基于 VAR 树的反向最近邻查询技术的研究

修建新[1,2],郝忠孝[1,3]

(1.哈尔滨理工大学 计算机科学与技术学院,黑龙江 哈尔滨 150080;
2.黑龙江东方学院 计算机科学与电气工程学部,黑龙江 哈尔滨 150086;
3.哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

摘 要:在空间数据库中,反向最近邻查询技术是最重要的查询技术之一,它是在最近邻查询技术的基础上提出的,如何有效地实现反向最近邻查询一直是人们研究的热点。以往都是基于类似 R 树索引结构的查询,在高维的情况下,使查询的速度急剧下降,形成"维数灾难"。因此引用了一种新的索引结构——VAR 树,并对 VAR 树进行了改进,引进了性能优越的 SR 树,并给出了基于这种索引结构的最近邻和反最近邻查询的算法。经实验验证基于 VAR 树的反向最近邻查询算法,在高维空间中的查询效率有了较大的提高。

关键词:SR-树;VAR 树;最近邻;反向最近邻查询
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2010)06-0051-04

Research of Reverse Nearest Neighbor Query Technology Based on VAR-Tree

XIU Jian-xin[1,2], HAO Zhong-xiao[1,3]

(1. College of Computer Science & Technology, Harbin University of Science and Technology, Harbin 150080, China;
2. Dept. of Computer Science and Electrical Eng., Heilongjiang East College, Harbin 150086, China;
3. College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: In the spatial database, the reverse nearest neighbor query is one of the most important queries, which is based on the nearest neighbor query, how to implement effectively the reverse nearest neighbor queries have been a hot issue. In the past, most are based on R-tree index structure similar to the query, in the case of high-dimensional, making the sharp decline in the rate of inquiries, a"dimension disaster". Propose a new index structure - VAR tree and introduce the SR tree of high performance after improving the VAR-tree, and give nearest neighbor queries and anti-nearest neighbor algorithms based on the structure. Experiments show that the algorithm of the reverse nearest neighbor queries based on VAR-tree enhances the query efficiency in high-dimensional space.

Key words: SR-tree; VAR-tree; nearest neighbors query; reverse nearest neighbors query

0 引 言

在空间数据库中,反向最近邻查询技术[1]是重要的查询技术之一,它是在最近邻查询技术[2]的基础上提出的。反向最近邻技术和人们平常所说的影响集问题是一致的,在市场预测和决策支持系统[3]中一个非常重要的任务是求得一个数据点相对于某一数据集的影响集,反最近邻查询技术正是为解决这个问题而产生的,可以很好地解决影响集求取问题,并且其自身对于空间数据库技术也有很重要的价值。

目前对反最近邻查询[4]的研究还很有限,所提出的查询方法也都不尽如人意,主要的缺陷是反最近邻查询所采用的索引结构都是从 R*-树演化而来的,高维情况下查询性能随着维数的增加急剧下降。因此,文中引用了一种新的索引结构——VAR 树,并给出基于这种索引结构的最近邻和反最近邻查询算法,在高维空间中可以提高反最近邻查询的性能。

收稿日期:2009-10-10;**修回日期**:2010-01-15
基金项目:黑龙江省自然科学基金资助项目(F200601)
作者简介:修建新(1979-),女,硕士,研究方向为空间数据库理论;郝忠孝,教授,博士生导师,研究方向为时空数据库理论、空值数据库理论、数据库数据组织的无环性理论研究。

1 VAR 树的索引结构

VAR 树最先是在《高维数据索引结构研究》这篇文献中提出的,这种索引结构将 VA-file 和 R-Tree 两类索引结构有机地结合起来,利用 R-Tree 管理

VA-file 中的近似数据。树形索引结构便于数据的存储，VA-file 可以对高维数据进行量化压缩并能够过滤出大量的无用数据，VAR 树正是结合了这两种索引结构的优点，有效地提高了查询效率。

