文章编号: 1000-5013(2011)02-0169-06

无索引空间数据库的基于最优点的集合最近邻查找算法

骆炎民[1,2], 陈维斌[1], 廖明宏[3]

(1. 华侨大学 计算机科学与技术学院, 福建 泉州 362021;
2. 厦门大学 信息科学与技术学院, 福建 厦门 361005;
3. 厦门大学 软件学院, 福建 厦门 361005)

摘要: 针对度量空间中的无索引空间数据库,提出一种基于最优点的集合最近邻查找算法及其改进算法. 采用真实数据集与人工生成的数据集对算法进行测试,评估所提出算法的效率. 实验结果表明,所提算法的效率优于组最近邻居查询算法,并且对于高维数据空间,所提出的算法有较高的稳定性. 由于查询区域中数据点的数量比较少,改进的基于最优点的集合最近邻查找算法的效率总体上要比改进前高.

关键词: 空间数据库; 最近邻; 集合最近邻; 查询区域

中图分类号: TP 311.131; TP 301.6 **文献标志码:** A

集合最近邻(Aggregate Nearest Neighbor, ANN)是由最近邻(Nearest Neighbor, NN)演变而来的. 它在许多研究领域中得到广泛的应用,如地理信息系统[1-2]、数据挖掘中的分类[3]和异常点检测[4]等. 空间数据一般具有高维、海量等特性,如何高效地在空间数据库中进行各种最近邻查询,已经成为空间数据库及数据挖掘中的一个重要研究内容. 目前,大多数的NN和ANN算法的主要思想是基于空间索引结构的,通过对空间数据库中的数据进行裁剪,以提高查询效率. 在众多的索引结构中,R-树[5-7]是最流行的一种. 这些算法大部分在低维数据空间都能获得理想的执行结果和效率,而有研究表明,传统的索引方法在高维数据空间中将失去效果. 文献[8]指出,在高维数据空间中,由于存在所谓的"维灾"的问题,传统的NN和ANN查询可能会失去意义. 在空间数据库的最近邻查找中,查找效率是一个关键问题. 因此,为了缩小查询空间范围,以减少查询所需要的时间,人们引进了各种各样的空间索引机制. 在空间数据库中,数据集一般通过高维空间索引(Spatial Access Methods, SAM)方法建立相关索引结构,如R-树[9],R^*-树[10]等. 因此,大部分的最近邻查询算法都利用这些索引机制来实现高维数据空间中的最近邻及 k 最近邻查询[11-13]. 此外,一些基于距离空间的索引结构,如 MB^+-树[14],GNAT[15]和MVP-树[16]等也用于相似性查询. 本文提出了一种无索引的集合最近邻查找算法,并以max函数为例讨论ANN查找算法的实现.

1 基本定义

一般情况下,空间数据库中的数据是均匀分布的. 在此前提下,设定 $f=\mathrm{sum}$,并以二维空间中的数据为例,分析算法的实现方法. 算法很容易推广到高维数据空间,在高维数据空间中也是适用的.

文中一些符号的含义如下:空间数据库数据集 $P=\{p_1,p_2,\cdots,p_N\}$;查询数据集 $Q=\{q_1,q_2,\cdots,q_n\}$;N,n 分别为空间数据库数据集和查询数据集中数据对象的数量;M 为最小边界矩形的面积;$d(p,q)$,$D(p,q)=|pq|$ 分别为数据对象 p,q 的距离函数和欧氏距离;集合距离函数 $\mathrm{adist}(p,Q)=f(|pq_1|,$

收稿日期: 2009-08-10

通信作者: 骆炎民(1974-),男,副教授,主要从事图形图像与人工智能的研究. E-mail:lym@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(2009J01288)

$|pq_2|,\cdots,|pq_n|)$, $C(q,r)=\{p|p\in M,|pq|\leqslant r\}$, $S(p)\equiv S(p,Q)=\bigcup C(q_i,|pq_i|)$.

由于空间数据的海量性,因此算法是在没有空间数据索引的情况下,对空间数据进行裁剪,获取一个足够小的查询区域,然后在查询区域中求解 ANN. 这样可以裁剪掉查询区域以外的数据对象,减少搜索空间,提高效率. 如何获得查询区域是算法实现的关键.

