一种频繁复合项目集的混合求解方法

李国和 赵沁平 王 喜

(北京航空航天大学 计算机学院, 北京 100083) (北京城市学院, 北京 100083)

摘 要: 关联规则挖掘的关键在于频繁项目集的求解,为了能够在含有数值类型数据的交易数据库中快速求解含有多值的频繁项目集,拓展了含有多种数值的交易数据库定义. 在此基础上,根据树的思想,建立含有交易项和交易数量的树,并结合 Apriori 算法和智能搜索,提出在各个较小的树枝路径中求解频繁项目集求解方法 FABCTA(Fast Algorithm By Candidate Transaction Tree and Apriori). 通过采用真实数据实验对比,FABCTA 效率明显优于 Apriori 算法.

关 键 词: 数据库; 树; 规则; 搜索理论; Apriori 算法

中图分类号: TP 181

文献标识码: A **文 章 编 号**: 1001-5965(2004)08-0791-06

Synthesizing algorithm for mining composite-frequent item sets

Li Guohe Zhao Qinping

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Wang Xi

(Beijing City College, Beijing 100083, China)

Abstract: It is very important to get the frequent item set in the associate rule mining. In order to fast obtain the frequent item set from a database that includes multiple values, the definition of transaction database was extended. And then by the tree concept, a special tree was built in which every node is formed by item and item's count. At last, on the foundation of Apriori Algorithm and Artificial Intelligent Search, FABCTA (fast algorithm by candidate transaction tree and apriori) was presented to solve the frequent item set in small branches of tree. By the test on real data, FABCTA is more efficient than Apriori algorithm.

Key words: databases; trees; rules; search theory; Apriori algorithm

关联规则的挖掘可分解为频繁项目集的求解和关联知识的表示两个子过程[1,2],而前者则是整个挖掘过程的主要计算开销. 目前,关于提高频繁项目集求解效率的代表性研究工作是 R. Agrawal 等人提出的 Apriori 算法. 本文提出了求解含有交易量频繁项目集的 FABCTA(Fast Algorithm By Candidate Transaction Tree and Apriori)方法,该方法在最小支持度较小时,效率明显优于 Apriori 算法.

1 基本概念

为了方便表述 FABCTA 方法,下面首先对 Apriori 算法中定义的交易数据库给予扩展,重新定义含有商品交易量的交易数据库.

定义 1 交易数据库 $D(ID,I,V)$,其中 ID 为顾客的交易标识集合,$I=\{i_1,\cdots,i_m\}$ $(m\geqslant 2)$ 为所有商品的集合,$k=\mathrm{Loc}(i_k)$ 表示商品 i_k 在 I 中的位置,$V=\{V_{i_1},V_{i_2},\cdots,V_{i_m}\}$ 为所有顾客购买所有商品量的集合,即存在映射 $f: ID\times I\rightarrow V$, $f(id,i)\in V_i$, $V_i\in V$, $f(id,i)$ 表示顾客 id 购买商品 i 的数量. $t_{i,id}=i(f(id,i))$ 表示顾客 id 进行了商品 i 及其商品交易量 $i(f(id,i))$ 的交易,称 $i(f(id,i))$ 为复合交易项目,交易 $T_{id}=\{t_{i_1,id},t_{i_2,id},\cdots,t_{i_m,id}\mid \forall k\in$

收稿日期: 2003-04-02

作者简介: 李国和(1965-),男,福建漳州人,博士生, guohe_li@sina.com.

$\{1,2,\cdots,m\}$, $i_k \in I$, $id \in ID$, 且 $t_{i_k,id}=i_k(f(id,i_k))\}$ 是顾客 id 进行购买的商品及其数量(即复合交易项目)的集合. 当一个顾客 $id \in ID$ 进行一次交易后,在交易数据库 D 中就追加一个交易 T_{id}. $t_{i(v)}=\{t_{i,id_1},t_{i,id_2},\cdots,t_{i,id_m} \mid i \in I,\ v=i(f(id_1,i))=i(f(id_2,i))=\cdots=i(f(id_m,i)),id_1,id_2,\cdots,id_m \in ID\}$ 表示进行商品 i 及其交易量为 v 的集合,$|t_{i(v)}|$ 为进行商品 i 及其交易量为 v 的交易总次数(即相同复合交易项目的个数). 如表1所示,其中0表示顾客没有进行实际商品交易.

