

# Voronoi 图的构建与受限区域内的最近邻查询方法研究

张丽平 赵纪桥 李松 经海东 崔环宇

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

**摘要** Voronoi 图在空间数据查询、数据挖掘、图像处理、模式识别和智能交通管理等方面具有重要的作用。为了简化构建的复杂性和提高构建效率,基于分治法、启发式局部优化策略和局部数据点的扫描线动态更新策略,提出了基于凸包的 Voronoi 图生成方法,给出了 Create\_Voronoi() 算法。进一步,为了弥补已有近邻查询方法无法处理受限区域内的最近邻查询的不足,基于 Voronoi 图研究了受限区域内的同质和异质最近邻查询方法,分别提出了 TVor\_NN() 算法和 YVor\_NN() 算法。理论研究和实验分析表明,提出的研究方法在 Voronoi 图的构建和受限范围的最近邻查询等方面具有较大的优势。

**关键词** Voronoi 图, Delaunay 三角形, 最近邻查询, 受限区域

**中图分类号** TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.041

## Research on Methods of Construction of Voronoi Diagram and Nearest Neighbor Query in Constrained Regions

ZHANG Li-ping ZHAO Ji-qiao LI Song JING Hai-dong CUI Huan-yu

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

**Abstract** The Voronoi diagram plays an important role in spatial data query, data mining, image processing, pattern recognition and intelligent traffic management etc. To reduce complexity and improve efficiency, based on the divide and conquer, local optimization and dynamic update of scan lines, the method to construct Voronoi diagram using the convex hulls was proposed. The Create\_Voronoi() algorithm was given. Furthermore, to remedy the defects of the existing methods for the near neighbor query, the identical-quality nearest neighbor query and the heterogeneous nearest neighbor query in the constrained regions were studied. The TVor\_NN() algorithm and the YVor\_NN() algorithm were presented. The theoretical study and the experimental results show that the redundant calculation is reduced and the algorithms hold large advantage at the construction of Voronoi diagram and the nearest neighbor query in constrained regions.

**Keywords** Voronoi diagram, Delaunay triangle, Nearest neighbor query, Constrained region

## 1 引言

Voronoi 图是计算几何领域中的一个重要分支, Voronoi 图的一些重要性质如近邻性质、最大空圆性质、控制范围性质等,使其在实际生产和科学研究领域受到了较多的关注。针对复杂的数据集, Voronoi 图在查找最近点、求最大空圆、求  $n$  个点的凸包、求最小树等问题方面具有重要作用。此外, Voronoi 图与其对偶图 Delaunay 三角剖分已被广泛地应用到几何形体重构、空间数据聚类、空间数据挖掘、图像处理、移动通信、模式识别、机器人运动规划、方向关系分析等领域<sup>[1]</sup>。在 GIS 和卫星定位与遥感领域, Voronoi 图可表达空间数据信息中的侧向邻近关系,并兼具矢量和连续铺盖数据模型的基本特点,可以很好地管理空间数据,因此成为 GIS 等领域近年的研究热点之一<sup>[2,3]</sup>。

根据构建过程的不同,平面点集的 Voronoi 图构造算法一般可划分为<sup>[3]</sup>:半平面的交、增量构造方法、分治法、减量算法、平面扫描算法等;其对偶图 Delaunay 三角网的算法可分为逐点插入法、三角网生长法和分治法等。近年来,国内外相继提出了 Voronoi 图的并行生成方法<sup>[4]</sup>和增删点情况下的生成方法<sup>[5]</sup>等。已有的 Voronoi 图生成方法能构建出对应的空间点集的 Voronoi 图,但其数据结构往往较为复杂,构建过程繁琐,计算难度较大,且具有一定的计算误差。为了弥补已有方法的不足,本文将数据点集凸包技术和 Delaunay 三角网络技术相融合,充分利用两者的优良性质构建 Voronoi 图,构建程序中利用了分治法、启发式局部优化策略和局部数据点的扫描线动态更新策略,提高了构建效率,减少了计算误差。

随着信息社会数据量“爆炸”式的增长,海量空间数据的有效查询、聚类和分析技术受到了越来越多的重视。利用

到稿日期:2013-11-30 返修日期:2014-02-24 本文受黑龙江省教育厅科学技术研究项目(12541122)资助。

张丽平(1976—),女,硕士,副教授,主要研究方向为数据库理论与技术、算法设计, E-mail: zhanglptg@163.com; 赵纪桥(1989—),男,硕士生,主要研究方向为数据结构和算法设计、数据库; 李松(1977—),男,博士,副教授,主要研究方向为数据结构和算法设计; 经海东(1989—),男,硕士生,主要研究方向为数据库理论及应用、算法设计与实现; 崔环宇(1988—),男,硕士生,主要研究方向为数据库理论及应用、算法设计与实现。

