

基于位集合的 Apriori 算法的改进

王 威, 陈 梅

(贵州大学 计算机科学与信息学院, 贵州 贵阳 550025)

摘 要:针对经典 Apriori 算法运行效率瓶颈问题,结合位集合占用内存空间少、逻辑运算快的特点,提出一种基于位集合的改进算法 ABS。该算法通过一次数据库扫描,构建事务集位集合;采用位集合逻辑“与”运算和位统计操作确定频繁项集;改进连接和剪枝策略,采用位集合的逻辑“或”运算,统计运算结果重复出现次数,生成候选项集。挖掘实例数据库 Northwind 的频繁项集,对比 Apriori 算法,改进算法运行时间明显减少。该算法避免了数据库的重复扫描和繁琐的连接减枝操作,进一步提高了 Apriori 算法的运行效率。

关键词:数据挖掘;关联规则;频繁项集;位集合;Apriori 算法

中图分类号:TP311.13

文献标识码:A

文章编号:1673-629X(2011)12-0070-03

Improvement of Apriori Algorithm Based on BitSet

WANG Wei, CHEN Mei

(College of Computer Science and Information, Guizhou University, Guiyang 550025, China)

Abstract: It puts forward an improved Apriori algorithm based on BitSet which avoids the bottleneck problem in the efficiency of the classic Apriori algorithm and combines with the BitSet characteristics of taking up less memory space and faster logical operation. This algorithm constitutes transactional BitSet by scanning database one time, using logical "and" operation of BitSet and bit-count operation to determine the frequent itemsets. Improving the strategies of connecting and pruning, it uses logical "or" operation of BitSet and counts the repeated number of operation result for generating candidate itemsets. By mining frequent itemsets of Northwind sample database, compared with Apriori algorithm, the running time of improved algorithm decreases sharply. The algorithm avoids repeated database scanning and complicated operations of connecting and pruning, and furthermore, increases the efficiency of Apriori algorithm.

Key words: data mining; association rule; frequent itemsets; BitSet; Apriori algorithm

0 引 言

随着收集、存储的数据量逐渐增大,从大型事务数据集发现项之间有趣的相关与关联性越来越受到人们的关注,频繁模式挖掘正是用于发现数据之间有趣、相关和关联的联系。

发现频繁项集的经典算法是 R. Agrawal 等在 1994 年提出的 Apriori 算法,它是为布尔关联规则挖掘频繁项集。算法采用逐层迭代的方法,产生候选项集,使用候选项集产生发现频繁项集,找出所有频繁项集^[1]。但是,当面对大型事务数据时,重复的扫描数据库以及复杂的候选项集生成策略,降低了 Apriori 算法的效率,为解决这一问题,文中提出了一种基于位集合的改进 Apriori 算法。

1 Apriori 算法分析

Apriori 算法总体分为两个步骤^[2]:

(1) 候选 $k-1$ ($k > 1$) 项集确定频繁 $k-1$ 项集: 扫描事务数据库中的事务,计算候选 $k-1$ 项集的集合 C_{k-1} 中每个候选项集的支持度,选择满足最小支持度的 C_{k-1} 中的候选 $k-1$ 项集组成频繁 $k-1$ 项集的集合 L_{k-1} 。

(2) 频繁 $k-1$ 项集生成候选 k 项集。

第一步: 连接步。频繁 $k-1$ 项集的集合 L_{k-1} 与自身连接产生候选项集的集合 C_k 。连接条件是: 设 l_1 和 l_2 是 L_{k-1} 中的项集, 假定事务或项集中的项是按照字典次序排序的, 当 l_1 和 l_2 前 $(k-2)$ 项相同时, 两项集可连接, 结果项集为 $l_1[1], l_1[2], \dots, l_1[k-1], l_2[k-1]$ 。

第二步: 剪枝步。根据 Apriori 性质: 任何非频繁项集的 $k-1$ 项集都不会是频繁 k 项集的子集。因此, 如果候选 k 项集的 $k-1$ 项子集不是 L_{k-1} 中的项集时, 将该候选 k 项集从 C_k 中删除。

经典的 Apriori 算法在运行效率上存在不足: 在大

收稿日期: 2011-05-06; 修回日期: 2011-08-15

基金项目: 贵州省科技计划工业攻关基金项目(黔科合 GY 字 [2008]3035)

