

M-相点数据索引 SPindex

陈 瑛¹ 陈钊滢² 叶小平²

(广州商学院计算机系 广州 511363)¹ (华南师范大学计算机学院 广州 510631)²

摘 要 空间数据索引是空间数据管理的关键技术,其性能决定空间数据库的使用效率。由于可将时间维度转化为空间维度处理,空间数据索引在时空数据库和移动对象数据库管理方面广泛的应用。因此研究空间数据索引具有理论意义和应用价值。现有空间索引多是基于 R-树技术,为快速有效地访问海量空间数据,提出一种基于空间相点分析的空间数据索引方法 SPindex。首先,将空间区域所对应的最小外接矩形(MBR)集合与相平面中相点集合建立对应;其次,通过相点关系对相应 MBR 进行相互位置分析,进而提出一种基于相点的空间数据结构 MROB;以此为基础,提出一种新的基于 M-相点分析的空间数据索引 SPindex;最后,通过与常规索引进行实验,评估表明了该工作的可行性与有效性。

关键词 相平面与 M-相点, MROB 数据结构, SPindex, 仿真评估

中图法分类号 TP311.13 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.046

SPindex: Spatial Index Based on M-phase Points

CHEN Ying¹ CHEN Zhao-ying² YE Xiao-ping²

(Department of Computer, Guangzhou Commercial School, Guangzhou 511363, China)¹

(School of Computer, South China Normal University, Guangzhou 510631, China)²

Abstract Spatial data indexing plays a critical role in spatial data management, and its performance determines the efficiency of spatial database directly. By transforming spatial-temporal issues into pure spatial ones, spatial index can be widely used in spatial-temporal and moving object databases. Therefore it is meaningful and necessary to study the spatial data index. Most of the current spatial indexes are based on R-tree, however, in order to access massive spatial data quickly and efficiently, this paper proposed a novel spatial index SPindex based on the analysis of phrase-points. Firstly, we built a congruent relationship between the MBRs of spatial regions and phase points in phase plane. Secondly, though analyzing the positions of MBRs by the characteristics of the corresponding phase points, we proposed spatial data structure about phase points named MROB. Then a new spatial data indexing SPindex based on analysis of M-phase points was put forward. Finally, we designed experiments to test the performance of SPindex and results show that SPindex is feasible and efficient.

Keywords Phase plane and M-phase points, Data structure of MROB, SPindex, Simulation and evaluation

1 引言

在过去几十年间,地理信息系统(Geography Information System, GIS)被广泛应用于交通、电力、水利、城市规划、公安和军事等重要领域。研究与发展 GIS 产业对于国民经济的发展和维护国家安全都具有重要意义,而作为 GIS 核心技术的空间数据库(spatial databases, SDB)也随之成为数据库领域中的一个研究热点。与传统数据管理不同, SDB 是集储存、管理各种空间或非空间信息为一体的计算机信息应用系统,具有传统关系型数据库所不具备的技术要求。由于空间信息量的海量性以及空间对象、空间查询的高度复杂性,如果直接

对其进行基于遍历的查询,将无法满足实际应用需要,为此,空间数据索引就成为 SDB 的一项关键技术^[1]。现有空间数据索引技术可分为基于多维点和多维区域查询两种主要类型。基于点查询的主要有 K-D 树^[2,3]、K-D-B 树^[4]、HB 树^[5]等,它们多是由二叉树发展而来,另外还有四叉树^[6],它具有基于空间划分组织索引结构,主要应用于栅格数据管理。基于区域查询主要是 R 树^[7]系列,其结构和相关性质都可看作是由 B 树发展而来。基于点查询的索引结构相对简洁,效率较高,但多需要通过比较精细的空间剖分来处理多维点的无序性,另外不能兼顾多维区域查询;基于区域查询的 R 树索引是空间索引技术的主流,作为一种平衡树, R 树和 B 树具有

到稿日期:2014-03-11 返修日期:2014-05-06 本文受国家自然科学基金(60673135, 60373081), 国家自然科学基金重点项目(60736020), 广东自然科学基金(9151027501000054, s2011010003409)资助。

陈 瑛(1979—),女,讲师,CCF 会员,主要研究方向为新型数据管理技术、软件工程, E-mail: taoxiaotong@163.com; 陈钊滢(1990—),女,硕士生,CCF 会员,主要研究方向为空间数据库技术; 叶小平(1955—),男,教授,博士生导师,CCF 高级会员,主要研究方向为数据库与大数据环境信息服务, E-mail: dbzsu@163.com(通信作者)。

