

动态数据集环境下的强邻近对查询

李 松¹ 张丽平¹ 郝忠孝^{1,2}

¹(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

²(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(lisongbeifen@163.com)

Strong Neighborhood Pair Query in Dynamic Dataset

Li Song¹, Zhang Liping¹, and Hao Zhongxiao^{1,2}

¹(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

²(College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract The strong neighborhood pair query (SNP query) has important significance in the spatial data mining, big data processing, spatial database, geographic information system, similarity analysis and reasoning of data, etc. To remedy the deficiency of the existing work, according to the characteristic and complexity of the strong neighborhood pair query in dynamic dataset, a strong neighborhood pair query algorithm (VR_SNP algorithm) in the double data sets is proposed based on the Voronoi diagram and R tree. According to the irregular regions and uneven density of the points, the Voronoi diagram is used to query the strong neighborhood pair, conversely, the R tree is used to query the pair. Based on the secondary calculation and filtration for the initial neighborhood pair set, the VR_SNP_DA algorithm and VR_SNP_DE algorithm are given respectively to deal with the strong neighborhood pair query as the data sets increase and decrease dynamically. Furthermore, the VR_SNP_DL algorithm to deal with the query about the moving objects is studied. The theoretical study and the experimental results show that the algorithms have great advantages when the scale of the datasets are large and the positions of the points usually change.

Key words spatial database; Voronoi diagram; dynamic dataset; near neighbor query; strong neighborhood pair query (SNP query)

摘 要 数据集中的强邻近对查询在空间数据挖掘、大数据处理、空间数据库、地理信息系统、数据的相似分析和推理等方面具有重要的作用。已有的数据查询方法无法有效处理动态数据集中的强邻近对查询问题,针对动态数据集中的强邻近对查询的特点和复杂性,基于 Voronoi 图和 R 树空间索引结构提出了处理初始数据环境下的双数据集中的强邻近对查询算法 VR_SNP。针对分布区域不规则且数据点分布密度差异较大的情况利用 Voronoi 图进行计算查询,反之,则利用 R 树进行查询。通过对初始强邻近对集和候选邻近对集进行二次判断计算,筛选出有效结果,给出了数据集动态增加和动态减少环境下的强邻近对查询算法 VR_SNP_DA 和算法 VR_SNP_DE。进一步提出了移动点位置变化情况下的强邻近对查询算法 VR_SNP_DL。理论研究和实验比较表明在数据集的数据量、新增点集和删除点集的规模较大、移动点的位置变化次数较多等情况下,所提出的算法具有较为明显的查询优势。

收稿日期:2013-10-27;修回日期:2014-07-21

基金项目:国家自然科学基金项目(61370084);黑龙江省自然科学基金项目(F201302);黑龙江省教育厅科学技术研究项目(12531Z004, 12541128)

关键词 空间数据库; Voronoi 图; 动态数据集; 最近邻查询; 强邻近对查询

中图法分类号 TP311.13

随着信息技术的高速发展, 数据信息量成爆炸式增长. 数据的空间定位、海量数据的智能分析、网络多媒体信息的智能查询、交通智能管理等领域的应用需求使得空间海量数据的查询与分析技术成为计算机信息技术领域研究的一个重点和热点.

作为数据查询的一个重要问题, 数据集中的数据对象的近邻查询技术在空间数据挖掘、聚类分析、空间定位、智能导航、图像处理、数据相似性分析、地理信息系统和空间数据库等领域具有重要的作用. 国内外学者对近邻查询问题进行了广泛研究, 取得了一些研究成果^[1-2]. 近年来, 最近邻查询的研究进一步拓展到针对组最近邻查询^[3]、反向最近邻查询^[4]、连续障碍最近邻查询^[5]、不确定数据集中的 k 最近邻查询^[6]、可视最近邻查询^[7-8]、高维数据近似最近邻查询^[9]、不确定数据集的概率频繁最近邻查询^[10]、不确定图的 k 最近邻查询^[11]、基于 Voronoi 图的反向最近邻查询^[12]和组最近邻查询^[13]、线段最近邻查询^[14]、单纯型连续近邻链查询^[15-16]等方面. 所取得的成果解决了最近邻查询领域的一系列重要问题.

在近邻关系查询领域, 强邻近对查询问题是一个新问题, 所谓强邻近对查询即是在数据集中统一查询互为最近邻的点对和最近对. 强邻近对查询和最近邻查询具有各自不同的特点, 这 2 类查询的差异性主要在于: 最近邻查询主要解决数据对象的单向近邻查询问题, 例如在空间数据集中查询某移动数据点的最近邻, 该移动数据点为查询点, 数据集的其他数据对象为待处理的目标点, 通过对目标点过滤和精炼等步骤找到满足条件的数据点即可返回查询结果. 强邻近对查询主要处理数据集中的数据对象的双向邻近关系查询问题, 即需要确定互相为最近邻的点对和数据集中的最近对. 若将最近邻查询的一般方法直接用于强邻近对查询, 则需要将数据集中每个数据点均作为查询点反复调用查询算法进行查询, 确定每个数据对象点的最近邻, 然后再根据强邻近对的“相互”性进行筛选判断, 最后得出查询结果, 其计算量是相当大的. 因此, 探寻适合强邻近对查询的有效方法是有必要的.

强邻近对查询在数据对象的相似性分析、空间数据挖掘中的近邻关系分析、空间数据决策分析和图像相似性查询等方面具有重要作用. 例如卫星系

统采集的空间数据信息是海量而多样的, 计算机系统在挖掘和分析复杂的空间数据信息的内在联系和规律时, 往往需要判定和查询空间数据对象的邻近关系, 从而更有效地进行决策. 这类邻近关系既有通常的单向关系, 也有本文所研究的双向邻近关系. 数据集的强邻近对查询和分析不仅用于空间数据信息, 还可用于其他类型的复杂数据, 其数据对象之间的邻近关系不再局限于空间欧氏距离关系, 呈多样化和复杂化. 例如在网络商务信息分析中, 根据大量的交易信息、商品信息和消费者信息, 确定具有紧密“相互联系”的消费者和商品是有意义的. 如消费者和商品满足这样的条件: 消费者对某类商品更有偏好, 同时, 该商品是该消费者较适合的商品. 这里的“偏好”和“适合”即构成了“双向”邻近关系. 进一步, 还可提取“偏好”和“适合”相互联系最紧密的商品和消费者(即最近对). 显然, 有效处理和查询该类双向邻近关系能更好地帮助商业分析和决策. 目前, 数据信息的复杂性、海量性、多样性使得数据对象之间的关系变得更为复杂, 探寻数据集的强邻近对查询的有效方法具有一定的实际意义.