VAR 树的构造过程包括对原始数据的近似和对近似数据构造索引结构两个步骤。首先对原始数据进行量化压缩，得到比特数远少于原始数据的近似矢量，量化方法和 VA-file 文件一样；VA-file 是将上面量化后的近似数据按顺序排列而得到的，在查询时需要扫描所有的近似数据，这样会花费很大的代价，所以文中不再采用 VA-file 的顺序结构，而是在用类似 R 树的索引结构来管理这些近似的数据。VAR 树和类 R 树一样包括叶子节点和目录节点，不过 VAR 树的叶子节点不再包含原始数据，而是量化后的近似数据，目录节点的 MBR 也不再是原始数据的最小邻接矩形，而是近似数据的对应的网格的最小邻接矩形。

VAR 树的索引结构如图 1 所示。

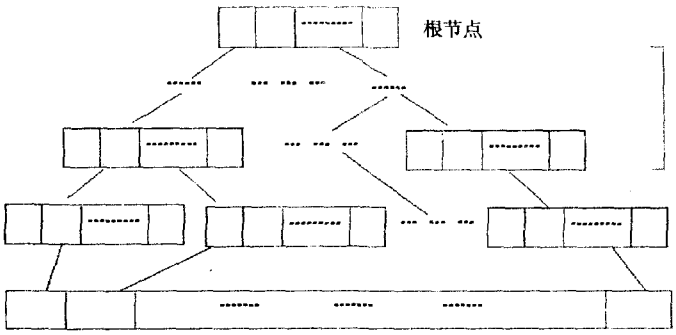

图 1 VAR 树的索引结构

2 基于 VAR 树的最近邻查询算法

2.1 VARNN 算法的思想

基于 VAR-tree 的最近邻查询算法[5]是在现有的最近邻查询算法的基础上做了一定的改进，主要的改进是采用了基于 VAR 树的索引结构，把查询矢量进行量化压缩为近似矢量，扫描每一个近似矢量，根据查询矢量同每个近似矢量所代表的网格之间的距离过滤掉大量无用数据，保留候选者，然后再读取所有候选者对应的原始数据精确计算得到最近邻查询结果。

2.2 VARNN 算法的实现

该算法首先用根节点中所有项的 mbr 初始化列表 mbrlist，对 mbrlist 以 Vq(查询矢量)与各 mbr 之间的距离下界 di 进行排序，如果 mbrlist 中的节点是叶子节点，就计算查询矢量与近似矢量之间的距离下界，读出所指的真实距离，得到最近邻查询点，并删除 mbrlist 中所有与 Vq 的距离下界大于 Ndist 的 mbr；如果不是叶子节点，将该 mbr 替换为对应的子节点中的所有 mbr，对 mbrlist 重新按照 Vq 与各 mbr 之间的距离下界 di 进行排序。

具体算法如下：

```
NNQuery(Vq)
Begin
NN=0
Ndist=Max // Ndist 是最近邻与查询点之间的距离
Mbrlist= base. mbr //用根节点中所有项的 mbr 初始化列表 mbrlist
order(d(i)) //对 mbrlist 以 Vq 与各 mbr 之间的距离下界 di 进行排序
While (mbrlist<>null)
If (mbrlist. first is leaf node) then
For (every leaf node. emtry) do // empty 是叶子节点中的每一数据项
dist=di(Vq,entry. appr) //计算查询矢量与近似矢量之间的距离下界
If (dist<Ndist) then
        Vdata= empty. pointer(Vq) //读出 empty. pointer 所指的原始数据
        dist=d(Vq,Vdata) //计算查询矢量与该数据的距离
        If (dist<Ndist) then
        NN=Vdata
        Ndist=dist
        Dele mbrlsit. mbr
        //删除 mbrlist 中所有与 Vq 的距离下界大于 Ndist 的 mbr
        Else if (mbrlist. first is a index node) then
将该 mbr 替换为对应的子节点中的所有 mbr
order(d(i))
//对 mbrlist 重新按照 Vq 与各 mbr 之间的距离下界 di 进行排序
NN 即为最近邻
Ndist 为最近邻到 Vq 之间的距离
End
```

