基于质心片的不确定高维索引研究

庄 毅[1],胡海洋[2],胡 华[2]
(1.浙江工商大学计算机与信息工程学院,浙江杭州 310018; 2.杭州电子科技大学计算机学院,浙江杭州 310018)

摘 要: 提出一种基于质心片的(CU-Tree)不确定高维索引结构.对于高维空间中的不确定数据对象,首先通过k平均聚类算法将其聚成若干类,然后分别计算每个不确定超球进行质心"切片",并对其进行复合编码得到对应的统一索引键值,并且用B+树建立索引.这样,高维空间的概率查询就转变成对一维空间的启发式的范围查询及求精运算.实验证明该方法能更有效地缩小搜索空间,减少积分计算的代价.实验都表明,CU-Tree索引在查询效率方面要明显优于其它的索引方法,尤其适合海量高维不确定数据的查询.

关键词: 概率范围查询; 分片; 不确定超球; 质心片
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2011) 05-1136-07

Centroid-Slice-Based Uncertain High-Dimensional Indexing Structure

ZHUANG Yi[1], HU Hai-yang[2], HU Hua[2]
(1. *College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, Zhejiang* 310018, *China*;
2. *School of Computer, Hangzhou Dianzi University. Hangzhou, Zhejiang* 310018, *China*)

Abstract: This paper proposes a centroid-slice-based uncertain high-dimensional indexing algorithm, called CU-Tree. In the CU-Tree, all (n) data objects are first grouped into some clusters by a k-Means clustering algorithm. Then each object's corresponding uncertain sphere is "sliced" in terms of the centroid-distance. Finally a unified key of each data object is computed by adopting composite encoding scheme, which are inserted by a B^+-tree. Thus, given a query object, its probabilistic range search in high-dimensional spaces is transformed into the search in the single dimensional space with the aid of the CU-Tree. Extensive performance studies are conducted to evaluate the effectiveness and efficiency of the proposed scheme.

Key words: probabilistic range query; partition; uncertain sphere; centroid-slice

1 引言

近几十年来,随着计算技术的发展,数据管理技术也得到了迅猛的发展.以Oracle、DB2、SQL Server等为代表的关系数据库管理系统(Relational Database Management System, RDBMS)已成为诸多大型信息管理系统不可或缺的核心部分.同时,以可扩展标记语言(Extensible Markup Language, XML)为代表的半结构化数据管理技术也在数据交换和缺乏严格结构的数据管理方面占据一席之地.上述技术均对数据质量、待处理数据的准确性要求非常高.当原始数据的质量不高时,需要先经过预处理过程提升数据质量.数据的不确定性在诸如经济、军事和电信等领域普遍存在,其存在性未知而且各属性值存在误差.近年来,相继开发出一些不确定数据管理系统,比较有名的如,美国Stanford大学的Trio系统[1]、Washington大学的MystiQ项目[2]、Purdue大学的Orion项目[3]和牛津大学的MayBMS[4]等.国内对不确定数据的研究尚处于起步阶段[5],丁晓峰等人[6]提出移动环境的不确定移动对象索引方法.

目前,不确定数据,特别是高维不确定数据的查询处理正受到了越来越多的关注[7],已成为国内外学术界关注的一个重要研究课题,具有较强的理论研究价值及现实应用前景[5].与高维确定数据相似查询[8](similarity query)不同,高维不确定数据的研究将概率引入到高维数据模型中来衡量不确定对象成为结果集中元素的可能性,即每个不确定对象可表示为具有一定概率密度函数(Probability density function)的记录.因此对高维不确定数据采用传统的查询方法(如R-Tree[9]、VA-File[11]等)难以对其进行有效处理,往往会导致查询结果出现偏差,不能满足用户的需求.同时,由于其概率查询存在

收稿日期:2009-10-02;修回日期:2010-03-10
基金项目:国家自然科学基金(No.61003074, No.60873022, No.60903053);浙江省自然科学基金(No.Z1100822, No.Y1080148, No.Y1090165, No.Y1110644, No.Y10969);浙江工商大学青年人才基金重点项目(No.Q09-7);南京大学软件新技术国家重点实验室开放基金;温州市科技计划项目(No.2010G0066)

大量积分运算,因此处理代价非常之高[2].而且随着数据量的增长,其查询效率往往并不理想.因此,国内外众多学者纷纷提出一些索引方法减少概率查询中的CPU计算代价,提高查询效率.