表1 交易数据库

ID	I			
	a	b	c	d
u_1	1	1	0	2
u_2	2	4	1	5
u_3	1	1	0	2
u_4	2	2	1	2
u_5	3	3	1	1
u_6	1	1	0	2
u_7	1	1	0	1
u_8	3	0	1	1

定义2 $D(ID,I,V)$ 为交易数据库. $|D|$ 表示交易数据库的交易总次数. 交易数据库 D 中,由若干复合项目构成的集合 CFs,且 $CFs \neq \varnothing$, $i(v_i) \in CFs$, $i \in I, v_i \in V_i, V_i \in V$. 如果集合的基数 $|CFs|>1$, 那么 $\forall i_1(v_{i_1}),i_2(v_{i_2}) \in CFs, i_1,i_2 \in I$, $v_{i_1} \in V_{i_1}, v_{i_2} \in V_{i_2}$, $V_{i_1},V_{i_2} \in V, i_1 \neq i_2$, 称 CFs 为复合项目集. $Support_{(CFs)}=|CFs|/|D|$ 称为 CFs 的支持度. D 中任一复合项目集 CFs 的支持度 $Support_{(CFs)}$ 不小于某一给定的支持度 minSupport(称为最小支持度),称该复合项目集 CFs 为频繁复合项目集,记为 $CFs_{(Freq)}$. 如果复合项目集 CFs 的任一子集(除空集$\varnothing$和自身外)都是频繁复合项目集,称该复合项目集 CFs 为候选复合项目集,记为 $CFs_{(Cand)}$.

有了上述定义后,下面给出交易树的定义[3].

定义3 交易树 TransTree(D,N,E),其中 D 为交易数据库,$i(v):c$ 表示交易树中的节点(其中 $i \in I, v \in V_i$, $V_i \in V, c=|t_{i(v)}|$), $N=\{i(v):c \mid i \in I, v \in V_i, V_i \in V, c=|t_{i(v)}|\}$ 为交易树的节点的集合,$E=\{<i_1(v_1):c_1,i_2(v_2):c_2>|i_1(v_1):c_1,i_2(v_2):c_2 \in N, c_1 \geqslant c_2$ 且 $\{i_1(v_1),i_2(v_2)\} \subseteq T_{id}, id \in ID\} \cup \{<i_1(v_1):c_1,i_2(v_2):c_2>|i_1(v_1):c_1,i_2(v_2):c_2 \in N, c_1=c_2, Loc(i_1)<Loc(i_2)$ 且 $\{i_1(v_1),i_2(v_2)\} \subseteq T_{id}, id \in ID\}$ 为交易树的边的集合. 图1为 $\{u_1,u_3,u_6,u_7\}$ 构造的交易树.

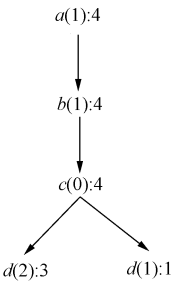


图1 部分交易树

从交易树的定义中,可得到以下性质:

性质1 每个顾客 id 的交易 T_{id} 中的任何一个复合项目必在交易树的唯一一条路径上(忽视交易次数 c),但一条路径可能对应多个顾客的交易.

性质2 交易树的节点交易次数也是交易树在该节点下拥有的最大分支数.

性质3 交易树的路径从根节点到其他节点,节点的交易次数是非严格单调递减的.

性质4 交易树的最大分支数不大于交易数据库 D 的交易总次数 $|D|$.

性质5 交易树的深度为交易数据库 D 的商品数 $|I|$.

