重建数据立方体的数据覆盖方法*

曹蓟光, 王申康

(浙江大学 计算机科学与工程系 人工智能研究所,浙江 杭州 310027)
E-mail: cjggjc@263.net
http://ai.zju.edu.cn

摘要: 数据切片体现了数据立方体某方面的数据特征,当产生数据切片的数据立方体由于某种原因不可访问时,通过已知的多个数据切片尽可能地恢复数据立方体中的原始信息,有利于对数据的深入分析和理解.提出了一种基于数据切片重建数据立方体的方法,通过数据切片的连接生成多维细粒度空间,利用数据覆盖操作逐步细分每个数据切片所对应的多维空间,以逼近多维细粒度空间.提出了重建后数据立方体的可查询判据.该方法直接利用数据库中的基本操作来实现,高效地支持大数据量的应用环境.

关 键 词: 数据切片;数据立方体;数据覆盖;多维空间
中图法分类号: TP311 文献标识码: A

由数据立方体经过聚类、切片、旋转、上钻、下挖等操作所形成的数据切片体现了数据立方体在某一个"侧面"、某一个概念级别上的数据特征,它把用户关心的信息呈现给用户,是用户使用数据仓库中的信息的有力工具.当产生数据切片的数据立方体由于某种原因不能访问时(如存储介质的物理损坏),或出于性能方面的考虑要尽量减少对服务器的访问时,数据切片成为查询和分析数据的主要信息来源.这些分别体现原始数据立方体某方面特征的数据切片中通常蕴涵着大量的有用信息,通过数据切片尽可能地恢复数据立方体中原始信息的过程称为"基于数据切片的数据立方体的重建".

文献[1,2]从统计数据库的安全性角度研究了存在由查询结果推知私有数据的可能性的条件.文献[3]在讨论用多个分类表建立一个支持统一访问的数据视图时,把问题转化为用多个低阶概率分布推导出联合概率分布的问题,进而用最大熵原理来恢复原始细节信息.它实际上是关系数据库的连接操作在统计意义上的扩展,要求各分类表的分类属性间不能存在包容关系,因此不能应用于数据立方体的重建问题.文献[4]在讨论不同数据源的数据集成时,用最大似然估计来处理从不同数据源的综合数据中发现细节数据的问题,但所给出的方法一般只对从单个分类的求和结果中获得细节数据的问题有效.通过这些方法所得到的结果都是基于统计意义上的,所得到的解不是确定的值,一般用于对准确程度要求不高的情况.文献[5]在讨论用同源数据统计结果来分析原始细节数据时,采用了图论的知识,提出了"intersection hypergraph"的概念,并用它来化简、集成统计结果,力图把用户的查询通过这些已知统计结果表示出来.文献[6]则用解线性方程组的方法来求解细节信息.这些方法不支持增量式的计算,新信息的加入要重新计算,计算代价大.另一方面,这些方法均是针对数据量较小、可以在内存中处理的情况,对于数据仓库中的大数据量的情况不适用.本文从数据切片向数据立方体重建的角度,给出了细粒度多维空间的生成操作——多数据切片的连接操作,定义了数据覆盖的概念及其操作,在此基础上提出了逐步切分泛化多维空间以逼近细粒度多维空间的数据立方体重建算法,并给出了重建后数据立方体的可查询判据.算法支持增量式计算,支持大数据量的应用环境,并且实现简单.

* 收稿日期: 2000-12-08; 修改日期: 2001-05-18

作者简介: 曹蓟光(1973－),男,天津蓟县人,博士,工程师,主要研究领域为数据库技术,决策支持,智能信息处理,人工智能;王申康(1945－),男,浙江绍兴人,教授,博士生导师,主要研究领域为数据库理论,智能信息系统,人工智能,多媒体技术.

1 问题的描述

定义 **1**. 多维模型是一个二元组$(Ddom,D)$,其中:

$Ddom=\{Ddom_1,...,Ddom_i,...,Ddom_n\}(1\le i\le n)$ 称为维的论域集;

$D=\{D_1,...,D_i,...,D_n\}(1\le i\le n)$ 称为维度 i 的标识集.