作者简介: 王 威(1987-), 女, 辽宁彰武人, 硕士研究生, 研究方向为数据库技术; 陈 梅, 教授, 硕士生导师, 研究方向为数据库技术。

型数据库关联规则挖掘,或处理规模庞大的候选项集时,重复扫描数据库计算候选项集支持度,生成候选项集时依次进行连接判断以及复杂的剪枝策略,增加了系统的输入输出操作,严重影响了算法的执行效率。

2 基于位集合的 Apriori 算法的改进

因为位集合具有占用内存空间少、逻辑运算速度快等特点,多用于数据编码、数据压缩,常见的基于位集合的关联规则算法 MBSA (Map-based BitSet Association Rule) 是通过数据库的一次扫描来构建位集合,该算法并不产生候选项集,虽然避免了繁琐的连接和剪枝步,但直接拼接产生大量无用的候选项集,增加了无意义的逻辑“与”操作^[3];基于 Apriori 和位集合的关联规则算法是根据 Apriori 性质改进 MBSA 算法,大大减少了无用的候选项集,避免了许多无意义的“与”操作,但是它仅仅是在 MBSA 算法基础上引入 Apriori 性质,并没有将位集合与 Apriori 性质相结合^[4]。

文中提出基于位集合的改进 Apriori 算法 ABS (AprioriBitSet),它结合位集合和经典 Apriori 算法的特点,通过扫描一次数据库,构建事务集和频繁项集位集合,改进复杂的候选项集生成策略,设计实现基于位集合的连接剪枝策略,本算法继承了 Apriori 算法简单直观的优点,并且解决了 Apriori 算法运行效率的瓶颈问题。

2.1 ABS 基本内容

采用位集合数据结构实现 Apriori 算法。构建每个项的事务集位集合和频繁项集的位集合,对事务集位集合进行逻辑“与”、位统计操作确定频繁项集,对频繁项集的位集合进行逻辑“或”,统计结果重复出现次数产生候选项集。

算法依据频繁项集的性质^[5-8]:
性质 1:任何非频繁的 $(k-1)$ 项集都不是频繁 k 项集的子集^[1]。

性质 2:频繁 k 项集的 k 个 $(k-1)$ 子项集都是频繁的。

证明:性质 1 的逆否命题成立,即频繁 k 项集的所有 $(k-1)$ 子项集都是频繁的,频繁 k 项集共有 $C_k^{k-1} = k$ 个 $(k-1)$ 子项集,因此它们都是频繁的。

性质 3:候选 k 项集的集合 C_k 中每个项集,在 L_{k-1} 与自身连接时,重复出现 $k \cdot (k-1)/2$ 次。

证明:由性质 1 可知,候选 k 项集的集合 C_k 中每个项集的 $k-1$ 子项集都包含在频繁 $k-1$ 项集的集合 L_{k-1} 中,根据集合的唯一性,在 L_{k-1} 与自身连接时, $C_k^{k-1} = k$ 个频繁 $k-1$ 项集两两进行连接时产生相同的 k 项集,且重复出现 $C_k^2 = k \cdot (k-1)/2$ 次。

性质 4:如果频繁 k 项集的集合 L_k 的项集个数小

于 $k+1$,频繁 $k+1$ 项集不存在。
证明:由反证法可得,假设频繁 $k+1$ 项集存在,由性质 2 可知, $k+1$ 个 k 子项集都是频繁的,与已知 L_k 的项集个数小于 $k+1$ 矛盾。

第一步:扫描数据库,生成 $|C_1|$ 个项的事务集位集合,对每个位集合进行位统计(位统计是统计每一个位集合中位“1”出现的次数)得到支持度计数,选择满足最小支持度计数的候选 1 项集,生成频繁 1 项集的集合 L_1 。

第二步:根据 L_1 建立项序列 S 。
第三步: L_1 中的项集两两连接,生成 $C_{|L_1|}^2$ 个候选 2 项集的集合 C_2 。

第四步: $C_k(k \geq 2)$ 中每个项集所对应项的事务集位集合进行逻辑“与”和位统计操作,将满足最小支持度的项集加入集合 L_k 。根据项序列 S 对 L_k 中的项集进行二进制位编码,构建 $|L_k|$ 个频繁 k 项集的位集合。频繁 k 项集的位集合两两进行逻辑“或”操作,所得 $C_{|L_k|}^2$ 个结果进行位统计,将统计数为 $k+1$,并且重复出现 C_{k+1}^2 次的结果按照项序列 S 解码后加入 C_{k+1} 中。循环执行,当 $C_{k+1} = \varnothing$ 与 $|C_{k+1}| < k+1$ 时算法终止^[9-12]。

