DOI: 10.13656/j.cnki.gxkx.2005.02.010
广西科学 Guangxi Sciences 2005, 12(2): 115~117, 122

# Apriori算法的复杂性研究*
# Studies on Complexity of Apriori Algorithm

袁鼎荣[1,2], 严小卫[1,2]
Yuan Dingrong[1,2], Yan Xiaowei[1,2]

(1. 广西师范大学数学与计算机科学学院, 广西桂林 541004; 2. 悉尼科技大学信息技术学院, 澳大利亚悉尼)
(1. Coll. of Math. & Comp. Sci., Guangxi Normal Univ., Guilin, Guangxi, 541004, China; 2. Faculty of Info. Tech., Univ. of Tech. Sydney, Sydney, Australia)

摘要: 介绍关联规则挖掘及 Apriori 算法, 分析事务数据库的特性及 Apriori算法的复杂性, 指出频繁项集挖掘算法的优化途径.

关键词: Apriori算法 数据挖掘 事务数据库 频繁项集
中图法分类号: TP301.6 文献标识码: A 文章编号: 1005-9164(2005)02-0115-03

**Abstract** Association rules mining and Apriovi algrithm was introduced. The characteristics in tran-saction database and the complexity of the Apriori algorithm is also analyzed. Approaches of improving the algorithms for mining frequent item sets are given.

**Key words** Apriori algorithm, data mining, transaction database, frequent itemset

数据挖掘技术近十年得到迅猛发展, 它吸引了数据库、人工智能、数理统计等方面专家的广泛兴趣. 数据挖掘的一个重要领域—— 关联规则的挖掘主要研究事务数据库中事务项间的因果关系, 因此关联规则在决策支持系统中具有较好的应用前景.

计算机技术的发展和条形码录入应用的扩展, 为事务数据库的数据采集与事务数据库的形成提供优越的技术条件, 从而形成了海量的事务数据库. 商业竞争的白热化, 使得决策层需要从海量的数据中寻找一些关联性的规则, 进一步推理出一些商业规律, 既方便客户, 也方便商业企业的管理, 以使商业利润最大化. 这促使了关联规则挖掘研究的兴起.

现今, 关于关联规则的研究已有许多成果[1~4]. 这些成果中, 核心技术是频繁项集的挖掘方法[1,4,5]. 经典的频繁项集挖掘方法是 Apriori算法. 所有频繁项集的挖掘方法所处理的数据与现实事务数据库中所处理的海量数据从质和量来说都有较大的差距. 为进一步将理论推向应用, 以使频繁项集挖掘算法能处理实际海量数据, 本文对事务数据库的事务数据特性及 Apriori算法的复杂性作分析研究, 并指出了频繁项集挖掘算法的优化途径.

## 1 关联规则挖掘的概述

关联规则挖掘就是从海量的事务数据库中, 发现支持度与可信度分别大于域值 $minsupp$ 与 $minconf$ 的事务项之间相互关联的规则[1~10]. 例如: $I=\{i_1, i_2, \cdots, i_n\}$ 为项目集; $D=\{T_1, T_2, \cdots, T_n\}$ 为事务数据库, $T_i$ 是一个事务, 也即一次交易, 为某些项的集合, $T_i \subseteq I$. 对于给定的项集 $A$ 和事务 $T$, 若有: $A \subseteq T$, 则事务 $T$ 包涵了项集 $A$.

如果 $X \subseteq I, Y \subseteq I, X \cap Y = \varnothing$, 对于规则 $X \rightarrow Y$, 若满足:

(1) $supp(X \cup Y) \geqslant minsupp$,

(2) $conf(X \rightarrow Y) \geqslant minconf$,

则称 $X \rightarrow Y$ 是强关联规则, 其中条件(1)是关联规则挖掘的核心问题. 满足(1)的集合 $X \cup Y$ 称为频繁集. 因此, 关联规则的挖掘问题就转化为频繁集的挖掘问题. 挖掘的对象是现实中海量的事务数据库[1~3,6~9]. 挖掘的方法以 Apriori算法为代表, 而且 Apriori算法有多种变型[4,10].