许多类似之处,它能与传统关系数据库相融合,同时可兼顾空间点查询,但 R 树结构复杂且对所处理数据的维数相当敏感,当空间目标数量增多时,R 树深度逐渐增大,树内目录矩形的重叠会迅速增长,导致空间开销增大,失败查找路径增长,降低查询效率。空间数据和关系数据的重大区别是难以“排序”,而索引需要建立在某种合适的序关系基础之上,因此,无论是四叉树的空间剖分还是 R 树的 MBR 技术,本质上都可以看作是对无序性的一种隐式而有效的处理。论文研究一种“显式”处理“序关系”的空间数据索引技术 SPindex,其要点是将各个空间区域所对应的最小外接矩形(Minimum Bounding Rectangle,MBR)表示映射成相平面上的 M-相点,通过相点之间的“拟序关系”分析对应 MBR 之间的位置关系,从而建立一种基于相点分析的空间数据结构——MROB 结构,以此为基础,建立了一种不同于现有经典技术的新的空间数据索引方法。本文工作的创新点是通过将二维 MRB 映射为一维 M-相点实现了空间数据的降维处理,再基于 M-相点自身特性建立其拟序关系,最后根据拟序关系实现了空间数据索引机制。SPindex 能够实现 R 树“点”和“区域”查询的基本功能,另外对三维时空数据“降维”到二维提供了借鉴,而这可应用在时空数据库特别是移动对象数据库。相关工作将另文讨论或见参考文献[8,9]。

本文第 2 节讨论 MBR 与 M-相点之间相互映射及相关特性;第 3 节提出构建 MROB 结构算法和基于 SPindex 查询算法;第 4 节是相关实验仿真与基本评估。

2 M-相点分析

空间索引中通常用最小外接矩形(Minimum Bounding Rectangle,MBR)来近似空间区域。通过对 MBR 的存储和查询就可近似地查询所需要的空间信息。

2.1 M-数与 M-相点

在二维欧氏空间中,坐标轴上区间 $[x_1, x_2]$ ($[y_1, y_2]$)称为一阶区间;两条相邻边分别平行于 x 和 y 坐标轴的矩形称为二阶区间。设 (x_1, x_1) 和 (y_2, y_2) 分别为二阶区间 w 的左下和右上顶点坐标,则可用一个二阶区间 w 来表示一个 MBR。

定义 1(MBR-数和 MBR 相点) 设 $w=[x_1, y_1; x_2, y_2]$, 则 w 对应两个 MBR-数:

$$\begin{aligned} M_1(w) &= M_1([x_1, y_1; x_2, y_2]) \\ &= m_{11}([x_1, y_1; x_2, y_2]) \times 10^a + m_{12}([x_1, y_1; x_2, y_2]) \\ M_2(w) &= M_2([x_1, y_1; x_2, y_2]) \\ &= m_{21}([x_1, y_1; x_2, y_2]) \times 10^a + m_{22}([x_1, y_1; x_2, y_2]) \end{aligned}$$

这里, m_{11} , m_{12} 和 m_{21} , m_{22} 分别是 w 中对角线顶点坐标 x_1, y_1, x_2, y_2 的函数, a 是所考虑 w 中 x_1, y_1, x_2, y_2 最大者的数位。

$w=[x_1, y_1; x_2, y_2]$ 对应相点 $M(w) = \langle M_1(w), M_2(w) \rangle$ 。

本文中 MBR 数和 MBR 相点也称为 M-数和 M-相点。

设 $w=[x_1, y_1; x_2, y_2]$, 在定义 1 取 $m_{11} = x_2, m_{12} = x = x_2 - x_1; m_{21} = y_2, m_{22} = y = y_2 - y_1$, 此时,

$$\begin{aligned} M(w) &= \langle M_1(w), M_2(w) \rangle = \langle x_2 \times 10^a + x, y_2 \times 10^a + y \rangle \\ &= \langle x_2 \times 10^a + (x_2 - x_1), y_2 \times 10^a + (y_2 - y_1) \rangle \end{aligned}$$

由所考虑 M-相点组成集合 T 所在的平面称为 M-相平面。

设 Γ 是所考虑 MBR 集合。由定义 1, 建立映射 $\sigma: \forall w \in \Gamma, w=[x_1, y_1; x_2, y_2], M(w) = (M_1(w), M_2(w))$, 则 σ 是由 Γ 到 $M(\Gamma)$ 1-1 对应, w 对应 M-相平面上 M-相点 $M(w)$, Γ 对应 M-相平面上 M-相点集 $M(\Gamma)$ 。

理论研究和实际应用中的数据对象通常会涉及到 MBR 间的相交、相离、包含等各种关系,其中“包含”是基本关系,在一定意义下,其它关系可转化为包含关系。本文主要研究区间之间的包含关系。