2.3 VARNN 算法的理论证明

定理 1：VARNN 算法是正确的、可终止的，时间复杂度为 $O(n^2)$。

证明：

算法的正确性是显然的，基于 VAR 树的 NN 查询算法中对扫描近似矢量列表判断是否为叶子节点，是叶子节点读出原始数据得到最近邻，并删除无用的数据；不是叶子节点就重新排序列表再进行循环这一过程，最终可以得到最近邻查询点。

算法是可终止的，从算法中可以看到直到列表为空时就终止算法。

算法对列表进行排序的时间复杂度为 $O(n^2)$，对列表中的近似矢量进行循环判断，时间复杂度为

$O(n)$,如果节点是叶子节点的时间复杂度为$O(n^2)+O(n)$;如果不是叶子节点时间复杂度为$O(n^2)+O(n)+O(n^2)$,所以总的时间复杂度为$O(n^2)$。

3 基于VAR树的反向最近邻查询算法

3.1 RNN算法的研究现状

目前提出的RNN算法大概有以下几种:在1999年,Flip Korn和S. Muthukrishnan最先提出的基于Rnn-(R*-)树的查询方法[6];在2001年,C. Yang和K. I. Lin提出的基于Rdnn-(SS-)树的查询方法[7];还有人在近几年提出了基于SRdnn-(SR-)树的查询算法[8]。

第一种查询算法是最先提出了反向最近邻查询的概念和算法,有着重要的历史意义;第二种查询算法是在第一种的基础上引入了最近邻,提高了查询的速度;第三种是在第二种的基础上采用了SR-树这种索引结构,结合了R-树和SS-树的优点,SR-树的最突出的特点是节点记录中存储了下层节点或数据点的最小包围矩形和最小包围圆,也就是说下层数据存放在最小包围矩形和最小包围圆得到的相交区域,大大提高了查询的效率。但是这三种查询算法都是基于类R树的索引结构,势必导致了在高维空间中查询性能的显著下降,甚至是"维数灾难"。

3.2 VARRNN算法的思想

文中是在现有的反向最近邻查询算法的基础上做了改进,提出了基于VAR-tree的反向最近邻查询算法。本算法主要的改进是VAR-tree的索引结构中类R树采用了SR-树,用SR-树去管理VA-file中的近似数据,首先对原始的数据进行量化压缩得到近似数据,然后用SR-树来管理这些近似数据。与其它算法不同的是,VARRNN算法中VAR树的叶子节点不再包含原始数据,而是量化后的近似数据,目录节点也不再是原始数据的最小邻接矩形和最小邻接圆的相交区域,而是近似数据对应的网格的最小邻接矩形和最小邻接圆的相交区域。

VAR树的索引结构包括叶子节点和目录节点,叶子节点的记录形式为(p,dnn),p为任意点,dnn是p与它最近邻之间的距离,目录节点的记录形式为(S,R,W,childptr,max-dnn),其中S为下层节点的最小邻接矩形,R为下层节点的最小邻接圆,W为最小邻接矩形和最小邻接圆的相交部分,max-dnn=max(dnn(p)),p是以childptr指向的节点为根的子树内的点。

3.3 VARRNN算法的实现

该查询算法首先把根节点的所有数据项初始化到列表mbrlist中,对列表进行排序,如果mbrlist中的节点是叶子节点,计算查询矢量与近似矢量之间的距离下界,计算任意点p与查询点q之间的距离$D(p,q)$,如果$D(p,q)\leqslant$dnn,那么就读出所指的真实距离,并得到p就是q的一个反最近邻查询点,最后删除mbrlist中所有与Vq的距离下界大于Ndist的mbr;如果当前节点是目录节点,将该mbr替换为对应的子节点中的所有mbr,对mbrlist重新按照Vq与各mbr之间的距离下界di进行排序,再计算q和每一个记录(S,R,W,childptr,max-dnn)中的S的距离$D(q,S)$,如果$D(q,S)>$max-dnn,那么该记录childptr所指的子树将被剪除,否则递归调用本算法。