首先引入算法中的两个概念:r-近邻域和数据集的最小边界矩形(MBR).

定义 1 给定一个查询对象 $q\in O$ 和一个非负半径 r,查询对象 q 的 r-近邻域定义为包含在以 q 为圆心,r 为半径的空间数据点所组成的集合,如图 1 所示. 它可表示为

$$RN(q,r)=\{o\mid o\in O\wedge d(q,o)\leqslant r\}.$$

定义 2 能够包含查询数据集 Q 中所有对象的最小矩形,称为查询数据集的最小边界矩形(MBR),如图 2 所示.

由于空间数据库中的对象是均匀分布的,因此在查询数据集的 MBR 中总有空间数据库中的对象存在. 如图 2 所示,查询对于 $f=\mathrm{sum}$,在数据集 P 对于查询点集合 Q 的集合最近邻点将以极大的概率落在 Q 的 MBR 中. 一般情况下,只需在 Q 的 MBR 中查询其集合最近邻点,把 Q 的 MBR 以外的区域裁剪掉. 这个区域称之为查询区域.

如果一个查询区域足够小,并且又能保证 ANN 查询结果一定落在该查询区域内时,则称这个查询区域是一个"满意"的查询区域.

2 基于最优点的集合最近邻查找算法

给定一个点 p 和半径 r,将数据空间分成两部分,一部分是 p 点的 r-近邻区域,如图 1 中的阴影部分;另一部分是 p 的 r-近邻区域以外的区域,称为 p 的 r-近邻外区域. 如果半径 r 足够小,使得 p 的 r-近邻区域刚好能包含查询数据集合 Q 中的所有对象$\{q_1,q_2,\cdots,q_n\}$,即 p 是查询数据集的最小外接圆的圆心,称这个 p 点为查询集合的最优点,记为 v_p 点,如图 1 所示.

对于 $f=\mathrm{sum}$,如果 p 是 Q 的集合最近邻点,则 p 使得 $\mathrm{adist}(p,Q)=\mathrm{sum}(|pq_1|,|pq_2|,\cdots,|pq_n|)$ 的值最小. 很显然,如果 r 值合适的话,p 一定落在 v_p 的 r-近邻区域,而不可能落在 v_p 的 r-近邻外区域,即 $\mathrm{adist}(v_p,Q)<\mathrm{adist}(s,Q)$,如图 3 所示.

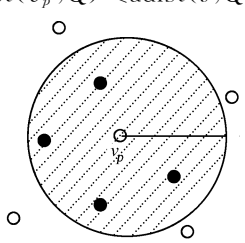


图 1 v_p 点的 r 近邻区域

Fig. 1 Example of the r-neighbor region of v_p

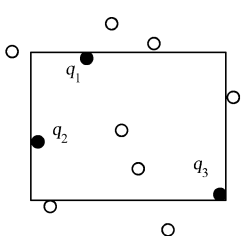


图 2 查询数据集 Q 的最小边界矩形

Fig. 2 Minimum boundary rectangle of query data set Q

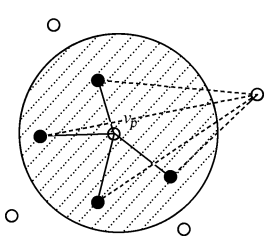


图 3 r-近邻区域的点集合最近邻比较

Fig. 3 Comparison of ANN between the point of r-neighbor and out-neighbor

在 v_p 确定的前提下,寻找合适 r 的简便方法是,让 $r=\max(|v_pq_i|)(i=1,\cdots,n)$. 当 Q 中存在某个对象距离 v_p 点比较远时,求出的查询区域会很大,影响算法的效率. 因此,可以采用另一种方法. 由于 Q 的 ANN 点 p 使得 $\mathrm{adist}(p,Q)=\mathrm{sum}(|pq_1|,|pq_2|,\cdots,|pq_n|)$的值最小,故对于每个 q_i,$|pq_i|$的值应尽可能小,由此可以断定 p 一定落在每个 q_i 的 r-近邻区域. 根据定义可知,v_p点的 r-近邻区域可以表示为 $C(v_p,r)=\{p|p\in P,|p,v_p|\leqslant r\}$. 对于查询数据集 Q 中的每个对象 q_i,通过定理 1 可以求得 v_p_ANN 算法的查询区域.

