广西科学院学报
Journal of Guangxi Academy of Sciences

2010,26(1):19～22
Vol. 26,No. 1 February 2010

一种新的频繁项集挖掘算法 DS-ECLAT*

A New Mining Algorithm of Frequent Itemsets DS-ECLAT

张 毅,杨 颖,陆瑞兴
ZHANG Yi,YANG Ying,LU Rui-xing

(广西大学计算机与电子信息学院,广西南宁 530004)
(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:在 ECLAT 算法的基础上,提出一种新的频繁项集挖掘算法——DS-ECLAT 算法。该算法使用回写集和深度搜索最长项集两项新技术,在每次迭代中,无须扫描整个数据库,对于(K+1)项集的探索仅依赖于 K 项集,并生成 K 项回写集,下一次迭代时吸取这些回写集,减少了交运算的次数,提高了算法的执行效率。相对于 ECLAT 算法,新算法减少了内存的需要,具有更好的可伸缩性。

关键词:挖掘算法 频繁项集 回写集
中图法分类号:TP181 **文献标识码**:A **文章编号**:1002-7378(2010)01-0019-04

Abstract: DS(deep search)-ECLAT algorithm is a new frequent itemsets algorithm proposed on the fundation of ECLAT algorithm. In algorithm two new technologies writing-back sets and depth search for the longest itemsets are applied. Through scanning of the whole database becomes evitable in each iterative. Depending only on the K key set, the exploring operation of (k + 1) itemsets generates write-back sets of K items, which could be utilized in the next iterative. In this way intersection operations are reduced and algorithm efficiency is improved. Comparing with ECLAT algorithm, DS-ECLAT algorithm requires less memory while acquires better scalability.

Key words: mining algorithm, frequent itemsets, write-back set

随着网络、计算机软硬件的发展,信息量迅速增加,很多经典的数据挖掘方法很难适应大型数据库的需要。文献[1]提出的 Apriori 算法,是事务数据库中发现关联规则挖掘频繁项集最有影响的算法之一。但是因扫描数据库过于频繁,Apriori 算法处理大型数据集并不实际。文献[2]提出一种基于垂直数据格式的 ECLAT 算法,使用 TID 项集存储事务集,整个挖掘过程仅需扫描一遍数据库,极大地提高了剪枝步的效率,但是该算法仍存在诸多问题,如 TID 集可能很长,需要的空间将呈平方数增长,自连接的交运算消耗大量时间,遇到大型数据库时,在可伸缩性与执行效率方面显得力不从心等。ECLAT 算法的研究受到国内众多学者的重视,如文献[3]在 ECLAT 基础上,提出多数约束规则,并赋之以相应的算法;文献[4]基于划分的思想,改进了 ECLAT 算法。本文在 ECLAT 算法的基础上,提出一种新的 DS-ECLAT 算法,以期弥补 ECLAT 算法时空复杂度与可伸缩性的不足。

1 算法推导

我们首先以实例分析 ECLAT 算法的特点和不足,进而进行 DS-ECLAT 算法的推导。在算法推导中将用到如下两条 Apriori 基本性质[5]:

性质 1 频繁项集的所有非空子集也是频繁的。

性质 2 非频繁项集的所有超集也是非频繁的。

1.1 ECLAT 算法分析

表 1 是垂直数据格式,TID 集是包含项集的事

收稿日期:2009-12-10

作者简介:张 毅(1980-),男,硕士研究生,主要从事数据库、数据挖掘研究。

*广西自然科学基金项目(桂科青 0731023)资助。

务标识符的集合。例如：I_1 是某种商品的名称，T_1，T_2，…，T_k 是某次交易活动。某种商品包含很多个交易记录，某次交易可能包含很多商品。频繁项集挖掘的目的是获取某类商品的集合，这类集合的 TID 集项的长度必须大于或等于最小支持度 min _ sup。

表 1 垂直数据格式的 1 项集

项集	TID 集
I_1	T_1,T_3,T_4,T_5,T_6
I_2	T_2,T_3,T_5,T_6
I_3	T_1,T_2,T_5,T_6
I_4	T_2,T_3,T_5

例如：通过扫描一次数据集将水平格式的数据格式转换成垂直数据格式的事物数据库 D，存储于表 1 中，为一项频繁项集，各项支持度计数等同于 TID 集的长度，它们都大于等于 min _ sup(值为 2)。主要任务是找出 D 中所有的频繁项集 L。使用 ECLAT 算法，主要步骤如下：