收稿日期: 2004-07-21
作者简介: 袁鼎荣(1967-), 男, 广西全州人, 讲师, 主要从事数据挖掘和人工智能研究.
* 广西科学基金(桂科基0448093)、清华大学智能技术与系统国家重点实验室开放课题和广西师范大学科研基金资助项目.

## 2 Apriori算法

Apriori算法是一种经典的算法,它采用自底向上的搜索方法,即先搜索发现1-项、2-项3-项频繁项集,最后发现最大频繁项集.设 $DB$ 为事务数据库,$C_K$ 表示长度为 $k$ 的候选频繁项集,$L_k$ 表示第 $k$ 次扫描数据库时所得的长度为 $k$ 的频繁项集,$k$ 为扫描次数记录器.该算法定义了一个重要过程,即 **Apriori-gen** ($L_k$),用于从 $k$ 阶频繁项集生成 $k+1$ 阶频繁项集. **Apriori-gen**($L_k$)调用了2个子过程 **join**()与 **prune**(),前者用于合并 $L_k$ 中前 $k-1$ 项相同,第 $k$ 项不同的频繁项集产生长度为 $k+1$ 的侯选频繁项集 $C_{k+1}$,后者用于剪掉 $C_{k+1}$ 中的项的任意子集不在 $L_k$ 中的项,得到第 $k+1$ 阶候选集 $C_{k+1}$[1-3].

Apriori算法详细描述如下[1,4,5]:

```
Algorithm Apriori
input: DB, minsupp;
output: all frequent itemsets;
method
begin
  L0 := ∅;
  k := 1;
  C1 := {{i} | i ∈ I};
  FrequentItemsets := ∅;
  while Ck ≠ ∅ {
    Read DB and count supports for itemsets in Ck;
    Lk := {frequent itemsets in Ck};
    Ck+1 := Apriori-gen(Lk);
    k := k+1;
    FrequentItemsets := FrequentItemsets ∪ Lk;
  }
  return FrequentItemsets;
end.
Procedure join
input: Lk;
output: preliminary candidate set Ck+1;
begin
  for i from 1 to |Lk-1|
    for j from i+1 to |Lk|
    if Lk-itemset i and Lk-itemset j have the same (k-1)-prefix then
    Ck+1 := Ck+1 ∪ {Lk-itemset i ∪ Lk-itemset j};
    else break;
  end.
Procedure prune
input: preliminary candidate set Ck+1;
output: Ck+1;
begin
  for all itemsets c in Ck+1
    if all k-subset s of c is not in Lk
      delete c from Ck+1;
end
```

## 3 事务数据库的特性

本文讨论的事务数据库中的记录为事务,数据项为事务项.设 $I=\{i_1, i_2, \cdots, i_n\}$ 为项目集;$DB=\{T_1, T_2, \cdots, T_n\}$ 为事物数据库,$|I|=n$ 为项目集元素总数,即商品项目总数,$|DB|$ 为事务数据库记录总数,即交易数[1,5,10].事务数据库有如下性质:

性质 3.1 $|I|$ 相对稳定,$|I|$ 的大小依超市规模的大小而定,且远远大于频繁项集的最大维数.

如果超市规模相对稳定,商品项目的数量也相对稳定,虽然一些有较好市场前景的产品项目加入项目集,但也会有失去市场前景的淘汰产品项目从项目集中消除,所以 $|I|$ 将相对稳定.

性质 3.2 $|DB|$ 近似线性增长.

任何超市在任何作业时段内都存在交易,交易的数量依其规模的大小而定,少则每天几百条,多则上万条交易记录加入事务数据库.针对某一特定 $DB$,其长幅在某一幅度内变化,从而使得 $|DB|$ 随时间成近似线性增长.