Voronoi 图进行空间数据的查询和分析是近年来的一个研究热点,例如基于 Voronoi 图的反向最近邻查询<sup>[6]</sup>、线段最近邻查询<sup>[7]</sup>和移动环境下的范围最近邻查询<sup>[8]</sup>等。在数据的查询领域,空间数据的近邻查询是一个重要的内容,国内外学者对数据集中的近邻关系查询进行了研究,在简单近邻查询、最近对查询、邻近对查询、组最近邻查询、反向最近邻查询等方面取得了一些重要的研究成果<sup>[1,2]</sup>。近年来,由于数据信息的复杂性和查询需求的不断提高,查询条件变得日益复杂,有关近邻关系的查询进一步扩展到可视最近邻查询<sup>[9]</sup>、可视反向  $k$  最近邻查询<sup>[10]</sup>、聚集最近邻查询<sup>[11,12]</sup>、路网双色连续反向  $k$  近邻查询<sup>[13]</sup>、概率频繁最近邻查询<sup>[14]</sup>、高维近似最近邻查询<sup>[15]</sup>、连续近邻链查询<sup>[16,17]</sup>和球面最近邻查询<sup>[18]</sup>等方面。实际查询中,在受限区域内查询数据点的最近邻问题是一个新问题,目前,已有的近邻查询方法无法有效地处理受限区域内的最近邻查询问题。为了弥补已有方法的不足,我们利用所生成的 Voronoi 图进一步给出了受限区域内的同质和异质最近邻查询方法,分别提出了 TVor\_NN 算法和 YVor\_NN 算法。

## 2 基本定义

**定义 1**(Voronoi 图)<sup>[3]</sup> 给定一组数据点集  $Q = \{q_1, \dots, q_n\} \subset R^2$ , 其中  $2 < n < \infty$ , 当  $i \neq j$  时,  $q_i \neq q_j$ 。Voronoi 区域由以下公式给出:  $VH(q_i) = \{q | D(q, q_i) \leq D(q, q_j)\}$ 。  $D(q, q_i)$  为  $q$  与  $q_i$  之间的最小距离。  $q_i$  称为 Voronoi 生成点, 由  $q_i$  所决定的 Voronoi 区域  $VH(q_i)$  称为 Voronoi 多边形, Voronoi 多边形的棱记为  $VL(q_i)$ 。由  $V(H) = \{VH(q_1), \dots, VH(q_n)\}$  所定义的图形被称为 Voronoi 图。与  $VH(q_i)$  共享相同的棱的 Voronoi 多边形被称为  $VH(q_i)$  的邻接多边形, 它们的生成点被称为  $q_i$  的一级邻接生成点, 简记为  $LJG(q_i)$ 。依次可定义多级邻接生成点。图 1 展示了部分 Voronoi 图。

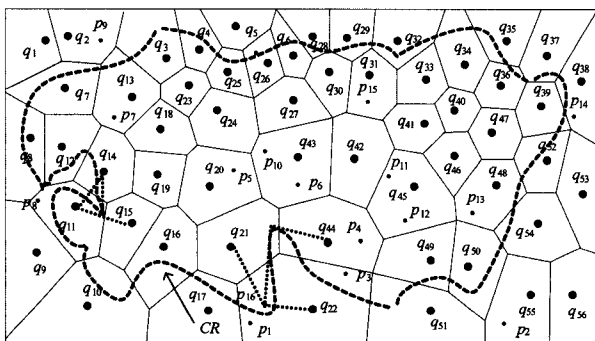


图 1 部分 Voronoi 图及受限区域示例

**定义 2**(Delaunay 三角形/三角网)<sup>[3]</sup> Voronoi 图中的生成点连接所得到的三角形, 称为 Delaunay 三角形; 一个 Voronoi 图对应的所有 Delaunay 三角形构成一个连续的无重叠的空间网状结构, 将该网状结构称为该 Voronoi 图对应的 Delaunay 三角网。

**性质 1**<sup>[3]</sup> 离 Voronoi 生成点  $q_i$  最近的生成点  $q_j$  必是  $q_i$  的邻接生成点。

**性质 2**<sup>[3]</sup> 一个 Voronoi 多边形  $V(q)$  内任何点  $o(o \notin q)$  到  $q$  的距离必小于  $o$  到其他 Voronoi 生成点的距离。Voronoi 图与 Delaunay 三角网对偶。

**性质 3**<sup>[3]</sup> 在数据点集  $Q$  所形成的 Delaunay 三角网中,

每个 Delaunay 三角形的外接圆均不包含  $Q$  中的其他任意数据点; 在由点集  $Q$  所能形成的三角网中, Delaunay 三角网中的三角形的最小角度是最大的。

**定义 3**(最近邻查询)<sup>[1]</sup> 假设有一  $d$  维空间的点集  $Q$  和一个查询点  $o$ , 最近邻查询就是找出的子集  $NN(o, Q): NN(o, Q) = \{q \in Q | \forall s \in U: D(o, q) \leq D(o, s)\}$ 。若要找出  $k$  个最近邻, 该定义则可扩展成  $k$  个最近邻的查询, 即  $kNN(o, Q) = \{q_1, q_2, \dots, q_k\}$ 。其中,  $\forall h \in Q - kNN(o, Q), u \in kNN(o, Q)$  且  $D(o, q) \leq D(o, h); D(q_i, o) \leq D(q_j, o), 1 \leq i < j \leq k$ 。