$D_i=(All_i,T_i,Y_i)$为一个三元组,表示维度 i.All_i 为维度 i 中包含所有数据元素的集合,称为维度 i 上的最粗粒度数据集,Y_i 是维度 i 中所有不可再分的数据元素组成的集合,称为最细粒度数据集;$T_i=(T_{i1},\ldots,T_{ij},\ldots,T_{im})$是以 All_i 为公共根、Y_i 为公共叶的 m 棵概念子树,T_{ij} 表示维度 i 上的第 j 棵概念树,T_{ij}^k 表示维度 i 上的第 j 棵概念树的第 k 概念层的数据元素的集合,($1\le k\le K$,K 为此概念树的深度)满足:

- $T_{ij}^1=All_i;T_{ij}^K=Y_i$;
- $All_i=\bigcup Y_i=\bigcup T_{ij}^k$;
- $All_i\in Ddom_i;T_i\in Ddom_i;Y_i\in Ddom_i$;

则称 $w^*=\triangleright_i\triangleleft Y_i$ 为细粒度多维论域空间;$w'=\triangleright_i\triangleleft T_{ij}^k$ 为泛化多维论域空间;$w=\bigcup_j\bigcup_k\triangleright_i\triangleleft T_{ij}^k$ 为多维论域空间.显然有 $w^*\subset w;w'\subset w$.

定义 **2**. 数据立方体为一个三元组

$$\text{cube}=(w^*,M,f),$$

其中 M 为指标的论域集;$f:w^*\to M$ 为细粒度多维论域空间 w^*到指标论域集 M 的映射;数据立方体中的元素表示多维论域空间 w^*中的点在指标论域集 M 中的取值.

定义 **3**. 数据切片为一个三元组

$$\text{slice}=(w',M,f),$$

其中 M 为指标的论域集;$f:w'\to M$ 为泛化多维论域空间 w'到指标论域集 M 的映射;数据切片中的元素表示泛化多维论域空间 w'中的点在指标论域集 M 中的取值.

可以证明:数据立方体和数据切片定义中的映射 f 为多维论域空间到指标论域空间的线性映射;f 为 w 到 M 的线性泛函,M 为线性算子 f 的值空间,则由多个数据切片重建数据立方体的过程就是在线性泛函 f 约束下由空间 w 重建空间 w^*的过程.

2 细粒度多维空间的生成

设数据切片 S_1 和 S_2 为由同一个数据立方体产生的数据切片,维度 i 中 S_1 与 S_2 对应的元素的集合为 ${}^1T_{is}^t$ 和 ${}^2T_{ij}^k$.记 S_1 与 S_2 连接后所得的数据切片的维度 i 对应的元素集为 ${}^{12}T_{ir}^l$,则

- 当概念子树 $s=j$ 时,$r=s=j;l=\max(t,k)$;
- 当 S_1 与 S_2 在维度 i 中对应的概念子树 $s\ne j$ 时,设 ${}^1T_{is}^t$ 与 ${}^2T_{ij}^k$ 是对此维度 i 上的最细粒度空间 Y_i 的不同划分:$Y_i/{}^1T_{is}^t$ 与 $Y_i/{}^2T_{ij}^k$;

则 $Y_i/{}^{12}T_{ir}^l=(Y_i/{}^1T_{is}^t)/{}^2T_{ij}^k=(Y_i/{}^2T_{ij}^k)/{}^1T_{is}^t$ 是由 S_1,S_2 对最细粒度空间 Y_i 共同划分后形成的所有非空元素集合所组成的集合.

例 1:S_1 在“年龄组”维度上的数据元素集为 ${}^1T_{is}^t=\{(15,20),(30,40),(40,60)\}$;$S_2$ 在“年龄组”维度上的数据元素集为 ${}^2T_{ij}^k=\{(15,25),(30,35),(35,45)\}$,则 $S_1\triangleright\triangleleft S_2$ 后在“年龄组”维度上的数据元素集为

$${}^{12}T_{ir}^l=\{(15,20),(20,25),(30,35),(35,40),(40,45),(45,60)\}.$$

- 由 $\triangleright_i\triangleleft{}^{12}T_{ir}^l$ 所形成的空间为由 S_1 与 S_2 所生成的细粒度多维论域空间.