与确定性高维索引不同的是,不确定高维索引所针对的数据对象是随着时间推移而变化的、不确定的,如移动对象的运动轨迹[8]、传感器采集得到的数据[12]等.较确定性高维相似查询来说,由于不确定高维概率查询计算代价非常之大.为了加快其检索效率,需要对其建立索引机制.不确定高维数据索引属于高维索引范畴,根据数据模型的不同,该技术分为确定性高维索引和非确定高维索引[13].其中确定性高维索引大体分为三类:(1)基于数据和空间分片的树形索引,如R-tree[13]及其变种[9]等.由于"维数灾难"的存在,这些方法仅适合维数较低的情况;(2)采用近似的方法来表示原始向量,如VA-file[14]及其变种[11,15]等;(3)采用基于距离尺度的方法,如iDistance[7].Cheng等人[19]较早提出一种基于R-tree的最近邻概率查询(PNN)的索引方法.Kriegel等人[17]提出另一种加速执行PNN查询的方法,其中每个对象表示为一组从该对象对应的连续的概率密度函数采样得到点构成.最近,文献[16,21]分别提出采用一个对象存在于数据库中的概率(称为存在概率)来推导出下界和上界,从而对求得其对应的最近邻对象进行有效的"裁剪".Tao等人[20]提出一种多维不确定索引——U-Tree以支持概率范围查找,但该方法对高维概率查询效果不十分理想.在文献[18]中,Lian等人提出一种高效检索离查询对象集的聚合距离最小的数据对象的算法.为了提高PNN查询的判定概率的计算,Cheng等人[19]又提出PNN查询的变种,它使用概率阈值作为结果的约束条件,已开发出一种有效的验证方法用于推导出对象判定概率的下界和上界.这些方法不太适合 k 近邻概率查询(k-PNN)(其中 $k\geqslant 1$).针对不确定数据的 k 近邻概率查询(k-NN),Soliman等人[16]提出一种新型的查询类型,它会对每个对象作为查询对象 q 的最近邻的概率进行排序,返回出现概率最高的 k 个对象.在文献[12]中,Ljosa等人提出一种高效的索引结构——APLA-tree,用以加速k-NN查询.该方法采用每个对象离查询对象 q 的不确定pdf的期望距离(如L1-norm)作为ranking约束条件.这样,其k-NN查询是基于期望距离,在其查询结果中无概率.

针对概率范围查询中的高计算代价问题,提出一种基于质心片的不确定高维索引方法——CU-Tree(Centroid-Slice-based Uncertain High-Dimensional Indexing Tree),以支持高效的概率范围查询.该方法通过对每个不确定超球进行基于质心距离的"切片"编码,同时借助双重尺度(即编码和存在概率),表达成一个统一的索引键值,这样可以将高维空间的范围概率查询转化成一维空间的基于启发式的范围查询及求精运算.较传统索引,如顺序检索、U-Tree[20],CU-Tree能显著缩小搜索空间,从而更有效地过滤掉不相关的点.实验表明该方法能有效提高查询效率,尤其适合海量不确定高维数据的检索.

2　CU-Tree索引

2.1　问题定义及动机

表1是本文将要用到的符号.

表1　常用符号

符号	意义	符号	意义
Ω	高维不确定数据库且 $\Omega=\{\boldsymbol{V}_1,\boldsymbol{V}_2,\cdots,\boldsymbol{V}_n\}$	pdf	概率密度函数
n	不确定对象个数	Prob	概率
$\boldsymbol{V}_i$	第 i 个不确定对象	$Vol(\cdot)$	$\cdot$ 的体积
$\boldsymbol{V}_q$	查询点	$d(\boldsymbol{V}_i\boldsymbol{V}_j)$	相似距离
r	查询半径	$\Theta(\boldsymbol{V}_q,r)$	查询超球

定义1(高维不确定对象)　高维不确定对象 $\boldsymbol{V}_i$ 是一个 D 维空间中的数据点,并且满足以下两个条件:(1) $\boldsymbol{V}_i$ 存在一个概率密度函数(pdf);(2) $\boldsymbol{V}_i$ 对应一个不确定区域,即其在该不确定区域内活动,出现概率满足 pdf,其中 $\boldsymbol{V}_i\in\Omega$ 且 $i\in[1,n]$.

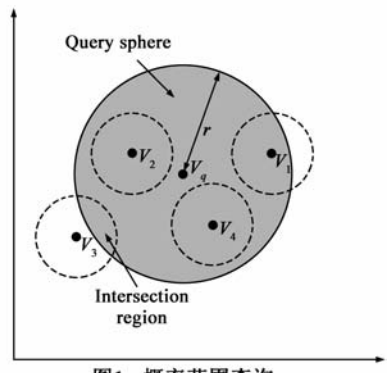


图1　概率范围查询

定义2(高维不确定区域)　给定一个高维不确定对象 $\boldsymbol{V}_i$,其对应不确定区域可表示为以 $\boldsymbol{V}_i$ 为中心,ε 为半径的一个超球,记作 $UR(\boldsymbol{V}_i)=\Theta(\boldsymbol{V}_i,\varepsilon)$,其中 ε 是对应不确定区域的活动半径且 $\boldsymbol{V}_i\in\Omega$ 同时 $\boldsymbol{V}_i$ 在 $\Theta(\boldsymbol{V}_i,\varepsilon)$ 里的出现满足一个概率密度函数(pdf_i).