定理 1 假设 $R_i=C(q_i,r_i)$表示 q_i 的 r-近邻区域,则 $R=Y_{i=1,\cdots,n}R_i$ 是查询数据集 Q 中各对象 q_i 的 r_i-近邻区域的并集. 若所选取的每个 r_i 的值能保证 $I_{i=1,\cdots,n}R_i$ 不是空集,则 Q 的集合最近邻点一定落在 R 中.

根据定理 1 可知,最优点的集合最近邻查找算法的查询区域就是 R. 要保证 R 不是空集,同时又要让 R 所覆盖的区域尽可能地小,必须合理地计算每个 r_i 的值,而获取合理各个 r_i 值的关键是 v_p 点的选

择．因此，v_p 点的选择是算法的关键．

获取 v_p 点的最简单的方法是随机生成一个 v_p 点，然后取 $r_i=|v_pq_i|$．但是这样做的随机性太大，有可能使 R 覆盖几乎整个数据空间，使算法失去意义．考虑到 $f=\mathrm{sum}$，合理的 v_p 点应该使得 $\mathrm{adist}(v_p,Q)=\mathrm{sum}(|v_pq_1|,|v_pq_2|,\cdots,|v_pq_n|)$ 的值尽可能地小或者是最小，因此，最佳的 v_p 点应该是 Q 的质心，P 中距离该质心最近的点就是 Q 的集合最近邻．但是，对于不同的函数 f，Q 的质心是不一样的．

设 (x,y) 是 Q 的质心 q 的坐标，(x_i,y_i) 为 Q 中的查询点 q_i 的坐标，为使 $\mathrm{adist}(p,Q)$ 最小，必须满足 $\mathrm{adist}(p,Q)$ 在 $q(x,y)$ 上的偏导数为 0[5]．由此可以得出计算 $q(x,y)$ 的公式，即

$$\frac{\partial \mathrm{adist}(q,Q)}{\partial x}=\sum_{i=1,\cdots,n}\frac{(x-x_i)}{\mathrm{sqrt}((x-x_i)^2+(y-y_i)^2)}=0, \tag{1}$$

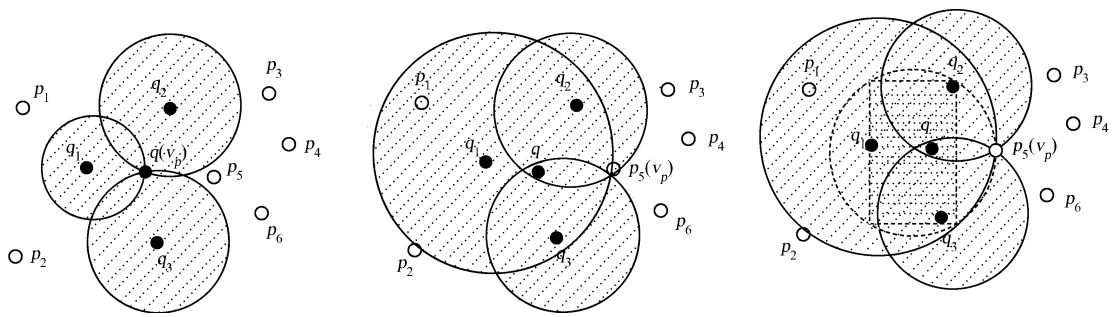
$$\frac{\partial \mathrm{adist}(q,Q)}{\partial y}=\sum_{i=1,\cdots,n}\frac{(y-y_i)}{\mathrm{sqrt}((x-x_i)^2+(y-y_i)^2)}=0. \tag{2}$$

然而，上式在 $n>2$ 的时候没有一个固定形式的解，故利用上式也很难得到所需要的几何中心．

为了降低计算复杂度及提高效率，在基于最优点的集合最近邻查找算法中使用几何中心来代替质心．Q 的几何中心 $q(x,y)$ 的计算公式为

$$x=\frac{1}{n}\cdot\sum_{i=1,\cdots,n}x_i,\qquad y=\frac{1}{n}\cdot\sum_{i=1,\cdots,n}y_i. \tag{3}$$

