象数目对 U-PCRS 算法和 RSASO 算法的运行时间的影响。显而易见，同等条件下 U-PCRS 算法的运行时间远小于 RSASO 算法的运行时间，因为对每一更新时刻，U-PCRS 算法不必重新遍历整个数据集，而是仅对与 process 相关的数据点进行更新操作，减少了遍历与计算的时间开销。还可以看出，随着数据规模 N 的扩大，算法平均处理时间不断增加，这是因为数据规模的扩大导致反支配概率计算次数的增加，使得需要处理的 process 事件的增多，从而使运行时间增加。

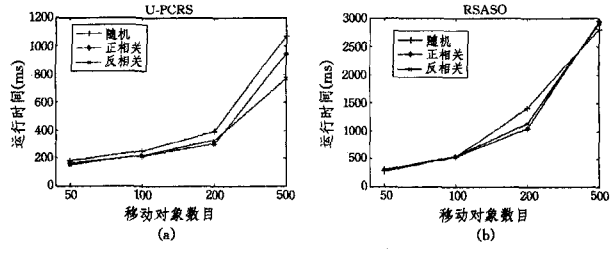

图 5 确定移动对象数目 N 对运行时间的影响

图 6(a)和(b)分别显示了 $N=200$，$R=10$ 时静态维度对 U-PCRS 算法和 RSASO 算法运行时间的影响。算法的平均处理时间和 process 队列规模都随着静态属性维度 d 的增加而不断减少。这是因为当静态维度增大时，静态属性上成为反 Skyline 候选点的可能性降低，导致在动态属性上存在反支配关系的移动对象数目的减少，相应的算法产生的 process 队列规模会减少。对每一个移动对象而言其从属的队列长度会减少，计算每个移动对象的反 Skyline 概率的开销会减少，因此响应时间会减少。

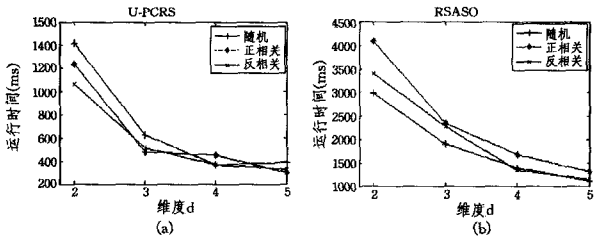

图 6　静态属性维度 d 对运行时间的影响

图 7(a),(b)和(c)分别显示了当 $R=10$ 时,当数据分别是随机数据、正相关数据以及反相关数据时静态维度 d 和移动对象数目 N 对 U-PCRS 算法产生的 process 数目的影响。从图中可以看到随着 N 的增大,process 数目增大,随着 d 的增大,process 数目减少。这主要是因为,当 N 增大时,移动对象距离函数相交的可能性会增大,导致产生 process 个数会增大。当 d 增大时,静态属性上存在支配关系的移动对象的数目会减少,从而产生的 process 数目会减少。

