异常情况下维结构的变换方法*)

陆昌辉[1] 吴晓华[2] 刘青宝[1] 邓 苏[1] 张维明[1]
(国防科学技术大学信息系统与管理学院 长沙 410073)[1]
(中国人民解放军第二炮兵装备研究院第四研究所二室 北京 100085)[2]

摘 要 多维数据模型是数据仓库与OLAP(On-Line Analytical Processing)的一个重要基础,在许多实际情况中,由于一些异常的发生,使得原有模型中的维结构不能准确地捕获其真正含义,满足不了分析需求。本文基于这样一个研究背景,首先给出了维结构的一些相关定义;然后用规则的形式统一对这些异常给维结构带来的影响进行了描述;接着从算法的总体思想、具体实施步骤等角度对维结构变换算法进行了详细介绍;并以军事应用里部队编成问题中部队维结构的变换实例对该算法的应用进行了分析;最后还与相关的研究进行了比较。

关键词 数据仓库,多维数据模型,维结构,异常,变换算法

Method for Revising Dimension Structure in Exceptions

LU Chang-hui[1] WU Xiao-hua[2] LIU Qing-bao[1] DENG Su[1] ZHANG Wei-ming[1]
(College of Information System and Management, National University of Defense Technology, Changsha 410073, China)[1]
(The Fourth Institute of the Second Artillexy Equipment Academy, Beijing 100085, China)[2]

Abstract Multidimensional data model is an important base of data warehouse and OLAP. In many real-life cases, particularly when exceptions arise, and their presence may turn the old dimension structure unsuitable for representing its real meaning. Based on this situation, this paper first gives some definitions about the dimension structure, and represents the effect brought by these exceptions with rules. Then describes the algorithm for revising the dimension structure in detail, including its idea and concrete processes. Finally, introduces the use of the revision algorithm with the practical example of military application.

Keywords Data warehouse, Multidimensional data model, Dimension structure, Exceptions, Revision algorithm

1 引言

多维数据建模技术是数据仓库、多维数据库、OLAP应用的基础[1]。随着管理信息系统的广泛应用,各大公司都积累了大量的历史操作数据,怎么利用这些历史数据来为他们的决策活动服务已成为当前的研究重点,多维数据建模技术是其关键技术之一。在多维数据模型中,维是一个非常重要的概念,由于其具有一定的层次结构,它允许人们从不同的粒度对所关心的事实进行分析,从而为各级管理人员提供辅助决策信息[2]。在进行OLAP分析时,往往存在一个根据已有经验建立的相应多维数据模型,但由于政策因素、临时紧急任务、不可靠数据或者一些不确定性因素等引发异常时,使得原有模型的维结构不能准确地捕获其真正含义,满足不了分析需求。