2.2 实例分析

例如:计算交易数据库 D (见表 1)的频繁项集,最小支持度 $\min_sup=2$ 。

表 1 交易数据库 D	
TID	item
001	A、B、E
002	B、D
003	B、C
004	A、B、D
005	A、C
006	B、C
007	A、C
008	A、B、C、E
009	A、B、C

(1)扫描交易数据库 D ,生成 5 个项的事务集位集合,进行位统计 count 得到候选 1 项集的支持度计数(见表 2),由于 $\min_sup=2$,得频繁 1 项集的集合 $L_1 = \{A、B、C、D、E\}$ 。

表 2 事务集位集合		
item	TBitSet	count
A	100110111	6
B	111101011	7
C	001011111	6
D	010100000	2
E	100000010	2

(2)根据 L_1 建立项序列 $S = ABCDE$ 。
(3) L_1 中的项集两两连接,生成 10 个候选 2 项集

的集合 $C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$ 。

(4) C_2 中每个项集所对应项的事务集位集合进行逻辑“与”和位统计操作,例如:项集 AB 中 $A = \{100110111\}$ 和 $B = \{111101011\}$ 进行逻辑“与”操作结果为 $AB = \{100100011\}$, 位集合 $\text{count} = 4 > \text{min_sup}$, 加入 L_2 ; 项集 AC 中 $A = \{100110111\}$ 和 $D = \{010100000\}$ 进行逻辑“与”操作结果为 $AD = \{000100000\}$, 位集合 $\text{count} = 1 < \text{min_sup}$, 删除, 依次计算得到频繁 2 项集的集合 $L_2 = \{AB, AC, AE, BC, BD, BE\}$, $|L_2| = 6 > 2 + 1$ 计算 L_3 。

(5) 根据 S 对 L_2 进行编码, 即 $L_2 = \{11000, 10100, 10001, 01100, 01010, 01001\}$, 构建数据结构 $C_3 = \{\text{BitSet}, \text{repeat}\}$ 。 $L_2[0] = \{11000\}$ 和 $L_2[1] = \{10100\}$ 进行逻辑“或”操作结果为 $\{11100\}$, 位统计 $\text{count} = 3 = 2 + 1$, C_3 中不存在, 加入 $C_3, \text{repeat} = 1$; $L_2[0] = \{11000\}$ 和 $L_2[3] = \{01100\}$ 进行逻辑“或”操作结果为 $\{11100\}$, 位统计 $\text{count} = 3 = 2 + 1$, C_3 中存在, $\text{repeat} + 1$; $L_2[1] = \{10100\}$ 和 $L_2[4] = \{01010\}$ 进行逻辑“或”操作结果为 $\{11110\}$, 位统计 $\text{count} = 4 > 2 + 1$, 删除等等, 计算得 $C_3 = \{\{11100, 3\}, \{11001, 3\}, \{10101, 1\}, \{01110, 1\}, \{01101, 1\}, \{01011, 1\}\}$, 选择 C_3 中 $\text{repeat} = 3$ 的位集合, 按照 S 解码得 $C_3 = \{ABC, ABE\}$ 。

(6) C_3 中每个项集所对应项的事务集位集合进行逻辑“与”和位统计操作, 例如: 项集 ABC 中 $A = \{100110111\}$, $B = \{111101011\}$ 和 $C = \{001011111\}$ 进行逻辑“与”操作结果为 $ABC = \{000000011\}$, 位集合 $\text{count} = 2 = \text{min_sup}$, 加入 L_3 ; 项集 ABE 中 $A = \{100110111\}$, $B = \{111101011\}$ 和 $E = \{100000010\}$ 进行逻辑“与”操作结果为 $ABE = \{100000010\}$, 位集合 $\text{count} = 2 = \text{min_sup}$, 加入 L_3 , 得到频繁 3 项集的集合 $L_3 = \{ABC, ABE\}$, $|L_3| = 2 < 3 + 1$ 算法结束。