定义 1 中,  $D(o, q)$  和  $D(o, s)$  等表示两个数据点之间的距离。在不强调数据集的情况下,  $NN(o, Q)$  也可简写为  $NN(o)$ , 表示  $o$  的最近邻。

## 3 基于凸包的 Voronoi 图生成方法

平面数据点的凸包是计算几何中的一个重要内容, 被广泛用于网格生成、聚类分析、数据查询、图像处理、交通控制和机器人智能等领域。数据点的凸包被定义为包含这些平面数据点的最小凸多边形。在凸包中, 连接任意两点的线段必须完全位于多边形内。数据点的最短路径相当于由凸包进行了完全的包围。由于凸包具有优良的几何特性和易构性, 本节着重研究基于凸包的 Voronoi 图生成方法(简称为 TVor 方法)。

我们给出的基于凸包的 Voronoi 图生成方法的关键步骤是首先基于凸包的性质和特点生成给定数据点集  $Q$  的 Delannay 三角网; 生成过程中利用分治法、启发式局部优化策略和局部数据点的扫描线动态更新策略; 进而确定三角网中每个 Delannay 三角形的外接圆, 确定其圆心; 最后依据 Voronoi 图的定义和性质生成数据点集  $Q$  的 Voronoi 图。

其中, Delannay 三角网的高效生成是本文方法的关键, Delannay 三角网的具体生成方法的核心步骤如下:

第 1 步 针对给定数据集构建空间凸包, 将空间数据点逐个加入到凸包中, 并动态地修改凸包。

第 2 步 利用启发式方法, 每次将数据点集  $Q$  分成近似相等的两个子集  $QL$  和  $QR$ , 重复第 1 步, 构建  $QL$  与  $QR$  的凸包;  $QL$  与  $QR$  中的点集少于预定数值时, 进行第 3 步。

第 3 步 在  $QL$  和  $QR$  中生成三角网, 利用动态局部优化算法对三角网进行优化进而生成 Delaunay 三角网。

第 4 步 找出  $QL$  和  $QR$  中两个凸包底线和顶线, 由底线至顶线合并  $QL$  和  $QR$  中两个三角网, 反复执行第 4 步, 直到一个凸包中包含所有的数据集, 利用局部动态更新策略对三角网进行局部更新。

根据以上论述, 基于凸包生成给定数据点集  $Q$  的 Voronoi 图的具体算法如算法 1 所示。

### 算法 1 Create\_Voronoi(Q)

输入: 数据点集  $Q$ ;

输出: 数据点集  $Q$  的 Voronoi 图  $VT$ ;

begin

1. Create\_TBao();

2. Loop  $QL, QR$ //初始化  $QL$  集合与  $QR$  集合

3. for  $i=1$  to  $nPoint$

4. Point  $\leftarrow nPoint(i)$ ;

5. TBao.add(point);

6. LvDelaunay  $\leftarrow$  TBao.get(QL) //对  $QL$  递归处理

```

7. if (|QL| < flag) then
8.   return; //当 QL 中的点集较少时,递归结束
9. LrDelaunay ← TBao.get(QR)
10. if (|QR| < flag) then
11.   return;
12. LOP(QR, QL); //对 QR, QL 进行启发式局部优化;
13. TBao ← CombinV_R(QR, QL); //进行递归合并
14. if (TBao.getCountPoint = n) then
15.   DT ← 构建成功的三角网;
16.   Delannay ← TBao.getDelaunay;
17. else if (RegionTriMesh = NULL) then
18.   return (点集不存在);
19.   if (部分数据点发生变化) then
20.     生成动态数据点的扫描线;
21.     DT ← 局部动态更新 Delannay 三角网;
22.     生成每个 Delannay 三角形的外接圆;
23.     连接外接圆的圆心,生成 Voronoi 图 VT;
24.     return VT;
end

```

时间复杂度分析:设空间数据点的个数为  $n$ ,算法 1 基于凸包利用分治法生成 Delannay 三角网的时间复杂度为  $O(n \log n)$ ,利用生成的 Delannay 三角网生成外接圆的时间复杂度为  $O(n)$ ,连接外接圆的圆心生成各个 Voronoi 多边形的棱的时间复杂为  $O(n)$ ,故该方法总的复杂度为  $O(n \log n)$ 。该方法充分利用了凸包的特性进行数据点的计算,相比其他方法更简捷,易于实现,效率较高。

#### 4 基于 Voronoi 图的近邻查询方法

近年来,随着数据信息的爆炸式增长,空间数据的查询和分析技术越来越受到人们的重视。作为空间数据查询的重要分支,空间数据的近邻关系查询具有较为重要的作用。已有的最近邻查询方法<sup>[6-18]</sup>主要适用于无受限区域环境下的最近邻查询问题,但无法有效处理受限区域内的最近邻查询问题。本节基于第 3 节所构建的 Voronoi 图着重研究受限区域内的最近邻查询问题,首先给出以下相关定义。