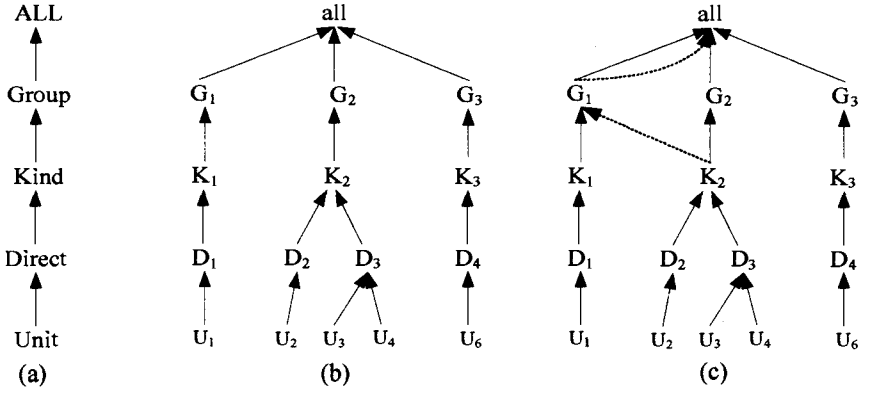

图1 (a)部队维的层次定义 (b)部队维的一个实例结构 (c)修改后的维实例结构

军事领域是数据仓库应用中的另一个重要领地,在为快速构建各种作战集群提供决策支持的分析主题中,部队维是

*)国家自然科学基金(60172012)。陆昌辉 博士,讲师,主要研究方向为数据仓库与决策支持技术。

一个非常重要的维度。为了根据部队的类别来赋予不同的作战任务，从而使组建的部队编组战斗力达到最佳，部队维(Army)通常包含如图1(a)所示的层次定义(其中Unit是部队基层单位层，它处于最底层；Direct表示的是该部队的上级单位；Kind表示所属作战编队(如重点火力打击编队、重要地点防护编队等)；Group表示各种作战编队组成的作战集群(如远东作战集群、大西洋作战集群等))。

在实际工程中，笔者遇到了这样的情况：某些部队(如图1(c)中的U_2)在平时隶属于作战集群G_2，但由于战况的突然变化，需要将其编入作战集群G_1。这样在根据原有模型(如图1(b)所示)对作战集群G_1的战斗力进行评估时，就没有包含这一支部队，从而得出不准确的结果，严重影响了领导的决策。对于这类引发维结构变化的事件，这里统一定义为多维数据模型中的异常。本文主要研究在这样一些异常的影响下，多维数据模型的维结构变换方法。

2 维结构描述

在下面的描述中，设D为维集，L为维层次集，C是由维层次的取值所构成的常量集。

2.1 维结构定义

关于多维数据模型的详细描述，可以参考相关文献[3-5]，这里就不再重复了，下面主要对维结构的一些相关概念进行定义。

定义1 维模式D_{schema}可以定义成这样一个四元组$(T_D, \overline{\prec}_D, \perp_D, ALL)$，其中$T_D=\{T_D, i=1,\cdots,m\}$为其维层次集合；$\overline{\prec}_D$是定义在$T_D$上的一个偏序关系，其传递闭包定义为$\overline{\prec}_D^*$；$ALL$是该维模式的顶层维层次，它具有最高抽象级别，含有唯一的元素all；$\perp_D$为该维的底层维层次，其粒度最细，在一个特定的维模式中，$\perp_D$是唯一的。

维就是符合维模式定义的某一具体实例，维与维模式的关系类似于面向对象程序设计中对象与类的关系。

定义2 为了描述某一维中某个维层次的取值情况，可以用关系$\rho\in D\times L\times C$来表示，$\forall(d,l,c)\in\rho$意味着维$d$中层次$l$的取值为$c$。集合$Iset(d:l)=\{c\mid c\in C,(d,l,c)\in\rho\}$称作维$d$中层次$l$的实例集。

定义3 设d为一给定的维，维路径$path_d$定义成这样一个序列$\langle d:l_1:c_1,\cdots,d:l_k:c_k\rangle$，其中维层次$l_1$是维$d$的底层维层次，$l_k\overline{\prec}_D ALL$，当$1\leqslant j\leqslant k-1$时，存在$c_j\in Iset(d:l_j)$，$c_{j+1}\in Iset(d:l_{j+1})$，且$l_j\overline{\prec}_D l_{j+1}$。

定义4 维路径的唯一性定义如下：设$path_d$，$path'_d$是定义在维d中同一维层次序列$l_1,\cdots,l_k$上的两条路径，若这两条路径在维层次l_1上的取值相同，则$path_d=path'_d$，也就是说定义在某一特定维层次序列上的维路径由底层维层次的取值唯一决定。

2.2 异常规则描述

在规则的定义中，规则通常由规则体和规则头组成。其中规则体就是该规则发生作用时必须满足的前提条件，规则头就是满足这些条件后得出的结果[6]。根据规则的定义，显然可以统一用它来定义这些异常给维结构带来的影响。在对这些异常规则进行描述之前，先进行层次表达式的定义。