2.3 算法描述

算法描述如下: An Improved Apriori Algorithm based BitSet

算法: 使用 BitSet 基于候选项集产生频繁项集

输入: 交易数据库 D ; 最小支持度计数阈值 min_sup

输出: 所有频繁项集 L

方法:

(1) $C_1 = \text{genTBitSet}(D)$;

//扫描 D 生成事务集位集合

(2) $k = 1$;

(3) while($|C_k| \neq \varnothing$)

(4) $L_k = \text{GFititemsets}(C_k, C_1, \text{min_sup})$; //与操作生成 L_k

(5) if ($|L_k| > k + 1$)

(6) if ($k = 1$)

(7) $S = \text{gen_order}(L_k)$; //生成项集序列 S

(8) $C_{k+1} = \text{gen_C}_2(L_k)$; //连接: 生成 C_2

(9) }

(10) else $C_{k+1} = \text{gen_CK}(L_k, S)$; //连接: 位集合操作, 产生 $C_k (k > 2)$

(11) }

(12) $L = L \cup L_k$;

(13) $k++$;

(14) }

(15) return L 。

3 算法比较分析

在相同条件下, 采用实例数据库 Northwind 作为数据源, 使用 C# 实现, 对 Apriori 算法与基于位集合的 Apriori 算法进行测试比较。算法运行时间比较如图 1 所示。

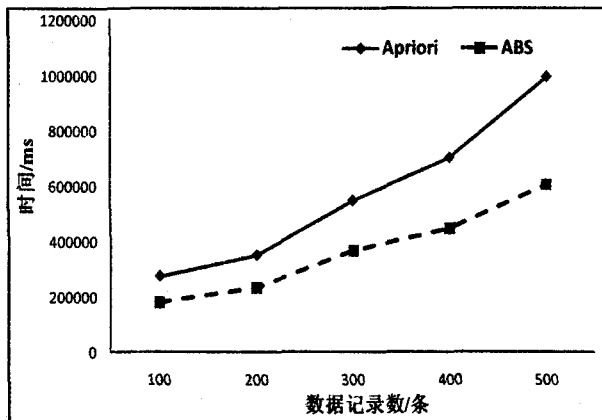


图 1 运行时间比较

基于位集合的改进 Apriori 算法 ABS 与 Apriori 算法相比, 效率的提高主要体现在:

(1) 只扫描数据库一次, 生成事务集位集合。每次生成频繁项集时, 只需对位集合进行逻辑“与”和位统计操作, 减少了扫描数据库的时间;

(2) 直接连接生成候选 2 项集;

(3) 按照项序列 S 生成项集的位集合, 对位集合进行逻辑“或”操作, 统计结果出现次数, 生成候选 $k (k > 2)$ 项集, 避免了复杂的连接与剪枝。

4 结束语

文中的创新点在于通过对位集合的操作生成频繁项集和候选项集, 提高了 Apriori 算法的运行时间。到目前为止只研究了基于位集合的 Apriori 算法的单机实现, 下一步的目标是结合分布式计算技术研究实现

(下转第 76 页)

从差别矩阵可以看出,该信息系统的核为 CORE (C) = {P0, P9}。计算可得属性约简为 {P0, P3, P4, P9}。这样知道,在研究的 10 个页面中, P0 和 P9 非常重要, P3 和 P4 也比较重要,这四个页面经常被访问,可以通过修改网站的结构,把他们放在比较容易访问的位置,便于学习者浏览。

3.3 学习者访问路径

通过对每个会话的分析,得到远程学习者在本站点的访问序列(见表 2),也就是他在这个网站的访问路径。通过对路径的分析可以知道学习者在网站的访问习惯、访问偏好等,可以帮助我们调整网站的布局,提高学习者的学习效率。

表 2 用户访问序列

SessionID	UserID	Web 访问序列
S0	U1	P1, P2, P5, P6, P7, P8, P9, P0, P012, P0, P013, P014, P015, P016, P017, P018, P019, P020, P021, P022, P023, P024, P025, P9, P0, P012, P014, P0
		P057, P058, P014, P073, P0, P010, P014, P076, P077, P078, P079, P080, P081, P082, P083
S457	U112	P9, P011, P3, P4, P9, P011, P3, P4
.....