通常, 若利用已有查询数据集中单个最近邻的方法对数据集中的数据点采取逐一处理, 则需要对每个点的最近邻情况进行计算判断, 需要反复调用相关算法, 计算量非常大, 在数据集较大的情况下该方法效率较低. 为了处理数据集中的强邻近对查询问题, 文献[17]提出了过滤区域的概念, 分情况对数据点对进行查询处理. 但文献[17]主要针对 2 个静态的相似数据集进行处理, 所提出的方法无法处理数据集动态变化的情况, 在实际应用中具有局限性. 由于数据对象的变化(如位置变化、形状变化、数量变化)较为常见, 动态数据集中的强邻近对查询是一个更为复杂的新问题, 为了弥补已有方法的不足, 本文对动态数据集中的强邻近对查询问题进行了系统的创新性研究.

1 基本定义

为了研究动态数据集中的强邻近对查询方法, 本节首先给出文中所需的基本定义.

定义 1^[1]. 设有一 d 维空间的点集 S 和一个查询点 o , 最近邻查询就是找出的子集 $NN(o, S)$:

$NN(o, S) = \{s \in S \mid \forall h \in S; D(o, s) \leq D(o, h)\}$.

定义 1 中, $D(o, s)$ 和 $D(o, h)$ 等表示 2 个数据点之间的距离函数. 在不强调数据集的情况下, $NN(o, S)$ 也可简写为 $NN(o)$, 表示 o 的最近邻.

基于定义 1, 数据集中强邻近对查询的形式化定义如下:

定义 2^[2]. 给定 2 个对象点集 S 和 $P, s \in S, p \in P, D(s, p)$ 表示点 s 到 p 的最短距离. 若 $NN(s, P) = p$ 或 $NN(p, S) = s$, 则称点对 (s, p) 为近邻对, 其中, 若 $NN(s, P) = p$ 且 $NN(p, S) \neq s$, 则称点对 (s, p) 为单向近邻对, 记为 $(s, p>);$ 若 $NN(s, P) = p$ 且 $NN(p, S) = s$, 则称点对 (s, p) 为对偶近邻对, 记为 $\langle s, p \rangle; \forall s' \in S, p' \in P$, 若 $D(s, p) \leq D(s', p')$, 则称 (s, p) 为最近对, 记为 $[s, p]$. 将对偶近邻对和最近对统称为强邻近对. 在 S 和 P 中寻找强邻近对的查询称为强邻近对查询.

2 动态数据集集中的强邻近对查询

现实中, 数据集 S 和 P 往往不是静止的, 随着时间和应用需求的改变经常会发生动态变化(例如数据集中数据点的增多和减少等). 数据集合中的强邻近对查询(SNP query)依据数据集的动态改变类型可分为数据集动态增加情况下的强邻近对查询(SNP_DA query)、数据集动态减少情况下的强邻近对查询(SNP_DD query) 和数据点位置动态改变情况下的强邻近对查询问题(SNP_DL query).

2.1 初始数据集集中的强邻近对查询

对于空间数据集 S 和 P , 本节首先提出一种利用空间索引结构 R 树^[1] 处理强邻近对查询问题的方法(简称 R_SNP 方法). R 树是分级数据结构, R 树及其变形树是空间数据库应用最多的数据结构. R 树是用来存储多维对象的最小的边界框, 而不是最初的空间对象. R_SNP 方法的主要步骤如下:

步骤 1. 针对集合 S 和集合 P , 分别构建 S 和 P 中空间数据点的 R 树索引结构;

步骤 2. 基于构建的 R 树索引结构, 分别以 S 和 P 中的各个空间数据点为查询点进行最近邻的查询, 得出单向近邻对;

步骤 3. 对步骤 2 中的查询结果进行比较筛选, 进一步确定出强邻近对.

R_SNP 方法基于 R 树进行空间数据的组织, R 树以最小外包矩形对空间数据对象进行近似和聚类, 对于数据分布区域较为规则, 数据分布较为均匀

合理的空间数据集, 基于 R 树进行查询和处理具有较大的优势. 但实际应用中, 空间数据分布区域和分布密度等往往较为复杂多样, 从而导致 R 树的构建难度加大, 且在构建 R 树时, 其最小外包矩形所产生的空白区域往往会较大, 使得查询往往需要反复多分支进行, 查询的效率不理想. 为了能有效处理多样的空间数据分布区域和分布不均匀情况下的强邻近对查询问题, 本节基于 R 树和计算几何中的 Voronoi 图给出了强邻近对查询的方法(简称 VR_SNP 方法).

Voronoi 图的定义如定义 3 所示.

定义 3^[18]. 给定 2 维空间中的一组数据点 $S = \{s_1, \dots, s_n\}$, 其中 $2 < n < \infty$, 当 $i \neq j$ 时, $s_i \neq s_j$. Voronoi 区域形式化定义为 $VP(s_i) = \{s \mid D(s, s_i) \leq D(s, s_j)\}$. $D(s, s_i)$ 为 s 与 s_i 之间的最小距离. s_i 称为 Voronoi 生成点, 由 s_i 所决定的 Voronoi 区域 $VP(s_i)$ 称为 Voronoi 多边形, Voronoi 多边形的棱记为 $VL(s_i)$. 由 $V(S) = \{VP(s_1), \dots, VP(s_n)\}$ 所定义的图形被称为 Voronoi 图. 共享相同棱的 Voronoi 多边形被称为邻接多边形, 它们的生成点被称为邻接生成点.

性质 1^[18]. 二维空间中, 每个 Voronoi 生成点最多有 6 个邻接的生成点. 离 Voronoi 生成点 d_i 最近的生成点 d_j 必是 d_i 的邻接生成点.

性质 2^[18]. 一个 Voronoi 多边形 $V(d)$ 内任何点 $p(p \in d)$ 到 d 的距离必小于 p 到其他 Voronoi 生成点的距离.

图 1 展示了以数据集 S 中的部分数据点为生成点所构建的部分 Voronoi 图. 图 1 中, 集合 S 中的部分数据点 $s_1, \dots, s_{26}$ 是生成点, 包围生成点的多边形即为 Voronoi 多边形, p_1, p_2 等数据点是处于 Voronoi 图中的集合 P 中的部分数据点. 基于 Voronoi 图的定义和性质, 空间数据点和生成点之间的位置邻近关系可进行较好的表示. 例如由性质 2 可知, p_{38} 的最近邻即为 s_{21} . 通常, 数据集中的数据点不是静止不动的, 往往会发生变化, 例如数据对象点 p_{39} 在某时间段内沿着图 1 中的曲线由 s_1 对应的 Voronoi 多边形内移动到 s_{20} 对应的 Voronoi 多边形内. 显然, p_{39} 的最近邻即由 s_1 改变为 s_{20} .

基于定义 3 和性质 1, 2, 我们进一步给出利用 Voronoi 图的性质进行判断最近对和单向近邻对的推论.