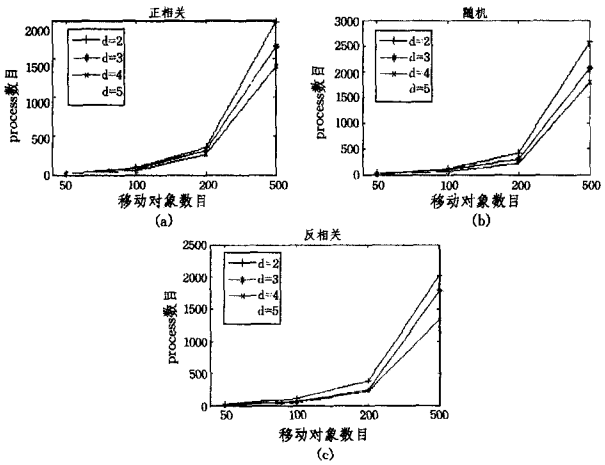

图 7　静态属性维度 d 和不确定移动对象数目 N 对 U-PCRS 算法和 RSASO 算法运行时间的影响

图 8(a)和(b)分别显示了 $N=200$,$d=3$ 时不确定区域半径变化在 U-PCRS 算法中对 process 数目以及运行时间的影响。

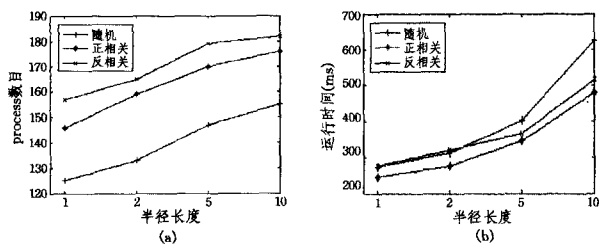

图 8　半径 R 对 U-PCRS 算法和 RSASO 算法运行时间的影响

由于本实验中的 process 持续时间段都较长,所以半径 R 的大小对 process 的数量影响并不是很明显,但随着半径 R 的增大,U-PCRS 算法的运行时间都变长,这是因为半径的增大使得反 Skyline 概率的计算开销会变大。

对于真实数据集,我们采用希腊雅典地区的 50 辆卡车的真实数据集作为测试,评估 U-PCRS 算法在真实环境下的有效性。我们为每辆卡车随机地产生速度的方向,同时为每辆卡车设置不同的半径 R 以产生不确定区域,静态维属性采用在[0,500]之间随机产生的 3 维数据,实验结果如图 9 所示。

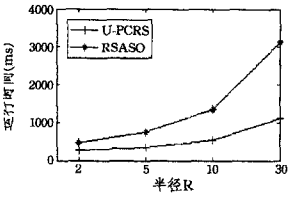

图 9　半径 R 在 U-PCRS 算法中对 process 数目以及运行时间的影响

对于任意的 process 而言,随着半径 R 的增大,在移动对象速度一定的情况下,$|t_{end}-t_{begin}|$ 增大,在连续更新的情况下对 process 跟踪计算的时间会增加。U-PCRS 只需对在剪枝后的 process 进行跟踪计算,而 RSASO 算法要每个时刻进行计算,平均处理时间将大大增加。

结束语　本文研究了移动环境下,查询点确定、目标点位置不确定时移动对象间的反支配概率的表示形式,以及移动对象间反 Skyline 概率的表示。定义了影响 p-$RSky$ 集合的 process,并根据其对移动对象反 Skyline 概率的影响分成了 down process 和 up process,并提出了基于 process 的算法,实现了对不确定移动对象的连续概率反 Skyline 计算。进一步的研究将关注查询点和目标点都在移动的概率反 Skyline 查询,即在查询点和目标对象的位置都具有不确定性的条件下的反 Skyline 查询。

参 考 文 献

[1] Dellis E, Seeger B. Efficient Computation of Reverse Skyline Queries[C]// Proc of VLDB' 07. New York: ACM, 2007, 291-302

[2] Lian X, Chen L. Reverse Skyline Search in Uncertain databases [J]. ACM Trans on Database Systems (TODS), 2010, 35(1): 1-49

[3] Zhu L, Li C, Chen H. Efficient Computation of Reverse Skyline on Data Stream [C]// Proc of CSO'09. Piscataway, NJ: IEEE, 2009: 735-739

[4] Wang G, Xin J, Chen L, et al. Energy-Efficient Reverse Skyline Query Processing over Wireless Sensor Networks [OL]. http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.64, 2011-06-01

[5] 白梅,信俊昌,东韩,等. 不确定数据流上的概率反轮廓查询处理[J]. 计算机研究与发展,2011,48(10):1842-1849

[6] Wu Xiao-bing, Tao Yu-fei, Wong R C-W, et al. Finding the Influence Set through Skylines[C]// EDBT 2009, Saint Petersburg. Russia, March 2009

[7] Deshpande P M, Deepak P. Efficient Reverse Skyline Retrieval with Arbitrary Non-Metric Similarity Measures[C]// EDBT 2011. Ppsala, Sweden, March 2011

[8] Borzsonyi S, Kossmann D, Stoker K. The Skyline operator[C]// Proceeding of The International Conference on Data Engineering (ICDE). Heidelberg, Germany, 2001: 421-430

[9] 张彬,蒋涛,乐光学,等. 一种最优的相互 skyline 查询算法[J]. 华中科技大学学报:自然科学版,2010,38(8)

[10] Han Ah, Li Zhong-he, Won D K, et al. An Efficient Method for Processing Reverse Skyline Queries over Arbitrary Spatial Objects[C]// Mobile Congress (GMC). Shanghai, China 2010 Global

[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)

[12] 曹金凤,董一鸿,王勇,等. 移动环境下的连续概率 Skyline 查询[J]. 计算机科学与探索,2012,6(5):443-444