2.2 M-相点分析

集合 T 上满足自反性和传递性的关系 R 称为 T 上的一个拟序关系。

定义 2(相点拟序) 设 T 为 MBR 相点集合, 定义 T 上关系“ $\leq$ ”和“ $>$ ”如下:

$$\begin{aligned} ① \forall M(u), M(v) \in T, M(u) \leq M(v) \\ \Leftrightarrow (x_2(u) \leq x_2(v) \wedge (y_2(u) \leq y_2(v) \vee (x_2(u) = x_2(v) \wedge y_2(u) \leq y_2(v))) \\ ② \forall M(u), M(v) \in T, M(u) > M(v) \\ \Leftrightarrow (x_2(u) > x_2(v) \vee (x_2(u) = x_2(v) \wedge y_2(u) > y_2(v))) \end{aligned}$$

可验证“ $\leq$ ”和“ $>$ ”都是 T 上的拟序关系。

定理 1(相点与拟序) $\forall u, v \in M(\Gamma)$, ① $u \leq v \Leftrightarrow v \in UR(u)$; ② $v \leq u \Leftrightarrow v \in DL(u)$; ③ u, v 不相容 $\Leftrightarrow v \in OUL(u) \vee v \in ODR(u)$ 。

证明: ① 设 $u=[i, j], v=[k, l], u \leq v \Leftrightarrow i \leq k, j \leq l \Leftrightarrow v \in UR(u)$ 。

同理可证②和③。

定理 2(MBR 包含关系必要条件) 设 $w(i)=[x_1(i), y_1(i); x_2(i), y_2(i)], w(j)=[x_1(j), y_1(j); x_2(j), y_2(j)]$, 其中, $x(i)=x_2(i)-x_1(i), y(i)=y_2(i)-y_1(i), x(j)=x_2(j)-x_1(j), y(j)=y_2(j)-y_1(j)$, 若 $w(i) \subseteq w(j)$, 则 $(x_2(i) \leq x_2(j) \wedge (y_2(i) \leq y_2(j) \vee (x_2(i) = x_2(j) \wedge y_2(i) \leq y_2(j)))$ 。

证明: 只需证明, 当 $w(i) \subseteq w(j)$ 时, ① $M(i) \leq M(j)$; ② $M(i) \in DL(M(j))$; ③ $M(j) \in UR(M(i))$ 成立。设 a 为坐标值中最大位数, 由 MBR 包含关系可知 $x(i) \leq x(j) \wedge y(i) \leq y(j)$ 。从而有

$$\begin{aligned} w(i) \subseteq w(j) &= x_1(i) \leq x_2(j) \wedge y_1(i) \leq y_2(j) \\ \Leftrightarrow (x_2(i) \times 10^a + x(i) \leq x_2(j) \times 10^a + x(j)) \wedge (y_2(i) \times 10^a + y(i) \leq y_2(j) \times 10^a + y(j)) \\ \Leftrightarrow (x_2(i) \leq x_2(j) \wedge (x(i) \leq x(j) \vee (x_2(i) = x_2(j) \wedge x(i) \leq x(j))) \wedge (y_2(i) \leq y_2(j) \vee (y(i) \leq y(j) \vee (y_2(i) = y_2(j) \wedge y(i) \leq y(j)))) \end{aligned}$$

推论 若 $(x_2(i) \leq x_2(j) \wedge (y_2(i) \leq y_2(j) \vee (x_2(i) = x_2(j) \wedge y_2(i) \leq y_2(j))))$, 则 $\neg(w(i) \subseteq w(j))$, 也就是说 $\neg(M(j) \in UR(M(i)))$, 则 $\neg(w(i) \subseteq w(j))$ 。

3 索引 SPindex

3.1 MROP

定义 3(相点线序划分 MROP) 设 $L \subseteq \Gamma, L$ 为 Γ 中相点分支 $MROB \Leftrightarrow \forall u, v \in L, u \leq v \vee v \leq u \vee u = v$ 。 L 中最大、最小元分别记为 $\max L$ 和 $\min L$ 。设 MROP 是 MROB 组成的集合, 且若 $\forall MROB_i, MROB_j \in MROP, i \neq j, MROB_i \cap MROB_j = \emptyset$, 且 $\cup MROB = \Gamma$, 则称 MROP 是 Γ 上一个 M-相点线序

划分 MROP(Γ)。

算法 1(MROP 算法)

输入: 空间点集合 $\Gamma = \{m_1, m_2, \dots, m_n\}$

输出: 空间划分 MROP

Step 1 将 Γ 所有元素按照“>”排序, 不妨设排序后为 $\Gamma = \{m_1, m_2, \dots, m_n\}$, 令 $i=1, j=1$;

