一种新的改进 Apriori 算法*

杨金凤,刘 锋
(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:通过对 Apriori 算法的核心思想进行研究分析,结合 Apriori 性质,对 Apriori 中连接的步骤进行了改进。通过该方法,可以有效地减少连接步产生的大量无用项集并减少判断项集子集是否是频繁项集的次数。

关键词:Apriori;候选项集;频繁项集

中图分类号:TP311.13 **文献标识码**:A

New improved Apriori algorithm

YANG Jin Feng, LIU Feng
(School of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: The core thought of Apriori algorithm is analyzed, and the nature of Apriori is take into account, in this foudation, Apriori algorithm is improved. Applying this improved algotithm, less items than classic Apriori algorithm can be get, and the times of judgment are lower.

Key words: Apriori; candidate items; frequent items

数据挖掘 DM(Data Mining)出现于 20 世纪 80 年代后期,是在数据库技术的基础上,结合人工智能、机器学习、统计学、神经网络等多种学科技术产生的具有很强生命力的新研究领域。其中关联规则挖掘研究[1-2]是一项重要的内容,目的是发现大规模数据集中项集之间有趣的关联关系或模式。

频繁项集的挖掘是关联规则挖掘的核心,如何高效地从海量数据库中找出频繁出现的项集是世界范围内的热门研究课题。

1 相关概念[1]

设 $I=\{I_1,I_2,\cdots,I_m\}$ 是项的集合,称为项集,包含 k 个项的项集称为 k 项集。D 是数据库事务的集合,数据库中的每个事务 T 是项的集合,$T\subseteq I$,TID 是事务 T 的标识符。设 A 是一个项集,事务 T 包含 A,当且仅当 $A\subseteq T$,一个包含 k 个项的事务 T 可以产生 2^k 个非空的子项集。

规则 $A\Rightarrow B$ 的支持度 s 是 D 中同时包含 A 和 B 的事务占总事务的百分比。规则 $A\Rightarrow B$ 的支持度 c 是 D 中同时包含 A 和 B 的事务占包含 A 的事务的百分比。项集的出现频率是包含项集的事务数,称为项集的支持度计数。项集的支持度计数除以总的事务数就是相对支持度计数,如果项集 I 的相对支持度计数不小于预定义的最小支持度阈值,则称此项集是频繁项集,否则是不频繁项集。

2 Apriori 算法和优化

2.1 经典的 Apriori 算法[2-6]

Apriori 是一种逐层搜索的迭代方法,即用 k 项频繁项集探索$(k+1)$项频繁项集。首先,扫描数据库找出频繁 1 项集的集合,该集合为 L_1。用 L_1 找频繁 2 项集的集合 L_2,用 L_2 找 L_3,如此下去直到不能再找到频繁项集。找每个 L_k 需要 1 次数据库全扫描。

为了提高频繁项集逐层产生的效率,一种称作 Apriori 的重要性质用于压缩搜索空间。Apriori 性质为频繁项集的所有非空子集也必须是频繁的。

Apriori 算法是由连接步和剪枝步组成。(1)连接步:为找 L_k,通过将 L_{k-1} 与自身连接产生候选 k 项集的集合。该候选项集合为 C_k。(2)剪枝步:C_k 是 L_k 的超集,即 C_k 的成员可能是频繁的;也可能是不频繁的。扫描数据

* 基金项目:安徽省自然基金项目(070412051);安徽省高等学校省级重点教学研究项目(2007jyxm020)

库确定 C_k 中每个候选项集的计数，从而确定 L_k（即根据定义，计数值不小于最小支持度计数的所有候选项集是频繁的，从而属于 L_k）。然而 C_k 可能很大，因此所涉及的计算量就很大。为压缩 C_k，可以使用 Apriori 性质。任何非频繁的 $(k-1)$ 项集都不是频繁 k 项集的子集。因此，如果候选 k 项集的 $(k-1)$ 项子集不在 L_{k-1} 中，则该候选项集也不可能是频繁的，从而可以从 C_k 中删除。

