# 关联规则挖掘 Apriori 算法的优化①

王胜和　(安徽公安职业学院 公安科技系 安徽 合肥 230031)

朱玉全　(江苏大学 计算机科学与通信工程学院 江苏 镇江 212000)

**摘　要：** 针对 Apriori 算法存在的不足，提出了一种新的优化 Apriori 的方法。该方法通过优化频繁项集修剪策略，减少无效候选项集的产生；优化连接策略，减少连接次数，避免相同项目的多次重复比较；结合事务数据库逐步压缩技术，减少对无用事务的扫描次数。实验结果表明，经过优化的 Apriori 算法具有更好的运行效率。

**关键词：** 关联规则; apriori 算法;事务压缩;数据挖掘

## Optimized Apriori Algorithm for Mining Association Rules

WANG Sheng-He (Department of Public Security Technology, Anhui Police Professional College, Hefei 230031, China)

ZHU Yu-Quan (School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China)

**Abstract:** To address disadvantages of the Apriori algorithm, a new method is presented to optimize the Apriori. It can reduce the number of invalid candidate item sets through optimizing the strategy of frequent item sets pruning. In order to reduce the connections of items, avoid repeated comparison of the same items, it optimizes the joining strategy. And it also removes the useless transactions from database step by step in order to reduce the times of scanning transactions. The results of experiment show that the optimized algorithm is more efficient.

**Keywords**: association rule; apriori algorithm; transaction reduction; data mining

## 1　引言

数据挖掘是一种从大量数据中提取出隐含的、未知的、潜在的和有用的信息的过程。数据挖掘是解决当今“数据爆炸而知识贫乏”的一种有效办法。虽然数据挖掘技术的发展仅有短短十余年的历史，但发展势头却相当迅猛，现已在实际领域中得到广泛的应用，并且产生了良好效果。关联规则是由 R. Agrawal 等人于 1993 年提出的，是数据挖掘问题中的一个重要研究内容。Apriori 算法[1]是最有影响的挖掘布尔关联规则频繁项集的算法之一。其基本思想是：采用自底向上的逐层迭代方法，利用已知的(K-1)(K>1)维频繁项集来生成 K 维候选项集，再通过扫描数据库来计算所有 K 维候选项集的支持度，从而求出 K 维频繁项集。通过对 Apriori 算法的分析，可以发现算法存在明显的不足:(1)算法在连接前，没有排除不可能成为频繁项集的项目，将产生巨大的候选项集。(2)算法在连接过程中，通过模式匹配来决定两项集是否连接，重复比较的项目太多。(3)没有对事务数据库及时压缩，对无用事务重复扫描。针对 Apriori 算法的不足，一些学者从不同的角度改进了 Apriori 算法。文献[2]从项集约简、连接策略优化、事务压缩方面对 Apriori 进行了优化，但其约简和连接策略不全面；文献[3]只改进了 Apriori 算法的连接策略；文献[4,5]只通过约简无用事务来提高 Apriori 效率。这些算法尚不全面。

① 基金项目:安徽省高等学校青年教师科研资助计划(2005jq1168)
收稿时间:2009-11-17;收到修改稿时间:2009-12-21

文章将提出一种新的算法，该算法采用连接前修剪繁项集及优化连接策略，结合数据库逐步压缩技术，从而减少候选项集的数目，避免了连接过程中过多的模式匹配，同时削减数据库的规模，减少对无用事务的扫描，整体提高算法效率。

本文第 2 节介绍相关工作。第 3 节给出标志位线性分析优化算法。第 4 节给出实验数据以及分析结果。第 5 节给出结论以及未来工作。

### 1.1 基本介绍

在拥有标志寄存器的处理器中，绝大多数的运算指令都会对标志位定值，虽然这些定值并不一定会使用到。如果对源指令中标志位的每次定值都进行翻译，那么所产生的目标代码的效率就会大幅度下降。平均而言，在翻译所产生的代码中，对每一个标志位的定值所需要的代码与翻译一条指令基本功能的代码数量差不多，而每条指令都会对多个标志位进行修改，那么，如果完全把标志位定值翻译出来，而这些定值并没有被使用，那么浪费就显而易见了。

## 2 算法基本思想

### 2.1 频繁项集修剪策略

性质 1[2] 如果(K-1)维频繁项集 $L_{k-1}$ 中包含单个项目 i 的集合数量小于 k-1，则 i 不可能包含在频繁 K-项集中。

在第 K 步中，根据 K-1 步生成的 K-1 维频繁项集来产生 K 维候选项集，由于在产生 K-1 维频繁项集时，可以顺便实现对该项集中出现项目的个数进行计数处理。根据性质 1，若某项目的计数小于 K-1，则该项目不可能出现在 K 维频繁项集中，没必要再让其参与连接。所以在连接前，可以先删除 Lk-1 中所有包含该项目的项集，从而排除由该项目所引起的大规模组合，减少候选项集的数量。