Step 2 若 $MROB_i = \emptyset$, 添加 m_j 到 $MROB_i$ 队列末尾, $j=j+1$;

Step 3 若 $j > |\Gamma|$, 转 Step 4; 否则, $k=1$, $\text{Tail}(MROB_k)$ 表示 $MROB_k$ 队尾元素;

Step 3.1 若 $k < i$ 且 $m_j \leq \text{Tail}(MROB_k)$, 添加 m_j 到 $MROB_k$ 队列的末尾, $j=j+1$, 转 Step 3.3;

Step 3.2 若 $k > i$, 则 $i=i+1$, 转 Step 2;

Step 3.3 若 $m_i \notin \text{Tail}(MROB_k)$, $k=k+1$, 转 Step 3.1。

Step 4 分别输出各 $MROB_m$ ($1 < m < i$) 中的所有元素, 这些元素组成一个线序分支, 且所有 $MROB_m$ ($1 < m < i$) 组成 Γ 上的一个 MROP。

算法 1 时间复杂度: Step 1 时间复杂度为 $O(n \log n)$, Step 2 为常数复杂度, Step 3 时间复杂度为 $O(n^2)$, Step 4 的时间复杂度为 $O(n)$, 故时间复杂度为 $O(n^2)$ 。

例 1 设 $\Gamma = \{(52, 51), (41, 62), (72, 62), (42, 51), (51, 63), (53, 51), (62, 72), (52, 41), (41, 72), (62, 73)\}$, 基于算法 1 构建 MROP 过程, 如图 1 和图 2 所示, 由此得到 3 条 MROB:

$MROB_1 = \langle (41, 72), (41, 62) \rangle$;

$MROB_2 = \langle (62, 73), (62, 72), (51, 63), (42, 51) \rangle$;

$MROB_3 = \langle (72, 62), (53, 51), (52, 51), (52, 41) \rangle$ 。

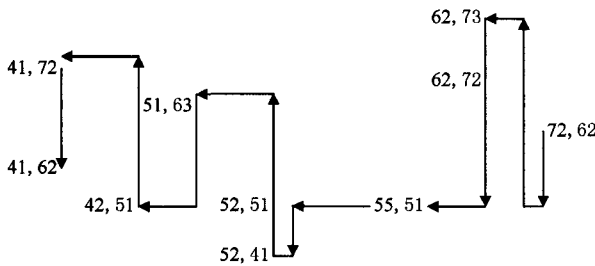


图 1 Γ 元素按照“>”排序

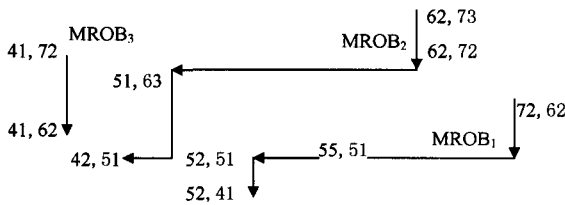


图 2 MROB 构建

3.2 空间索引 SPindex

以下为书写简洁, MROP 记为 LOP, MROB 记为 LOB 或 L 。设 $L \in \text{LOP}(\Gamma)$, L “首元素”记为 $\max(L)$, “尾元素”记为 $\min(L)$ 。设 $\Gamma_{\max}(\text{LOP}(\Gamma))$ 是 $\text{LOP}(\Gamma)$ 中“ $\max(L)$ ”列表, 简记为 $\Gamma_{\max}$, 由算法 1 在 $\Gamma_{\max}$ 上得到的线序划分记为 $\text{LOP}(\Gamma_{\max})$, 其中的线序分支为 $L_i(\Gamma_{\max})$ ($1 \leq i \leq |\text{LOP}(\Gamma)|$)。类似定义 $\Gamma_{\min}(\text{LOP}(\Gamma))$, 线序划分记为 $\text{LOP}(\Gamma_{\min})$, 其中线序分支记为 $L_r(\Gamma_{\min})$ ($1 \leq r \leq |\text{LOP}(\Gamma)|$)。 $L_i(\Gamma_{\max})$ 和 $L_r(\Gamma_{\min})$ 也称为“最值 LOB”, 而定义 1 中线序分枝称为“数据 LOB”。

定义 4(相点分析空间数据索引) 空间点集合 Γ 上索引 SPindex(Γ) 定义如下:

① 根结点 $L_{\text{root}} = \langle G_1, G_2, \dots, G_m \rangle$, $G_i = \max(L_i(\Gamma_{\max}))$ 。

② $L(\Gamma_{\max})$ 结点层: 在数据 LOP 中构建 $|\text{LOP}(\Gamma_{\max})|$, 其中 LOB 记为 $L_i(\Gamma_{\max})$ 。本层次中结点为 $\text{LOP}(\Gamma_{\max})$ 中 $L_i(\Gamma_{\max})$ ($1 \leq i \leq |\text{LOP}(\Gamma_{\max})|$)。