**定义 4(受限区域中的最近邻查询)** 设二维空间中的平面区域为  $CR$ ,数据对象集为  $Q$ 。 $CR$  的边界线记为  $CR_l$ 。针对  $Q$ ,在  $CR$  中查询某个或某些数据对象点的最近邻称为受限区域中的最近邻查询。处于受限区域  $CR$  之内的数据点称为域内点;处于  $CR$  之外的数据点称为域外点。

图 1 中,虚线包围的区域即为受限区域  $CR$ ,数据点  $q_{30}$ ,  $p_{15}$  等均为域内点; $q_{51}$ ,  $q_{55}$  等为域外点。如无特殊说明,本文中的查询点为域内点。

**定义 5(可达判定圆域)** 设  $p$  和  $q$  之间的最短距离为  $MIN(p, q)$ ,则以  $q$  为圆心、 $MIN(q, p)$  为半径生成的圆域称为  $q$  到  $p$  的可达判定圆域,记为  $Circle(q, MIN(q, p))$ 。若  $p$  是  $q$  的最近邻,则称该圆域为  $q$  的最近邻判定圆域,记为  $Circle(q, NN\_MIN(q, p))$ 。

**定义 6(域内阻断)** 对于空间数据集中的两个数据点  $q_j$  和  $q_i$ ,若  $q_j$  和  $q_i$  均在受限区域  $CR$  内,但  $q_j$  和  $q_i$  之间的直线段被  $CR$  的边界线  $CR_l$  截断,即  $q_j$  和  $q_i$  之间不能直线可达,则称  $q_j$  和  $q_i$  被域内阻断,表示为  $IZD(q_j, q_i)$ 。若  $q_j$  和  $q_i$  被域内阻断,则  $q_j$  和  $q_i$  通过折线可达,其最短折线距离表示为  $D_z(q_j, q_i)$ 。图 1 中,  $q_{11}$  和  $q_{14}$  即被域内阻断。

通常,查询点和数据点集的关系可分为“同质”和“异质”两类。查询点是数据点集的元素称为查询点和数据点集具有“同质”关系,反之则是“异质”关系。例如,某出租车  $q$  查询离其位置最近的出租车,出租车  $q$  和出租车集合之间的关系即为“同质”关系;若出租车  $q$  查询离其最近的修理厂,则出租车  $q$  和修理厂之间的关系即为“异质”关系。显然,在具体查询中,这种“同质”和“异质”关系将会较大程度地影响数据的组织结构和索引查询方法。本文 4.1 节给出查询点和数据集为“同质”关系下的受限区域内的最近邻查询(简称受限区域内的同质最近邻查询)方法;4.2 节给出查询点和数据集为“异质”关系下的受限区域内的最近邻查询(简称受限区域内的异质最近邻查询)方法。

##### 4.1 受限区域内的同质最近邻查询方法

针对受限区域内的同质最近邻查询问题,首先给出一个基于“先判定后查询”确定给定数据点的最近邻的查询方法(本文简称 TJS\_NN 方法)。TJS\_NN 方法的主要步骤如下:

第 1 步 判定处于受限区域中的数据对象点;

第 2 步 对受限区域中的数据对象点集建立空间索引结构(如空间网格或空间索引树等);

第 3 步 基于建立的空间索引结构查找给定查询点的最近邻。

TJS\_NN 方法思想较为简单明了,但由于受限区域的形状和边界往往较为复杂,数据点的判定和空间索引结构的建立具有较大的难度,索引树节点的空白空间往往也较大,查询效率不够理想。为此,本文提出基于 Voronoi 图的受限区域中的同质最近邻查询方法(简称 TVor\_NN 方法)来提高查询效率,减少计算冗余。与 TJS\_NN 方法不同,TVor\_NN 方法主要基于“先查询后过滤”的思想。

TVor\_NN 方法的主要步骤如下:

第 1 步 构建整个数据对象空间中对象集合的全局 Voronoi 图;

第 2 步 利用 Voronoi 图的性质 1 计算出无受限区域环境下查询点  $q$  的最近邻  $q'$ ;

第 3 步 对第 2 步计算出的查询点  $q$  的最近邻  $q'$  按情况进行筛选判断。

(1)若  $q'$  处于受限区域  $CR$  中,且  $q$  和  $q'$  之间的直线段不被受限区域的边界线阻断,则  $q'$  即为  $q$  在  $CR$  中的最近邻;

(2)若  $q'$  处于受限区域  $CR$  中,且  $q$  和  $q'$  之间的直线段被受限区域的边界线阻断,则确定  $q$  到  $q'$  的可达判定圆域。若圆域和  $CR$  的相交区域内无其他数据点,则  $q'$  为  $q$  在  $CR$  中的最近邻;否则,针对相交区域内的数据点计算和更新  $q$  的最近邻;

(3)若  $q'$  不处于受限区域  $CR$  中,则确定处于  $CR$  中的  $q$  的邻接生成点集  $P$ ,在  $P$  集中计算  $q$  的最近邻  $p$ ,确定  $q$  到  $p$  的可达判定圆域。若圆域和  $CR$  的相交区域内无其他数据点,则  $p$  即为  $q$  在  $CR$  中的最近邻;否则,针对相交区域内的数据点计算和更新  $q$  的最近邻。