从交易数据库到交易树的转化,往往得到的是若干交易树(称为交易树林). 这导致后续处理比较麻烦,为此虚增一个商品 i_0,即 $I=\{i_0\} \cup \{i_1,\cdots,i_m\}$,并假设所有顾客都进行的含有商品 i_0 及其交易量为0的交易(即没有进行实际交易),可得交易树节点 $i_0(0):|D|$. 根据交易树的定义,交易树节点 $i_0(0):|D|$ 必为交易树的根节点,从而得到唯一的一棵完整的交易树,如图2所示. 不加说明,下面讨论的交易树均指这种交易树.

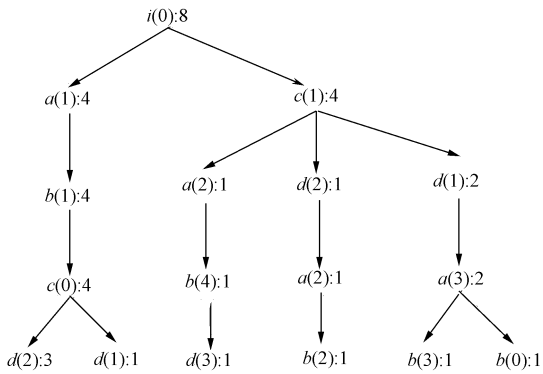


图2 完整交易树

定义4 minSupport 为最小支持度. 根据每个商品 $i \in I$ 和交易量 $v \in V_i$ 的总交易次数 $|t_{i(v)}|$,在

交易树中剪除所有交易总次数$|t_{i(v)}|$小于 minSupport $*|D|$的所有节点得到的交易树子树,称为候选交易树,记为

CandTree(D,N,E,minSupport)

根据性质 3,如取 minSupport=0.3,由图 2 交易树可得对应的候选交易树如图 3 所示.

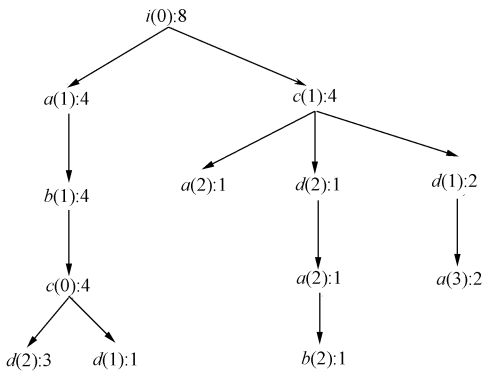


图 3　候选交易树

定理 1　如果在交易候选树中存在频繁复合项目集 $CFs_{(Freq)}$,则该频繁复合项目集 $CFs_{(Freq)}$ 也是交易数据库的频繁复合项目集.

证明　设 $CFs_{(Freq)}$ 为候选交易树 CandTree(D,N,E,minSupport)的频繁复合项目集. 依定义 4,$CFs_{(Freq)}$ 中的复合项目在候选交易树的若干路径上;依定义 3,候选交易树的路径是交易树的子路径;依定义 2,交易树的路径是交易数据库 D 中至少为一个顾客的复合交易项的有序序列. 因此,候选交易树 CandTree(D,N,E,minSupport)的频繁复合项目集 $CFs_{(Freq)}$ 也就是交易数据库 D 的频繁复合项目集.

例如:minSupport=0.25,候选交易树的频繁复合项目集 $CFs_{(Freq)}=\{a(2),c(1)\}$(Support=0.25),对应候选交易树的路径集 $Paths_{(Cand)}=\{(c(1):4,a(2):1),(c(1):4,d(2):1,a(2):1)\}$和交易树的路径集 Paths=$\{(c(1):4,a(2):1,b(4):1,d(3):1),(c(1):4,d(2):1,a(2):1,b(2):1)\}$. 其中,Paths 中任一路径构成的复合项目集分别是顾客 u_2 和 u_4 的交易.

根据这一定理,就可以在候选交易树的路径中求解频繁复合项目集.

2　频繁复合项目集的求解算法

为了便于在候选交易树中描述求解频繁复合项目集,下面给出有关的定义.