③ $L(\Gamma_{\min})$ 结点层: 在 $L_i(\Gamma_{\max})$ 对应数据 LOB 集合中构建 $\text{LOP}(\Gamma_{\min}(L_i(\Gamma_{\max})))$, 本层结点为 $\text{LOP}(\Gamma_{\min}(L_i(\Gamma_{\max})))$ 中 $L_r(\Gamma_{\min}(L_i(\Gamma_{\max})))$ ($1 \leq r \leq |\text{LOP}(\Gamma_{\min}(L_i(\Gamma_{\max})))|$)。

④ LOB 叶结点层: $L_r(\Gamma_{\min}(L_i(\Gamma_{\max})))$ 对应数据 LOB 构成本层结点。

其中, ①到③以及④分别称为 SPindex 的“导航层(Ori(SPindex))”和“数据层(Da(SPindex))”。

一个 SPindex 模式构成如图 3 所示。

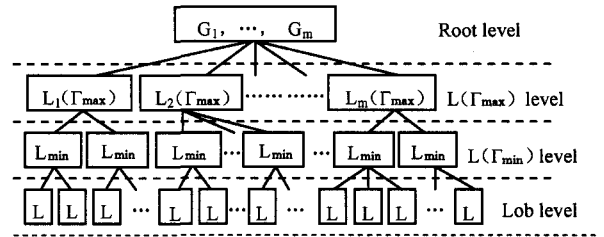


图 3 SPindex 模式

定理 3(MBR 相互关系的相点描述) 设 $w_i = [x_{i1}, y_{i1}; x_{i2}, y_{i2}]$, $w_j = [x_{j1}, y_{j1}; x_{j2}, y_{j2}]$, 则

$$w_i \subseteq w_j \Leftrightarrow (x_{i1} \geq x_{j1} \wedge y_{i1} \geq y_{j1}) \wedge (x_{i2} x(i) \leq x_{j2} x(j) \wedge y_{i2} y(i) \leq y_{j2} y(j))$$

证明:(充分性)若 $(x_{i2} x(i) \leq x_{j2} x(j) \wedge y_{i2} y(i) \leq y_{j2} y(j))$, 则 $x_{i2} \leq x_{j2}$ 且 $y_{i2} \leq y_{j2}$, 又 $(x_{i1} \geq x_{j1} \wedge y_{i1} \geq y_{j1})$, 可得 $w_i \subseteq w_j$ 。

(必要性)由 $w_i \subseteq w_j$ 得 $(x_{i1} \geq x_{j1} \wedge y_{i1} \geq y_{j1})$, 由定理 2 得 $x_{i2} x(i) \leq x_{j2} x(j) \wedge y_{i2} y(i) \leq y_{j2} y(j)$ 。证毕。

由定理 3 可得, 分析 MBR 之间的包含关系可以转化为考察相点之间的代数关系。

算法 2(SPindex 查询算法)

输入: 查询目标 $Q = [x_1, y_1; x_2, y_2]$

输出: 所有被 Q 包含的空间相点

Step 1 将查询目标 Q 转换为空间相点 $M(Q)$, 并令 $i=1, m = |\text{LOP}(\Gamma_{\max})|$ 。

Step 2 进入 $L(\Gamma_{\max})$ 层: 若 $L_i(\Gamma_{\max}) \subseteq M(Q)$, 则 $L_i(\Gamma_{\max})$ 包含 L 中元素为查询候选结果, 转 Step 5; 否则转 Step 3。

Step 3 进入相应 $L_{\min}$ 层, 依次考察 $L_i(\Gamma_{\max})$ 中子结点 $L_{\min_k}$ 。若 $k = |L_{\min}|$, 则转 Step 5。若 $L_{\min} \cap M(Q) = \emptyset$, $L_{\min}$ 包含 L 中元素都不查候选结果, $k=k+1$, 返回 Step 3, 否则转 Step 4。

Step 4 进入相应 LOB 层: 在 LOB 中实行二分查找:

Step 4.1 确定初始查找区间, 令 $\text{low}=0; \text{high}=\text{LOB.length}-1$ 。

Step 4.2 取中间点 $\text{mid}=(\text{low}+\text{high})/2$; 比较 $M(Q)$ 与 LOB_{mid} , 有以下 3 种情况:

Step 4.2.1 若 $\text{LOB}_{\text{mid}} \not\subseteq M(Q)$, $\text{low}=\text{mid}+1$; 查找在右半区进行, 返回 Step 4.2;