例 2:已知 3 个数据切片 S_1,S_2 和 S_3 如下:

Table 1 Data slices S_1

表 1 数据切片 S_1

Age group①	Select range②	Amount③
25~30	(*a*)	4
25~30	(*b*,*c*)	11
25~30	(*d*,*e*)	7
30~40	(*a*)	12
30~40	(*b c*)	7
30~40	(*d*,*e*)	10

①年龄组,②选择范围,③人数.

Table 2 Data slices S_2

表 2 数据切片 S_2

Age group①	Select range②	Amount③
25~30	(*a*,*b*)	12
25~30	(*c*,*d*)	9
25~30	(*e*)	1
30~40	(*a*,*b*)	14
30~40	(*c*,*d*)	8
30~40	(*e*)	7

①年龄组,②选择范围,③人数.

Table 3 Data slices S_3

表 3 数据切片 S_3

Age group①	Select range②	Amount③
25~35	(*a*,*b*)	21
25~35	(*c*)	7
25~35	(*d*,*e*)	9
35~40	(*a*,*b*)	5
35~40	(*c*)	1
35~40	(*d*,*e*)	8

①年龄组,②选择范围,③人数.

经连接操作后形成的多维细粒度空间 w^*为{(25~30),(30~35),(35~40)} $\triangleright\triangleleft$ {(*a*),(*b*),(*c*),(*d*),(*e*)},共包含 15 个空间点.

3 数据覆盖及其操作

数据切片 $S=(w',M,f)$中的 w'为 w^*空间中的点的集合,这个集合可以表示成多个"数据覆盖"所包含的多维空间点的集合.

3.1 数据覆盖的定义

定义 **4**. 数据覆盖 $L=(R,V)$.其中,$R\subset w'$,$V=\{[V_s,V_e]|f:R\rightarrow[V_s,V_e]\}$为 R 在值域 M 中的取值区间.

一个数据覆盖物理上对应于数据切片中的一条数据库记录.在上例中的数据切片 S_1 包含 6 个数据覆盖 L_1~L_6,如 $L_1=\{((25\sim30),(a)),[4,4]\}$.

对于两个数据覆盖 L_i,L_j,如果 $R_i\cap R_j\neq\varnothing$,则称两个数据覆盖有交集存在.

3.2 数据覆盖的操作

已知数据覆盖 $L_i=(R_i,[V_{si},V_{ei}])$和 $L_j=(R_j,[V_{sj},V_{ej}])$;令$[V_{\Phi s},V_{\Phi e}]$为根据领域知识确定的 w^*中的多维空间点在值域 M 中的取值范围,则定义操作 $L_i\cup L_j$:

若 $R_i=R_j$,则 $L_i\cup L_j=(R',[V_s',V_e'])$.其中,$R'=R_i=R_j$,

$$V_s'=\max(V_{si},V_{sj}),V_e'=\min(V_{ei},V_{ej}).$$

若 $R_i\neq R_j$,则返回一个数据覆盖集 L,它包含两个数据覆盖 L_i 和 L_j.

若 $R_i=\varnothing$,$R_j\neq\varnothing$则返回 L_j.

定义操作 $L_i+L_j=W$.其中,

$$W=(R_w,[V_{ws},V_{we}]),R_w=R_i\cup R_j,$$
$$V_{ws}=V_{si}+V_{sj},V_{we}=V_{ei}+V_{ej}.$$

定义操作 $L_i-L_j=W$.其中,

$$W=(R_w,[V_{ws},V_{we}]),R_w=R_i-R_j,$$
$$V_{ws}=\max(V_{\Phi s},(V_{si}-V_{ej})),V_{we}=\min(V_{\Phi e},(V_{ei}-V_{sj})).$$

4 重建算法

4.1 数据立方体重建算法

算法 **1**. $C=f(S_1,S_2)$.