经典的 Apriori 算法在产生候选项集上，由 L_{k-1} 自连接产生 C_k，然后再利用 Apriori 性质对 C_k 进行删减。设 l_1 和 l_2 是 L_{k-1} 中的项集。记号 $l_i[j]$ 表示 l_i 中的第 j 项（例如，$l_1[k-2]$ 表示 l_1 的倒数第 2 项）。Apriori 假定事务或项集中的项按字典次序排序。执行 L_{k-1} 的自连接，如果它们的前 $(k-2)$ 个项相同的话，则可以连接。例如，$(l_1[1]=l_2[1]) \wedge (l_1[2]=l_2[2]) \wedge \cdots \wedge (l_1[k-2]=l_2[k-2]) \wedge (l_1[k-1]<l_2[k-1])$，那么 l_1 和 l_2 是可以连接的，否则是不可以连接的，连接结果项集是 $l_1[1], l_1[2], \cdots, l_1[k-1], l_2[k-2]$。然后再利用 Apriori 性质进行判断项集 $l_1[1], l_1[2], \cdots, l_1[k-1], l_2[k-2]$ 是否可能是频繁的，这样需要先产生项集 $l_1[1], l_1[2], \cdots, l_1[k-1], l_2[k-2]$ 的 $(k-1)$ 个 $(k-1)$ 项子集，然后依次比较 $(k-1)$ 个子集是否在 L_{k-1} 中，如果出现不在，则删减项集 $l_1[1], l_1[2], \cdots, l_1[k-1], l_2[k-2]$；如果都在，再对数据库扫描，进行计数。

2.2 对 Apriori 算法的改进

通过上面的分析发现，为了生成 C_k，在连接步骤需要大量的比较，而且由连接产生的项集即使后来由 Apriori 性质确定了它不是候选项集，但在确定之前仍然需要对它生成子项集，并对子项集进行确定是否都在 L_{k-1} 中。这些步骤浪费了大量的时间，如果可以保证由连接步生成的项集都是候选项集的话，那么可以省掉不必要的连接比较和剪枝步骤。下面介绍改进后的算法。

首先对 L_{k-1} 中的每项进行扫描，记下项集 $\{l_1[1]\}, \{l_1[1], l_1[2]\}, \cdots, \{l_1[1], l_1[2], \cdots, l_1[k-2]\}, \{l_2[1]\}, \{l_2[1], l_2[2]\}, \cdots, \{l_2[1], l_2[2], \cdots, l_2[k-2]\}, \{l_3[1]\}, \cdots$。设 1 个 k 项集 $l_i=\{l_i[1], l_i[2], \cdots, l_i[k-1], l_i[k]\}$，由 Apriori 性质知道，如果 l_i 属于 C_k，那么以下的 $(k-1)$ 个 $(k-1)$ 项集就必须都出现在 L_{k-1} 中，所以 $\{l_i[1]\}$ 至少要出现 $(k-2)$ 次，$\{l_i[1], l_i[2]\}$ 至少要出现 $(k-3)$ 次，依次类推 $\{l_i[1], l_i[2], \cdots, l_i[k-1]\}$ 至少要出现 1 次。设扫描得到的 $\{l_i[1]\}$ 出现次数为 a，如果 $a<(k-2)$，则可以将 L_{k-1} 中所有以 $l_i[1]$ 开头的 $(k-1)$ 项集全部删除，如果 $a \geqslant (k-2)$，那么比较扫描得到的 $\{l_i[1], l_i[2]\}$ 出现次数 b 与 $(k-3)$ 的大小，若 $b<(k-3)$，则删除 L_{k-1} 中所有以 $\{l_i[1], l_i[2]\}$ 开头的项集，如果 $b \geqslant (k-3)$，则继续比较下一项。通过简单的数字比较，可以大量地从 L_{k-1} 中删除项集，这样可以大大地减少不必要的连接。连接生成的 k 项集，也只需要比较 1 次就可以确定是否属于 C_k，其算法如下：

输入：D（事物数据库）；

min_sup：最小支持度计数阈值。

输出：$L(D$ 中的频繁项集）。

(1) 扫描数据库 D，生成频繁 1 项集 L_1；

(2) 由频繁 1 项集生成频繁 2 项集 L_2；

(3) for$(i=1; l_i \in L_k; i++)$。

(4) 累加 $\{l_i[1]\}$、$\{l_i[1], l_i[2]\}$、…、$\{l_i[1], l_i[2], \cdots, l_i[k-1]\}$ 的出现次数；

(5) 将只含 $l_i[1]$ 项的项集出现次数 a 与 $(k-1)$ 比较大小，如果 a 小于 $(k-1)$，则删除 L_k 中的所有以 $l_i[1]$ 项开头的项集，从 $l_{i+1}[1]$ 项的项集出现次数开始比较；如果 a 大于 $(k-1)$，再比较以 $\{l_i[1], l_i[2]\}$ 开头的项集出现次数 b 与 $(k-2)$ 的大小。如果 b 小于 $(k-2)$，则删除 L_k 中的所有以 $\{l_i[1], l_i[2]\}$ 开头的项集，并将只含 $l_i[1]$ 项的项集出现次数赋值为 $(a-b)$，再对 $(a-b)$ 与 $(k-1)$ 进行比较；如果 b 大于 $(k-2)$，再对以 $\{l_i[1], l_i[2], l_i[3]\}$ 开头的项集出现次数 c 与 $(k-3)$ 的大小进行比较。依次类推，删除后 L_k 项集为 L_k'。