具体算法如下:

```
RNNQuery(n,q)
Begin
RNN=0
Ndist=Max // Ndist是最近邻与查询点之间的距离
Mbrlist= base. mbr //用根节点中所有项的mbr初始化列表mbrlist
order(d(i)) //对mbrlist以Vq与各mbr之间的距离下界di进行排序
While (mbrlist<>null)
If (mbrlist. first is leaf node) then
For (every leaf node. emtry) // empty是叶子节点中的每一数据项
dist=di(Vq,entry. appr) //计算查询矢量与近似矢量之间的距离下界
If (D(p,q)≤dnn) then
Vdata=empty. pointer(Vq) //读出empty. pointer所指的原始数据
dist=d(Vq,Vdata) //计算查询矢量与该数据的距离
If (D(p,q)≤dnn) then
RNN=Vdata
Ndist=dist
Dele mbrlsit. mbr
//删除mbrlist中所有与Vq的距离下界大于Ndist的mbr
Else if (mbrlist. first is a index node) then
将该mbr替换为对应的子节点中的所有mbr
order(d(i))
//对mbrlist重新按照Vq与各mbr之间的距离下界di进行排序
For (all entry(S,R,W,childptr,max-dnn) in n) do
If D(q,S)<max-dnn then
RNNQuery(childptr,q)
RNN即为反向最近邻
Ndist为反最近邻到查询点q之间的距离
End
```

3.4 VARRNN算法的理论证明

定理2:VARRNN算法是正确的、可终止的,时间

复杂度为 $O(n^2)$。

证明：

算法显然是正确的，基于 VAR 树的 RNN 查询算法中对扫描近似矢量列表判断是否为叶子节点，是叶子节点读出原始数据，再判断到查询点的距离是否小于最近邻 dnn，如果小于得到反最近邻；不是叶子节点就重新排序列表，计算 q 和每一个记录中的 S 的距离 $D(q,S)$，如果 $D(q,S) > \max - \text{dnn}$，那么该记录 childptr 所指的子树将被剪除，否则递归调用本算法，最终可以得到反向最近邻查询点。

算法是可终止的，从算法中可以看出直到列表为空时终止算法。

算法对列表进行排序的时间复杂度为 $O(n^2)$，对列表中的近似矢量进行循环判断，时间复杂度为 $O(n)$，如果节点是叶子节点的时间复杂度为 $O(n^2)+O(n)$；如果不是叶子节点时间复杂度为 $O(n^2)+O(n)+O(n^2)+O(n^2)$，所以总的时间复杂度为 $O(n^2)$。

4 实验与分析

4.1 VARNN 算法的查询性能

实验是在 Windows XP 系统上用 C 语言编程实现的，采用 GSTD 程序随机生成均匀分布的实验数据，根据数据集的大小输出不同的节点访问次数，数据的维数为 11 维。下面给出 11 维均匀分布数据集的 NN 查询性能曲线图，如图 2 所示。