推论 1. 对象点集 S 和 P 的 Voronoi 图为 $V(S)$ 和 $V(P)$, 若 $(d_i \in VP(p_j)) \wedge (p_j \in VP(s_i))$, 则 s_i 和

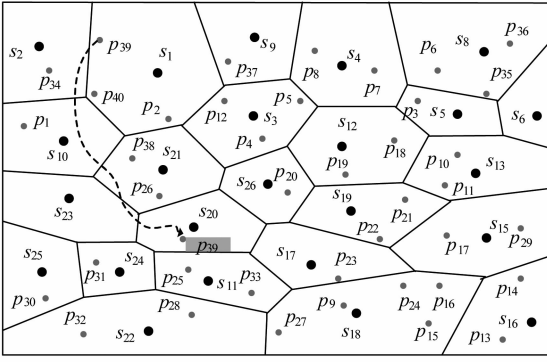


Fig. 1 Voronoi diagram and data points set.

图1 Voronoi图和数据点集

p_j 为最近对 $[s_i, p_j]$; 若 $(s_i \in VP(p_j)) \wedge (p_j \in VP(s_i))$, 则 s_i 和 p_j 为单向近邻对 $\langle s_i, p_j \rangle$; 若 $(p_j \in VP(s_i) \wedge (d_i \in VP(p_j)))$, 则 p_j 和 s_i 为单向近邻对 $\langle p_j, d_i \rangle$.

在 VR_SNP 方法中, 对于空间内的数据点集 S 和 P , 若 S 中数据点数目和 P 内的数据点数目相差较小, 且分布的区域不规则, 数据分布密度较复杂, 则可利用 Fortune 算法^[18] 分别生成 S 和 P 的 Voronoi 图, 再根据性质 1, 2 和推论 1 对 S 和 P 的强邻近对进行筛选查询. 若数据集分布的区域较为规则, 数据分布密度较均匀, 则可利用 R 树对强邻近对进行查询. 相应的 VR_SNP 算法如算法 1 所示:

算法 1. VR_SNP(S, P).

输入: 数据点集 S 和 P ;

输出: 对偶近邻对集 DN 、最近对集 CN 、候选邻近对集 AN .

$DN \leftarrow \emptyset, CN \leftarrow \emptyset, AN \leftarrow \emptyset, TN \leftarrow \emptyset, GN \leftarrow \emptyset$; /* 初始化 */

if 分布区域不规则、数据点分布密度差异较大
then

Generate_Voronoi($V(S)$);

Generate_Voronoi($V(P)$);

/* 生成 Voronoi 图 */

for $j=1$ to $|P|$ do

/* $|P|$ 表示 $V(P)$ 中生成点的个数 */

if $(p_j \in VP(s_k)) \wedge (s_k \in VP(p_j))$

/* 判断单向近邻对 */

then $AN \leftarrow \langle s_k, p_j \rangle$;

end if

if $(p_j \in VP(s_k)) \wedge (s_k \notin VP(p_j))$

/* 判断单向近邻对 */

then $AN \leftarrow \langle p_j, s_k \rangle$;

end if

if $(s_k \in VP(p_j)) \wedge (p_j \in VP(s_k))$

/* 判断对偶近邻对 */

then $DN \leftarrow \langle p_j, s_k \rangle$;

for $k=1$ to $|DN|$

计算 DN 中对偶近邻对的距离;

计算距离最小值 $|s_v, p_w|$;

$CN \leftarrow (s_v, p_w)$;

return DN, CN, AN ;

end for

end if

end for

else if

$Rtree_S \leftarrow Generate_Rtree(S)$;

$Rtree_P \leftarrow Generate_Rtree(P)$;

/* 分别构建集合 S 和 P 的 R 树结构 */

for $j=1$ to $|S|$ do /* 基于 R 树求最近邻 */

$TN \leftarrow (s_j, NN_{Rtree_P}(s_j, P))$;

end for

for $j=1$ to $|P|$ do /* 基于 R 树求最近邻 */

$GN \leftarrow (p_j, NN_{Rtree_S}(p_j, S))$;

end for

在 TN 和 GN 中的数据点对中筛选出对偶近邻对集 DN ;

在 DN 中筛选出最近对集 CN ;

$AN \leftarrow TN \cup GN - DN$;

/* 计算候选邻近对集 */

return DN, CN, AN ;

end if

VR_SNP 算法中, 针对分布区域不规则且数据点分布密度差异较大的情况利用 Voronoi 图进行计算查询, 本文将该部分的计算表示为 VR_SNP_{voronoi} , 所得的 DN, CN, AN 表示为 $VR_SNP_{\text{voronoi}}(DN, CN, AN)$; 针对分布区域规则且数据点分布较均匀的情况利用 R 树进行计算查询, 本文将该部分的计算表示为 VR_SNP_R , 所得的 DN, CN, AN 表示为 $VR_SNP_R(DN, CN, AN)$. 由于动态数据集的强邻近对查询算法需要用到初始环境下的单向近邻对信息, 故 VR_SNP 算法在返回强邻近对的同时将中间计算所得的候选邻近对集也返回了.

VR_SNP 算法的计算性能受数据分布区域和分布规律的影响较大, R 树在有些情况下构建和查询过程较易实现, 但时间复杂度相对较高; Voronoi

图由于具有优良的空间划分和定位性质,在一些情况下,其计算性能较优,但构建过程和细节计算过程有时不如 R 树简明、便捷。

2.2 数据集动态增加情况下的强邻近对查询

实际应用中,数据集 S 和 P 的规模经常会动态增大,新增点对初始数据集中的强邻近对查询结果往往会产生较大的影响。处理动态数据集的强邻近对查询问题的一个直接方法(本文称为 A_SNP_DA 方法)是在数据集每次动态增加点后,均更新数据集,基于新的数据集再调用 VR_SNP 算法进行查询。在 A_SNP_DA 方法中,由于数据集每次更新后均重建空间索引结构,且需对大量的近邻关系重新计算,冗余计算量较大、效率较低,不适合处理具有频繁更新的大数据集环境中的动态强邻近对查询需求。因此,基于 2.1 节的研究,本节进一步研究处理数据集动态增加情况下的强邻近对查询的有效方法,提出了 VR_SNP_DA 算法。

为了研究新增点 z_{new} 对初始数据集环境下的近邻对的影响情况,首先给出对内元素的定义。

定义 4. 针对近邻对 (s, p) , s 和 p 均称为对内元素,其中 s 称为首位元素, p 称为次位元素。

本文中,对内元素和其对应的近邻对集之间的关系可用符号 $\tilde{\in}$ 表示,形如: $s \tilde{\in} (s, p)$; 若一个数据元素和某近邻对集没有关系,则用符号 $\notin$ 表示,形如: $q \notin (s, p)$ 。

由强邻近对的定义可知,对于单向近邻对,次位元素是首位元素的最近邻,首位元素不是次位元素的最近邻;对于强邻近对,首位元素和次位元素互为最近邻。

针对初始数据集环境下查询得到的单向近邻对 (s, p) 或 (p, s) , 对偶近邻对 $\langle s, p \rangle$ 和最近对 $[s, p]$, 新增点 z_{new} 的加入往往会对其中的对内元素产生影响。设新增点 z_{new} 加入到集合 S (加入到集合 P 可类似讨论), 具体分为如下 4 种情况:

情况 1. z_{new} 的加入可使某个或某些单向近邻对 (p, s) 改变为 (p, z_{new}) 或 $\langle p, z_{\text{new}} \rangle$ 或 $[p, z_{\text{new}}]$, 即 z_{new} 成为 p 的新的最近邻;

情况 2. z_{new} 的加入可使某个或某些对偶近邻对 $\langle s, p \rangle$ 改变为 $\langle z_{\text{new}}, p \rangle$ 或 (z_{new}, p) 或 (p, z_{new}) 或 $[z_{\text{new}}, p]$;

情况 3. z_{new} 的加入可使某个或某些最近对 $[s, p]$ 改变为 $[z_{\text{new}}, p]$ 或 $\langle z_{\text{new}}, p \rangle$ 或 (z_{new}, p) 或 (p, z_{new}) ;

情况 4. z_{new} 的加入可生成新的单向近邻对 (z_{new}, p) 或 $\langle z_{\text{new}}, p \rangle$ 或 $[z_{\text{new}}, p]$ 。

情况 1~3 主要是新增点对初始单向近邻对和强邻近对的影响情况; 情况 4 是新增点加入后并没影响初始单向近邻对和强邻近对, 但以其为首位点形成了新的近邻对。实际处理中, 可分情况进行强邻近对的查询和更新, 而不用在每次数据集变化时都对整个数据集重新计算。

由此可得出 VR_SNP_DA 算法的主要思路如下:

1) 针对初始的集合 S 和 P 调用 VR_SNP 算法生成初始数据集中的强邻近对和候选邻近对;

2) 确定新增数据点 z_{new} 所属的数据集是 S 还是 P ;

3) 若 VR_SNP 算法是利用 Voronoi 图查询出初始强邻近对和候选邻近对, 则进一步执行以下步骤:

① 以新增点 z_{new} 为生成点, 对 z_{new} 归属的数据集(S 或 P) 的 Voronoi 图进行局部重构;

② 利用 Voronoi 图的性质计算判断 z_{new} 加入后对初始数据集中近邻对的影响情况;

③ 根据步骤②的计算结果更新强邻近对集。

4) 若 VR_SNP 算法是利用 R 树空间索引结构查询出初始强邻近对和候选邻近对, 则针对新增点 z_{new} 进一步执行以下步骤:

① 利用 R 树查询 z_{new} 的最近邻和反向最近邻;

② 利用步骤①的计算结果信息与初始单向近邻对, 强邻近对进行比较, 更新强邻近对集。

具体的 VR_SNP_DA 算法如下:

算法 2. VR_SNP_DA(S, P, Z).

输入: 数据点集 S, P 和新增点集 Z ;

输出: 新的对偶近邻对集 DN' 、最近对集 CN' 、候选邻近对集 AN' 。

$DN' \leftarrow \emptyset$; $CN' \leftarrow \emptyset$; $AN' \leftarrow \emptyset$; /* 初始化 */
 $\{DN, CN, AN\} \leftarrow \text{VR_SNP}(S, P)$;

/* 调用 VR_SNP 算法生成初始的对偶近邻对集 DN , 最近对集 CN , 候选邻近对集 AN */
 for $i=1$ to $|Z|$ do

 判断 $z_{\text{new}i}$ 所属的数据集;

 if VR_SNP_{Voronoi}(DN, CN, AN) = TRUE

 then

 Re_construct Voronoi($z_{\text{new}i}$);

 /* 局部更新 */

 end if

 for $j=1$ to $|P|$ do

 if $(z_{\text{new}i} \in VP(p_j)) \wedge (p_j \in VP(z_{\text{new}i}))$

then

$DN_1 \leftarrow \langle p_j, z_{newi} \rangle;$

/ * 判断对偶近邻对 * /

if $p_j \in \{DN, CN, AN\}$ then

动态调整 $\{DN, CN, AN\};$

生成新的 $DN', CN', AN';$

return $DN', CN', AN';$

end if

if $p_j \notin \{DN, CN, AN\}$

/ * 判断单向近邻对 * /

then

$AN_1 \leftarrow \langle p_j, z_{newi} \rangle;$

$AN_2 \leftarrow \langle z_{newi}, p_j \rangle;$

动态调整 $\{DN, CN, AN\};$

生成新的 $DN', CN', AN';$

return $DN', CN', AN';$

end if

end if

end for

if $VR_SNP_R(DN, CN, AN) = \text{TRUE}$ then

$AN_1 \leftarrow \langle z_{newi}, RNN_R(z_{newi}, P) \rangle;$

$AN_2 \leftarrow \langle RNN_R(z_{newi}, P), z_{newi} \rangle;$

if $RNN_R(z_{newi}, P) \in RNN_R(z_{newi}, P)$ or

$RNN_R(z_{newi}, P) \in \{DN, CN, AN\}$ or

$RNN_R(z_{newi}, P) \in \{DN, CN, AN\}$ then

动态调整 $\{DN, CN, AN\};$

生成新的 $DN', CN', AN';$

return $DN', CN', AN';$

end if

end if

end for

当获得初始数据集环境下的强邻近对后, VR_SNP_DA 算法的执行效率主要取决于算法针对新增点对强邻近对的影响情况的判断和强邻近对集的动态更新的计算量。

2.3 数据集动态减少情况下的强邻近对查询

2.2 节给出了数据集动态增加情况下的强邻近对查询算法, 本节进一步研究数据集动态减少情况下的强邻近对查询方法。与 2.1 节首先给出的 A_SNP_DA 方法类似, 本节首先也给出处理数据集动态减少情况下的查询问题的一个直接方法(本文称为 D_SNP_DE 方法)。D_SNP_DE 方法的基本思想

是在数据集每次动态减少点后, 均更新数据集, 基于新的数据集再调用 VR_SNP 算法进行查询。但由于 D_SNP_DE 方法在执行过程中每次均需重建索引结构, 同时需执行大量冗余计算, 效率较低。为此, 本节进一步提出 VR_SNP_DE 算法。

针对初始数据集环境下查询得到的单向近邻对 $\langle s, p \rangle$ 或 $\langle p, s \rangle$, 对偶近邻对 $\langle s, p \rangle$ 和最近对 $[s, p]$, 数据点 z_{del} 的减少经常会对其中的对内元素产生影响。VR_SNP_DE 算法的核心思想如下:

1) 针对初始的数据集 S 和 P 调用 VR_SNP 算法生成初始数据集中的强邻近对和候选邻近对;

2) 确定删除点 z_{del} 所属的数据集是 S 还是 P ;

3) 确定删除点 z_{del} 所属的初始数据集中的近邻对的类型;

4) 若删除点 z_{del} 属于 S (属于 P 可做类似判断), 则遍历初始对偶近邻对集 DN , 最近对集 CN 和候选邻近对集 AN ; 判断删除点 z_{del} 对 DN, CN 和 AN 的具体影响 2 种情况:

(1) 情况 1. 若 z_{del} 是 AN 中某些候选邻近对的对内元素, 则进行以下操作:

① 若 $\langle z_{del}, p \rangle \in AN$, 则将 $\langle z_{del}, p \rangle$ 直接删除, 更新 AN ;