Step 4.2.2 若 $\text{LOB}_{\text{mid}} - 1 \not\subseteq M(Q)$, 则 $\text{index}=\text{mid}-1$, 转 Step 4.3; 否则令 $\text{high}=\text{mid}-1$, 查找在左半区进行, 返回 Step 4.2;

Step 4.3 把位置标号大于 index 的元素放到候选查询结果集中, 令 $k=k+1$, 转 Step 3。

Step 5 通过定理 3 将所有候选结果 M_i 与 $w(Q)$ 中 (x_1, y_1) 比较, 若

$w(Q)$, $x \leq M_i$, $x \wedge w(Q)$, $y \leq M_j$, y , 则 M_j 为查询结果。若 $i=m$, 转 Step 6, 否则, 令 $i=i+1$, 返回 Step 2。

Step 6 输出结果。

Step 2 到 Step 4 的时间复杂度为 $O(\log n)$, Step 5 的时间复杂度最大为 $O(n)$, 算法 2 时间复杂度为 $O(n)$ 。

4 仿真评估

本文设计了相应仿真以测试 SPindex, 其比较对象为四叉树。仿真主要测试建立索引和基于索引查询的情形, SPindex 还可进行增量式更新, 限于篇幅将另文讨论, 但其基本思想可参见参考文献[10]。实验硬件环境为处理器 AMD E-350 Processor 1.60GHz, 内存 2GB; 软件环境为: 操作系统 Win7, 编程语言 Java, 集成开发环境采用 Eclipse 3.4。对实验数据涉及到的参数做如下设定: 随机生成包含在 $[0, \max\text{Point}; 0, \max\text{Point})$ 区域内的 MBR 和相应的 MBR 集合 Γ , 其中 $\max\text{Point}$ 为 MBR 最大的坐标取值。

4.1 SPindex 创建

建立索引的时间开销如图 4 所示, 随着数据量的增大, 建立 SPindex 所需时间会上升且耗时多于四叉树。这是由于在建立 SPindex 时需对全部数据进行排序, 且建立每条 MROB 时都要遍历相应数据, 时间复杂度为 $O(m \times n)$ (其中 m 为所建立的 MROB 条数); 而建立四叉树只在每一层对全部数据遍历一次, 时间复杂度为 $O(h \times n)$ (其中 h 为四叉树索引的深度)。

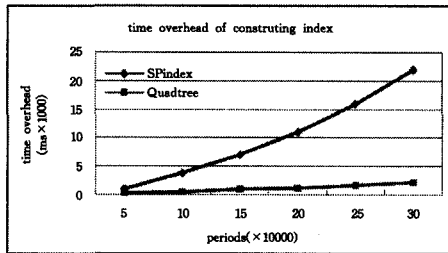


图 4 构建索引的时间开销

建立索引的内存开销如图 5 所示, SPindex 和四叉树索引所用内存空间均呈线性分布, 但 SPindex 比四叉树占用更少内存。这是因为 SPindex 存储的主要是待查询数据, 而四叉树索引把所有待查询数据存放在叶结点, 因此需要存储各类更为复杂的空间位置和结构信息。

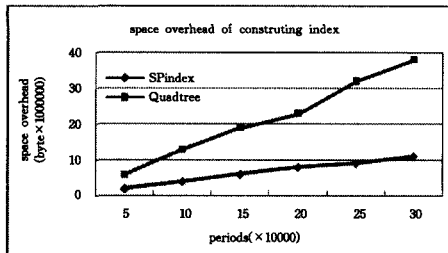


图 5 构建索引的空间开销

4.2 SPindex 查询

如图 6 所示, 随着数据量的增加, 平均一次随机查询的时间消耗也增长。其中 SPindex 比四叉树具有更好的查询效率。如图 7 所示, 当取 30 万数据时, 随着查询量增加, SPindex 查询效率也优于四叉树。

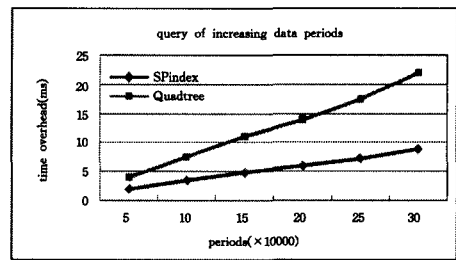


图 6 数据量增加时的查询

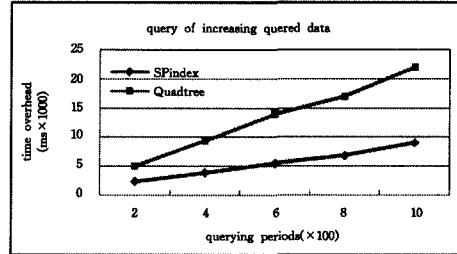


图 7 批量查询个数增加时的查询

对 30 万数据量, 查询域所在位置不同时的情形如图 8 所示。当查询域靠近开始点时, SPindex 和四叉树索引所需查询消耗均较少。而四叉树略优于 SPindex, 这是由于查询区域越靠近开始点, 四叉树遍历越少子树, 而 SPindex 无论是查询区域位于何处都要对所有 MROB 查询。随着查询位置后移, 四叉树所查询子树增多, SPindex 优势显现。