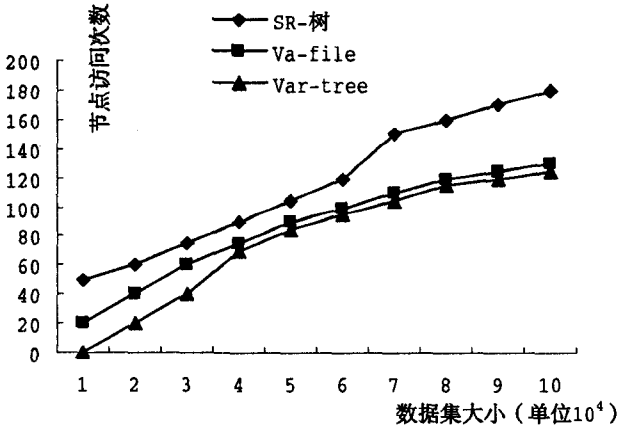

图 2 11 维均匀分布数据集的 NN 查询性能

4.2 VARRNN 算法的查询性能

实验环境与 NN 算法相同，下面给出 11 维均匀分布数据集的 RNN 查询性能曲线图，如图 3 所示。

图 3 显示了 NN 查询和 RNN 查询的数据集合的节点访问次数和数据大小的关系曲线，并与 VA－file 和 SR－树做了比较，曲线图中可以看出基于 Var－tree 的 NN 查询和 RNN 查询的节点访问次数少于基于 SR－tree 和 Va－file 的查询的节点访问次数，说明基于 Var－tree 的 NN 查询和 RNN 查询性能优于其他两种查询的性能。

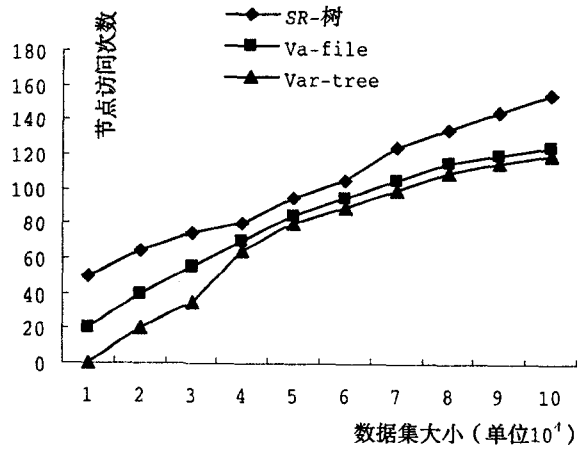

图 3 11 维均匀分布数据集的 RNN 查询性能

5 结束语

空间数据库中，反向最近邻查询技术有很重要的研究价值和应用价值。文中在以往研究的基础上，提出了新的索引结构 VAR－tree，并给出基于这种索引结构的最近邻和反向最近邻的查询算法，该索引结构解决了高维空间中树型索引结构的缺陷，提高了查询的效率。同时存在一些缺点和不足，如对于近似数据的排序增加时间复杂度等。

以后需要改进的方面：

(1)对 VAR－tree 索引结构做进一步的改进，提高索引能力；

(2)对反向最近邻查询算法进行进一步优化，提高查询效率。

参考文献：

[1] 郝忠孝，刘永山．空间对象的反最近邻查询[J]．计算机科学，2005，32(11)：115－118．

[2] 程 淼，胡 圣，袁正午．时空数据库中多个最近邻对象的查询算法[J]．计算机工程，2006，32(19)：60－62．

[3] 陈述彭，鲁学军，周成虎．地理信息系统导论[M]．北京：科学出版社，1999：1－8．

[4] 张 奋，肖政宏．基于 SR－树的空间对象反最近邻查询技术研究[J]．西华大学学报，2007，26(3)：44－47．

[5] Roussopoulos N, Kelley S, Vincent F. Nearest Neighbor Queries[C]//The 1995 ACM SIGMOD International Conference on Management of Data. SanJose, California, USA: [s. n.], 1995: 71－79.

[6] Korn F, Muthukrishnan S. Influence sets based on reverse nearest neighbor queries[C]//The 2000 ACM SIGMOD In-

（下转第 58 页）

3 仿真与结果分析

文中用 C++ 实现了 LEACH 和文中提出的改进路由算法，并设计了表 2 所示的仿真参数。

表 2 仿真参数表

参数名	值
网络范围	(0,0)到(300,300)
sink 位置	(150,200)
节点总数 N	400
节点初始能量	3J
E_{DA}	5nJ/bit/signal
E_{elec}	50nJ/bit
ε_{fs}	10pJ/bit/m2
ε_{amp}	0.0013nJ/bit/m4

由式(12)，得簇的数目最优值 $k_{opt}=23$；由于 LEACH 算法簇数目占节点数的 5%，故 $k_{leach}=20$。

图 2 给出了运行 LEACH 和改进算法的结果，可以看出，在第 325 轮时，LEACH 算法已几乎没有存活的节点，而改进算法还有大约 90 个活节点，可见改进而来的新算法比 LEACH 算法进一步延长了无线传感器网络的生存期。