根据定义2,如图2所示,不确定对象 $\boldsymbol{V}_i$ 会在阴影区域内按照一定概率密度函数(pdf_i)分布出现.简单起见,假设高维空间任意一点 $\boldsymbol{V}_i$ 在阴影区域的出现概率满足均匀分布,则其概率密度函数为 $pdf_i=1/Vol(\Theta(\boldsymbol{V}_i,\varepsilon))$.图1为4个不

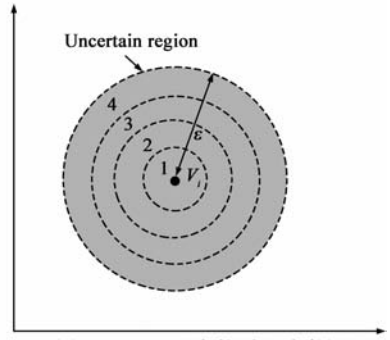


图2　$\Theta(\boldsymbol{V}_i,\varepsilon)$对应的质心片举例

确定对象(如 $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, $\boldsymbol{V}_3$ 和 $\boldsymbol{V}_4$)对应的不确定区域,用虚线圆表示.

定义 3(高维不确定数据库) 高维不确定数据库(Ω)由 n 个高维不确定对象 $\boldsymbol{V}_i$,构成,记作 $\Omega=\{\boldsymbol{V}_1,\boldsymbol{V}_2,\cdots,\boldsymbol{V}_n\}$且 $\boldsymbol{V}_i\in\Omega$.

定义 4(高维范围概率查询) 给定查询对象 $\boldsymbol{V}_q$、查询半径 r 和阈值 T,其范围概率查询返回向量 $\boldsymbol{V}_i$,使得 $\boldsymbol{V}_i$ 满足其出现在以 $\boldsymbol{V}_q$ 为中心 r 为半径的区域中的概率大于 T,记作 Prob($\boldsymbol{V}_i$ 出现在$\Theta(\boldsymbol{V}_q,r)$中) > T,其中 $\boldsymbol{V}_i\in\Omega$.

假设 $\boldsymbol{V}_i$ 在不确定区域($\Theta(\boldsymbol{V}_i,\varepsilon)$)出现概率满足均匀分布,即其概率密度函数为:$pdf_i=1/Vol(\Theta(\boldsymbol{V}_i,\varepsilon))$,则出现概率表示为:

$$\begin{aligned}\text{Prob}(\boldsymbol{V}_i\ \text{falls in}\Theta(\boldsymbol{V}_q,r)) &= \int_{\Theta(V_i,\varepsilon)\cap\Theta(V_q,r)} pdf_i\cdot \mathrm{d}v\\ &= \int_{\Theta(V_i,\varepsilon)\cap\Theta(V_q,r)} \frac{1}{Vol(\Theta(V_i,\varepsilon))}\cdot \mathrm{d}v\\ &= \frac{Vol(\Theta(V_i,\varepsilon)\cap\Theta(V_q,r))}{Vol(\Theta(V_i,\varepsilon))} \qquad (1)\end{aligned}$$

基于质心片的不确定高维索引(CU-Tree)的提出的目的是减少概率范围查询的 CPU 和 IO 代价.

2.2 基于质心片的不确定超球编码

由于概率查询涉及大量积分运算,为了提高查询效率,提出一种基于质心距离的分片编码方式.通过预先计算不同质心片的概率,使得尽可能减少查询中的概率计算量.

定义 5(质心距离) 给定一个点 $\boldsymbol{V}_i$,其质心距离(记为 Centroid-Distance, CD)是其到所在类 $\boldsymbol{C}_j$ 的质心 $\boldsymbol{O}_j$ 的距离,表示为 $CD(\boldsymbol{V}_i)=d(\boldsymbol{V}_i,\boldsymbol{O}_j)$且 $\boldsymbol{V}_i\in\Theta(\boldsymbol{O}_j,\boldsymbol{CR}_j)$,其中 $i\in[1,|C_j|]$且 $j\in[1,K]$.

假设 n 个对象通过 K 平均聚类得到 K 个类.对于任意一个类 $\boldsymbol{C}_j$,其中 $j\in[1,K]$,该类中对象的个数记为 $|\boldsymbol{C}_j|$且满足 $\sum_{j=1}^{K}|C_j|=n$.

定义 6(类半径) 对于任意一个类 $\boldsymbol{C}_j$,其质心 O_j 与该类中距离其最远的对象的距离,称为它的类半径,记作 CR_j,其中 $j\in[1,T]$.

定义 7(类超球) 给定任意一个类 $\boldsymbol{C}_j$ 和类半径 CR_j,类超球表示为 $\Theta(\boldsymbol{O}_j,\boldsymbol{CR}_j)$.

定义 8(质心片,Centroid-Slice) 对于任意一个不确定超球 $\Theta(\boldsymbol{V}_i,\varepsilon)$,按照质心距离对其均匀切分成 τ 片,将该类超球中的第 λ 个质心片表示为 $CS(\lambda,i)$,其中 $\lambda\in[1,\tau]$且 $i\in[1,n]$.

由于可以用不确定区域 $\Theta(\boldsymbol{V}_i,\varepsilon)$中的随机对象 $\boldsymbol{X}_t$ 来模拟 $\boldsymbol{V}_i$ 在不确定区域中的出现且$t\in[1,n_1]$,其中 n_1 表示 $\boldsymbol{V}_i$ 在不确定区域中出现的次数.因此质心片的编号基于以下原则:假设随机点 $\boldsymbol{X}_t\in\Theta(\boldsymbol{V}_i,\varepsilon)$,该点对应质心片的编号随着其质心距离的增加而减少.如图 2 所示.