② 若 $\langle p, z_{del} \rangle \in AN$, 则查询 p 的新的最近邻 p' , 将 $\langle p, p' \rangle$ 加入 AN ; 在 AN 中查找是否包含 $\langle p', p \rangle$, 若包含, 则将 $\langle p, p' \rangle$ 加入对偶近邻对集 DN ;

③ 将 $\langle p, p' \rangle$ 和初始最近对集 CN 中的元素比较, 确定新的 CN 。

(2) 情况 2. 若 z_{del} 是集合 DN 或 CN 中某些元素的对内元素, 则进行以下操作:

① 若 $\langle p, z_{del} \rangle \in DN$, 将 $\langle p, z_{del} \rangle$ 从 DN 删除; 查询 p 的新的最近邻 p' , 将 $\langle p, p' \rangle$ 加入 AN ; 在 AN 中查找是否包含 $\langle p', p \rangle$, 若包含, 则将 $\langle p, p' \rangle$ 加入对偶近邻对集 DN ; 进一步确定新的 CN 。

② 若 $[p, z_{del}] \in CN$, 将 $[p, z_{del}]$ 从 CN 中删除; 查询 p 的新的最近邻 p' , 将 $\langle p, p' \rangle$ 加入 AN ; 在 AN 中查找是否包含 $\langle p', p \rangle$, 若包含, 则将 $\langle p, p' \rangle$ 加入对偶近邻对集 DN ; 进一步确定新的 CN 。

具体的 VR_SNP_DE 算法如下:

算法 3. VR_SNP_DE(S, P, Z).

输入: 数据点集 S, P 和删除点集 Z ;

输出: 新的对偶近邻对集 DN' 、最近对集 CN' 、候选邻近对集 AN' 。

$DN' \leftarrow \emptyset; CN' \leftarrow \emptyset; AN' \leftarrow \emptyset; / * \text{初始化} * /$

$\{DN, CN, AN\} \leftarrow VR_SNP(S, P);$

```
/* 调用 VR_SNP 算法生成初始的对偶近邻对
   集  $DN$ , 最近对集  $CN$ , 候选邻近对集  $AN$  */
for  $i=1$  to  $|Z|$  do
    判断  $z_{deli}$  所属的数据集;
    遍历集合  $AN, DN, CN$ ;
    if  $(z_{deli}, p_j) \in AN$  then
         $AN' \leftarrow AN - (z_{deli}, p_j)$ ;
    end if
    if  $(p_j, z_{deli}) \in AN$  then
         $AN' \leftarrow AN - (p_j, z_{deli})$ ;
L1:  $p'_j \leftarrow NN(p_j)$ ;
         $AN' \leftarrow (p_j, p'_j)$ ;
        if  $(p'_{p_j}, ) \in AN'$  then
             $DN \leftarrow (p_j, p'_j)$ ;  $DN' \leftarrow DN$ ;
            在  $DN'$  中计算  $CN'$ ;
        end if
    end if
    if  $(z_{deli}, p_j) \in DN$  then
         $DN' \leftarrow DN - (z_{deli}, p_j)$ ; 转 L1;
    end if
    if  $(z_{deli}, p_j) \in CN$  then
         $CN' \leftarrow CN - (z_{deli}, p_j)$ ; 转 L1;
    end if
return  $DN', CN', AN'$ .
end for
```

当获得初始数据集环境下的候选邻近对和强邻近对后, VR_SNP_DE 算法针对删除点对强邻近对的影响情况的判断和强邻近对集的动态更新的计算量是影响该算法的执行效率的核心因素。

2.4 移动数据点的强邻近对查询

数据集中数据点的空间位置发生动态改变是数据集动态变化的一个重要方面. 例如超市、修理厂、救护车等空间对象的位置随时间往往会发生动态改变. 本节首先给出处理该问题的一个直接方法(本文称为 M_SNP_DL 方法), 其基本思想是在每个时刻点确定数据集的数据点的空间位置, 针对每个时刻点的数据集分别调用 VR_SNP 算法进行查询. 但由于 M_SNP_DL 方法在每个时刻点进行查询时均需重建整个数据集的空间索引结构, 每次查询均执行大量冗余计算, 没有充分利用已有的查询信息, 查询效率较低. 为此, 本节基于 Voronoi 图进一步提出 D_SNP_DL 算法. 为便于讨论, 本节着重研究数据集 P 中单个数据点位置变化情况下的查询算法, S 中的数据点和多个数据点的空间位置动态变化时可

类似处理.

针对初始数据集环境下查询得到的单向近邻对 $\langle s, p \rangle$ 或 $\langle p, s \rangle$, 对偶近邻对 $\langle s, p \rangle$ 和最近对 $[s, p]$, 数据点的位置的动态变化往往会对其中的对内元素产生影响. VD_SNP_DL 算法的核心思想如下:

1) 针对初始时刻 t_0 的 S 和 P , 调用 VR_SNP 算法生成时刻 t_0 的强邻近对和候选邻近对;

2) 确定时刻 t_1 位置发生变化的数据点 z_{mov} 与其新的空间位置;

3) 基于 Voronoi 图判定数据对象点 z_{mov} 在新的空间位置下在 S 中的最近邻集 NN 和反向最近邻集 RNN ;

4) 确定 NN 和 RNN 的交集 JZ ;

5) 针对 JZ , 分 4 种情况进行判定:

① 若 $JZ = \emptyset$, 且 z_{mov} 是时刻 t_0 的某对偶近邻对 dn 或最近对 cn 中的对内元素, 则将 dn 或 cn 从时刻 t_0 的强邻近对结果中删除. 返回更新后的强邻近对.

② 若 $JZ = \emptyset$, 且 z_{mov} 不是时刻 t_0 的某对偶近邻对 dn 或最近对 cn 中的对内元素, 则数据集在 t_0 时刻的强邻近对集即为 t_1 时刻的强邻近对集, 返回 DN 和 CN ;

③ 若 $JZ \neq \emptyset$, 且 z_{mov} 是时刻 t_0 的某对偶近邻对 dn 或最近对 cn 中的对内元素, 则进一步判断 dn 和 cn 中的另一个对内元素是否属于 JZ . 若该对内元素不属于 JZ , 则从 DN 和 CN 中删除 dn 和 cn , 将 JZ 中的各元素与 z_{mov} 组成新的对偶近邻对 DN 和 CN , 更新和计算 DN 和 CN , 返回新的结果集; 若该对内元素属于 JZ , 将 JZ 中的各元素与 z_{mov} 组成新的对偶近邻对 DN 和 CN , 更新和计算 DN 和 CN , 消除重复的近邻对, 返回新的结果集;

④ 若 $JZ \neq \emptyset$, 且 z_{mov} 不是时刻 t_0 的某对偶近邻对 dn 或最近对 cn 中的对内元素, 则将 JZ 中的各元素与 z_{mov} 组成新的对偶近邻对 DN 和 CN , 更新和计算 DN 和 CN , 返回新的结果集.