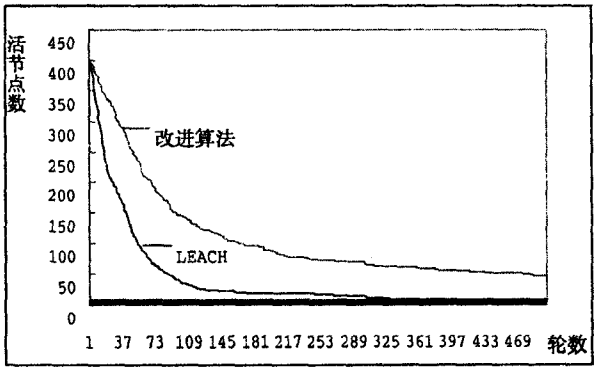

图 2 仿真结果

4 结束语

论文以节省能耗、延长网络寿命为目标，对无线传感网的路由算法进行了研究。基于 LEACH 的思想，通过改进，设计了一种新的多跳路由算法，描述了算法，分析了算法的能耗，用 C++ 进行了算法实现，比较了新算法与 LEACH 的运行结果，证明了算法在节能方面的作用。文中在延长无线传感网的生存期方面做了有益的研究工作。

参考文献：

[1] 孙利民，李建中，陈 渝，等. 无线传感器网络[M]. 北京：清华大学出版社，2005：3－4.

[2] Shah R C, Rabaey J M. Energy aware routing for low energy ad hoc sensor networks[C]//Proc. of IEEE Wireless Communications and Networking Conference. Piscataway, USA: [s. n.], 2002: 17－21.

[3] 毕俊蕾，任新会，郭拯危. 无线传感器网络路由协议分类研究[J]. 计算机技术与发展，2008，18(5)：131－134.

[4] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An Application－specific Protocol Architecture for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660－670.

[5] Guo Shujie, Zheng Jie, Qu Yugui, et al. Clustering and multi－hop routing with power control in wireless sensor networks [J]. The Journal of China Universities of Posts and Telecommunications, 2007, 14(1): 49－57.

[6] 杨 冕，秦前清. 基于无线传感器网络的路由协议[J]. 计算机工程与应用，2004(32)：130－132.

[7] 汪泉弟，李 彬，刘青松. 无线传感器网络能量多路径路由研究[J]. 信息与控制，2006，35(2)：130－132.

[8] Bandyopadhyay S, Coyle E J. Minimizing communication costs in hierarchically clustered networks of wireless sensors[J]. Wireless Communications and Networking, 2003(2): 1274－1279.

[9] 熊昊翔，李 峰，李 平. 基于节能的无线传感器网络 LEACH 协议改进[J]. 计算机技术与发展，2007，17(11)：237－240.

[10] 张 怡，李 云，刘占军，等. 无线传感器网络中基于能量的簇首选择改进算法[J]. 重庆邮电大学学报：自然科学版，2007，19(5)：613－616.

[11] 刘 芳，徐家品. 一种基于簇的无线传感器网络能力有效路由协议[J]. 传感器与微系统，2008，27(3)：34－36.

[12] 敬海霞，胡向东. 无线传感器网络的路由协议研究[J]. 计算机技术与发展，2007，17(10)：150－154.

(上接第 54 页)

ternational Conference on Management of Data. Dallas, Texas, USA: [s. n.], 2000: 201－212.

[7] Yang C, Lin K I. An Index Structure for Efficient Reverse Nearest Neighbor Queries[C]// Proceedings of the IEEE International Conference on Data Engineering. Washington: IEEE Computer Society, 2001: 485－492.

[8] Katayama N, Satoh S. The SR－tree: An Index Structure for High Dimensional Nearest Neighbor Queries[C]// Proc. ACM SIGMOD Int. Conf. on Management of Data. [s. l.]: [s. n.], 1997: 703－717.