定义 5　交易路径为从根节点 $i_0(0)*|D|$ 的下一个节点开始到指定节点 $i(v_i):c_i$ 的前一个节点为止的所有节点的复合项目集点构成的有序序列,且所有的复合项目的交易次数均取 c_i,记为 $p(i(v_i)):c_i=<a_1(v_{a_1}):c_i,\ a_2(v_{a_2}):c_i,\cdots,a_k(v_{a_k}):c_i>$.

由于(候选)交易树中,指定的节点 $i(v_i):c_i$ 有时不只一个,如对 $a(2)$,$|t_{a(2)}|=2$,对应的交易路径有<$c(1):4,a(2):1$>,<$c(1):4,d(1):1,a(2):1$>. 为此对具有相同终点的交易路径给出标识,以示交易路径的区别.

定义 6　某一交易路径($p(i(v_i)):c_i$)的标识路径为标识 label 与交易路径 $p(i(v_i)):c_i$ 构成的二元有序对,记为 $p_{label}(i(v_i)):c_i=(label,p(i(v_i)):c_i)$. 所有具有相同指定复合项目 $i(v_i)$ 的标识路径的集合称为关于复合项目 $i(v_i)$ 的标识路径集,记为 $ps(i(v_i))=\{p_{label_1}(i(v_i)):c_{i_1},p_{label_2}(i(v_i)):c_{i_2},\cdots,p_{label_j}(i(v_i)):c_{i_j}\}$.

在标识路径集中,可能出现相同的复合项目,但在不同的标识路径中. 为了快速求解频繁复合项目集,避免无效的候选复合项目集的生成,必须先对复合项目所处的路径进行标识.

定义 7　$ps(i(v_i))$ 为标识路径集,$a(v_a)$ 为 $ps(i(v_i))$ 中某些标识路径的复合项目,复合项目 $a(v_a)$ 的所有标识定义为一个集合 ls,且 ls=$\{label|\forall p_{label}(i(v_i)):c_i\in ps(i(v_i))$,且 $a(v_a):*\in p_{label}(i(v_i)):c_i\}$,称($a(v_a)$,ls)为标识复合项目,记为 $a(v_a).ls$.

定义 8　标识复合项目集 itemset.ls(其中,itemset 为复合项目的集合,ls 为标识的集合)的定义如下:如果 itemset=$\varnothing$,那么 ls=$\varnothing$;如果|itemset|=1,且 $\exists a(v_a).ls_a$,使得 $a(v_a)\in$itemset,那么 $ls=ls_a$;如果|itemset|>1,且 $\exists a(v_a).ls_a$,(itemset−$\{a(v_a)\}$).ls_{-1} 存在,那么 $ls=ls_a\cap ls_{-1}$. 标识路径集的确定可采用人工智能状态空间的搜索算法求解[4].

有了上述定义后,结合 Apriori 算法,下面给出在已知标识路径集 $ps(i(v_i))$ 中求解频繁复合项目集的算法:

'输入标识路径集 $ps(i(v_i))$,输出关于复合项目 $i(v_i)$ 的频繁复合项目集 $CFs_{(Freq)}(i(v_i))$

Sub GenFreqItemSet($ps(i(v_i))$,$CFs_{(Freq)}(i(v_i))$,minSupport$*|D|$)