具体的 VR_SNP_DL 算法如下:

算法 4. VR_SNP_DL(S, P, z_{mov}).

输入: 数据点集 S, P 和移动点 z_{mov} ;

输出: 新的对偶近邻对集 DN' 、最近对集 CN' 、候选邻近对集 AN' .

$DN' \leftarrow \emptyset$; $CN' \leftarrow \emptyset$; $AN' \leftarrow \emptyset$; /* 初始化 */
 $\{DN, CN, AN\} \leftarrow VR_SNP(S, P)$;

/* 调用 VR_SNP 算法生成时刻 t_0 的对偶近邻对集 DN , 最近对集 CN , 候选邻近对集 AN */

```

确定  $z_{\text{mov}}$  在时刻  $t_1$  的新的空间位置;
 $NN \leftarrow NN(z_{\text{mov}}); RNN \leftarrow RNN(z_{\text{mov}});$ 
 $JZ \leftarrow NN \cap RNN;$ 
遍历  $DN$  和  $CN$  中的近邻对  $dn$  和  $cn$ ;
if  $JZ = \emptyset$  and  $(z_{\text{mov}} \in dn \text{ or } z_{\text{mov}} \in cn)$  then
     $DN' \leftarrow DN - dn \text{ or } CN' \leftarrow CN - cn;$ 
    return  $DN', CN'$ ;
end if

if  $JZ = \emptyset$  and  $(z_{\text{mov}} \notin dn \text{ and } z_{\text{mov}} \notin cn)$  then
     $DN' \leftarrow DN, CN' \leftarrow CN;$ 
    return  $DN', CN'$ ;
end if

if  $JZ \neq \emptyset$  and  $(z_{\text{mov}} \in dn \text{ or } z_{\text{mov}} \in cn)$  then
    确定  $dn$  和  $cn$  中的另一个对内元素  $z'$ ;
    if  $z' \in JZ$  then
         $DN' \leftarrow DN - dn \text{ or } CN' \leftarrow CN - cn;$ 
L1: 遍历  $JZ$  中的数据点  $z_{jz}$ ;
         $DN' \leftarrow [z_{jz}, z_{\text{mov}}];$  更新  $CN'$ ;
        return  $DN', CN'$ ;
    end if
    if  $z' \in JZ$  then
        转 L1;
    end if
end if

if  $JZ \neq \emptyset$  and  $(z_{\text{mov}} \notin dn \text{ or } z_{\text{mov}} \notin cn)$  then
    遍历  $JZ$  中的数据点  $z_{jz}$ ;
     $DN' \leftarrow [z_{jz}, z_{\text{mov}}];$  更新  $CN'$ ;
    return  $DN', CN'$ ;
end if

```

3 方法分析和实验比较

为了处理动态数据集中的强邻近对查询问题,第2节分别给出了 VR_SNP 算法、VR_SNP_DA 算法、VR_SNP_DE 算法和 VR_SNP_DL 算法.本节进一步进行方法分析和实验比较.

3.1 方法分析

强邻近对查询是最近邻查询领域的一个新问题.本文着重研究在动态数据集中查询强邻近对(即互为最近邻的点对和最近对)的方法,本文的研究工作和已有的研究工作主要有以下2个区别:

1) 针对的问题不同.本文着重研究在2个动态

数据集中查询强邻近对,返回互相为最近邻的点对和最近对,邻近关系具有“双向”性,数据集具有动态性.已有的研究工作则主要是研究数据对象的“单向”邻近关系.例如文献[3-16]主要是查询满足一定条件的数据点的最近邻,不再需要考虑该查询点是否是其最近邻的最近邻,处理和分析的数据点间的邻近关系相对强邻近对间的关系较“弱”.

2) 处理过程中,数据对象点的“地位”不同.本文研究的强邻近对查询问题主要是在2个动态数据集中查询互相为最近邻的点对和最近对,需要全面考虑分析动态数据集中每对可能满足条件的数据点对,查询过程中没有特定的查询点.已有的最近邻查询问题主要是围绕特定的查询点进行处理,主要是寻求给定的查询点和数据集中其他数据点的邻近关系,不需要考虑每个数据点和其他数据点的邻近关系.

以上的主要区别使得处理动态数据集中的强邻近对查询的方法和通常最近邻查询方法有着显著不同,已有的方法不能直接用来查询动态数据集中的强邻近对.将已有方法间接用于强邻近对的查询的思路为:根据强邻近对查询的定义,为了准确获取动态数据集中的互相为最近邻的点对和最近对,需要针对2个数据点集中的每个数据点查询其最近邻和反向最近邻,查询最近邻和反向最近邻的方法可利用已有研究成果(例如文献[4]的算法),然后再进行判断筛选,最终得到满足条件的强邻近对.显然,用该方法查询强邻近对时需要计算查询2个数据集中所有数据点的最近邻,属于“蛮力”求解,在数据集规模较大时,计算代价非常大.本文所提出的 VR_SNP_DA 算法主要用来解决数据集中的数据点动态增加情况下的强邻近对查询问题,VR_SNP_DE 算法主要解决数据集中的数据点动态减少情况下的强邻近对查询问题,VR_SNP_DL 算法主要解决数据集中数据点的位置动态移动情况下的强邻近对查询问题.利用本文的算法可直接完成多种动态条件下的强邻近对查询任务,避免了较多的冗余计算.

3.2 实验比较

本节在 Pentium4、CPU 1.8 GHz、内存 8 GB、Windows XP 环境下,利用 C++ builder 6.0 等软件工具,基于空间数据信息(来源于 <http://freegisdata.rtwilson.com/> 网站提供的开源数据集 DIVA-GIS Country Data)对算法性能进行了实验比较和分析.

由 3.1 节可知,已有的方法不能直接用来查询动态数据集中的强邻近对.对动态数据集中的强邻

近对查询,尚没有针对该问题的已有研究成果,故无法直接在特定动态数据集上进行比较. 为了进一步分析本文方法的效率,本节着重分析将已有方法间接用于强邻近对的查询的方法(本文称为 JSNP 方法)和本文所提方法的比较情况. JSNP 方法主要是:根据强邻近对查询的定义,针对 2 个数据对象集中的每个数据点,利用已有最近邻查询的研究成果^[1-2]的方法逐一查询其最近邻和反向最近邻,根据数据集的动态变化再反复调用已有算法进行查询更新,对查询结果再进行判断筛选,最终得到满足条件的强邻近对.

为便于实验,我们将所获得的空间数据信息整合成 2 个主要的数据集:路网结点数据信息集(简称集合 RN)和人口对象分布数据集(简称集合 PD). 集合 RN 和集合 PD 包含的数据对象数目在实验中可增删调整. 集合 RN 包含的数据点数表示为 n ,集合 PD 包含的数据点数表示为 m . 用 RZ_{new} 表示新增对象集,所包含的新增数据对象的数目为 t ,用 RZ_{del} 表示删除对象集,所包含的删除点的数目为 r ,移动对象点位置改变的次数为 w . δ 表示在不同条件下本文所提的算法相对于 JSNP 方法的效率比较结果.