定义5 层次表达式定义了在层次$l(l\in d)$的实例集中，由符合某一条件的元素构成的子集，它可通过如下的方式来描述：

(a) $d:l:c$是一个层次表达式，其中$d\in D, l\in L, c\in C$与其对应的集合中仅含有唯一元素$\{c\}$。

(b) $d:l:v\wedge\varphi(v.A,t_1,\cdots,t_m)$也是一个层次表达式，其中$d\in D, l\in L$，$v$为类型$C$的变量；$\varphi(v.A,t_1,\cdots,t_m)$是由$v.A,t_1,\cdots,t_m$构成的条件表达式，$A$是层次$l$的描述属性，$t_1,\cdots,t_m, m\geqslant 0$是一些与$v.A$相对应的常量。该层次表达式描述了在层次$l(\in d)$的实例集中，由符合条件$\varphi(v.A,t_1,\cdots,t_m)$的元素构成的子集。

在上述两个表达式中，如果维d进行了明确指定，则可以将其省略掉。

范例1：假设*equipment*为维*Army*中*Unit*层次的一个描述属性，它用来表示部队的装备，则下面的表达式均属于层次表达式：

$Army:Unit:u_1$；

$Army:Unit:X\wedge(X=u_1\vee X=u_3)$；

$Army:Unit:X\wedge\neg(X.equipment=$装甲$)$；

$Army:Unit:X\wedge(X.equipment=$装甲$\vee X.equipment=$火炮$)$。

定义6 规则r统一定义成如下的形式：$B_1(l_1),\cdots,B_k(l_k)\rightarrow d:l:c$，其中$B_1(l_1),\cdots,B_k(l_k)$为规则体，记为Body$(r)$，$d:l:c$为规则头，记为Title$(r)$；$l_1,\cdots,l_k,l$为维$d$的层次，且$l_j\overline{\prec}_D l_{j+1}, 1\leqslant j\leqslant k-1, l_k\overline{\prec}_D l$；$B_1(l_1),\cdots,B_k(l_k)$分别为对应维层次$t_1,\cdots,l_k$的层次表达式；$c$是维$d$中层次$l$的取值。如果维$d$进行了明确指定，则规则头中的$d$可以省略。

范例2：为了描述图1(c)中的异常，可以用如下规则来表示：

$Army:Unitk:U_2; Army:Direct:D_2; Army:Kind:K_2\rightarrow Group:G_1$

本文采用的是基于优先级的默认推理[7]，故用规则来描述异常给维结构带来的影响时，必须对这些规则的作用次序进行明确指定。

定义7 根据偏序关系$\overline{\prec}_D^*$的传递性，可以对规则的作用次序作如下规定：设r_1, r_2为任意两条规则，若r_1的规则头定义在层次l_i上，r_2的规则头定义在层次l_j上，若存在$l_i\overline{\prec}_D^* l_j$，则规则$r_1$较规则$r_2$先发生作用；反之，则规则$r_2$较规则$r_1$先发生作用。

定理1 对维结构产生影响的异常均可以用定义6的规则来描述。

证明：设维路径$p=\langle d:l_1:c_1,\cdots,d:l_k:c_k\rangle$是任意一条受异常影响的路径，在该异常的影响下，维路径p变为$p'=\langle d:l_1:c_1, d:l_2:c'_2,\cdots,d:l'_k:c'_k\rangle$，根据定义6的规则描述形式，可以构造如下规则集：

$d:l_1:c_1\rightarrow d:l_2:c'_2$；

$d:l_1:c_1\rightarrow d:l_3:c'_3$；

……

$d:l_1:c_1\rightarrow d:l_k:c'_k$。

显然，在上述规则集的作用下，维路径p可以转换为p'，证明完毕。

定义8 若对于某一维路径p，存在多条规则在同一维层次l上对其有影响，且这些规则在l上的影响结果各不相同，我们就称维路径p存在规则冲突。

3 变换方法描述

在下面描述的维结构变换方法中，其输入是维实例表

d_{inst}(用关系表的形式来存储维实例数据[8, 9])以及一组描述异常的规则集 *Rules*,输出是修正后的维路径集 *Paths*。

在根据异常规则对维路径进行修正的过程中,我们遵循如下的约定:若维路径 p 在维层次 l 上存在规则冲突,则将它在维层次 l 上的取值置为 null。

维结构变换算法的总体思想是:首先根据定义 7 所描述的规则作用次序对该异常规则集进行排序;然后对排序后的规则集逐一进行扫描,对于受影响的任一维路径 p,找出所有影响该路径的规则集 S,再根据 S 中规则的描述情况对该维路径 p 进行更新。