'含有复合项目 $i(v_i)$,长度为 1 的频繁复合项目集

① FreqItemSet($i(v_i)$)=$\{\{i(v_i)\}\}$

'含有复合项目,长度为 2 的候选标识复合项目集

② CandItemSet = {itemset. ls | itemset = {$a(v_a)$, $i(v_i)$}, $a(v_a)$. ls 为 ps($i(v_i)$) 中的标识复合项目,且 ls≠∅}
'含有复合项目 $i(v_i)$,长度为 2 的频繁复合项目集
③ For Each itemset. ls in CandItemSet
count = 0
For Each label ∈ ls '选择交易路径
Get $p_{\text{label}}(i(v_i)):c \in \text{ps}(i(v_i))$ by label
If (itemset − {$i(v_i)$}): * ⊆ $p_{\text{label}}(i(v_i)):c$ Then ' * 表示任意值
count = count + c
End if
End For
If count ≥ minSupport * |D| Then
'追加频繁复合项目集
$\text{CFs}_{(\text{Freq})}(i(v_i)) = \text{CFs}_{(\text{Freq})}(i(v_i)) \cup \{\text{itemset}\}$
Else '剪除非频繁复合项目集
CandItemSet = CandItemSet-{itemset. ls}
End if
End For
'含有复合项目 $i(v_i)$,长度 $L>2$ 的频繁复合项目集
④ Do While |CandItemSet| > 1
TemCandItemSet = ∅
For Each itemset_1. ls_1, itemset_2. ls_2 in CandItemSet
$\text{itemset} = \text{itemset}_1 \cup \text{itemset}_2$
$\text{ls} = \text{ls}_1 \cap \text{ls}_2$
'确定长度 L-1 的候选复合项目集
If ls ≠ ∅ Then
If |itemset − itemset_1| = 1 and
|itemset − itemset_2| = 1 and ∀ items ∈ ρ_{-1}(itemset) and items ∈ FreqItemSet($i(v_i)$)
Then '与 Apriori 算法相似,ρ_{-1}(itemset) 为长度为|itemset|−1 的 itemset 的子集
count = 0
For Each label in ls '选择交易路径
Get $p_{\text{label}}(i(v_i)):c \in \text{ps}(i(v_i))$ by label
If {itemset − $i(v_i)$}: * ⊆ $p_{\text{label}}(i(v_i)):c$
count = count + c
End if
End For
End If
End If
'频繁复合项目集
If count ≥ minSupport * |D| Then
'追加频繁复合项目集
$\text{CFs}_{(\text{Freq})} = \text{CFs}_{(\text{Freq})} \cup \{\text{itemset}\}$
TemCandItemSet = TemCandItemSet ∪ {itemset. ls}
End if
End For
'用于产生长度为 L+1 的候选复合集
CandItemSet = TemCandItemSet
End Do
End Sub

GenFreqItemSet 算法可求得含有指定复合项目的频繁项目集. 由于(候选)交易树是根据复合项目出现频率和商品的次序来建立的,因此,通过指定符合项目求解频繁项目集也是唯一的,即如果 $\exists a(v_a), b(v_b)$,且 $a(v_a) \neq b(v_b)$,那么 $\text{CFs}_{(\text{Freq})}(a(v_a)) \cap \text{CFs}_{(\text{Freq})}(b(v_b)) = \varnothing$.

3 频繁复合项目集的求解过程

根据交易树和候选交易树的上述特点,求解交易数据库中的频繁复合项目集 FrequentSet 过程如下:

1) 求解交易数据库 D 中,每个复合项目 $i(v)$ ($i \in I, v \in V_i$) 的交易总次数 $|t_{i(v)}|$,删除交易总次数 $|t_{i(v)}|$ 小于 minSupport * |D| 的复合项目 $i(v)$ ($i \in I, v \in V_i$) 后,根据交易总次数 $|t_{i(v)}|$ 和商品 i 进行从大到小排序,并把结果保留在 SItemSet_1 中. 该原型为 Gen _ SortedItemSet (D, minSupport, SItemSet_1).

实际上,SItemSet_1 为长度为 1 的频繁复合项目集(称频繁单复合项目集).

2) 根据 SItemSet_1,遍历交易数据库 D,逐次读取一个交易 T,从 T 中删除所有不在 SItemSet_1 中的复合项目,并根据 SItemSet_1 的先后顺序,对 T 中保留的复合项目进行排序后形成顾客的复合项目有序序列,再插入候选交易树 CandTree 中. 该过程的原型为 Gen _ CandTree (D, SItemSet_1, CandTree).

3) 根据复合项目 $i(v) \in \text{SItemSet}_1$ 求解标识路径集 ps($i(v)$),其原型为 Gen _ Label _ Paths (CandTree, $i(v)$, ps($i(v)$)).