对于每个不确定超球 $\Theta(\boldsymbol{V}_i,\varepsilon)$来说,将其均匀切分成 τ 个质心片,则该不确定超球中的任意一随机点 X_t 可以用三元组表示:

$$\boldsymbol{X}_t::=<t,CID,CD\_ID> \qquad (2)$$

其中 CID 表示 $\boldsymbol{X}_t$ 所在类的编号,CD_ID 表示 X_t 所在的质心片编号且 $CD\_ID(\boldsymbol{X}_t)=1+\lceil\frac{\varepsilon-CD(X_t)}{\varepsilon/\tau}\rceil$.

一般来说,对于 τ 个分片,其编码的组合共有 $\tau(1+\tau)/2$.由于查询超球与不确定超球相交部分的质心片是连续的,所以假设其相交部分的质心分片编号为 $[LBC,UBC]$,即从第 LBC 片到第 UBC 片,则其对应的编码为:

$$H=LBC^2+UBC^2 \qquad (3)$$

表 2 为不确定超球 $\Theta(\boldsymbol{V}_i,\varepsilon)$的编码举例.如表 2 所示,将不确定超球分成 4 个质心片,共有 10 组.基于以上的编码规则,得到高维数据的编码算法,如算法 1 所示.

表 2 质心片编码举例

CD_ID	组合	LBC	UBC	编码值(H)
1	{1}	1	1	2
2	{1,2}	1	2	5
3	{1,2,3}	1	3	10
4	{1,2,3,4}	1	4	17
	{2}	2	2	8
	{2,3}	2	3	13
	{2,3,4}	2	4	20
	{3}	3	3	18
	{3,4}	3	4	25
	{4}	4	4	32

算法 1 基于质心片的编码

输入: Ω:n 个高维不确定对象集

输出: $H[1,n\tau(1+\tau)/2]$:$n\tau(1+\tau)/2$ 个质心片对应的编码;

```
for V_i ∈ Ω and do
    for X_t ∈ Θ(V_i, ε) do
        计算 X_t 对应的质心距离;
        根据式(3),得到 X_t 对应的质心片编码;
```

2.3 数据结构

定义 9 (质心片的下界编号,Low Bound Id of Centroid-Slice) 给定两个相交的超球 $\Theta(\boldsymbol{V}_q,r)$和 $\Theta(\boldsymbol{V}_i,\varepsilon)$,$\Theta(\boldsymbol{V}_i,\varepsilon)$对应质心片的下界编号是指离类质心 $\boldsymbol{O}_j$ 最远的质心片编号,表示为 $LBC(i)$.

定义 10 (质心片的上界编号,Upper Bound Id of Centroid-Slice) 给定两个相交的超球 $\Theta(\boldsymbol{V}q,r)$和 $\Theta(\boldsymbol{V}_i,\varepsilon)$,$\Theta(\boldsymbol{V}_i,\varepsilon)$对应质心片的上界编号是指离类质

心 O_j 最近的质心片编号,表示为 $UBC(i)$.

为了能有效地将存在概率值($Prob(\boldsymbol{V}_i)$)与不确定区域(超球)中分片的统一编号($H(\boldsymbol{V}_i)$)结合起来组成一个有效的索引键值,在分片编码基础之上,提出一种索引键值的统一表达方法.该方法将 $H(\boldsymbol{V}_i)$与 $Prob(\boldsymbol{V}_i)$通过线性组合表达成一个统一的索引键值,如下所示:

$$key(\boldsymbol{V}_i) = H(\boldsymbol{V}_i) + Prob(\boldsymbol{V}_i) \tag{4}$$

其中 $H(\boldsymbol{V}_i)$为 $\boldsymbol{V}_i$ 所在的质心分片对应的编码且为大于 1 的整数. $Prob(\boldsymbol{V}_i)$表示为随机点 X_i 落入从第 LBC 个质心片到第 UBC 个质心片的不确定区域的概率且 $Prob(\boldsymbol{V}_i) < 1$.因此可以将它们合并在一个索引键值表示.另外,为了得到 $Prob(\boldsymbol{V}_i)$,假设在该区域随机产生 n_1 个点($\boldsymbol{X}_t$),则当前的 $Prob(\boldsymbol{V}_i)$可表示为: $\sum_{t=1}^{n1} pdf(\boldsymbol{X}_t)$且 $t \in [1, n_1]$.

由于式(4)不包含任何对象及其对应类信息,为了将这些信息包含其中,将上式改写为式(5)所示:

$$\begin{aligned} key(\boldsymbol{V}_i) &= c_1 \times CID + c_2 \times i + H(\boldsymbol{V}_i) + Prob(\boldsymbol{V}_i) \\ &= c_1 \times CID + c_2 \times i + LBC^2 + UBC^2 + \sum_{t=1}^{n1} pdf(\boldsymbol{X}_t) \end{aligned} \tag{5}$$

其中常数 c_1 和 c_2 分别是两个较大的整数,使得每个类中的对象对应的键值进一步线性放大,使其值域不重叠,其中 $c_1 \gg c_2$. CU-Tree 索引的创建步骤如算法 2 所示.