输入:数据切片 S_1,S_2

输出:数据立方体 C

过程:

(1) 由 S_1,S_2的连接操作生成细粒度空间 w^*,即求 $\triangleright_i\triangleleft S_i$ 所对应的细粒度多维空间 w^*;

(2) 对 S_2 中的每一个数据覆盖 $M=(R_m,V_m)$,其中包含 p 个细粒度多维空间点 $r_i\in R_m(i=1,\ldots,p)$,把 S_1 中与 M

有交集的数据覆盖集合 S_m 分成 p 个组 $N_i \in S_m(i=1,\ldots,p)$,使得 N_i 中的每一个数据覆盖 $N_{ij}=(R_{ij},V_{ij}) \in N_i$,均有 $r_i \in R_{ij}$ 成立.

(a) 若 $R_{ij} \subset R_m$,则

$S_m=S_m-N_i$

$S_2=(S_2-M)\cup(M-N_{ij})$ /*对应数据库中的一条 Update 语句*/

$S_1=S_1\cup(M-N_{ij})$ /*对应数据库中的一条 Insert 语句*/

$M=M-N_{ij}$

(b) 若 $R_{ij}=R_m$,则

$S_m=S_m-N_i$

$S_2=(S_2-M)\cup(M\cup N_{ij}),S_1=(S_1-N_{ij})\cup(M\cup N_{ij})$ /*对应数据库中的两条 Update 语句*/

(c) 若 $R_{ij} \supset R_m$,则

$S_m=S_m-N_i$

$S_1=(S_1-N_{ij})\cup(N_{ij}-M)$ /*对应数据库中的一条 Update 语句*/

$S_2=S_2\cup(N_{ij}-M)$ /*对应据库中的一条 Insert 语句*/

(d) 若 S_m 中的 N_i 均不为空时,分别从 N_i 中取出一个数据覆盖 N_{ij},形成一个数据覆盖集 $N^0=\bigcup_i N_{ij}$.

对于存在的所有数据覆盖集 N^0 做如下处理:若 $N_i^0 \in N^0$ 满足且 $\bigcap_i N_i^0=\varnothing$,则

$S_1=S_1\cup(\sum_i N_i^0-M)$,$S_2=S_2\cup(\sum_i N_i^0-M)$ /*对应数据库中的若干条 Insert 语句*/

(3) 对 S_1 中的每一个数据覆盖 $N=(R_n,V_n)$,其中包含 q 个多维细粒度空间点 $r_s \in R_n(s=1,\ldots,q)$,把 S_2 中与 N 有交集的数据覆盖集合 S_n 分成 q 个组 $M_s \in S_n(s=1,\ldots,q)$,使得 M_s 中的每一个数据覆盖 $M_{st}=(R_{st},V_{st})$均有 $r_s \in R_{st}$ 成立.

(a) $R_{st} \subset R_n$,则

$$S_n=S_n-M_s,S_2=S_2\cup(N-M_{st}),S_1=(S_1-N)\cup(N-M_{st}),N=N-M_{st}.$$

(b) 若 $R_{st}=R_n$,则

$$S_n=S_n-M_s,S_2=(S_2-M_{st})\cup(N\cup M_{st}),S_1=(S_1-N)\cup(N\cup M_{st}).$$

(c) 若 $R_{st} \supset R_n$,则

$$S_n=S_n-M_s,S_1=(S_1-N)\cup(M_{st}-N),S_2=S_2\cup(M_{st}-N).$$

(d) 若 S_n 中的 M_s 均不为空时,分别从 M_s 中取出一个数据覆盖 M_{st},形成一个数据覆盖集 $M^0=\bigcup_s M_{st}$.

对于存在的所有数据覆盖集 M^0 做如下处理:若 $M_i^0 \in M^0$ 满足 $\sum_i M_i^0 \supset N$ 且 $\bigcap_s M_t^0=\varnothing$,则

$S_2=S_2\cup(\sum_i M_i^0-N)$,$S_1=S_1\cup(\sum_i M_i^0-N)$ /*对应数据库中的若干条 Insert 语句*/

(4) 返回 $C=S_1\cup S_2$.

4.2 增量式重建方法

算法 **2**. $C'=f(C,S)$.