表 1 中，选取每对频繁单个项的 TID 集的交，可以对该数据集进行挖掘。由于每个项是频繁的，总共进行 $C_4^2=6$ 次交运算，导致生成 6 个非空二项集，所有二项集的事务个数大于等于 2，因此表 2 为二项频繁项集的集合。同理，对表 2 进行 $C_4^3=4$ 次交运算，得出表 3 的结果，唯有项集$\{I_1,I_3,I_4\}$的 TID 集仅为 1，不是三项频繁集。据性质 1，一个(K+1)项集是频繁(K+1)项集，当且仅当(K+1)个 K 项子集都是频繁的，例如产生四项集$\{I_1,I_2,I_3,I_4\}$，其任意三项子集也必须满足最小支持度。如果不满足，不对三项集进行交集运算，并不生成四项集。这里项集$\{I_1,I_3,I_4\}$支持度为 1，并不生成$\{I_1,I_2,I_3,I_4\}$。至此算法终止。

表 2 垂直数据格式的 2 项集

项集	TID 集
I_1,I_2	T_3,T_5,T_6
I_1,I_3	T_1,T_5,T_6
I_1,I_4	T_3,T_5
I_2,I_3	T_2,T_5,T_6
I_2,I_4	T_2,T_3,T_5
I_3,I_4	T_2,T_5

表 3 垂直数据格式的 3 项集

项集	TID 集
I_1,I_2,I_3	T_5,T_6
I_1,I_2,I_4	T_3,T_5
I_1,I_3,I_4	T_5
I_2,I_3,I_4	T_2,T_5

分析：ECLAT 算法共扫描数据库一次，交集运算利用 Apriori 连接步规则[5]，即：频繁集 L_k 为得到候选集 C_{k+1}，须对 L_k 进行自连接，设 L_k 的所有项都是按字典顺序排列，L_k 的元素 l_1 和 l_2 是可连接的，如果$(l_1[1]=l_2[1])\wedge(l_1[2]=l_2[2])\wedge\cdots\wedge(l_1[k-1]=l_2[k-1])\wedge(l_1[k]<l_2[k])$。条件 $l_1[k]<l_2[k]$ 可以保证不产生重复。例中共进行 10 次交运算。该算法的优点是求(K+1)项集的支持度并不要求扫描整个事务数据库 D，但是长的 TID 集做交运算消耗大量时间。

1.2 改进的 DS-ECLAT 算法

项集$\{I_1,I_2,\cdots,I_m\}$中，按照深度优先搜索频繁项集原则，第一次迭代自连接频繁一项项集 L_{11}(表示第一次迭代的一项集的频繁项集合，该项以 I_1 起始)，求出 I_1 为首的候选二项项集 C_{12}及其 TID 集，统计支持度计数，剪去不满足条件的项，得到频繁二项项集 L_{12}。按上述方法依次求出 C_{13}和 L_{13}，此时将 L_{13}各项集去除 I_1 项后的二项集分别写入回写集(Wb)，回写集以首项下标的数字升序排列，再以项的个数升序排列。余下按 L_{13}步骤中的方法求出频繁项集及其回写集，直到求出最长的频繁项集。

第二次迭代求 I_2 为首的频繁项集的集合。第一步从回写集中取出 I_2 为首的二项集直接写入 L_{22}，其 TID 集为空，如果 L_{11}自连接产生的以 I_2 为首的项跟 L_{22}已有的项不等，则求出 TID 集，写入 C_{22}，否则忽略该项。最后对 C_{22}中满足最小支持度的项记入 L_{22}。第二步在第一步基础上求出 L_{23}，生成 C_{23}过程中，L_{22}自连接的两个项集由于 TID 集是否为空将分成 3 种不同的处理方法，这个处理是减少交集运算的关键所在，该步骤暂称为最长项查找，具体将在后文中单独列出。同理，第二步结束前，也要把满足条件的集合写入回写集。以后各步与第二步同，直至求出 I_2 为首的最长频繁项集。

第三次迭代开始与第二次迭代方法一致，直至判断$\{I_{m-1},I_m\}$是否为频繁项集。

最长项查找：长度为 m 的项 l_i 与 l_j。若两项都有 TID 集，则直接连接，做交集运算求出 TID 集并判断是否满足最小支持度。若项 l_i 的 TID 集不为空、l_j 的 TID 集为空，则将 l_i 与 l_j 的最后一项连接，做交集运算，生成候选项及其 TID 集。如果 l_i 与 l_j 的 TID 集都为空，则两项集不同的末项做交运算，在频繁项集中查找相同部分 l 是否有 TID 集，如果没有，则丢弃最后一项，直到找到含有 TID 集的最长项集 l。丢弃项与前项一起(包括 l)，两两做交运算，逐步转化成频繁 2^n(n 为整数，$0<n\leqslant\log m$)项集，直到不满足最小支持度退出或生成 m 项频繁项集。