算法 2　索引创建

输入:　Ω:高维数据库;
输出:　bt:CU-Tree 不确定高维索引;
1.　采用 K 平均聚类算法将高维对象聚成 K 类;
2.　对于每个类超球中的任意不确定超球 $\Theta(\boldsymbol{V}_i, \varepsilon)$来说
3.　　将其均匀分为 τ 个质心片;
4.　　根据算法 1,得到对应的编码;
5.　　根据公式(5),计算该对象对应的键值用 B + tree 建立索引;
6.　返回 CU-Tree 索引 bt;

3　不确定概率范围查询算法

上节介绍了 CU-Tree 索引的创建算法.本节讨论运用该索引进行概率查询.不失一般性,首先假设 $\Theta(\boldsymbol{V}_q, r)$与不确定超球 $\Theta(\boldsymbol{V}_i, \varepsilon)$相交,研究超球 $\Theta(\boldsymbol{V}_i, \varepsilon)$中的哪些分片与 $\Theta(\boldsymbol{V}_q, r)$相交.

如图 3 所示,不确定超球 $\Theta(\boldsymbol{V}_i, \varepsilon)$被切分成 4 个质心片.与 $\Theta(\boldsymbol{V}_q, r)$相交的质心片共 2 个,是第 1 和第 2 个质心片,表示为 $LBC(j) = 1$, $UBC(j) = 2$.对于查询超球 $\Theta(\boldsymbol{V}_q, r)$,对应的质心距离的查询范围为[$CD(\boldsymbol{V}_q) - r$, $CD(\boldsymbol{V}_q) + r$].同时,假设不确定超球$\Theta(\boldsymbol{V}_i, \varepsilon)$分别被切分成 τ 个质心片.当查询超球与该不确定超球相交时,对于质心片来说,一定存在一些连续的分片(如从第 $LBC(i)$个质心片到第 $UBC(i)$个质心片,其中 $LBC(i) \leqslant UBC(i)$被相交.因此得到质心片的下界(LBC)、上界(UBC)的编号:

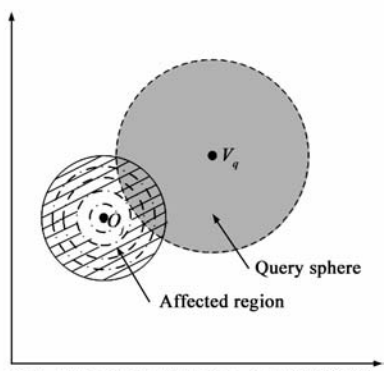


图3　基于质心片的$\Theta(\boldsymbol{V}_q, r)$对应的搜索区域

$$LBC(j) = \begin{cases} \lceil \dfrac{CD(\boldsymbol{V}_q) - r}{2\varepsilon/\tau} \rceil + 1, & \text{if} \quad 0 < CD(\boldsymbol{V}_q) - r < \varepsilon \\ 1, & \text{if} \quad CD(\boldsymbol{V}_q) - r \leqslant 0 \end{cases} \tag{6}$$

$$UBC(j) = \begin{cases} \lceil \dfrac{CD(\boldsymbol{V}_q) + r}{2\varepsilon/\tau} \rceil + 1, & \text{if} \quad CD(\boldsymbol{V}_q) + r < \varepsilon \\ \tau, & \text{if} \quad CD(\boldsymbol{V}_q) + r \geqslant \varepsilon \end{cases} \tag{7}$$

其中$\lceil \cdot \rceil$表示$\cdot$的整数部分.

如图 4 所示,查询前,由于已经对 n 个不确定对象进行聚类,得到 K 个类超球.因此可以先判断查询超球与这 K 个类超球是否相交.如不相交,则可以很快排除这些类中包含的不确定对象.否则,将每个与查询超球相交的类超球中不确定超球与查询超球判断是否相交,对相交的不确定对象,通过 CU-Tree 索引快速进行得到其相交部分的近似概率.具体如图 6 所示,由于相交部分小于或等于从第 LBC 到第 UBC 质心片的部分,所以概率表示为:

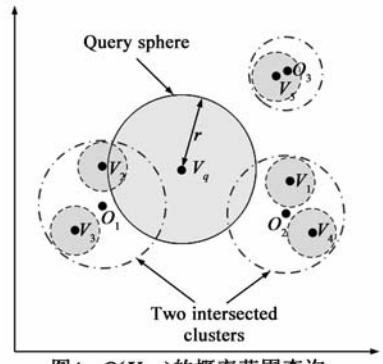


图4　$\Theta(\boldsymbol{V}_q, r)$的概率范围查询

$$\text{Prob} = \int_{\Theta(V_i,\varepsilon) \cap \Theta(V_q,r)} pdf \cdot \mathrm{d}v \simeq \sum_{i=1}^{n_1} pdf(\boldsymbol{X}_i) \tag{8}$$

对任意一个不确定超球 $\Theta(\boldsymbol{V}_i, \varepsilon)$来说,当 $\sum_{i=1}^{n_1} pdf(\boldsymbol{X}_i) < T$,则放弃概率计算.否则进一步进行求精计算(refinement),即精确求得出现概率.假设查询超球与不确定超球相交部分的质心片为[LBC, UBC],则其对应的编码为 $LBC^2 + UBC^2$.又因为出现概率取值范围为[0, 1],所以求索引键值的取值范围为:[$c_1 \times CID + c_2 \times i +$

$LBC^2 + UBC^2, c_1 \times CID + c_2 \times i + LBC^2 + UBC^2 + 1]$.