TVor\_NN 方法中,首先针对整个空间中的数据集生成一个全局 Voronoi 图,再利用 Voronoi 图计算出无受限区域环境下查询点的最近邻。进一步,结合受限区域特点和 Voronoi 图的性质对其进行二次判断和更新。由于落在判定圆域内的数据点和查询点邻接数据点的个数较少,故二次判

断和更新的计算量较少。该方法在查询点和受限区域边界频繁变化的情况下具有明显的优势。

基于以上讨论,我们提出的基于 Voronoi 图的受限区域内的同质最近邻查询算法如算法 2 所示。

#### 算法 2 TVor\_NN(Q, q, CR<sub>t</sub>)

输入:空间数据集 Q,查询点 q,受限区域边界线 CR<sub>t</sub>

输出:查询点 q 在受限区域 CR 中的最近邻;

```
begin
1. if Q 的全局 Voronoi 图没有生成 then
2.   Create_Voronoi(Q);
3. else
4.   q' ← NNvor(q, Q); //利用 Voronoi 图的性质 1 计算出 q 的最近邻
5.   if (q' in CR) and ((IZD(qi, q')=false) then
6.     return q';
7.   elseif (q' in CR) and ((IZD(qi, q')=true) then
8.     CIR ← Circle(q, MIN(q, q'));
9.     CJR ← CIR ∩ CR;
10.    if CJR 内没有其他生成点 then
11.      return q';
12.    else
13.      计算 CJR 中 q 的最近邻;
14.      更新 q';
15.      return q';
16.   elseif q' not in CR then
17.     在 CR 中确定 q 的邻接生成点集 Q';
18.     qt ← NN(q, Q'); //在 Q' 中计算 q 的最近邻 qt;
19.     CIR' ← Circle(q, MIN(q, qt));
20.     CJR' ← CIR' ∩ CR;
21.     if CJR' 内没有其他生成点 then
22.       return qt;
23.     else
24.       计算 CJR' 中 q 的最近邻;
25.       更新 qt;
26.       return qt;
27. if 查询点更新为 qnew then
28.   q ← qnew;
29.   TVor_NN(Q, q, CRt);
end
```

时间复杂度分析:设空间数据点的个数为  $n$ ,算法 2 基于 Voronoi 图可较好地处理受限区域内的给定数据点的最近邻查询问题。若没有生成数据点集的 Voronoi 图,算法 2 调用 Create\_Voronoi() 算法生成 Voronoi 图的时间复杂度为  $O(n \log n)$ ;无受限区域环境下利用 Voronoi 图确定给定查询点的最近邻的时间复杂度为  $O(n)$ ;在受限区域环境下进一步按情况判定更新的平均复杂度是常数级的,故算法 2 核心步骤的平均时间复杂度为  $O(n \log n)$ 。算法 2 中,查询点的邻接生成点个数、可达判定圆域和受限区域的相交区域中的生成点的数量往往较少,从而最近邻更新的代价往往较低,算法的冗余计算较少。由算法 2 可知,整个数据集的 Voronoi 图构建后,在查询点更新较为频繁的情况下,该算法具有较大的查询优势。

#### 4.2 受限区域内的异质最近邻查询方法

与 4.1 节类似,针对受限区域内的异质最近邻查询问题,我们首先也可给出一个基于“先判定后查询”确定给定数据点的最近邻的查询方法(本文简称 YJS\_NN 方法)。YJS\_NN

方法的思想为:首先计算落在受限区域内的数据点,再基于受限区域内的数据点建立空间索引结构,最后根据查询点的位置,利用建立的空间索引结构进行索引查询。由于受限区域的形状往往较为复杂多样,导致数据点的筛选、索引结构的建立和查询中的剪枝过滤等关键步骤的效率较低。基于“先查询后过滤”的策略,本节提出了基于 Voronoi 图的受限区域内的异质最近邻查询方法(简称 YVor\_NN 方法)。

YVor\_NN 方法的主要思想为:首先针对数据集 Q,构建 Q 的全局 Voronoi 图(由于查询点  $p$  和  $Q$  是“异质”关系, $p$  并不作为生成点参与 Voronoi 图的构造, $p$  处于某个 Voronoi 多边形内或棱上);利用 Voronoi 图的性质 2 计算出无受限区域环境下查询点  $p$  的最近邻  $q$ ;再根据查询点  $p$  和其最近邻  $q$  被受限区域边界线的阻隔情况进行二次计算和判断,最后返回更新后的最近邻。基于 Voronoi 图的受限区域内的异质最近邻查询算法如算法 3 所示。