在低维数据空间中，选择 Q 的几何中心 q 作为 v_p 点，基本上能保证查询区域 R 不是空集．但在高维数据空间中，由于数据点比较稀疏，相互之间的距离比较大，可能会导致 R 是空集，如图 4(a)所示．为了避免这种情况发生，算法取数据集 P 中距离 Q 的几何中心最近的点作为 v_p 点（图 4b 中的 p_5），这样就能保证 R 中至少有一个 v_p 点，避免了空集的情况，如图 4(b)所示．由于 R 是 Q 中各个对象 q_i 的 r_i-近邻区域的并，因此查询区域 R 总是能包含 Q 的 MBR，这样能保证在空间数据均匀分布的情况下，Q 的集合最近邻点总能存在于查询区域 R 中．



(a) v_p 为 Q 的几何中心　　(b) v_p 为 Q 的几何中心 q 的最近邻　　(c) 以包含 Q 的 MBR 外接圆

图 4　最优点的集合最近邻查找算法查询区域示例

Fig. 4　Examples of search region by vantage point-bases region pruning algorithm of ANN

基于最优点的集合最近邻查找算法，主要有如下 5 个步骤：

(1) 初始化 $R=\Phi$；

(2) 按式(3)计算 Q 的几何中心，可得到 q；

(3) 在空间数据库 P 中查找 q 的最近邻作为最优点 v_p；

(4) 查询数据集 Q 中的每个点 q_i，再分别计算 $r_i=|q_iv_p|$，$R_i=C(q_i,r_i)$，并更新 $R=R\cup R_i$；

(5) 在查询区域 R 中搜索集合最近邻点，并返回结果．

3　基于最优点的集合最近邻查找的改进算法

由于空间数据库 P 中的对象数量要远远大于查询点集合 Q 中对象的数量，即 $N\gg n$．因此在一般情况下，Q 中数据点的分布比较集中，基于最优点的集合最近邻查找算法找出的查询区域也会比较小，算法的效率比较高．但是，当 Q 中数据点的分布比较分散时，对算法效率的影响就比较大．为了提高算法效率，需要对基于最优点的集合最近邻查找算法进行改进，以进一步缩小查询区域，扩大裁剪区域．

当空间数据库中的对象是均匀分布的，对于 $f=\mathrm{sum}$，数据集 P 对于查询点集合 Q 的集合最近邻点

将以极大的概率落在 Q 的 MBR 中. 显然,在基于最优点的集合最近邻查找算法中找出来的查询区域(图 4(c)中的点状阴影区域),不仅一定要覆盖查询点集合 Q 的 MBR(图 4(c)中的虚线矩形框区域),而且要比 Q 的 MBR 区域大得多. 因此,当 Q 中的数据对象分布比较分散的情况下,可以选取 Q 的 MBR 区域作为查询区域.

但是,当 Q 中的数据对象分布比较集中时,有可能存在 Q 的 MBR 区域中没有包含空间数据库 P 中的对象. 此时,如果以 Q 的 MBR 作为查询区域,会导致查询区域的数据对象为空集,如图 4(c)所示. 即 p_5 是 Q 的集合最近邻点,但它却落在 Q 的 MBR 外. 因此,将查询区域扩大为 Q 的 MBR 的最小外接圆,如图 4(c)所示的矩形阴影背景区域,该区域就包含了 p_5. 当然,如果 Q 中的数据对象分布非常集中,以至于其 MBR 的最小外接圆中也没有包含空间数据库 P 中的对象时,可以对外接圆的半径再进行适当的放大,让它足够包含有适当的 P 中的数据点.

在算法的实现中,查询数据集合 Q 的最小外接圆的圆心仍然使用 Q 的几何中心. 设 $q_i(x_i,y_i)(i=1,\cdots,n)$ 为 Q 中的点,令 $\min\_x=\min\_x_i$,$\min\_y=\min\_y_i$,$\max\_x=\max\_x_i$,$\max\_y=\max\_y_i$,$i=1,\cdots,n$,且 $(x_i,y_i)\in Q$,则查询数据集合 Q 的最小外接圆的半径 r_{min} 的计算公式为

$$r_{\min}=\frac{1}{2}\cdot(\max((|\max\_x-\min\_x|),(|\max\_y-\min\_y|))). \tag{4}$$

改进的基于最优点的集合最近邻查找算法,有如下 7 个主要步骤:

(1) 初始化 $R=\Phi$;

(2) 按式(3)计算 Q 的几何中心,可得到 q;

(3) 在空间数据库 P 中查找 q 的最近邻作为最优点 v_p;

(4) 求 Q 的数据集的最小边界矩形(MBR),即 $\min\_x=\min\_x_i$,$\min\_y=\min\_y_i$,$\max\_x=\max\_x_i$,$\max\_y=\max\_y_i$;

(5) 计算 Q 的 MBR 的最小外接圆,圆心为 q,按式(4)计算半径 $r_{\min}$;

(6) $R=C(q,r_{\min})$,当 v_p 点不在 R 中时,按增量比例 d 放大 $r_{\min}$,即 $r_{\min}=r_{\min}\times d$,$d>1$,更新 R;

(7) 最后,在查询区域 R 中搜索集合最近邻点,并返回结果.

4 算法的分析与实验结果

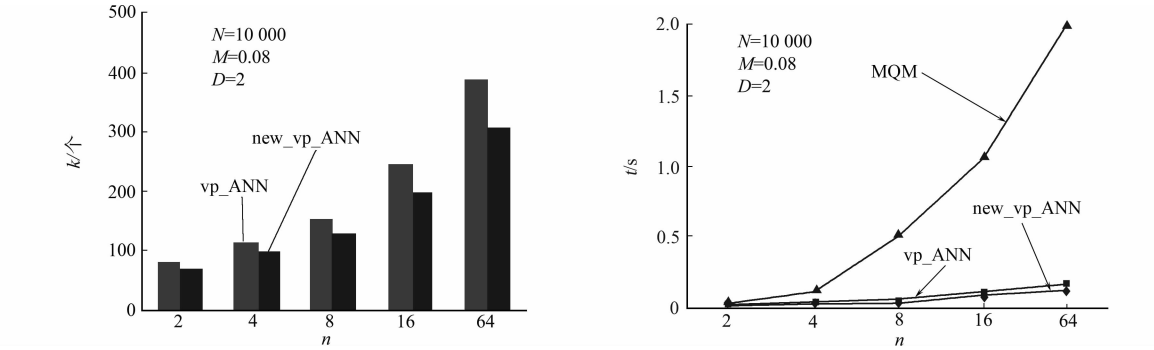
在算法的时间复杂度方面,求 Q 的几何中心的时间复杂度为 $O(n)$,其中 n 为 Q 中包含的点的数量;求 Q 的几何中心在 P 中的最近邻的时间复杂度为 $O(N)$,其中 N 为 P 中包含的点的数量;求最小包围矩形 MBR 的左上顶点和右下顶点的时间复杂度为 $O(n+1)$;计算 MBR 的圆心和半径的时间复杂度都为 $O(1)$;对集合 P 的裁剪的时间复杂度为 $O(N)$;在裁剪区域 R 中求 ANN 的结果的时间复杂度最大不会超过 $O(N)$. 由于集合 P 中所包含的数据点一般远远大于 Q 的中的点,即 $N\gg n$,故最优点的集合最近邻查找算法及其改进算法的时间复杂度都近似为 $O(N)$.

实验环境:Linux 操作系统,Inter(R),Pentium(R),4 CPU,2.66 GHz 处理器,512 MB 内存;采用 C^{++} 进行编程. 实验数据采用空间数据集 P 和查询数据集 Q,数据集中的数据都经过规范化到(0,1]区间内. 实验中,空间数据集 P 和查询数据集 Q 都驻留在内存,实验结果数据取 50 次查询的平均值. 数据来源:(1) 美国西部地区一些公共地名数据集 WUpppoint,包含有 10 493 个数据点,即它们是一些二维坐标点,数据可从 www.rtreeportal.org 下载获得;(2) 人工随机生成的高维数据,这些数据在数据空间中是均匀分布的.