算法 3 为以 $\boldsymbol{V}_q$ 为中心，r 为半径和 T 为阈值的范围概率查询函数，其中函数 *Refinement*($\boldsymbol{V}_q, r, \boldsymbol{V}_i$)用于对候选对象精确求得其出现概率；*BRSearch*(*left*, *right*, *j*)用于对第 *j* 个子索引进行标准的范围查询.

算法 3 *pR*Search($\boldsymbol{V}_q, r, T$)

输入：查询点 $\boldsymbol{V}_q, r, T$

输出：查询结果 S

```
1.  S←Φ, S1←Φ;          /* 初始化 */
2.  for each cluster sphere Θ(Oj, CRj) do
3.    if Θ(Oj, CRj) dose not intersect with Θ(Vq, r) then
4.      break;
5.    else if Θ(Oj, CRj) is contained by Θ(Vq, r) then
6.      return the all points in Θ(Oj, CRj) to S1;
7.      break;
8.    else
9.      for each uncertain sphere Θ(Vi, ε)∈Θ(Oj, CRj) do
10.       if Θ(Vi, ε) intersects with Θ(Vq, r) then
11.         S1←GetProb(Vq, r, CID);
12.         if S1 < T then end loop
13.         else Refinement(Vq, r, Vi); /* 计算出现概率 */
14.   S←S∪S1;
15. return S;          /* 返回候选点 */

GetProb(Vq, r, j)
16. get the LBC(j), UBC(j);
17. left←c1 × CID + c2 × j + LBC^2 + UBC^2;
18. right←c1 × CID + c2 × j + LBC^2 + UBC^2 + 1;
19. S3←BRSearch[left, right, j];
20. return S3 中的小数部分;

Refinement(Vq, r, Vi)
21. prob = 0;
22. for each Xj∈Θ(Vq, r) and Xj∈Θ(Vi, ε) do
23.     prob = prob + Σ_{j=1}^{n1} pdf(Xj)
24. return prob
```

4 实验分析

本节通过实验来验证该算法的有效性，同时与其它索引，如 U-Tree 和顺序检索作比较.我们用 C 语言实现了基于质心片的不确定高维索引——CU-Tree，同时实现或下载了 U-Tree 等高维索引算法.采用 B+树作为单维索引结构.所有实验的运行环境为 Pentium IV CPU 2.4GHz，1 GB 内存，硬盘大小为 120G 且 7200 转/分，同时索引页大小设为 4096 Bytes.实验中的测试不确定高维数据分为两类：(1)对 UCI 提供的颜色直方图数据[22]进行改进，使其包含从 Corel 图片库提取 68040 个 32 维的颜色直方图特征，每个直方图数据对应一个不确定区域，其中不确定区域半径 $\varepsilon = 0.1$，每一维的值的范围都在 0 和 1 之间；(2)计算机随机产生的 100,000 个 64 维的均匀分布的合成数据，其中每一维值的范围也在 0 和 1 之间且每个数据点对应一个不确定区域，其中不确定半径 $\varepsilon = 0.3$.以下实验分别将索引磁盘块访问数及 CPU 运算开销作为衡量查询性能的两个指标.

4.1 分片数(τ)对查询的影响

第一组实验研究分片数对查询性能的影响.实验采用 100,000 个合成数据作为测试数据，其中维数为 32.从图 5 中看出随着分片数的增加，CU-Tree 索引的查询效率逐步提高，当分片数超过 8 时，查询效率就不再提高了.这是因为分片数的增加会使实际查询区域越接近理想的查询区域(即查询超球与不确定超球相交部分)，因此在下面的实验中将 τ 设为 8.

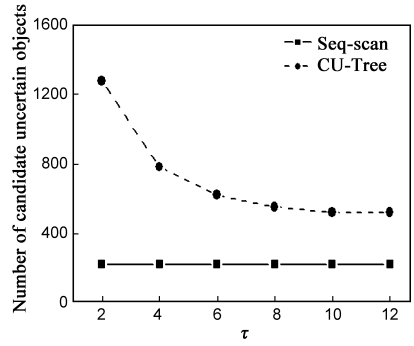
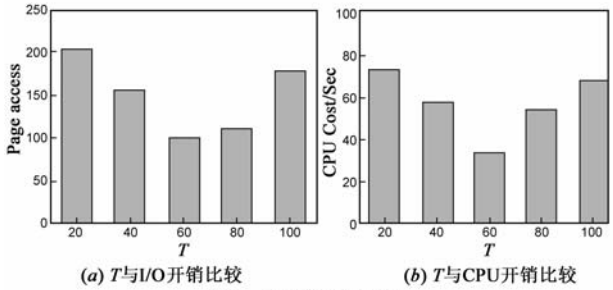


图5 分片数对查询的影响

4.2 聚类数(T)对查询的影响

在第二组实验中，研究研究聚类个数 T 对 *pRSearch* 查询性能的影响.从图 6(*a*)和(*b*)看出，随着聚类数(T)的增加，查询效率(包括 I/O 和 CPU 开销)开始是缓慢减少.因为随着聚类数增加，平均搜索空间在减少，但减少的幅度是缓慢的.当 T 超过一定数目(60)时，会使得各个类超球相互重叠导致查询的 I/O 和 CPU 代价提高.因此可以将 T 作为一个查询性能优化的调整因子.因此在下面的实验中将 T 设为 60.