#### 算法 3 YVor\_NN(Q, q, CR<sub>t</sub>)

输入:空间数据集 Q,查询点 p,受限区域边界线 CR<sub>t</sub>

输出:查询点 p 在受限区域 CR 中的最近邻;

```
begin
1. if Q 的全局 Voronoi 图没有生成 then
2.   Create_Voronoi(Q);
3. else
4.   q ← NNvor(p, Q); //利用 Voronoi 图的性质 2 计算出 p 在 Q 集中的最近邻
5.   判断 q 是否为 CR 的域内点;
6.   if (q 为 CR 的域内点) then
7.     判断查询点 p 和 q 是否被域内阻断;
8.     if (IZD(p, q)=false) then
9.       return q;
10.    else
11.      IR ← CR ∩ Circle(p, MIN(p, q));
12.      if IR 内包含有 Q 集中的数据点集 Q' then
13.        qnew ← NN(p, Q');
14.        return qnew;
15.      else return q;
16.    else
17.      确定 q 的域内邻接生成点集 TQ;
18.      tq ← NN(p, TQ);
19.      q ← tq;
20.      转 11;
21. if 查询点更新为 pnew then
28.   p ← pnew;
29.   YVor_NN(Q, p, CRt);
end
```

时间复杂度分析:设空间数据点的个数为  $n$ ,算法 3 调用 Create\_Voronoi() 算法生成 Voronoi 图的时间复杂度为  $O(n \log n)$ ;无受限区域环境下利用 Voronoi 图确定给定查询点的异质最近邻的时间复杂度为  $O(n)$ ;在受限区域环境下进一步按情况判定更新的平均复杂度是常数级的,故算法 3 核心步骤的平均时间复杂度为  $O(n \log n)$ 。算法 3 利用 Voronoi 图的性质 2 对查询点的异质最近邻进行了初始的计算,对最近邻再进行二次筛选和判定时所涉及的数据点较少,计算冗余较少,具有较高的查询效率。

## 5 实验分析

本文第2至4节研究了基于凸包的Voronoi图生成方法和基于Voronoi图的受限区域内的最近邻查询方法,分别给出了Create\_Voronoi算法、TVor\_NN算法和YVor\_NN算法。本节在AMD FX、4G CPU、8G内存、Windows XP环境下,利用C++ builder6.0对所提算法的性能进行了实验比较。所用空间数据集是由空间数据生成器(GEDATA5.0)随机生成的空间模拟数据。

图2展示了空间数据集规模逐渐增大情况下本文给出的基于凸包的Voronoi图生成方法(简称为CVor方法)与典型的平面扫描算法(简称PScan方法)、增量构造方法(简称ZG方法)的生成效率的比率。图2中,曲线 $l_1$ 表示CVor方法与PScan方法的比较情况,曲线 $l_2$ 表示CVor方法与ZG方法的比较情况。

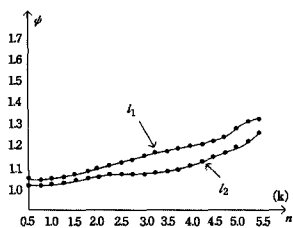
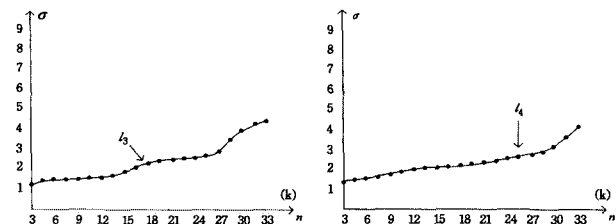


图2 Voronoi图生成方法效率比较

由图2可知,在数据量较小的情况下,本文的方法相对于PScan方法和ZG方法的优势并不明显,但随着数据集的增大,所提方法的查询效率相对于PScan方法和ZG方法的比率将逐渐增大,在数据量较多的情况下,本文所研究的Voronoi生成方法具有一定的优势。

由于已有的最近邻查询方法不适合处理受限区域内的最近邻查询问题,无法直接用来查询受限区域内的最近邻。因此本节着重对所提出的TVor\_NN方法和TJS\_NN方法,YVor\_NN方法和YJS\_NN方法进行比较分析。图3(a)给出了在数据集逐渐增大情况下,TVor\_NN方法和TJS\_NN方法效率的比较结果。图3(b)给出了YVor\_NN方法和YJS\_NN方法效率的比较结果。曲线 $l_3$ 表示TVor\_NN方法与TJS\_NN方法的比较情况; $l_4$ 表示YVor\_NN方法与YJS\_NN方法的比较情况。



(a) TVor\_NN方法和TJS\_NN方法的效率比较 (b) YVor\_NN方法和YJS\_NN方法的效率比较

图3 受限区域内最近邻查询方法效率比较

由图3可知,在数据集规模较小的情况下,TVor\_NN方法与TJS\_NN方法,YVor\_NN方法与YJS\_NN方法的查询效率的差距并不明显,但随着数据量的增大,TVor\_NN方法的查询效率明显高于TJS\_NN方法,YVor\_NN方法的查询效率明显高于YJS\_NN方法。若查询次数较为频繁,则由叠

加效应可知,TVor\_NN方法和YVor\_NN方法将会较大幅度地提高查询性能。通常,在数据量相似的情况下,受限区域的边界越复杂,查询点的位置更新越快,TVor\_NN方法和YVor\_NN方法的查询优势越明显。