性质 3.3 $I$ 中事务项 $i$ 发生的次数呈近似线性增长,其支持度在某个稳定的区间内波动.任何交易都是事务项的集合,随着 $|DB|$ 的增长 $i$ 的发生也随之增长.$i$ 的支持度满足: $supp(i)=i.count/|DB|$.根据销售实践可知 $supp(i)$ 是个相对稳定的值.它的变化反映该事务项的销售状态,$supp(i)$ 大于某个临界值时可定性该产品为旺销产品,小于某个临界值时可定性为滞销产品.

性质 3.4 交易 $t$ 的长度 $|t|$ 满足: $1 \leqslant |t| \leqslant large$,$large$ 是交易的最大维数.大于 $large$ 的交易可以看成是例外交易.平均交易长度 $len=\sum len.t_i/|DB|$ ($1 \leqslant i \leqslant |DB|$).交易 $t$ 的长度 $|t|$ 服从正态分布,$len=(1+large)/2$.

性质 3.5 $minsupp$ 是人为的阈值,可用于定界一个良性发展的超市各个项目的周转是否频繁,故 $minsupp$ 可定义为如下参考值:

(1) $minsupp=((len \times |DB|)/|I|)/|DB|=len/|I|$;

(2) $minsupp=(len-s) \times |DB|/|I|/|DB|=(len-s)/|I|$,其中 $s$ 为平均长度的标准差;

(3) $minsupp = (len - 3s) \times |DB| / |I| / |DB| = (len - 3s) / |I|$.

依统计理论,满足(1)的 $minsupp$ 可使得 50% 的项目定性为频繁项,满足(2)的 $minsupp$ 可使得 79.12% 的项目定性为频繁项,满足(3)的 $minsupp$ 可使得 99.85% 的项目定性为频繁项,$minsupp$ 的具体取值由用户依挖掘目的及对超市经营状况决定.

## 4 Apriori算法复杂性分析

Apriori算法采用自底向上的方法,首先扫描 $DB$,计算长度为 1的频繁项集的支持度,根据 $minsupp$ 产生 1-频繁项集.在 1-项频繁集的基础上调用 $Apriori\text{-}gen(L_k)(k > 1)$,产生频繁 2-项集,3-项集,…,$k$-项集,直到 $C_k$ 为空.其复杂性主要表现在对事务项集的访问上,包括 $DB$ 的访问及 $L_k$ 与 $C_k$ 的访问[1,4,5]. Apriori算法有如下特性:

性质 **4.1** 当 $|I| > \lambda, k <_{-} (\lambda,_{-}$ 为某一整数)时,扫描 $L_k$ 的开销比扫描 $DB$ 的开销要大.

证明 设 $minsupp = len \times |DB| / |I| / |DB| = len / |I|$.根据 Apriori算法及性质 3.5有:

$C_1 = I, |L_1| = 0.5 \times |I|$.

当 $k = 2$,生成 2-频繁候选项集时,至少循环扫描 $L_1$ 两次,则访问 $L_1$ 的元素(即事务项集)的次数为: $0.25 \times |I|^2$.取 $\lambda = 20000$,令 $0.25 \times |I|^2 = |DB|$ 则 $|DB| = 10000$(万).若事务数据库以 10万条/天的记录增加也要近 3a的时间.而 3a对超市来说,商业环境发生了较大的变化,这是不切实际的.

当 $k = 3$,生成 3-频繁项集时,需循环扫描 $L_2$ 三次,访问 $L_2$ 的元素(即事务项集)的次数为: $|L_2|^3$.

当 $|L_2| = \overline{|L_1|^2}$ 时,访问 2-频繁项集的复杂性与 1-频繁项集的相同.故当 $|I| > \lambda, k <_{-}$ 时,扫描 $L_k$ 的开销比扫描 $DB$ 的开销要大.