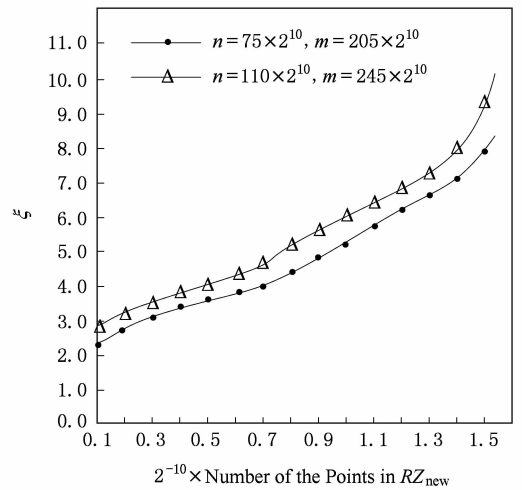
图 2 主要展示了 3 种典型情况下的实验结果:

情况 1. VR_SNP_DA 算法相对于 JSNP 方法计算效率的比较.(图 2(a)中, n, m 分别取 2 组数, $(n=75 \times 2^{10}, m=205 \times 2^{10})$, $(n=110 \times 2^{10}, m=245 \times 2^{10})$,纵坐标 ξ 表示 VR_SNP_DA 算法相对于 JSNP 方法查询效率的比率;横坐标表示数据集 RZ_{new} 包含的数据量).

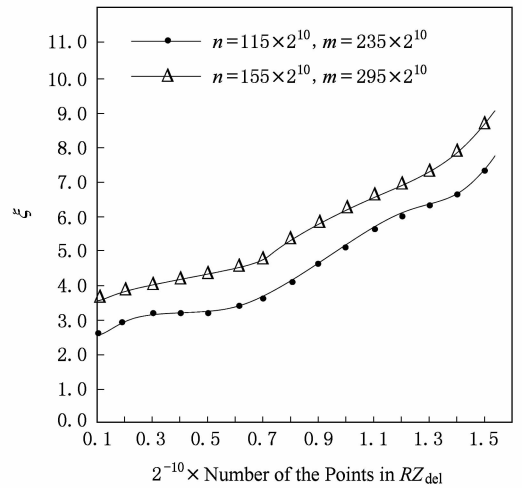
情况 2. VR_SNP_DE 算法相对于 JSNP 方法计算效率的比较.(图 2(b)中, n, m 分别取 2 组数, $(n=115 \times 2^{10}, m=235 \times 2^{10})$, $(n=155 \times 2^{10}, m=295 \times 2^{10})$,纵坐标 ξ 表示 VR_SNP_DE 算法相对于 JSNP 方法查询效率的比率;横坐标表示数据集 RZ_{del} 包含的数据量).

情况 3. VR_SNP_DL 算法相对于 JSNP 方法查询效率的比较.(图 2(c)中, n, m 分别取 2 组数, $(n=135 \times 2^{10}, m=255 \times 2^{10})$, $(n=215 \times 2^{10}, m=375 \times 2^{10})$,纵坐标 ξ 表示 VR_SNP_DL 算法相对于 JSNP 方法查询效率的比率;横坐标表示移动对象点的位置改变的次数).

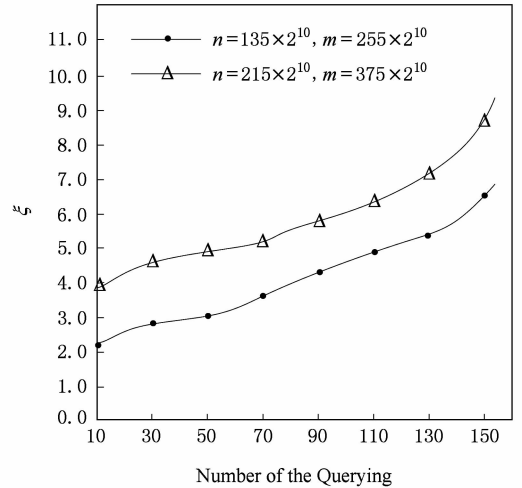
图 2(a)中,曲线分别表示在 n, m 分别取 2 组数 $((n=75 \times 2^{10}, m=205 \times 2^{10}), (n=110 \times 2^{10}, m=245 \times 2^{10}))$ 的情况下,随着 t 的逐渐增大, VR_SNP_



(a) Performance comparison of VR_SNP_DA algorithm and JSNP method



(b) Performance comparison of VR_SNP_DE algorithm and JSNP method



(c) Performance comparison of VR_SNP_DL algorithm and JSNP method

Fig. 2 Performance comparison.

图 2 性能比较

DA 算法相对于 JSNP 方法计算效率的比较情况. 由图 2(a)可知, VR_SNP_DA 算法比 JSNP 方法具有较高的计算效率, 随着新增点数目 t 增多, 也即更新次数增多, VR_SNP_DA 算法相对于 JSNP 方法计算效率的优势将明显增大. 图 2(b)中, 曲线分别表示在 n, m 分别取 2 组数 ($(n=115 \times 2^{10}, m=235 \times 2^{10}), (n=155 \times 2^{10}, m=295 \times 2^{10})$) 的情况下, 随着 r 的逐渐增大, VR_SNP_DE 算法相对于 JSNP 方法计算效率的比较情况. 由图 2(b)可知, VR_SNP_DE 算法比 JSNP 方法具有较高的计算效率, 删除对象集 RZ_{de} 的规模越大, VR_SNP_DE 算法相对于 JSNP 方法计算效率的优势将越明显. 图 2(c)中, 曲线分别表示在 n, m 分别取 2 组数 ($(n=135 \times 2^{10}, m=255 \times 2^{10}), (n=215 \times 2^{10}, m=375 \times 2^{10})$) 的情况下, 随着移动对象点的位置改变的次数逐渐增大, VR_SNP_DL 算法相对于 JSNP 方法计算效率的比较情况. 由图 2(c)可知, VR_SNP_DL 算法比 JSNP 方法具有较高的计算效率, 移动对象位置改变的次数越多, VR_SNP_DL 算法相对于 JSNP 方法的平均计算效率的优势将越明显.

由于 VR_SNP_DA 算法、VR_SNP_DE 算法和 VR_SNP_DL 算法在获得初始强邻近对时均调用了本文提出的 VR_SNP 算法, 有效避免了 JSNP 方法利用已有最近邻查询的研究成果逐一查询 2 个数据集中每个数据对象的最近邻和反向最近邻所产生的冗余计算问题, 从而使得针对动态数据集, 本文的方法在不同情况下的查询优势较为明显.