例如：项$\{I_3,I_4,\cdots,I_8,I_9\}$、$\{I_3,I_4,\cdots,I_8,I_{10}\}$TID

集为空,则先将{I_9,I_{10}}做交集运算得频繁项集 l_1,若余项{I_3,I_4,…,I_8}TID 集仍为空,则丢弃 I_8。如果 I_7 也被丢弃,须将{I_7,I_8}做交集运算得频繁项集 l_2。若项集{I_3,I_4,I_5,I_6}TID 不为空,则将与 l_2 做交运算,结果与{I_3,I_4,I_5,I_6}交,得出 8 项集。同时须将过程中产生的 l_1、l_2 及其合并项和 TID 集写入回写集,如果 8 项集频繁{I_4,…,I_8,I_9,I_{10}}及 TID 集也要写入回写集。

为体现迭代过程,DS-ECLAT 算法的步骤标记为 b1 的形式,表示第二次迭代的第一步骤。

步骤 a1:表 1 已求出垂直数据格式的一项频繁集 L_{11}集,对其自连接生成二项侯选集 C_{12},得图 1a,因其所有项满足最小支持度,L_{12}等于 C_{12}。

步骤 a2:图 1b 是 L_{12}自连接并进行交集运算的结果 C_{13},项{I_1,I_3,I_4}TID 项仅有一个,须剪除,得到频繁项集 L_{13}。

步骤 a3:利用性质 1,在 L_{13}中去掉 I_1 项推导出图 1c 的回写集。

步骤 a4:对 L_{13}中连接并做交运算得四项集,其最小支持度为 1。第一次迭代终止。

步骤 b1:从回写集中取出 I_2 项为首的项集。表 1 中,以 I_2 项为起始项与下标大于 2 的项连接,依次对比回写集,如果项集相同,则将此项写入 L_{22},TID 集用"*"号替代。否则将此项做交集运算,支持度大于等于 min _ sup 的项记入 L_{22}。最后得到图 1d 的 L_{22}。

步骤 b2:L_{22}的 TID 集为空,先对去除的末项 I_3,I_4 做交集运算,得出图 1e,并将该项集写入回写集。

步骤 b3:最长项集查找的最长项为{I_2},与集合{I_3,I_4}做交集得出图 1f。应将图 1f 的项集去掉 I_2 后的集合{I_3,I_4}写入回写集,但是已经存在。

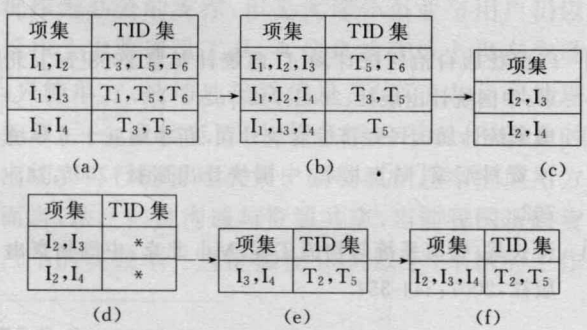

图 1　DS-ECLAT 算法过程

步骤 c1:第 3 次迭代,从回写集中读取以 I_3 项为起始的项集,只存在一个二项集{I_3,I_4},因此算法终止。

算法比较:因 ECLAT 算法和 DS-ECLAT 算法的 TID 集的不确定性,无法用常用的大 O 表示法准确估算时间复杂度。ECLAT 算法共进行 10 次交运算,DS-ECLAT 算法共进行 9 次交运算,另须进行 3 次回写集写入,与回写集进行 3 次对比。相对于繁冗的交运算,回写集的写入与查找比较付出的时间代价要小的多。就空间复杂度来说,DS-ECLAT 算法明显优于 ECLAT 算法,因为回写集的出现,其 TID 集为空,节约了大量存储空间。

ECLAT 算法及 DS-ECLAT 算法的结果都是基于比较的。前者每一频繁项集必须经历一次比较才能获得。而 DS-ECLAT 算法由于使用了回写集与最长项查找两个步骤,有效的减少了比较次数。为验证两个 TID 集为空项数为 m 的回写集的交集是频繁的,须从 L 中查找最长的带有 TID 集的项数为 n 的项集,生成(m+1)项频繁项集共比较(m−n)次,这个比较生成新的频繁项,而不影响本次迭代继承的回写集比较次数,因此 DS-ECLAT 算法在这一步中减少了(m−n)次比较次数。