**结束语** 为了提高Voronoi图的构建效率,减少数据结构的复杂性,本文将数据点集凸包技术和Delaunay三角网技术相融合,利用分治法、启发式局部优化策略和局部数据点的扫描线动态更新策略给出了基于凸包的Voronoi图构建方法,给出了Create\_Voronoi()算法。已有的近邻查询方法无法有效处理受限区域内的最近邻查询问题,为了弥补已有方法的不足,本文利用所生成的Voronoi图给出了受限区域内的同质和异质最近邻查询方法,首先针对整个空间中的数据集生成一个全局Voronoi图,再利用Voronoi图计算出无受限区域环境下查询点的最近邻。进一步,结合受限区域特点和Voronoi图的性质对其进行二次判断和更新,分别提出了TVor\_NN算法和YVor\_NN算法。基于理论和技术研究,我们给出了实验分析。未来的研究重点主要集中在以下方面:

1. 基于Voronoi图的受限区域内的组最近邻查询问题的研究。
2. 基于Voronoi图的不确定区域内(如Vague区域<sup>[19]</sup>)的模糊最近邻查询问题。

## 参考文献

- [1] 郝忠孝. 时空数据库查询与推理[M]. 北京: 科学出版社, 2010
- [2] 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] Sacl J R, Urrutia J. Voronoi diagrams. handbook on computational geometry[M]. Ottawa: Elsevier Science, 2006: 201-290
- [4] Dong L, Wu Y, Zhou S. Constructing the Voronoi Diagram of Planar Point Set in Parallel[C] // International Conference on Computational Intelligence and Software Engineering, 2009 (CiSE 2009). IEEE, 2009, 1-5
- [5] 张娟, 杜全叶. 增删点后的Voronoi图生成算法[J]. 图学学报, 2013, 34(1): 46-49
- [6] 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
- [7] 杨泽雪, 郝忠孝. 基于Voronoi图的线段最近邻查询[J]. 计算机科学, 2012, 39(6): 143-146
- [8] Kekeng X, David T. Voronoi-based range and continuous range query processing in mobile databases[J]. Journal of Computer and System Sciences, 2011, 77(4): 637-651
- [9] Nutanong S, Tanin E, Zhang Rui. Incremental evaluation of visible nearest neighbor queries[J]. IEEE Transactions on Knowledge Engineering, 2010, 22: 665-681
- [10] Gao Yun-jun, Zheng Bai-hua, Chen Gen-cai, et al. Visible Reverse k-Nearest Neighbor Query Processing in Spatial Databases[J]. IEEE Trans. Knowl. Data Eng., 2009, 21: 1314-1327
- [11] Hu Ling, Jing Yi-nan, Ku Wei-Shinn, et al. Enforcing k Nearest Neighbor Query Integrity on Road Networks [C] // ACM SIGSPATIAL GIS'12. 2012: 346-349
- [12] Elmongui H G, Mokbel M F, Aref W G. Continuous aggregate nearest neighbor queries[J]. GeoInformatica, 2013, 17(1): 63-95
- [13] 李艳红, 李国徽, 杜小坤. 路网中双色数据集上连续反向k近邻查询处理的研究[J]. 计算机科学, 2012, 39(11): 131-136

(下转第247页)

表5 Colon数据集上5折交叉验证精度

方法	基因数			
	20	74	123	199
MMIGA-Selection/ELM	0.87872	0.90122	0.91223	0.9840
t-test/ELM	0.87800	0.88711	0.88711	0.9719
MMIGA-Selection/SVM	0.90321	0.91244	0.91222	0.9841
t-test/SVM	0.87811	0.90244	0.9879	0.9793
MMIGA-Selection/BPNN	0.90128	0.92221	0.91221	0.89221
t-test/BPNN	0.88122	0.87162	0.88971	0.83871
MMIGA-Selection/Bayes	0.91943	0.88702	0.90177	0.91117
t-test/Bayes	0.90942	0.88712	0.88712	0.88710

表6 SRBCT数据集上5折交叉验证精度

方法	基因数			
	18	62	108	250
MMIGA-Selection/ELM	1.0000	0.99122	0.97822	0.98110
f-test/ELM	1.0000	0.97012	0.98221	0.95330
MMIGA-Selection/SVM	1.0000	1.00000	0.98391	0.98357
f-test/SVM	1.0000	0.98110	0.98101	0.97112
MMIGA-Selection/BPNN	1.0000	0.99001	0.98410	0.99212
f-test/BPNN	1.0000	0.98704	0.99100	0.97222
MMIGA-Selection/Bayes	1.0000	0.99778	0.97912	0.98777
f-test/Bayes	1.0000	0.99811	0.96812	0.96222

由上述图和表可以看出,MMIGA-Selection算法在不同分类器上都有较高的分类精度。由于较好的初始种群提供了较优的搜索起点,在选择过程中为遗传算法节省了时间,结合遗传算法本身特点,MMIGA-Selection算法比 t-test/f-test 法相对更耗时,但是差别较小,在基因选择领域中可以接受。