有了上述的主要过程和 GenFreqItemSet 过程,交易数据库 D 的频繁复合项目集 FrequentItemSet 的求解过程 FABCTA,其原型为 FABCTA

(D,FrequentItemSet),具体过程如下:

```
Sub FABCTA(D,FrequentItemSet)
    FrequentItemSet=∅ '频繁项目集
    '求解长度为1的频繁复合项目,并排序
    Gen _ SortedItemSet(D, minSupport, SItemSet1)
    '没有或只有一个长度为1的频繁复合项目
    If|SItemSet1|≤1 Then
        FreqItemSet=SItemSet1
        Exit Sub
    End If
    '构造交易候选树
    Gen _ CandTree(D, SItemSet1,CandTree)
    '每一个频繁项目 i(v_i)
    For Each i(v_i) in SItemSet1
    '复合项目 i(v_i)的交易路径集
    Gen _ Label _ Paths(i(v),ps(i(v_i))
    '含有复合项目 i(v)的频繁项目集的集合
    CFs(Freq)(i(v_i))=∅
    GenFreqItemSet(ps(i(v_i)),CFs(Freq)(i(v_i)),
minSupport * |D|)
    '追加频繁复合项目集
    FrequentItemSet = FrequentItemSet ∪
CFs(Freq)(i(v_i))
    End For
End Sub
```

4　实验结果与分析

在石油勘探开发乃至生产中,为了了解和掌握地层中油气分布情况必须进行多种测井,并建立相应的数据库. 为了深入研究不同测井之间的相关性,通过关联规则挖掘,进行测井组合的优化[5]. 本文在 Pentium Ⅲ机上,采用 Access 2000 数据库和 Visual Basic 6.0 实现了上述方法 FABCTA,并与 Apriori 算法在求解频繁项目集过程的时间开销进行对比. 新疆克拉玛依陆梁地区一口井的测井数据库 D,$|D|=6\ 842$,$ID=\{\text{DEPTH}\mid \text{DEPTH}\in[3\ 429..\ 4\ 280.5]$,且间距 step=0.25$\}$(即地层测试点的集合),数据库模式 $I=\{$AC, CALI, CALs, CNL, DEN, GR, RI, RT, RXO,SP$\}$(分别为声波测井、井径测井、微差井径测井、中子测井、密度测井、自然伽码测井、侵入带电阻率测井、原状电阻率测井、冲洗带电阻率测井和自然电位测井),对所有测井采集的数据(均为实数)按等距离进行两次离散化,而且所有数据值域分别转换为 $V_1=\{1,2,3,4,5,6,8,7,8,9,10\}$ 和 $V_2=\{1,2,3,4\}$. 为了对比 FABCTA 和 Apriori 算法执行效率,分别采用 0.25,0.5,0.75 和 0.9 为最小支持度进行频繁复合项目集求解时间的比较,测试结果如图 4 所示,图中 V_1 曲线对应 V_1 的离散化,V_2 曲线对应 V_2 的离散化. 从图中任一曲线测试点可以看出:当数据源相同时,最小支持度越小,FABCTA 求解效率越好于 Apriori 求解效率,反之,两者效率越接近. 当数据源不同,但记录数和最小支持度相同时,相同记录数越多,FABCTA 求解效率越好于 Apriori 求解效率,反之,两者效率越接近. 下面对实验结果进行分析.