(6) 用 L_k' 中的项集自连接生成 C'_{k+1}；

(7) for$(i=1; l_i \in C'_{k+1}; i++)$

if$(\{l_i[2], l_i[3], \cdots, l_i[k]\} \in L_k)$

l_i 是候选项集，放入 C_{k+1} 中；

(8) 扫描数据库，对 C_{k+1} 中项集进行计数，如果大于 min_sup，就是频繁项集，放入 L_{k+1} 中。

3 实例说明

通过由频繁 3 项集生成 4 项候选集的例子，说明是如何通过改进的算法在连接前对频繁 3 项集进行删减的。设 $L_3=\{\{I_1, I_2, I_3\}, \{I_1, I_2, I_6\}, \{I_2, I_3, I_4\}, \{I_2, I_3, I_5\}, \{I_2, I_4, I_5\}, \{I_3, I_4, I_5\}, \{I_3, I_5, I_6\}, \{I_4, I_5, I_6\}\}$。扫描 L_3，得到相关的计数。表 1 所示为删减 L_3 中项集的过程。

经过删减得 $L_k'=\{\{I_2, I_3, I_4\}, \{I_2, I_3, I_5\}\}$，$L_k'$ 自连接只生成 1 个 4 项集 $\{I_2, I_3, I_4, I_5\}$。$\{I_3, I_4, I_5\} \in L_3$，所以，得到候选项集 $C_4=\{I_2, I_3, I_4, I_5\}$。通过验证，结果是正确的。

如果采用经典的 Apriori 算法，先连接生成 2 个 4 项集 $\{I_1, I_2, I_3, I_6\}$，$\{I_2, I_3, I_4, I_5\}$，再进行剪枝，最坏情况下，需要扫描 8 个子项集是否在 L_3 中，才能确定，$\{I_1, I_2, I_3, I_6\}$，$\{I_2, I_3, I_4, I_5\}$ 是否为候选项集，最好的情况下也需要扫描 2 个子集。而采用新的算法，只连接生成 1 个 4 项集，再进行剪枝步，只需要扫描 1 个子项集 $\{I_3, I_4, I_5\}$ 是否在 L_3 中。

本文运用新的算法，从另一个先删减再连接的新视角来生成频繁项集，可以减少大量的无用连接，进而也减少了剪枝步需要判断是否为候选项集的数量，在时间上提高了效率。但是对 Apriori 算法改进的并不彻底，依然需要大量的数据库扫描，在未来的研究工作中着力解决多次扫描数据库的问题。

参考文献

[1] HAN Jia Wei，KAMBER M.数据挖掘概念与技术[M].范

表 1 删减 L_3 中的项集

扫描项	出现次数	比较数	操作
I_1	2	3	2<3。从 L_3 中删除所有以 I_1 开头的项集。不再比较以 I_1 开头的项集出现次数，从 I_2 开头的项集出现次数开始比较。
I_1, I_2	2	2	无操作。
I_2	3	3	3≥3。比较以 I_2 开头的项集出现的项集次数。
I_2, I_3	2	2	2≥2。保留 L_3 中以 I_2, I_3 开头的项集，继续下一步比较。
I_2, I_4	1	2	1<2。删除 L_3 中以 I_2, I_4 开头的项集。进行 I_3 的比较。
I_3	2	3	2<3。从 L_3 中删除所有以 I_3 开头的项集。不再比较以 I_1 开头的项集出现次数，从 I_4 开头的项集出现次数开始比较。
I_3, I_4	1	2	无操作。
I_3, I_5	1	2	无操作。
I_4	1	3	1<3。从 L_3 中删除所有以 I_4 开头的项集。结束。

明，孟小峰，译.北京：机械工业出版社，2007.

[2] 杨明，孙志挥，宋余庆.快速更新全局频繁项目集[J].软件学报，2004，15(8)：1189-1196.

[3] 郭健美，宋顺林，李世松.基于 Apriori 算法的改进算法[J].计算机工程与设计，2008，29(11)：2814-2815.

[4] 冯兴杰，周谆.Apriori 算法的改进[J].计算机工程，2005，31(21)：172-173.

[5] 朱其祥，徐勇，张林.基于改进 Apriori 算法的关联规则挖掘研究[J].计算机技术与发展，2006，16(7)：102-104.

[6] MING Cheng Tseng，WEN Yang Lin，RONG Jeng. Dynamic mining of multi-supported association rules with classification ontology[J]. 网际网路技术学刊，2006，7(14)：399-406.

(收稿日期：2009-09-01)

作者简介：

杨金凤，女，1984 年生，硕士，主要研究方向：数据挖掘技术。