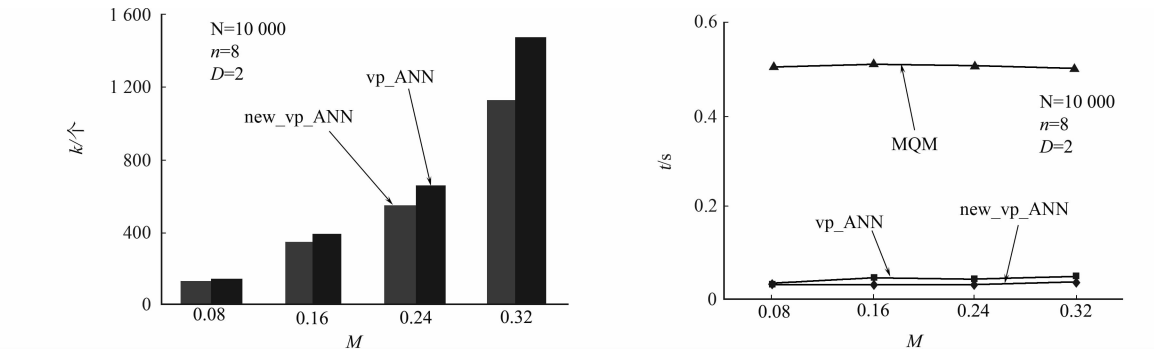
对不同的 n,M 及数据空间的维数(D)参数进行分析比较,结果如图 5 所示. 图 5 中:k 为查询区域中数据点的数量,t 为 CPU 执行时间. 从图 5 可知,基于最优点的集合最近邻算法查询区域中的数据点的数量总是比其改进的算法要多,而且随着 M 和 n 的增加,查询区域数据点的差别就更大. 但是,随着空间数据维数的增加,差别越小. 这是因为基于最优点的 ANN 的查询区域,总是覆盖且大于其改进算法的查询区域,随着维数的增加,数据空间中的数据分布比较稀疏. 所以,它们包含的点数的差别就越小.

实验结果表明,所提算法的效率要优于组最近邻居查询算法(Multiple Query Method,MQM),且

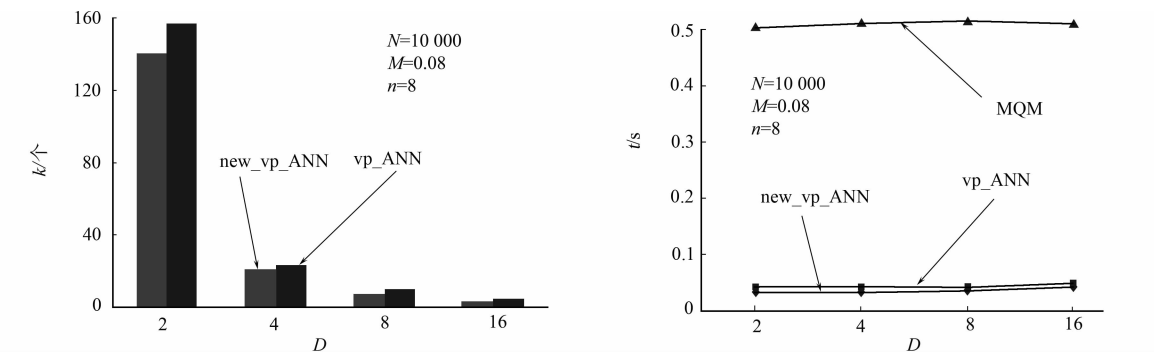
对于不同的 M，n 和数据维数 D，特别是对于高维数据空间，算法有比较高的稳定性. 由于查询区域中数据点的数量比较少，因此，改进的 ANN 算法的效率总体要比基于最优点的 ANN 算法要高一点.



(a) 数据点数量



(b) 最小边界矩阵的面积



(c) 数据空间维数

图 5　不同的查询数据集的算法比较

Fig. 5　Comparison on the different data set

5　结束语

提出一种无索引的，基于最优点集合的最近邻查找算法及其改进的算法. 将算法与其他查询方法进行比较，分析算法的基本特性. 结果表明：所提出的算法最大的特点在于不采用任何索引的情况下，也能高效地对空间数据库中的数据对象进行裁剪，缩小查询区域、提高集合最近邻查询的效率. 在下一步，将继续研究其他函数的集合最近邻算法，并且将算法扩展到向量空间中.

参考文献：

[1] YIU Man-lung, MAMOULIS N, PAPADIAS D. Aggregate nearest neighbor queries in road networks[J]. IEEE Trans Knowl Data Eng, 2005, 17(6): 820-833.