输入:数据立方体 C,数据切片 S

输出:新的数据立方体 C'

过程:Begin

Return $f(C,S)$

End

4.3 算法分析

算法中的主要计算耗费是判断两个数据覆盖之间的包含关系,也就是多维空间点集间的包含关系.设两个数据切片中分别包含 n_1 和 n_2 个数据覆盖,则在最坏情况下,计算多维空间点集间包含关系的次数的上限为

$(n_1+n_2)^2$.但在一般情况下,由于重建后的数据立方体中包含的数据覆盖数小于 n_1+n_2,且数据切片中的数据覆盖的分裂是渐近的,则空间点集间包含关系的计算次数远小于$(n_1+n_2)^2$.另外,为了进一步减少空间点集间包含关系的计算次数,可以充分利用前次的判断结果:两互不相交的数据覆盖分裂后所产生的数据覆盖间不相交;已经判断为不相交的两数据覆盖,下次不必再重复计算.

在增量式算法中,当添加的数据切片只包含一个数据覆盖时,计算复杂度为 $O(n)$,n 为已知的数据立方体中所包含的数据覆盖数.

算法直接面对数据库中的记录进行处理,利用基本的数据库操作(Insert,Update,Delete)来实现数据覆盖间的操作,避免了数据库中数据的导入、导出,适合大数据量的应用环境.

算法支持增量式计算,可以在较小代价的情况下利用获得的信息来完善数据立方体;在进行增量计算时,由于数据立方体始终保持数据一致状态,因此不需要对数据立方体加排他锁,保证了数据更新时服务不中断.

5 可查询判据

一个查询 $Q=(R_Q,V_Q)$为一个数据覆盖,R_Q 为此查询在 w^*中的对应点集;V_Q 为 R_Q 在对应值域中的取值.

5.1 查询判据

判据. 若重建后得到的数据立方体 C 中存在一个数据覆盖集 C^*,满足 $R_Q=\bigcup_i R_{C_i^*}$, $C_i^*=(R_{C_i^*},V_{C_i^*})\in C^*$,则称此查询 Q 为可应答查询,查询结果为 $V_{Qs}=\sum_i V_{C_{is}^*}$, $V_{Qe}=\sum_i V_{C_{ie}^*}$.

5.2 实现方法

算法 **3**. 可应答判据.

输入:重建后的数据立方体 C,一个用户查询 $Q=(R_Q,V_Q)$

输出:True/False;一个取值区间$[V_{Qs},V_{Qe}]$

步骤:$Q'=\varnothing$

对 C 中与 Q 有交集的所有数据覆盖 C_i^* 作如下处理:

若 $C_i^*\subset Q$,则 $Q'=Q'\cup C_i^*$,$Q=Q-C_i^*$;若 $Q=\varnothing$,则此查询为可应答查询,返回 True 和$[\sum_i Q'_{is},\sum_i Q'_{ie}]$;若 $Q\neq\varnothing$,则此查询不可应答,返回 False.

6 实例分析

基于上述方法重建的数据立方体 $f(f(S_1,S_2),S_3)$为表 4.

Table 4 Reconstructed data cube

表 **4** 重建后的数据立方体

Age group①	Select range②	Amount③
25~30	(*a*)	4
25~30	(*b*)	8
25~30	(*c*)	3
25~30	(*d*)	6
25~30	(*e*)	1
30~40	(*a*)	12
30~40	(*b*)	2
35~40	(*c*)	1
30~40	(*d*)	3
30~40	(*e*)	7
35~40	(*a*,*b*)	5
35~40	(*c*)	4
35~40	(*d*,*e*)	8
30~35	(*d*,*e*)	2
30~35	(*a*,*b*)	9
30~35	(*c*)	1

①年龄组,②选择范围,③人数.

对于一个查询 Q:“年龄在 25~40 之间、选择范围在(b,d)中的人数”,则

$$R_Q=\{((25\sim30)(b)),((30\sim35)(b)),((35\sim40)(b)),((25\sim30)(d)),((30\sim35)(d)),((35\sim40)(d))\}.$$

运用判据后得到数据覆盖集 Q':

$$Q'=\{((25\sim30)(b)),8\}\cup\{((30\sim35)(b),(35\sim40)(b)),2\}\cup\{((25\sim30)(d)),6\}\cup\{((30\sim35)(d),(35\sim40)(d)),3\}.$$

$$R_Q=\varnothing.$$

则此查询为重建后的数据立方体的可应答查询,查询结果为 $V_{Qs}=V_{Qe}=8+2+6+3=19$.