**结束语** 本文提出了一种基于互信息最大化的模型无关遗传算法的特征选择方法。算法结合互信息最大化方法(Filter法)对基因进行初步筛选,去除大量噪声以及不相关特征,为遗传算法提供一个更好的种群初始化环境。理想的搜索起点能够加快遗传算法的搜索速度。由于将类间距离与类内距离定义为遗传算法的适应度函数,使得算法与分类器无关,适应程度与运行效率相对较高。实验结果显示,MMIGA-Selection在基因特征选择上取得了较好的结果,算法有两个主要优点:1)能够有效减少冗余基因;2)模型无关性,泛化程度高。

### 参考文献

[1] Kang H N, Chen I M, Wilson C S. Gene expression classifiers for relapse-free survival and minimal residual disease improve risk classification and outcome prediction in pediatric B-precursor acute lymphoblastic leukemia[J]. Blood, 2010, 115: 1394-1405

[2] 任江涛,黄焕宇,孙婧昊. 基于相关性分析及遗传算法的高维数据特征选择[J]. 计算机应用, 2006, 26(6): 1403-1405

[3] 裘国永,王娜,汪万紫. 基于互信息和遗传算法的两阶段特征选择方法[J]. 计算机应用研究, 2012, 29(8): 2903-2905

[4] Liu Hui-qing, Li Jin-yan. A Comparative Study on Feature Se-

lection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns[J]. Genome Informatics, 2002, 13: 51-60

[5] Zhao Zheng, Wang Lei, Liu Huan. Efficient spectral feature selection with minimum redundancy[J]. Proceedings of the National Conference. 2010, 1: 673-678

[6] 王凌,陈震,危水根,等. 基于改进最大互信息法的MR切片图像配准[J]. 生物医学工程学杂志, 2012, 29(2): 201-205

[7] 杨虎,马斌荣,任海萍,等. 基于最大互信息的人脑MR-PET图像配准方法[J]. 北京生物医学工程, 2001, 20(4): 246-251

[8] Michalewicz Z. A Modified Genetic Algorithm for Optimal Control Problems[J]. Computers Math, 1992, 23(12): 83-94

[9] Huang Jin-jie, Cai Yun-ze, Xu Xiao-ming. A Hybrid Genetic Algorithm for Feature Selection Wrapper Based on Mutual Information[J]. Pattern Recognition Letters, 2007, 28(13): 1825-1844

[10] 李建中,杨昆,高宏,等. 考虑样本不平衡的模型无关的基因选择方法[J]. 软件学报, 2006, 17(7): 1485-1493

[11] Lu Hui-juan, Chen Wu-tao, Ma Xiao-ping, et al. Model-free Gene Selection Using Genetic Algorithms [J]. International Journal of Digital Content Technology and its Applications, 2011, 5(1): 195-203

[12] 陆慧娟. 基于基因表达数据的肿瘤分类算法研究[D]. 徐州: 中国矿业大学, 2012

[13] 王明怡. 微阵列数据挖掘技术的研究[D]. 杭州: 浙江大学, 2004

[14] 刘庆和,梁正友. 一种基于信息增益的特征优化选择方法[J]. 计算机工程与应用, 2011, 47(12): 130-132

[15] Hu Y, Loizou P C. Speech enhancement based on wavelet thresholding the multitaper Spectrum [J]. IEEE Trans on Speech and Audio Processing, 2004, 12(1): 59-67

[16] Wang Zhi-teng, Zhang Hong-jun, Hang Ying. Fire distribution optimization based on quantum immune genetic algorithm[C]// 2011 International Conference of Information Technology Computer Engineering and Management Sciences. IEEE, 2011, 1: 95-98

[17] Jiang Feng-guo. The truss structural optimization design based on improved hybrid genetic algorithm[J]. Advanced Materials Research, 2011, 163-167: 2304-2308

[18] Holland J H. The psychology of vocational choice: A theory of personality types and model environments[M]. Oxford, England: Oxford University Press, 1965

[19] Bagley J D. The Behavior of Adaptive Systems which Employ Genetic and Correlation Algorithms[D]. The university of Michigan, 1967

[20] Holland P W. Discrete Multivariate Analysis: Theory and Practice[M]. MIT Press, 1975

(上接第224页)

[14] 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

[15] 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

[16] Zhang Li-ping, Li Song, Li Lin, et al. Simple Continues Near Neighbor Chain Query of the Datasets Based on the R Tree[J].

Journal of Computaional Information Systems, 2012, 8(22): 9159-916

[17] 张丽平,李林,李松,等. 预定数据链规模的单纯型连续最近邻链查询[J]. 计算机工程, 2012, 38(10): 51-53

[18] 张丽平,李松,郝忠孝. 球面上的最近邻查询方法研究[J]. 计算机工程与应用, 2011, 47(5): 126-129

[19] 张丽平,李松,郝晓红,等.  $J_m$ 粗糙 Vague 区域关系[J]. 浙江大学学报, 2012, 46(1): 122-128