### 2.2 优化连接策略

在对 Apriori 算法的分析中发现，Apriori 算法需要进行大量的操作是：判断两个(K-1)项集是否前 K-2 项相同且最后一项不同。这项操作占了较多的运行时间。如果能减少这项操作执行的次数，就可以提高算法的运行效率。Apriori 算法中各交易记录中的项目均是已经按字典排序的，所以由 Apriori 算法生成的候选项集也是有序的，生成的频繁项集也是如此。对于两个(k-1)项频繁项集 $l_1$ 和 $l_2$，若 $l_1$ 不能与 $l_2$ 连接则 $l_1$ 与 $l_2$ 之后的所有(k-1)项集都不能满足连接条件，所以只要 $l_1$ 与 $l_2$ 不能连接，就不需再判断 $l_1$ 与 $l_2$ 之后的所有(k-l)项集是否能连接。从而减少连接过程中项目匹配的次数，提高算法的运行效率。另外，根据文献[3]的思想，在 $L_1$ 进行连接时，若某个频繁 1 项集的支持度越大，那么该项集与其它频繁 1 项集连接生成的候选 2 项集成为频繁项集的可能性越大，所以在找出了频繁 1 项集后，可以对其中的项按支持度由小到大进行排序，使得支持度小的项排在前面，那么在 $L_1$ 连接生成候选集并采用支持度剪枝后，结果也使得支持度小的频繁 2 项集排在前面，依此类推。这样 $L_{k-1}$ 连接生成的候选集数量递减的速度较快。

### 2.3 库优化策略

性质 2[4] 设 $L_k$ 为(k-1)维频繁项集，若 $T \subseteq I$，且 $|T|=k-1$，则在后续的 k 项频繁项集挖掘中，T 为数据库中可删除事务。

性质 3[5] 如果某一事务 T 包含 Ck 中的项集个数小于 k+1，则在后续的(k+1)项频繁项集挖掘中，T 为数据库中可删除事务。

证明见文献[5]。

从 Apriori 算法可以看出，在由 $C_k$ 产生 $L_k$ 的过程中，需要对数据库扫描一次，来计算每个候选项的支持数，但是此时数据库中可能有些事务已经对频繁项集的生成不起作用，根据性质 2 和性质 3，对事务数据库 D 进行筛选，逐步将不符合条件事务项作删除标记，以减小后续挖掘中数据库 D 的规模。

## 3 NewApriori 算法描述

NewApriori 算法是对 Apriori 算法的优化，采用了一种数据库优化技术，不断缩小数据库的规模，并结合修剪频繁集和连接优化策略，大大提高了挖掘的效率。

关联规则挖掘 NewApriori 算法：

输入：事务数据库 D，最小支持度 min_sup。

输出：D 中频繁项集 Lk。

算法：

(1) $L_1$=find_frequent_1_itemsets(D)

(2) sort ($L_1$);//对频繁 1 项集按支持度由小到大排序

(3) for(k=2;$L_{k-1} \neq \phi$;k++){

(4) $L/_{k-1}$=delete($L_{k-1}$);//从 $L_{k-1}$ 中删除不可能

得到频繁项集的集合

(5) Ck= apriori_gen(L/k-1,min_sup);

(6) For each transaction t∈Dk-1{

(7) $C_t$=subset($C_k$,t);//得到每一个事务 t 包含 $C_k$ 的子集 $C_t$

(8) For each candidate c∈$C_t$){

(9) c.count++;

(10) }

(11) }

(12) $D_k$=reduce($D_{k-1}$);//压缩数据库

(13) $L_k$={c∈$C_k$|c.count>=min_sup};

(14) }

(15) Return L=∪$_k L_k$;

procedure delete($L_{k-1}$) //从 $L_{k-1}$ 删除包含项目出现次数少于 k-1 的项集

(1) for all l in Lk-1{

(2) if i∈l {

(3) i.count++;//统计 $L_{k-1}$ 中单个项目个数

(4) }

(5) }

(6) if (i.count<k-1){

(7) if i∈l

(8) delete l from $L_{k-1}$;//删除 $L_{k-1}$ 中所有包含项目 i 的项集

(9) }

(10) return $L_{k-1}$

procedure apriori_gen($L_{k-1}$,min_sup)//利用项集中项目的有序性优化连接

(1) for each itemset I1∈Lk-1{

(2) for each itemset I2∈$L_{k-1}$∧I2 behind I1{

(3) if (I1[1]= I2[1])∧(I1[2]=I2[2])∧…∧(I1[k-2]=I2[k-2])

(4) {c=I1∪I2;

(5) if has_infrequent_subset(c,$L_{k-1}$)

(6) delete c;

(7) else

(8) add c to $C_k$;

(9) }

(10) else break;