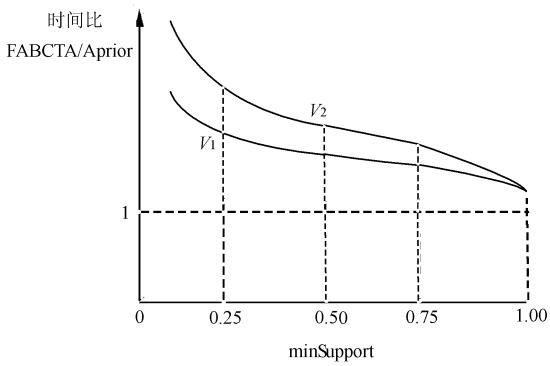


图 4　FABCTA 与 Aprior 执行对比

Apriori 算法的主要时间开销在于:①由长度为 $L-1$ 的频繁集生成长度为 L 的项目集;②对每个长度为 L 的项目集生成长度为 $L-1$ 的所有子集,并遍历当前频繁项目集验证长度 L 的项目集是候选集;③读取交易数据库的交易,遍历候选集进行统计,直到遍历整个交易数据库. 在交易数据库中,数据库中的交易数 $|D|$ 往往远大于商品和交易量总数 $|v_{i_1}|*\cdots*|v_{i_n}|$. 这样遍历交易数据库有可能成为求解频繁项目集的主要时间开销. 而 FABCTA 的主要时间开销在于:①一次遍历交易,求解长度为 1 的频繁复合项目集 SItemSet_1,并从大到小排序;②再次遍历交易数据库,生成候选交易树;③根据长度为 1 的频繁复合项目集 SItemSet_1,从候选交易树中确定标识路径 $\text{ps}(i(v))$;④通过候选复合项目集的标识集 ls,再在路径 $\text{ps}(i(v))$ 中选择更小的交易路径 $p_{\text{label}}(i(v))(\text{label}\in\text{ls})$,最后利用 Apriori 算法在 $p_{\text{label}}(i(v))$ 中求解含有 $i(v)$ 的频繁项目集,而不是在整个数据库中求解.

FABCTA 在候选复合项目集的生成过程中,一方面,通过标识集 $\text{ls}=\text{ls}_1\cap\text{ls}_2$,使得没有共同标识的复合项目不能进行候选复合项目集的生成;另一方面,利用 Apriori 算法中,由长度 $L-1$ 的频

繁集生成长度 L 的候选集的方法,避免了遍历数据库对明知不是频繁集的项目集的判断,同时,使求解频繁复合项目集的范围只局限于 ls 标识的交易路径中. 如果交易数据库中顾客进行的交易越多,而且越均匀,那么由交易数据库对应的候选交易树的分支数就越少,而且交易路径的复合交易项目数往往也比顾客的交易含有的复合项目少. 这就使得需要遍历的复合项目集数减少,最终表现为频繁项目集的求解更加快捷.

5 结束语

综上所述,本文扩展了交易数据库的定义,给出含有商品交易量复合项目、交易树和候选交易树的概念,实现原有数据"浓缩"转储. 同时,结合并保留求解频繁项目集的 Apriori 算法优点,并使之在若干更小范围内有效遍历已"浓缩"的数据,从而达到提高频繁复合项目集的求解快速. 当然,这种高效是以消耗一定的候选交易树空间取得的.

由于候选交易树是根据频繁单复合项目集来构造的,一旦针对某一交易数据库生成候选交易树后,随时增加交易往往会引起频繁单复合项目集的变化,将导致候选交易树重构,因此,FABC-TA 不适用于增量式的频繁项目集的求解.

参考文献 (**References**)

[1] Agrawal R, Srikant R. Fast algorithms for mining association rules[A]. Proceedings of the 20th International Conference on Very Large Databases[C]. Santiago, Chile, 1994. 487 ~ 499

[2] 李国和,吴卫江. 数据库中关联规则的挖掘[A]. 计算机科学[C]. 2001,28(5,专刊):108 ~ 110
Li Guohe, Wu Weijiang. Associate rule mining in database[A]. Computer Science[C], 2001, 28(5, monograph): 108 ~ 110 (in Chinese)

[3] Han Jiawei, Micheline Kambr. Data Mining—Concepts and techniques[M]. Beijing: Higher Education Press, 2001

[4] 林尧瑞,马少平. 人工智能导论[M]. 北京:清华大学出版社,1997. 50 ~ 60
Lin Yaorui, Ma Shaoping. Introduction of artificial intelligence [M]. Beijing: Tsinghua University Press, 1997. 50 ~ 60 (in Chinese)

[5] 李国和,吴卫江,刘延伟,等. 特征选取及其在测井系列选定中的应用[A]. 见:王珏. 计算机科学[C]. 2002,29(9,专刊):337 ~ 340
Li Guohe, Wu Weijiang, Liu Yanwei, *et al*. Feature selection and it's application in logging suit selection[A]. In: Wang Jue. Computer Science[C]. 2002, 29(9, monograph): 337 ~ 340 (in Chinese)