4 结束语

通过 Web 日志挖掘的方法可以跟踪学习者的网上学习行为,了解学生的学习偏好,便于我们更好地提高教学支持。另一方面,由于网络环境的复杂性,以及学习者上网习惯的关系,想从海量的 Web 日志中发现需要的知识有相当的难度,这需要不断探索新的方法。

(上接第 72 页)

基于分布式和位集合的频繁项集生成算法,解决在数据源分布、数据量巨大的情况下,如何高效地进行频繁项集挖掘的问题。

参考文献:

- [1] Han J, Kamber M. Data Mining: Concepts and Techniques [M]. Beijing: Higher Education Press, 2001.
- [2] 陈伟. Apriori 算法的优化方法[J]. 计算机技术与发展, 2009, 19(6): 80-83.
- [3] 黄端琼, 陈崇成, 黄洪宇, 等. 基于映射位集合的遥感图像关联规则挖掘[J]. 计算机应用, 2005, 25(7): 1592-1594.
- [4] 刘永彬, 秦亮曦, 王永卿, 等. 基于 Apriori 和位集合的关联规则应用[J]. 微计算机信息, 2007(33): 141-143.
- [5] Agrawal R, Srikant R. Fast algorithm for mining association rules[C]//Proc of the 21st VLDB Conference. Zurich, Switzerland: [s. n.], 1995: 487-499.

参考文献:

- [1] Han Jiawei. 数据挖掘: 概念与技术[M]. 北京: 机械工业出版社, 2004.
- [2] Liu Bing. Web 数据挖掘[M]. 北京: 清华大学出版社, 2009.
- [3] 周勇, 刘锋. 基于粗糙集的 Web 结构挖掘[J]. 计算机技术与发展, 2008, 18(3): 151-153.
- [4] 朱鹤祥. Web 日志挖掘中数据预处理算法的研究[D]. 大连: 大连交通大学, 2009.
- [5] 张春生, 庄丽艳. 基于兴趣的 Web 挖掘中用户身份的识别新方法[J]. 计算机技术与发展, 2009, 19(5): 62-64.
- [6] Kosala R, Blockeel H. Web Mining Research: A Survey (2000) [J]. SIGKDD Explorations - Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, 2000, 2(1): 1-15.
- [7] 许海洋. 模糊聚类分析在数据挖掘中的应用研究[J]. 计算机工程与应用, 2005, 17: 177-179.
- [8] 王伟, 高亮, 吴涛. 一种基于模糊聚类的离散化算法[J]. 计算机技术与发展, 2008, 18(3): 53-55.
- [9] 高新波. 模糊聚类分析及其应用[M]. 西安: 西安电子科技大学出版社, 2004.
- [10] 苗夺谦. 粗糙集理论、算法与应用[M]. 北京: 清华大学出版社, 2008.
- [11] Pawlak Z. Rough Set [J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341-356.
- [12] Ziarko W. Variable Predsion Rough Set Model [J]. Journal of Computer and System Sciences, 1993, 46(1): 39-59.
- [13] Bean C. Autonomous Clustering Using Rough Set Theory [J]. International Journal of Automation and Computing, 2008, 5(1): 90-102.
- [6] Ashok S, Edward O, Shamkant N. An efficient algorithm for mining association rules in large databases [C]//Proc of the 21st VLDB Conference. Zurich, Switzerland: [s. n.], 1995: 432-443.
- [7] 徐章艳, 刘美玲, 张师超, 等. Apriori 算法的三种优化方法[J]. 计算机工程与应用, 2004, 36(2): 190-202.
- [8] 钱光超, 贾瑞玉, 张然, 等. Apriori 算法的一种优化方法[J]. 计算机工程, 2008, 34(23): 196-198.
- [9] 董杰. 基于位表的关联规则挖掘及关联分类研究[D]. 大连: 大连理工大学, 2009.
- [10] 钱少华, 蔡勇, 钱雪忠. 基于数组的 Apriori 算法的改进[J]. 计算机应用与软件, 2006, 23(3): 111-113.
- [11] 郭云峰, 张集祥. 对关联规则挖掘中 Apriori 算法的一种改进[J]. 杭州电子科技大学学报, 2009, 29(2): 60-63.
- [12] 袁万莲, 郑诚, 翟明清. 一种改进的 Apriori 算法[J]. 计算机技术与发展, 2008, 18(5): 51-53.