性质 **4.2** 事务项集的访问复杂性为

$$f(k) = \sum_{i=1}^{k} |BD| \times C_i + L_2^2 + \sum_{j=3}^{k-1} L_{j-1}^3.$$

证明 因为 $C_1 = \{\{i\} | i \in I\}$,设扫描数据库的次数为 $k$,$k$ 是一个小于最大事务项长度且不小于 2的整数.则

(1) 产生 1-频繁项集时访问数据项集次数为: $|DB| \times C_1$;

(2) 产生 2-频繁项集时访问数据项集次数为: $|DB| \times C_2 + L \times L_1$;

(3) 产生 3-频繁项集时访问数据项集次数为: $|DB| \times C_3 + L_2 \times L_2 \times L_2$;

(4) 产生 $k$-项频繁集时访问数据项集次数为: $|DB| \times C_k + L_{k-1} \times L_{k-1} \times L_{k-1}$.

所以

$$f(k) = |DB| \times C_1 + |DB| \times C_2 + L_1^2 + |DB| \times C_3 + L_2^3 + \cdots + |DB| \times C_k + L_{k-1}^3 = \sum_{i=1}^{k} |DB| \times a + L_1^2 + \sum_{j=2}^{k-1} L_{j-1}^3.$$

推论 **4.1** Apriori算法不宜用于挖掘较长频繁项集.

设 $|L_k| = a$.根据最大频繁项集的子集是频繁集的理论可有 $a \times (2^k - 2)$ 个频繁子集,则访问 $DB$ 次数为 $|DB| \times (a \times (2^k - 1))$,访问 $L_k$ 次数为 $a \times 2 \times 2^k$.当 $k$ 增大时,访问最大频繁集的子集及计算子集的支持度复杂性开销显著增大.与直接挖掘最大频繁项集相比,就是访问冗余显著增加.

推论 **4.2** 控制 $|L_k|$ 的大小是降低复杂性的关键步骤.

根据 $C_{k+1}$ 的生成方法及性质 3.2可直接得到.

推论 **4.3** 数据库扫描次数及库结构直接影响复杂性.

推论 **4.4** $|C_k|$ 的大小决定数据库访问次数,是频繁项集挖掘算法复杂性的关键因素.

推论 **4.5** prune过程不适合对低阶候选频繁集 $C_k$ 产生时进行剪支.

低阶 $C_k$ 访问复杂性从 $L_k^2$ 变到 $L_k^3$,低阶的 $L_k$ 大于 $|DB|$ 的可能性要大,从而增加访问复杂性.

## 5 优化频繁项集挖掘方法的途径

根据前面对事务数据库的数据特性及 Apriori算法分析可知:事务数据库有庞大的数据记录及众多的候选频繁项 $C_k$ 及 $L_k$.根据性质 4.2,要减少访问复杂性,既要减少 $C_k$ 及 $L_k$,也要设法压缩数据记录.由于域值 $minsupp$ 直接决定 $|C_k|$ 及 $|L_k|$ 的大小.过大引起 $|C_k|$ 及 $|L_k|$ 膨胀,而得不到有意义的频繁项集;过小则发现不了频繁项集.本文提出以下优化频繁项集挖掘方法的途径.

(1)首先合理的确定 $minsupp$ 的大小,有效的控制频繁项集数量,可采用预挖掘方法或神经网络学习等方法;

(2)从 $k = 1$开始对 $C_k$ 进行有效的剪支,严格控制 $|C_k|$ 及 $|L_k|$ 的大小,可先挖掘最大频繁项集,从而降低对其子集处理访问的复杂性;

(3)对事务数据库进行处理以减少 $|DB|$ 的数量,或改进事物数据库的结构,如通过索引表、映射表、链

(下转第 122页 Continue on page 122)

和 Canopy技术的检测复制记录方法[8],其中排序邻居方法的窗口尺寸本文选择了 2种情况做比较: k = 16或 k = 8.

从表 4可以看出,经过预处理的复制记录检测方法的准确率要高于未经过预处理的复制记录检测方法,但由于所用的实验数据不大,所以未能充分显示出经过预处理的优越性.