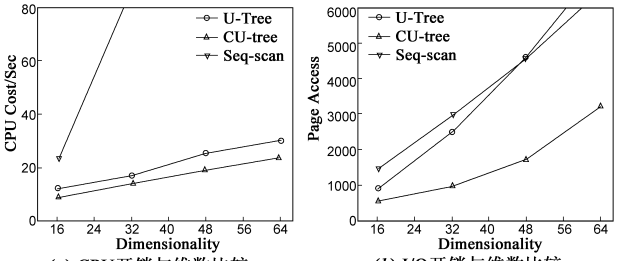
(*a*) *T*与I/O开销比较 (*b*) *T*与CPU开销比较

图6 *T*对查询的影响

4.3 维数对查询的影响

本次实验采用 100,000 个合成数据作为测试数据研究维数对查询性能的影响，其中维数从 16 到 64.从图 7 中看出随着维数的增加，CU-Tree 索引的查询效率最高，这是因为它能够有效地缩减查询过程中的搜索空间，较 U-Tree 来说，其查询时间及磁盘块的访问次数大大减少.U-Tree 对候选对象过滤能力在下降.而对于 CU-Tree 索引来说，维数对其查询效率影响相对较小，这

样使得两者在查询性能上的差异变大.从图中还可以看出随着维数的增加,U-tree 和顺序检索的查询性能越来越差,因为维数增加使得其搜索空间呈指数级增长.

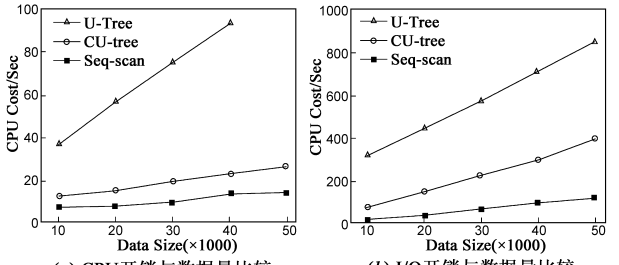


(*a*) CPU开销与维数比较　(*b*) I/O开销与维数比较

图7　查询效率与维数比较

4.4　数据量对查询的影响

本次实验研究数据量对查询性能的影响.采用真实数据作为测试数据集执行范围概率查询.图 8 分别从 CPU 和 I/O 开销方面比较了它们各自在查询性能上的差异.实验表明顺序检索要远远高于 U-Tree 和 CU-Tree,因为它在查询过程中需要进行 CPU 密集运算的概率积分操作.同时,在 I/O 的开销方面,CU-Tree 要优于其它 2 种方法.其原因与 4.3 节所述相似,不再另述.

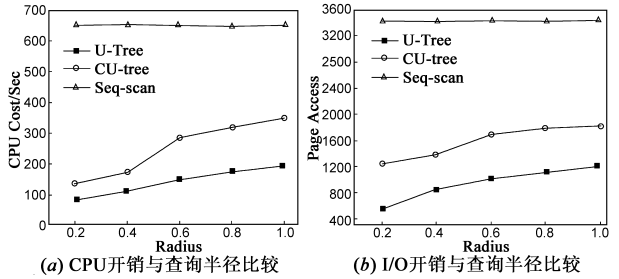


(*a*) CPU开销与数据量比较　(*b*) I/O开销与数据量比较

图8　数据量对查询影响

4.5　查询半径对查询的影响

最后实验采用合成数据来研究不同半径值对查询性能的影响.从图 9 可以看出,当查询半径从 0.2 到 1 时,CU-Tree 无论在 I/O 还是 CPU 计算代价方面都要明显优于其它方法.在这些索引中,尽管 U-Tree 索引采用区域分片方法来缩小高维不确定搜索空间,但其查询代价仍然非常高,仅次于顺序检索查询.



(*a*) CPU开销与查询半径比较　(*b*) I/O开销与查询半径比较

图9　半径对查询的影响

5　结语

本文提出一种基于质心片的高维不确定索引方法——CU-Tree.该方法通过预先对高维不确定对象进行聚类,然后对每个对象进行基于双尺度(即编码及存在概率)的统一编码,同时结合其对应的类信息生成统一索引键值,并采用 B + 树对其建立索引.较其它方法,理论和实验都表明 CU-Tree 能够有效地缩小搜索空间,从而明显减少概率积分运算的代价,优于其它同类索引方法,如 U-Tree 和顺序检索等.

参考文献

[1] Widom J. Trio: A system for integrated management of data, accuracy and lineage[A]. In Proc CIDR'05[C]. USA, ACM. 2005.262 – 276.