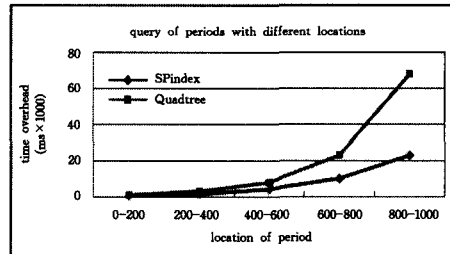


图 8 查询时间期间位置变化时的查询

对 30 万数据量, 查询区间变化时的情形如图 9 所示, SPindex 较四叉树显示出优势。

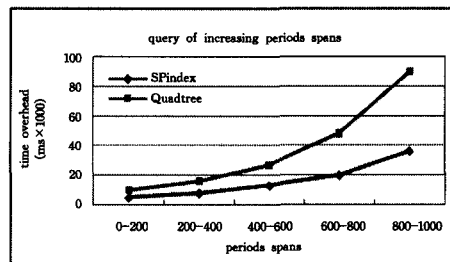


图 9 查询时间期间跨度变化时的查询

结束语 论文的主要贡献是通过将二维 MBR 映射为一维 M-相点, 由 M-相点分析建立起空间数据的拟序关系, 以此为基础建立了不同经典索引技术的新的空间数据索引 SPindex, 并与四叉树进行了比较评估。由仿真过程可知, SPindex 的创建时间耗费尽管比四叉树索引要大, 但换取了更高的查询效率且节省了空间耗费, 并且当数据量增大时, SPindex 整体性能也越好, 查询效率就越高。但当查询数据的相点大部集中在相平面的左下方时, 其查询效率略逊于四叉树索引。

(下转第 219 页)

(推理)发现 $\bar{w}(x)^F, \bar{w}(x)^F > w(x)$ 。利用式(35)、式(37)得到 $(\bar{x})^F$ 的智能挖掘系数 $\rho^F = 1.42 \in (0, 1]$; $\rho^F > 1$ 表明:在属性集合 α 内被补充属性,具有 α 的 (x) 内被补充信息元;或者, $\text{card}(\alpha)$ 增大, $\text{card}((x))$ 也被增大,表明了内逆 P-集合的基本特征。事实上,在另一类动态信息系统中,人们经常遇到属性增多、信息增大的现象。利用式(36)、式(38)与式(35)得到 $(\bar{x})^F$,它满足内逆 P-信息智能挖掘原理;利用 (x) 挖掘出 $(\bar{x})^F$ 。

结束语 利用一个基本事实,2012 年文献[1]提出逆 P-集合(inverse packet sets),逆 P-集合是另一类动态信息的数学模型表示;文献[2]提出函数逆 P-集合(function inverse packet sets),函数逆 P-集合是逆 P-集合的函数形式。在逆 P-集合(函数逆 P-集合)中,元素 x_i 与它的属性 α_i 满足属性析取特征,属性析取是另一类动态信息的基本特征,文献[1]给出了逆 P-集合的应用。与逆 P-集合相对应的是:利用一个基本事实,2008 年文献[3,4]提出 P-集合(packet sets),P-集合是一类动态信息的数学模型表示;文献[5,6]提出函数 P-集合(function packet sets),函数 P-集合是 P-集合的函数形式。在 P-集合(函数 P-集合)中,元素 x_j 与它的属性 α_j 满足属性合取特征,属性合取是一类动态信息的基本特征。文献[7-20]给出 P-集合、函数 P-集合在一类动态信息,函数 P-集合在一类动态信息,一类动态信息规律多个应用研究。无论逆 P-集合还是 P-集合,在一定条件下都被还原成有限普通集合(cantor set)。

利用逆 P-集合和逆 P-推理,本文给出在属性析取扩展条件下的内逆 P-信息智能(推理)挖掘,并给出几个基本理论结果与这些结果在金融系统中的一个通俗应用。本文中的例子稍加专业改进,能被应用到其他信息系统中。