## 5 结束语

本文利用外部源文件清除脏数据和标准化简写,保证了数据库中名称的一致性.在用外部源文件清除脏数据的过程中,使用领域权重计算来记录的相似性.本文还提出一种利用类似 SQL的语言对数据库转换成所需模式的新思路,该思路新颖、实用.在将来的工作中,我们将建立相关的数据字典,并把算法运用到大型的数据库中,以进一步检验该方法的优越性.

参考文献:


[1] Ktmball R. Dealing with dirty data[J]. DBMS, 1996, 9(10): 55-57.

[2] Lee M L, Lu H, Ling T W, et al. Cleansing data for mining and warehousing [A]. In: Proceedings of the 10th International Conference on Database and Expert Systems Applications, 1999. 751-760.

[3] Hernandez M, Stolfo S. The merge/purge problem for large databases [A]. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 1995, 127-138.

[4] Levenshtein V. Binary codes capable of correcting deletions [J]. Insertions and Reversals. Soviet Physics-Doklady10. 1966, 10: 707-710.

[5] Monge A, Elkan C. The field-matching problem: algorithm and applications [A]. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996.

[6] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval[J]. Information Processing and Management, 1988, 24(5): 513-523.

[7] Lee M L, Ling T W, Low W L. IntelliClean a knowledge-based intelligent data cleaner[A]. In: Proceeding of SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. 2000, 290-294.

[8] McCallum A, Nigam K, Ungar L. Efficient clustering of high-dimensional data sets with application to reference matching[A]. In: Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining, 2000. 169-178.


(责任编辑: 黎贞崇)

(上接第 117页 Continue from page 117)

表等方法入手.

## 6 结束语

本文对事务数据库的数据特性做了深入的研究,指出了: $|I|$与 $supp(i)$是相对稳定的,事务项 $i$发生的次数及 $|DB|$在事务数据库中近似线性增长,$minsupp$直接决定频繁项集数量的多少,对挖掘结果有决定性作用.在事务数据库特性的基础上分析了 Apriori算法的复杂性,提出访问产生低阶频繁项集时访问 $C_k$及 $L_k$复杂性更高的新观点,进一步阐述有效控制 $|C_k|$及 $|L_k|$的必要性, Apriori算法对挖掘长频繁项集存在访问冗余性,当然 $DB$进行预处理也应为一种有效途径.

参考文献:


[1] Chengqi Zhang, Shichao Zhang. Association rule mining modelandalgorithms[J]. SpringerI, 2002, 2307: 33-39.

[2] Wu Xindong, Chengqi Zhang, Shichao Zhang. Mining both positive and negative association rules [C]. Proceedings of 19th International Conference on Machine Learning, Sydney: Australia, 2002. 658-665.

[3] GIWebb. Efficientsearchforassociationrulesproc[C]. ACM SIGKDD Int l Conf. Knowledge Discovery and Data Mining, 2000: 99~ 107.

[4] 李绪成,王保保.挖掘关联规则中 Apriori算法的一种改进[J].计算机工程, 2002, 28(7): 104-105.

[5] 陆丽娜,陈亚萍,魏恒义,等.挖掘关联规则中的 Apriori算法的研究[J].小型微型计算机系统, 2000, 21(9): 940-943.

[6] Zhang C, Zhang S. Collecting quality data for database mining[C]. Proceedings of the 14th Australian Joint Conference on Artificial Intelligence, 2001, 593-556.

[7] Zhang S, Zhang C. Mining small database by collecting knowledge[C]. Proceedings of DASFAA01, 2001, 174-175.

[8] Wu X, Zhang S. Synthesizing high-frequency rules from different data sources [J]. IEE Transaction on Knowledge and Data Engineering, 2003, (15) 2: 353-367.

[9] Lee G, Lee K L, Chen A L P. Efficient graph-based algorithms for discovering and maintaining association rules in large databases[J]. Knowledge and Information Systems, 2001, (3): 338-335.

[10] 何炎祥,石 莉,张 戈,等.关联规则的几种开采算法及其比较分析[J].小型微型计算机系统, 2001, 22(9): 1065-1068.


(责任编辑: 黎贞崇)