根据规则集 S 对维路径 p 进行更新的过程是:采用由底向上的方式对所有维层次进行遍历,在对维层次进行扫描的过程中,检查规则头中包含有当前维层次的规则,根据规则的描述情况对该维路径 p 中的当前维层次取值进行修改(若存在规则冲突,则将该维层次的取值修正为 null);若没有规则头定义在当前维层次的规则,则根据规则头定义在该层次前的规则对其取值进行修改。

下面给出了该算法的具体步骤描述:

```
Revise (d_inst, Rules, Paths)
{
   Paths = {};
   按规则作用次序对 Rules 中的规则进行排序,得到 Rules';
   Rules1 = Rules'; // Rules1 用来记录没有发生作用的规则集
     For Rules'中的每一条规则 i Do
       Rules1 = Rules1 − {i};
       For each ((满足规则 i 定义条件的维路径 p) and (p ∉
           Paths)) Do
           从规则集 Rules1 找出影响维路径 p 的规则子集 Rules2;
           Modify(p,Rules2 + {i}); //根据相应的规则集对受到
           影响的维路径 p 进行修正
           Paths = Paths + {p};
       End For
   End For
   Return Paths; // 输出修正后的维路径集 Paths
}
Modify(p, Rules) // 根据规则集对路径 p 进行调整
{
   BtmLevel = 最低维层次; // 将 BtmLevel 初始化为 p 的最低维层
   次
   For Level j from 最底层 to 最高层 Do
     Count = 0;
   value = null;
   For each rule i in Rules Do
     If (Title(i) == j) Then // 对规则 i 的头是否定义在维层次 j
     进行判断
       If (j > BtmLevel) Then
         BtmLevel = j;
       If (value ! = Getvalue(i,j)) Then // Getvalue(i,j)为获取
         规则 i 在维层次 j 的取值
         Count = Count ++;
         value = Getvalue(i,j);
       End If
     End If
   End For
   If (Count > 1) Then // 是否存在规则冲突
     SetValue(p,j, null); // 若存在冲突,则将维路径 p 在维层次 j
     的值置为 null;
   Else If (Count == 1)
     SetValue(p,j,value); // 不存在冲突,且有规则头定义在该维
     层次上;
   Else If (j > BtmLevel)
     根据没有受规则影响的任一维路径 p',且 p'与 p 在维层次 j−1
     上的取值相
     同,则将 p 在维层次 j 上的取值置为 p'在该维层次上的取值;
     //如果有规则对维层次 l 的取值进行了修正,则必须对所有维
     层次 l'(l ≺D l')
      //的取值进行相应修正
   End If
End For
}
```

4 算法应用范例

下面以图 1 描述的部队维为例来进行分析。在该部队维中,包含有层次 Unit, Direct, Kind, Group,且有如下关系存在:Unit $\prec_D$ Direct, Direct $\prec_D$ Kind, Kind $\prec_D$ Group。在该部队编成应用范例中,维层次 Kind 的描述如表 1 所示。

表 1 维层次 Kind 的描述

KindID	Kind	Kind_Name
k_1	K_1	重点火力打击编队
k_2	K_2	远程导弹攻击编队
k_3	K_3	抢滩登陆编队
k_4	K_4	重要地点防护编队

其中 k_i 为行标识符,$K_i \in$ Iset(Army:Kind),Kind_Name 是其描述属性。类似地,在描述维层次 Unit, Direct, Group 的表中,u_i, U_i, d_i, D_i 以及 g_i 与 G_i 分别为其行标识符与相应维层次的取值。

表 2 是该部队维的一个实例,其中 PathID 为行标识符。

表 2 部队维实例

PathID	CustID	UnitID	KindID	GradeID
p_1	u_1	d_1	k_1	g_1
p_2	u_2	d_2	k_2	g_2
p_3	u_3	d_3	k_2	g_2
p_4	u_4	d_3	k_2	g_2
p_5	u_5	d_4	k_3	g_2
p_6	u_6	d_5	k_4	g_3
p_7	u_7	d_5	k_4	g_3
p_8	u_8	d_6	k_4	g_3