对于一个查询 Q:“年龄在 25~35 之间、选择范围在(b,c)中的人数”,则

$$R_Q=\{((25\sim30)(b)),((30\sim35)(b)),((25\sim30)(c)),((30\sim35)(c))\}.$$

运用判据后得到数据覆盖集 Q':

$$Q'=\{((25\sim30)(b)),8\}\cup\{((25\sim30)(c)),3\}\cup\{((30\sim35)(c)),1\},$$

$$R_Q=\{(30\sim35)(b)\},$$

则此查询为重建后的数据立方体的不可应答查询.

7 总 结

本文提出了一种由多个数据切片重建数据立方体的方法——数据覆盖法,利用定义在数据覆盖上的操作来逐步细分每个数据切片的多维空间 w',以期逼近数据立方体的多维细粒度 w^*空间.该方法的优点在于实现简单,可以在对数据库中的记录扫描时利用 Update,Insert,Delete 操作来完成重建过程,从而支持大数据量的应用环境;该方法支持增量式的运算,可以利用获得的数据切片或数据覆盖来完善重建的数据立方体.文中还给出了重建后的数据立方体上的查询可应答判据,利用它来判定重建后的数据立方体是否可以应答一个给定的查询.

本文提出的方法可以用来优化 OLAP(online analytical processing)应用.一些对数据立方体的查询与分析不必直接访问服务器,而可以利用客户端已经获得的数据切片来完成,从而减少了服务器的负载,缩短了查询响应时间,提高了系统的效率.

本文提出的方法可以用来实现脱机分析处理.在脱机的情况下尽可能地挖掘现有数据切片的隐含信息.当数据立方体受到破坏时,该方法可以用来对受损数据进行尽可能的恢复.

为了进一步提高算法的计算性能,将来的改进工作包括算法并行化改造和高效索引结构的利用.

References:

[1] Denning, D.E., Denning, P.J., Schwartz, M.D. The tracker: a threat to statistical database security. ACM Transactions on Database Systems, 1979,4(1):76~96.

[2] Dobkin, D., Jones, A.K., Lipton, R.J. Secure databases: protection against user interface. ACM Transactions on Databases Systems, 1979,4(1):97~106.

[3] Malvestuto F.M. A universal table model for categorical database. Information Sciences, 1989,49:203~223.

[4] Scotney, B., McClen, S. Efficient knowledge discovery through the integration of heterogeneous data. Information and Software Technology, 1999,41:569~578.

[5] Malvestuto, F.M. A universal-scheme approach to statistical databases containing homogeneous summary tables. ACM Transactions on Database Systems, 1993,18(4):678~708.

[6] Ribeiro, J.S., Kaufman, K.A., Kerschberg, L. Knowledge discovery from multiple databases. In: Fayyad, U., Uthurusamy, R., eds. Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining. Montreal: AAAI Press, 1995. 240~246.

A Data Cover Method of Data Cube Reconstruction Based on Data Slices*

CAO Ji-guang, WANG Shen-kang

(*Artificial Intelligence Research Institute*, *Department of Computer Science and Engineering*, *Zhejiang University*, *Hangzhou* 310027, *China*)

E-mail: cjggjc@263.net
http://ai.zju.edu.cn

Abstract: Data slices present some properties of data cube. When data cube cannot be accessed for some reasons, it is very useful to reconstruct data cube based on data slices to discover the implicit information. Therefore, in order to reconstruct data cube, a practical method named Data Cover Method is presented, which creates the multidimensional fine-granularity space by the join operation between data slices, and splits the multidimensional space of data slice into multidimensional fine-granularity space by the data cover operators. An answerable criterion of reconstructed data cube is also presented. This method can be implemented by the basic operations of the DBMS (database management system), and support the application environment of large data set efficiently.

Key words: data slice; data cube; data cover; multidimensional space

* Received December 8, 2000; accepted May 18, 2001

△△△