2　DS-ECLAT 算法描述

使用垂直数据格式挖掘频繁项集的 DS-ECLAT 算法描述如下。

输入:垂直数据库 D//项 I1、I2…Im

　　最小支持度 min _ sup

输出:L　//D 中满足最小支持度的频繁项集

方法:基于回写集的深度搜索最长频繁项集策略

```
L11=find _ vfrequent _ 1-itemsets(D);//Lik 是以 I1 开头的频繁 K 项集。目的是找到频繁一项集
for(k=2;k<=m;k++){
  for each 以 I1 为首的项集 l1 ∈L1(k-1);
    for each 以 I1 为首的项集 l2 ∈L1(k-1);
      if(l1 与 l2 可连接)then
        {c=l1 ⋈ l2; add c to C1k}
      if is _ infrequent(c,L1(k−1)) then
        delete c;
      else add c to L1k,ad (c-I1) to Wb;
}
  L1=∪kL1k
for(i=2; k<=m−i+1;i++){
  for(k=2; k<items _ count;k++){
   Lik =DS _ ECLAT _ gen(Li(k−1));
  }
```

```
  Li=∪kLik
}
return L=∪iLi
DS-ECLAT _ gen(Li(k-1)){
  将 Wb 中 Ii 为首的项集读入 Lik 中;
  for each 以 Ii 为首的项集 l1 ∈L1(k-1);
    for each 以 Ii 为首的项集 l2∈L1(k-1);
    if(l1 与 l2 可连接)then{
      if(l1 与 l2 的 TID 集都不为空)
        then{c=l1⋈l2; add c to Cik;}
      else if(l1 与 l2 的 TID 集一个为空)
        then{ c=l1⋈l2 尾项; add c to Cik;}
      else DSearch(l1, l2,Li);
      if is _ infrequent(c,Li(k-1)) then{   //进行
交运算,并判断支持度
       delete c;
    else add c to Lik,ad (c-Ii) to Wb;
Return Lik }
DSearch(l1,l2,Li){
if(c3=(l1 末项⋈l2 末项)频繁)c3 写入回写集;
for(l=l1-末项;l 不为空;l=l-末项)T=T∪末项;
link(T){
  if(T 不止一个项集)
    for each 邻近两项 l3,l4∈T,步长为 2;{
    l5=l3⋈l4;    l5 add to Wb;
    T1=T1∪l5;Link(T1);}
  else return T1;
l=l⋈l2;l add to Wb;   //l2 为 T 的唯一项
if(l=l⋈c3 频繁) then
  { (l-Ii) add to Wb;return l;}
else return -1;
}
```

3 结束语

频繁项集挖掘是关联规则挖掘的重要方法之一,本文提出的DS-ECLAT算法采用深度挖掘的方法,每次迭代处理以 I_k 为起始项的所有项的频繁项的集合,每次装入内存的信息量减少,回写集的出现更是减少了交集的次数。不过也因此带来了新的问题:首先,建立一个回写集,要花费新的时间,其次,为确定项集 l 是否为频繁集,须依次查找回写集。总的来说,这些代价与减少的交集次数相比,是很值得的。DS-ECLAT算法相对ECLAT在可扩展性与执行效率方面都有大幅度的提高。

参考文献:

[1] Agrawal R,Srikant R. Fast algorithms for mining association rules: proc 1994 Int Conf Very Large Databases(VLDB'94)[C]. 1994:487-499.

[2] Zaki MJ. Scalable algorithms for association mining[J]. IEEE Transactions Knowledge and Data Engineering, 2000,12(3):372-390.

[3] 李宏,陈松乔,陈建二,等. 基于ECLAT算法的多种约束关联规则挖掘算法研究[J]. 计算机测量与控制, 2006,14(7):934-936.

[4] 熊忠阳,耿晓斐,张玉芳. 一种新的频繁项集挖掘算法[J]. 计算机科学,2009,36(4a):42-44.

[5] Han Jiawei,Kamber M. Data mining:concepts and techniques[M]. San Francisco: Morgan Kanfmann, 2006.

(责任编辑:韦廷宗)

(上接第18页)

参考文献:

[1] 刘明菲,李兰. 区域物流与区域经济互动作用机理分析[J]. 工业技术经济,2004(3):40-43.

[2] 何秋,桂寿平. 区域物流系统动态模型的建立与合理性检验[J]. 交通与计算机,2002(3):30-33.

[3] 王俊. 中国物流业对经济增长作用的实证分析[J]. 科技情报开发与经济, 2004(1):69-70.

[4] 鄢飞,董千里. 陕西区域物流网络的构建研究[J]. 西北农林科技大学学报,2008(8):51-57.

[5] 广西壮族自治区统计局. 广西统计年鉴2008[M]. 北京:中国统计出版社,2008:745-762.

[6] 国家统计局国民经济综合统计司. 新中国五十五年统计资料汇编[M]. 北京:中国统计出版社,2005:745-762.

[7] 蒋长兵. 物流系统与物流工程[M]. 北京:中国物资出版社,2007:344-354.

(责任编辑:尹 闯)