在表 2 描述的维实例中,存在的这样一些异常:把给隶属于 D_2 的部队编入作战集群 G_1;把部队 u_7 列入编队 K_2;把隶属于 D_5 的部队列入编队 K_3;把部队 U_7 编入作战集群 G_2。为了定义上述异常,构造了下述规则集:

Rule1 Direct:Y∧Y. Direct $=D_2 \rightarrow$ Group:G_1

Rule2 Unit:X∧X. Unit$=U_7 \rightarrow$ Kind:K_2

Rule3 Direct:Y∧Y. Direct $=D_5 \rightarrow$ Kind:K_3

Rule4 Unit:X∧X. Unit $=U_7 \rightarrow$ Group:G_2

根据定义 7 所描述的规则作用次序,则上述规则集的作用次序为:Rule3→Rule2→Rule4→Rule1。

首先根据 Rule3 来进行修正:{ p_6, p_7 }受其影响,对 p_6 进行修正,修正后的结果如表 3 所示;接着对 p_7 进行修正,由于其在 Kind 层存在规则冲突,故其在该层次的取值为 null,由于篇幅有限,这里就不对整个算法的计算过程进行描述,最后算法的输出结果如表 3 所示(其中圆括号内的值为修改前的值)。

表 3 算法的最终结果

PathID	UnitID	DirectID	KindID	GradeID
p_2	u_2	d_2	k_2	$g_1(g_2)$
p_6	u_6	d_5	$k_3(k_4)$	$g_2(g_3)$
p_7	u_7	d_5	null(k_4)	$g_2(g_3)$

结束语 在现实世界中,由于一些异常的发生,通常会引发多维数据模型中相应维结构的变化。如果不对其进行处理,往往会影响分析结果的正确性。本文从这样一个问题背景出发,用规则的方法统一对这些异常进行了详细描述,并证明了这些异常给维结构带来的影响均可用本文给出的规则形式来描述;还从算法的总体思想、具体步骤等方面对维结构变换算法进行了详细说明;最后还用部队编成多维数据模型中部队维结构的变换实例对该算法的应用进行了介绍。

在已有的研究中,文献[10]对 OLAP 的维约束进行了描述,但其目的主要是用来推导维结构的可汇总性,并没有对维结构的变换问题进行研究。文献[11]对动态维进行了深入研究,定义了一套维度更新操作,还描述了一个支持动态维的

(下转第 239 页)

满意(认为过多),经过新一轮的查询后,减少了这样的元组数。同时也进行了其他实验,如查询"本人成分:干部"、"健康状况:良好"等,结果都比较理想。

表 3 模糊查询结果

姓名	性别	出生年月	出生地	本人成分	健康状况
张 姚	女	1978.03.18	河南	干部	良好
刘美国	男	1983.10.12	河南	学生	良好
赵 婷	女	1975.12.14	河北	工人	良好
万佳成	男	1979.01.15	河北	工人	良好
王 伟	男	1985.08.19	河北	学生	虚弱

从表 3 可以看出,最匹配的记录在最前面。在比较大型的数据库中,如果系统一次呈现很多记录,从前往后找,也是很容易找到最相关的结果的。因此,该方法同时实现了精确和模糊查询。产生最不确定记录即相关度接近于"0"时,可以用来减少反馈的次数,加快查询。

结束语 本文在关键词实现精确查询的基础上,扩展了模糊查询。通过改进 Rocchio 算法和定义对象之间的相关度,实现了结果集的降序排列,同时也方便了用户的查看。模糊查询的过程大概分三步完成。首先是提交的关键词去匹配数据库表的关系名或属性名,然后计算相异度值并存储,最后返回非"0"相异度值所在的行对应的元组记录并按相关度值降序排列。改进的 Rocchio 算法可以增强用户与结果集的相互交互,不断满足更高要求的查询。

不足的是在实验过程中存在多种主观和人为因素。利用矩阵存储相异度,使得系统的响应时间过长。将来的工作中我们着重从算法的时间复杂度和空间复杂度入手,提高算法的效率和稳定性。