[2] http://www.cs.washington.edu/homes/suciu/project-mystiq.html[DB/CD]

[3] http://www.cs.purdue.edu/probdb/[DB/CD]

[4] http://www.comlab.ox.ac.uk/projects/MayBMS/[DB/CD]

[5] 李建中,于戈,周傲英.不确定性数据管理的要求与挑战[J].中国计算机学会通讯,2009,5(4):6 – 14.
Li Jian-zhong, Yu Ge, Zhou Ao-ying. Requirement & Challenges of Unlertain Data Management. Communication of EEF [J].2009,5(4):6 – 14.(in Chinese)

[6] 丁晓锋,卢炎生,等.基于 U-tree 的不确定移动对象索引策略[J].软件学报,2008,19(10):2696 – 2705.
Ding Xiao-feng, Lu Yan-sheng, et al. U-Tree-based indexing method for Uncertain movie object[J]. Journal of Software, 2008,19(10):2696 – 2705.(in Chinese)

[7] H V Jagadish, B C Ooi, K L Tan, et al. iDistance: An adaptive B + -tree based indexing method for nearest neighbor search [J]. ACM Trans on Data Base Systems, 2005, 30(2): 364 – 397.

[8] Cheng R, Xia Y, Prabhakar S, Shah R, *V*itter JS. Efficient indexing methods for probabilistic threshold queries over uncertain data[A]. In Proc VLDB'04[C]. Toronto, 2004. 876 – 887.

[9] N Beckmann, H P Kriegel, R Schneider, et al. The R * -tree: An Efficient and Robust Access Method for Points and Rectangles [A]. In Proc ACM SIGMOD'90[C]. Atlantic City: SIGMOD Record, 1990. 19(2): 322 – 331.

[10] C Bohm, S Berchtold, D. Keim. Searching in high-dimensional spaces: index structures for improving the performance of multimedia databases[J]. ACM Computing Surveys, 2001, 33(3): 322 – 373.

[11] S Berchtold, C Bohm, H P Kriegel, et al. Independent quantization: An index compression technique for high-dimensional data spaces[A]. In Proc ICDE'00[C]. USA: IEEE Computer Society, 2000. 577 – 588.

[12] V Ljosa, A K Singh. APLA: Indexing arbitrary probability distributions[A]. In Proc ICDE'07[C]. Turkey: IEEE Computer Society, 2007. 946 – 955.

[13] A Guttman. R-tree: A dynamic index structure for spatial

searching[A]. In Proc ACM SIGMOD'84[C]. ACM Press, 1984.47 – 54.
[14] R Weber, H Schek, S Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces[A]. In Proc VLDB'98[C]. New York: Morgan Kaufmann Publishers, 1998.194 – 205.
[15] 薛向阳,罗航哉,等. LIFT:一种用于高维数据的索引结构[J].电子学报,2001,29(2):192 – 195.
Xue Xiang-yang, Luo Hang-zhai, et al. LIFT: An index structure for high-dimensional data[J]. Acta Electronic Sinica, 2001, 29(2):192 – 195. (in Chinese)
[16] G Beskales, M Soliman, I Ilyas. Efficient search for the top-k probable nearest neighbors in uncertain databases[A]. In Proc VLDB'08[C]. New Zealand, VLDB Endowment Inc.. 2008. 326 – 339.
[17] H Kriegel, P Kunath, M Renz. Probabilistic nearest-neighbor query on uncertain objects[A]. In Proc DASFAA'07[C]. Thailand, Springer. 2007. 337 – 348.
[18] X Lian, L Chen. Probabilistic group nearest neighbor queries in uncertain databases[J]. IEEE Trans on KDE, 2008, 20(6):809 – 824.
[19] R Cheng, J Chen, M Mokbel, C Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data[A]. In Proc ICDE'08[C]. Mexico, IEEE Computer Society. 2008. 973 – 982.
[20] Y Tao, R Cheng, XXiao, W KNgai, B Kao, S Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions[A]. In Proc VLDB'05[C]. Norway, ACM. 2005. 922 – 933.
[21] Y Qi, S Singh, R Shah, S Prabhakar. Indexing probabilistic nearest-neighbor threshold queries[A]. In Proc Workshop on Management of Uncertain Data[C]. New Zealand, VLDB Endowment Inc, 2008. 87 – 102.
[22] www.Kdd.ci.du, 2006. [DB/CD]

作者简介

庄 毅 男,1978 年生于浙江杭州.博士,浙江工商大学副教授,硕士生导师,获 2008 中国计算机学会优秀博士论文奖.研究方向为不确定数据管理、多媒体数据库等.
E-mail: zhuang@mail.zjgsu.edu.cn

胡海洋 男,1977 年生于江苏扬州.博士,杭州电子科技大学副教授,硕士生导师.研究方向为软件工程理论、不确定理论等.
E-mail: huhaiyang@hdu.edu.cn

胡 华 男,1964 年生于江西南昌.博士,杭州电子科技大学教授,博士生导师.研究方向为软件工程、工作流技术及数据库理论等.
E-mail: huhua@hdu.edu.cn