参考文献

- [1] 史开泉. 逆 P-集合[J]. 山东大学学报:理学版,2012(1):98-103
- [2] 史开泉. 函数逆 P-集合与信息规律融合[J]. 山东大学学报:理学版,2012,47(8):73-80
- [3] 史开泉. P-集合[J]. 山东大学学报:理学版,2008,43(11):77-84
- [4] Shi Kai-quan. P-sets and its applications[J]. An International Journal Advances in Sytem Science and Applications, 2009, 9(2):209-219

- [5] 史开泉. 函数 P-集合[J]. 山东大学学报:理学版,2011,46(2):62-69
- [6] Shi Kai-quan. Function P-sets[J]. International Journal of Machine Learning and Cybernetics, 2011, 2(4):281-288
- [7] 史开泉. P-集合,逆 P-集合与信息智能融合-过滤辨识[J]. 计算机科学,2012,39(4):1-13
- [8] 赵树理,吴松丽,史开泉. 内 P-推理信息恢复与属性潜藏推理发现[J]. 计算机科学,2013,40(4):209-213
- [9] 史开泉. P-集合与它的应用特征[J]. 计算机科学,2010,37(8):1-8
- [10] 史开泉. P-推理与信息的 P-推理发现-辨识[J]. 计算机科学,2011,38(7):1-9
- [11] 林宏康,范成贤,史开泉. 倒向 P-推理与属性剩余发现-应用[J]. 计算机科学,2011,38(10):189-193
- [12] 李豫颖,范成贤,史开泉. 混合记忆信息与记忆信息筛选[J]. 系统工程与电子技术,2011,38(8):1824-1828
- [13] 汪洋,耿红琴,史开泉. P-集合与动态信息的依赖-发现[J]. 系统工程与电子技术,2011(9):2035-2038
- [14] 林宏康,李豫颖,阮群生. 数据依赖与异常数据分离-应用[J]. 计算机科学,2011,38(5):203-207
- [15] Zhang Guan-yu, Li En-zhong. Information gene and identification of its information Knock-out/Knock-in[J]. An International Journal Advances in Systems Science and Applications, 2010(2):308-315
- [16] 阎红灿,王坚,刘保相. 基于 P-集合的本体形式背景抽取[J]. 计算机应用研究,2012(6):2196-2204
- [17] Fan Cheng-xian, Lin Hong-kang. P-sets and the Reasoning-Identification of Disaster Information[J]. Journal of Convergence Information Technology, 2012, 7(1):337-345
- [18] Lin Hong-kang, Fan Cheng-xian. The dual form of P-reasoning and identification of unknown attribute[J]. International Journal of Digital Content Technology and its Applications, 2012, 6(1):121-131
- [19] Lin Rong, Fan Cheng-xian. P-sets and identification of inward-Convergence information [J]. An International Journal of Convergence Information Technology, 2012, 7(7):157-164
- [20] Lin Hong-kang, Li Yu-ying. P-sets and its P-separation theorems [J]. An International Journal Advances in Systems Science and Applications, 2010, 10(2):209-215

(上接第 209 页)

因此,SPindex 主要是适用于查询区间分散的情况或者是查询区域集中于右方的情况。这些都是需要改进之处,另外还将另文讨论 SPindex 的插入、删除算法以及结合时态信息的整合机制。

参考文献

- [1] 郭薇,郭善,胡志勇. 空间数据库索引技术[M]. 上海:上海交通大学出版社,2006
- [2] Bentley J L. Multidimensional Binary Search Trees used for Associative searching[J]. Communication of the ACM, 1975, 18(9):509-517
- [3] Bentley J L. Multidimensional Binary Search Trees in Database Application[J]. IEEE Transactions on Software Engineering, 1979, 5(4):333-340
- [4] Robinson J T. The K-D-B-tree; a search structure for large mul-

tidimensional dynamic indexes[C]// Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data. New York: ACM, 1981:10-18

- [5] Lomet D D B, Salzberg B. The HB-Tree: A Multiattribute Indexing Method with Good Guaranteed Performance [J]. ACM Trans Database Syst, 1990, 15(4):625-658
- [6] Samet H. The Quadtree and Related Hierarchical Data Structures[J]. ACM Comput Surv, 1984, 16(2):187-260
- [7] Guttman A I R - Trees: A Dynamic Index Structure for Spatial Searching[C]// SIGMOD Conference. 1984:47-57
- [8] 叶小平,郭欢,汤庸,等. 基于相点分析的移动数据索引技术[J]. 计算机学报,2011,34(2):256-274
- [9] 叶小平,陈瑞鑫,周旋珍,等. 移动对象索引 ST-tree[J]. 华南师范大学学报:自然科学版,2014,46(3):44-48
- [10] 叶小平,汤庸,林衍崇,等. 时态拟序数据结构研究及应用[J]. 软件学报,2014,25(11):2587-2601