(11) }

(12) }

(13) return $C_k$;

procedure reduce($D_{k-1}$)//由 $C_k$ 求 $L_k$ 的过程扫描压缩数据库

(1) if （ |t|=k ） or (|$C_t$|<k+1)//删除长度为 k 的事务，删除包含 $C_k$ 子集个数小于 k+1 的事务，t 为 $D_{k-1}$ 中的事务项。

(2) delete t from $D_{k-1}$;

## 4 性能分析

该算法首先根据优化的频繁(K-1)项集来生成候选项目集，避免了不必要的组合，减少了候选项集的产生。其次利用项集有序特点改进了 Apriori 算法的连接过程，避免了进行大量的重复比较操作和连接操作，节省了时间。最后充分利用候选项集支持度计数过程中的中间结果来标记无用事务，逐步减小数据库的规模，以减少下一次扫描数据库所花费的时间，从而提高了整个算法的效率。

为进一步验证 NewApriori 算法的有效性，采用标准数据集 mushroom(119 个不同项目, 8124 条事务，平均每个事务 23 项,文件大小为 558 KB)，进行频繁项集挖掘实验。实验的硬件配置是 Pentium2.93GHz CPU，1G 内存，操作系统是 Windows XP，编程工具为 VC++6.0。分别实现了 NewApriori 算法和 Apriori 算法。性能对比分析指标是两种算法在不同支持度和事务数下的运行时间。图 1 为两种算法在固定事务数(8124 条事务)，不同最小支持度情况下的比较结果；图 2 为两种算法在固定最小支持度为 30%，不同事务数情况下的比较结果。

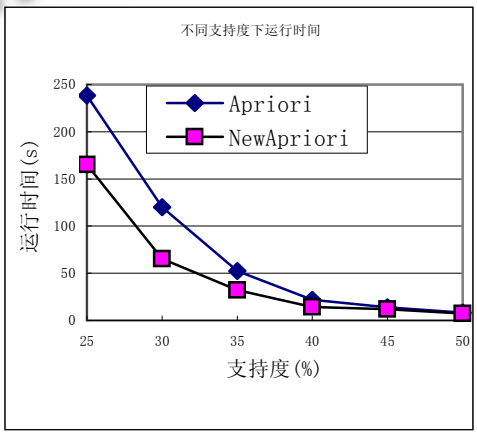


图 1　不同支持度下的运行时间

从图 1、2 可以看出，NewApriori 算法比 Apriori 算法

有更好的时间性能，在小支持度时，优势更为明显。另外，随着事务数的增加，NewApriori 算法运行时间的增长幅度相对较小。主要是因为 NewApriori 算法产生更小规模的候选项集，并且扫描逐步压缩的事务数据库。

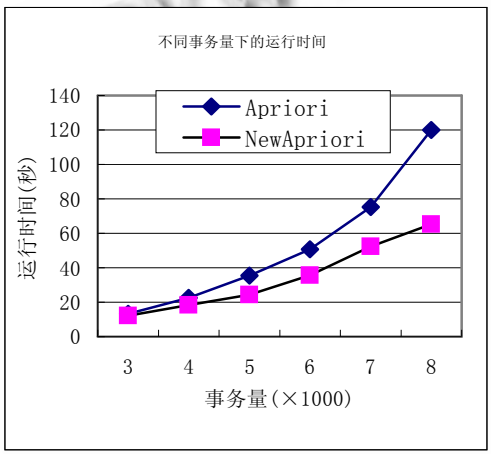


图 2 不同事务量下的运行时间

## 5 结语

文章首先分析了 Apriori 算法及存在的不足， 主要从三个方面对 Apriori 算法进行了优化，一是优化了频繁项集修剪策略，避免了不必要有项目组合；二是利用项集有序性改进了原来算法连接判断方式，减少比较次数，减少候选项目集的元素的数目；三是基于事务压缩，逐步删除无用事务，减少算法扫描数据库的次数。提出了 NewApriori 算法，给出了实验结果。实验表明 NewApriori 算法较 Apriori 算法有更好的时间性能。但针对特定领域的数据集，如何优化算法将是下一步的研究目标。

## 参考文献

1 Agrwal R, Srikan R. Fast Algorithms for Mining Association Rules in Large Databases. Proc. of the Twentieth International Conference on Very Large Databases,Santiago,Chile 1994,9:487－499.

2 徐章艳,刘美玲,张师超,卢景丽,区玉明.Apriori 算法的三种优化方法.计算机工程与应用, 2004,40(36):190－192.

3 刘美玲,徐章艳,卢景丽,区玉明,袁鼎荣,吴信东.利用项集有序特性改进 Apriori 算法.广西师范大学学报(自然科学版), 2004,22(1):33－37.

4 刘培奇,李增智,王云岚,朱海萍,赵银亮.基于数据库约简的关联规则挖掘算法.西安交通大学学报(自然科学版), 2003,37(8):836－839.

5 王曙光,施小英. 一种改进的相联规则提取算法.计算机工程与应用, 2002,38(15):173－174.