[2] JENSEN C S, KOLÁRVR J, PEDERSEN T B, et al. Nearest neighbor queries in road networks[C]// Proc of the 11th ACM International Symposium on Advances in Geographic Information Systems. New York: ACM, 2003: 1-8.

[3] JAIN A K, M. MURTY N M, FLYNN P J. Data clustering: A review[J]. ACM Comput Surv, 1999, 31(3): 264-323.

[4] AGGARWAL C C, YU P S. Outlier detection for high dimensional data[C]// Proc of the 2001 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2001: 37-46.

[5] PAPADIAS D, TAO Yu-fei, MOURATIDIA K, et al: Aggregate nearest neighbor queries in spatial databases[J]. ACM Trans on Database Syst, 2005, 30(2): 529-576.

[6] PAPADIAS D, SHEN Qiong-mao, TAO Yu-fei, et al. Group nearest neighbor queries[C]// Proc of the 20th International Conference on Data Engineering. Washington D C: IEEE Computer Society, 2004: 301-312.

[7] LI Hong-ga, LU Hua, HUANG Bo, et al. Two ellipse-based pruning methods for group nearest neighbor queries[C]// Proc of the 13th Aannual ACM International Workshop on Geographic Information Systems. Washington D C: IEEE Computer Society, 2005: 192-199.

[8] BEYER K, GOLDSTEIN J, RAMAKRISHNAN R, et al. When is "nearest neighbors" meaningful? [C]// Proc of the 7th International Conference on Database Theory. London: Springer-Verlag, 1999: 217-235.

[9] GUTTMAN A. R-trees: A dynamic index structure for spatial searching[C]// Proc of the 1984 ACM SIGMOD International Conference on Management of Data. San Francisco: Morgan Kaufmann Publishers Inc, 1984: 47-57.

[10] BECKMANN N, KRIEGEL H P, SCHNEIDER R, et al. The R*-tree: An efficient and robust access method for points and rectangles[C]// Proc of the 1990 ACM SIGMOD International Conference on Management of Data. New York: ACM, 1990: 322-331.

[11] PAPADOPOULOS A, MANOLOPOULOS Y. Performance of nearest neighbor queries in R-trees[C]// Proc of the 6th International Conference on Database Theory. London: Springer-Verlag, 1997: 394-408.

[12] UHLMANN J K. Satisfying general proximity/similary queries with metric trees[J]. Information Processing Letters, 1991, 40(4): 175-179.

[13] ROUSSOPOULOS N, KELLEY S, VINCENT F. Nearest neighbor queries[C]// Proc of the 1995 ACM SIGMOD International Conference on Management of Data. New York: ACM, 1995: 71-79.

[14] ISHIKAWA M, CHEN Han-xiong, FURUSE K, et al. MB+tree: A dynamically updatable metric index for similarity searches[C]// Web-Age Information Management. Berlin: Springer, 2000: 356-373.

[15] BRIN S. Near neighbor search in large metric spaces[C]// Proc of the 21th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers Inc, 1995: 574-584.

[16] BOZKAYA T, OZSOYOGLU M. Distance-based indexing for high-dimensional metric spaces[C]// Proc of the 1997 ACM SIGMOD Conference on Management of Data. New York: ACM, 1997: 357-368.

An Algorithm of Aggregate Nearest Neighbor Query for Non-Index Spatial Database

Luo Yan-min[1,2], CHEN Wei-bin[1], LIAO Min-hong[3]

(1. College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China;
2. School of Information Science and Technology, Xiamen University, Xiamen 361005, China;
3. School of Software, Xiamen University, Xiamen 361005, China)

Abstract: The vantage point-based aggregate nearest neighbor query algorithm and it's improved algorithm are proposed for non-index spatial database in metric space. The algorithms are tested by using both real datasets and synthetic datasets to evaluate efficiencies of the proposed algorithms. The experimental results show that the efficiencies of proposed algorithms are better than that of multiple query method. Furthermore, the proposed algorithms have higher stabilization in high-dimensional data space. Since there are less data points in it's search region, efficiency of the improved vantage point-based aggregate nearest neighbor query algorithm is generally higher than the vantage point-based aggregate nearest neighbor query algorithm.

Keywords: spatial database; nearest neighbor; aggregate nearest neighbor; search region

(责任编辑: 陈志贤 英文审校: 蔡灿辉)