4 结束语

数据集中的强邻近对查询问题在空间数据查询、空间数据挖掘、模式识别、数据相似分析、智能查询等方面具有重要的作用. 动态数据集中的强邻近对查询是一个新的复杂查询问题, 为了有效地处理动态数据集环境下的强邻近对查询问题, 我们对动态数据集中的强邻近对查询问题进行了探索, 提出了基于 Voronoi 图和 R 树的 VR_SNP 算法, 进一步研究了数据集动态增加和动态减少情况下的强邻近对查询方法, 通过对初始强邻近对集和候选邻近对集进行二次判断计算, 筛选出有效结果, 分别给出了 VR_SNP_DA 算法和 VR_SNP_DE 算法. 理论研究和实验表明数据集的数据量, 新增点集、删除点集的规模和移动对象点的移动次数较大时, VR_SNP 算法、VR_SNP_DA 算法、VR_SNP_DE 和 VR_

SNP_DL 算法具有较为明显的查询优势. 未来的研究重点主要集中在以下 2 个方面:

- 1) 不确定区域(例如 Vague 区域^[19])内的强邻近对查询问题.
- 2) 高维空间中的强邻近对查询问题.

参 考 文 献

- [1] Hao Zhongxiao. Query and Reasoning in Spatio-Temporal Database [M]. Beijing: Science Press, 2010 (in Chinese)
(郝忠孝. 时空数据库查询与推理[M]. 北京: 科学出版社, 2010)
- [2] Li Song, Zhang Liping, Sun Dongpu. Spatial Relation Query and Analysis [M]. Harbin: Harbin Institute of Technology Press, 2011(in Chinese)
(李松, 张丽平, 孙冬璞. 空间关系查询与分析[M]. 哈尔滨: 哈尔滨工业大学出版社, 2011)
- [3] Papadias D, Shen Q, Tao Y, et al. Group nearest neighbor queries [C] //Proc of the 20th Int Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society, 2004: 301-312
- [4] Tao Yufei, Yiu M L, Mamoulis N. Reverse nearest neighbor search in metric spaces [J]. IEEE Trans on Knowledge Data Engineering, 2006, 18(9): 1239-1252
- [5] Gao Y, Zheng B. Continuous obstructed nearest neighbor queries in spatial databases [C] // Proc of the Int Conf on Management of Data and the 28th Symp on Principles of Database Systems. New York: ACM, 2009: 577-589
- [6] Zhang Y, Lin X, Zhu G, et al. Efficient rank based kNN query processing over uncertain data [C] //Proc of the 26th Int Conf on Data Engineering (ICDE 2010). Piscataway, NJ: IEEE, 2010: 28-39
- [7] Sarana N, Eqemen T, Zhang Rui. Incremental evaluation of visible nearest neighbor queries [J]. IEEE Trans on Knowledge Engineering, 2010, 22(4): 665-681
- [8] Gao Yunjun, Zheng Baihua, Chen Gencai, et al. Visible reverse k -Nearest neighbor query processing in spatial databases [J]. IEEE Trans on Knowledge Engineering, 2009, 21(8): 1314-1327
- [9] Yuan Peisen, Sha Chaofeng, Wang Xiaoling, et al. c -Approximate nearest neighbor query algorithm based on learning for high dimensional data [J]. Journal of Software, 2012, 23(8): 2018-2031 (in Chinese)
(袁培森, 沙朝锋, 王晓玲, 等. 一种基于学习的高维数据 c -近似最近邻查询算法[J]. 软件学报, 2012, 23(8): 2018-2031)
- [10] Miao Dongjing, Shi Shengfei, Li Jianzhong. An algorithm on probabilistic frequent nearest neighbor query over snapshots of uncertain database with locally correlation [J]. Journal of Computer Research and Development, 2011, 48(10): 1812-1822 (in Chinese)

- (苗东菁, 石胜飞, 李建中. 一种局部相关不确定数据库快照集合上的概率频繁最近邻算法[J]. 计算机研究与发展, 2011, 48(10): 1812-1822)
- [11] Zhang Xu, He Xiangnan, Jin Cheqing. Processing k -Nearest neighbors query over uncertain graphs [J]. Journal of Computer Research and Development, 2011, 48(10): 1871-1878 (in Chinese)
- (张旭, 何向南, 金澈清. 面向不确定图的 k 最近邻查询[J]. 计算机研究与发展, 2011, 48(10): 1871-1878)
- [12] Li Song, Hao Zhongxiao. Reverse nearest neighbor queries using voronoi diagram [J]. Journal of Harbin Engineering University, 2008, 29(3): 261-265 (in Chinese)
- (李松, 郝忠孝. 基于 Voronoi 图的反向最近邻查询方法研究[J]. 哈尔滨工程大学学报, 2008, 29(3): 261-265)
- [13] Sun Dongpu, Hao Zhongxiao. Group nearest neighbor queries based on Voronoi diagram [J]. Journal of Computer Research and Development, 2010, 47(7): 1244-1251 (in Chinese)
- (孙冬璞, 郝忠孝. 基于 Voronoi 图的组最近邻查询[J]. 计算机研究与发展, 2010, 47(7): 1244-1251)
- [14] Hao Zhongxiao, Wang Yudong, He Yunbin. Line segment nearest neighbor query of spatial database [J]. Journal of Computer Research and Development, 2008, 45(9): 1539-1545 (in Chinese)
- (郝忠孝, 王玉东, 何云斌. 空间数据库平面线段近邻查询问题研究[J]. 计算机研究与发展, 2008, 45(9): 1539-1545)
- [15] Zhang Liping, Li Song, Li Lin, et al. Simple continues near neighbor chain query of the datasets based on the R tree [J]. Journal of Computational Information Systems, 2012, 8(22): 9159-916
- [16] Li Song Zhang Liping, Hao Xiaohong. Simple continues near neighbor chain query with obstacles [J]. Journal of Computational Information Systems, 2013, 9(19): 7593-7601
- [17] Zhang Liping, Li Song. New methods of strong neighborhood pair query [J]. Computer Engineering and Applications, 2009, 45(27): 123-127 (in Chinese)
- (张丽平, 李松. 强邻近对查询的新方法[J]. 计算机工程与应用, 2009, 45(27): 123-127)
- [18] Sacl J R, Urrutia J. Voronoi Diagrams. Handbook on Computational Geometry [M]. Ottawa: Elsevier Science, 2010: 201-290
- [19] Li Song, Hao Zhongxiao. Region relations of the irregular vague regions with holes based on vague sets [J]. Journal of Computer Research and Development, 2009, 46(5): 823-831 (in Chinese)
- (李松, 郝忠孝. 基于 Vague 集的含洞不规则 Vague 区域关系[J]. 计算机研究与发展, 2009, 46(5): 823-831)



Li Song, born in 1977. PhD. Associate professor in the College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China. His research interests include spatial and spatio-temporal database, data mining, information data security.



Zhang Liping, born in 1976. Master. Associate professor in the College of Computer Science and Technology, Harbin University of Science and Technology. Her research interests include spatial and spatio-temporal database, information data security.



Hao Zhongxiao, born in 1940. Received his BSc in the Department of Mathematics from Heilongjiang University in 1964. Professor and PhD supervisor in the Department of Computer Science and Technology, Harbin Institute of Technology and Harbin University of Science and Technology. His research interests include relational database, null database, acyclic database, active database, spatial database and temporal database.