参 考 文 献

[1] Agrawal S, Chaudhuri S, Das G. DBXplorer: A System for Keyword-based Search over Relational Databases // Proc. of the 18th Int'l Conf on Data Engineering. San Jose, 2002: 5-16

[2] Hristidis V, Papakonstantinou Y. DISCOVER: Keyword Search in Relational Databases // Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong, 2002: 670-681

[3] Hristidis V, Gravano L, Papakonstantinou Y. Efficient IR-style Keyword Search over Relational Databases // Proc. of the 29th Int'l Conf. on Very Large Data Bases. Berlin, 2003: 850-861

[4] 文继军,王珊. SEEKER:基于关键词的关系数据库信息检索. 软件学报,2005,16(7): 1270-1281

[5] Rocchio J J. Relevance Feedback in Information Retrieval. in SMART Retrieval System Experiments in Automatic Document Processing, 1971: 313-323

[6] 战学刚,林鸿飞,姚天顺. 中文信息检索中的相关反馈. 计算机科学,2000,27(7): 39-41

(上接第 222 页)

OLAP 服务器结构,但其研究的内容主要停留在如何根据人们的主观想法来进行维度的更新,比如创建一个维度,增加一个维层次,删除一个维层次等,并没有涉及到由于一些外在因素的影响,如何对维结构进行自动调节的内容。文献[9]虽然也对维结构的变换问题进行了相关研究,并给出了相应的变换算法,但其在对维路径进行修正的过程中,是按照维层次的级别来进行的。这就导致了下述问题的出现:对于某一维路径,这次修改了其在维层次 l 上的取值,下次可能还得修改其在 $l'(\overline{l \prec_D^s} l')$ 上的取值,用关系表的形式来存储维实例数据的情况下,就导致了对同一条记录进行多次修改与提交。与文献[9]相比,本文采用规则的形式来描述异常情况给维结构带来的影响,并对描述异常的规则的作用次序进行了严格定义,很好地保证了在进行转换的过程中结果的正确性,而且本文所提出的变换算法花费的时间明显较短,具有更高的执行效率。

参 考 文 献

[1] Firestone J M. Dimensional Modeling and E-R Modeling in the Data Warehouse[R]. DKMS-White Paper No. Eight, June 1998

[2] 段云峰,等. 数据仓库基础 Data Warehousing Fundamentals[M]. 北京:电子工业出版社, 2004

[3] Pederson T B, Jensen C S. Multidimensional data modeling for complex data[A] // Proceedings of the 15th International Conference on Data Engineering (ICDE'99)[C]. Sydney, Australia: IEEE Computer Society, 1999: 336-345

[4] 李建中,高宏. 一种数据仓库的多维数据模型[J]. 软件学报, 2000,11(7): 908-917

[5] Jensen C S, Kligys A, Pedersen T B, et al. Multidimensional Data Modeling for Location-Based Services[J]. The International Journal on Very Large Data Bases, 2004, 13(1): 1-21

[6] 史忠植,等. 人工智能:复杂问题求解的结构和策略 Artificial Intelligence: Structures and Strategies for Complex Problem Solving [M]. 4E. 北京:机械工业出版社, 2004

[7] Antoniou G. A tutorial on default logics[J]. ACM Computing Surveys, 1999, 31(4): 337-359

[8] Agosta L. 数据仓库技术指南 The Essential Guide to Data Warehousing[M]. 潇湘工作室,译. 北京:人民邮电出版社, 2000: 118-141

[9] Minuto E M, Vaisman A. Efficient Intentional Redefinition of Aggregation Hierarchies in Multidimensional Databases[C] // Proceedings of the 4th International Workshop on Data Warehousing and OLAP. Atlanta, Georgia, USA, 2001: 1-8

[10] Hurtado C A, Mendelzon A O. OLAP Dimension Constraints [C] // Proc. ACM PODS. Madison, USA, 2002: 169-179

[11] Vaismana A A, Mendelzonb A O, Ruaroa W, et al. Supporting dimension updates in an OLAP server[J]. Information Systems, 2004, (29): 165-185