一种基于单事务项集组合的频繁项集挖掘算法*)

曾 波
(重庆工商大学商务策划学院 重庆 400067)

摘 要 Apriori是挖掘频繁项集的基本算法,目前该算法及其优化变种都没有解决候选项及重复扫描事务数据库的问题。文章通过对Apriori及其优化算法的深入探究,提出了一种基于单事务组合项集的挖掘算法,该算法在一个事务内部对"数据项"进行组合,在事务数据库中对所有相同"项集"进行计数。不经过迭代过程,不产生候选项集,所有频繁项集的挖掘过程只需对事务数据库一次扫描,提高了频繁项集挖掘效率。

关键词 频繁项集,Apriori,单事务项集组合,候选项

Algorithm of Frequent Itemsets Mining Based on Single Transaction Itemsets Combination

ZENG Bo
(Strategical Planning College, Chongqing Technology and Business University, Chongqing 400067)

Abstract Apriori is a basic algorithm for frequent itemsets mining. At present, neither Neither Apriori nor its variations resolve some problems which is candidate item and scans transaction database repeatedly. This paper makes a profound research on Apriori and proposes a novel algorithm based on single transaction combination itemsets for mining. The algorithm combines data item to form an itemsets in one transaction database and counts the same itemsets in all transaction databases. Moreover, there is no iteration and candidate itemsets produced by the algorithm, and the mining process scans the traction database only one time, therefore, this algorithm is more effective.

Keywords Frequent itemsets, Apriori, Single transaction itemsets combination, Candidate item

1 引言

关联规则挖掘是数据挖掘领域的一个重要子课题,是目前数据挖掘领域应用最为广泛和成熟的分支之一。寻找给定数据集中数据项之间的关联或相互关系,并通过支持度和置信度两个参数来发现这些数据项之间关联的强弱。这些关联关系可以为商业决策者提供有价值的信息,从而实现商务决策的制定,如分类设计、交叉购物等。

挖掘关联规则的过程需要分解为两个步骤,首先是找出所有的频繁项集,其次是根据频繁项集产生强关联规则。这两步中,第二步非常容易实现,挖掘关联规则的总体性能由第一步决定。所以频繁项集的挖掘是关联规则挖掘的核心。目前挖掘频繁项集的算法主要有Apriori算法及其变种,如:基于散列技术[1]、事务压缩[2,3]、划分[4]、选样[5]、动态项集计数的Apriori算法[6]以及不产生候选挖掘频繁项集的FP-Growth算法等。

2 问题的描述及Apriori算法

2.1 问题的描述

设 $I=\{i_1,i_2,\cdots,i_m\}$ 是项的集合,$T=\{t_1,t_2,\cdots,t_n\}$ 是数据库事务的集合。T 中的每个事务 $t_x(x=1,2,3,\cdots,n)$ 中的项是 I 的一个子集,有 $t_x\subseteq I$。设 A 是一个项集,$A\subseteq I$,事务 T 包含 A,有 $A\subseteq T$。关联规则是形如 $A\Rightarrow B$ 的规则,其中 $A\subset I$,$B\subset I$ 且 $A\cap B=\Phi$。规则 $A\Rightarrow B$ 在事务集合 T 中成立,具有支持度 s,其中 s 是 T 中事务包含 $A\cup B$ 的百分比,即 $s=P(A\cup B)$;规则 $A\Rightarrow B$ 在 T 中成立具有置信度 c,如果 T 中包含 A 的事务同时也包含 B 的百分比是 c,即 $c=P(B|A)$。

支持度和置信度分别体现了蕴涵规则的事务发生的可能性和可靠性,如果支持度和置信度分别满足最小支持度阈值和最小置信度阈值,则这种规则称作强规则。引言中所述,挖掘关联规则的核心是找出所有的频繁项集。所谓频繁项集,就是满足最小支持度的项集。

2.2 Apriori算法

Apriori[7]是一种寻找频繁项集的基本算法。算法使用频繁项集性质的先验知识,使用一种称为逐层搜索迭代的技术,给定 k-项集,我们只需要检查 $(k-1)$ 项集是否频繁即可。整个算法分为两个过程——连接和剪枝。首先找出频繁1-项集的集合 L_1,通过 L_1 与自身的连接用于寻找 L_2,L_2 再与自身的连接寻找 L_3,依此类推,直到不能找到频繁 k-项集。在连接前每次通过剪枝去掉那些不满足最小支持度阈值和最小置信度阈值的项集。

Apriori算法的缺陷是:需要对事务数据库的重复扫描以及产生大量的候选项集。为了提高Apriori的算法效率,减少数据库扫描的范围,对Apriori算法进行了优化,并提出了一些比较巧妙算法,但是这些算法都没有从根本上解决Apriori算法需要产生大量候选项集以及多次扫描数据库这样一个关键问题。

3 S3CA算法

3.1 S3CA(Scan Combinatorial Count Comparison Algorithm)算法简介

在Apriori算法中,为找 L_k,通过 L_{k-1} 与自己连接产生候

*)基金项目:重庆市自然科学基金(2006BA6015)重点资助项目。曾 波 硕士,主要研究方向为算法、数据仓库与数据挖掘。

选 k-项集的集合，记做 C_k。C_k 由 $C^k_{L_{k-1}}=m$ 个 k-项集组成，为了计算每一个 k-项集在事务数据库 $T=\{t_1,t_2,\cdots,t_n\}$ 中出现的次数，需要 m 次扫描 T。

本文提出的单事务项集组合不产生候选项挖掘频繁项集的算法，整个过程分为四个步骤，即扫描、组合、计数、比较，其中前三个步骤同步进行。这也是 S3CA 算法名称的由来，在挖掘频繁项集时不产生候选项集且仅扫描数据库一次，从根本上解决了 Apriori 算法需要产生候选项以及重复扫描事务数据库的问题，提高了频繁项集的挖掘效率。

3.2 算法的描述

为计算第一个 k-项集的次数，扫描并记录其在 T 中的出现总次数，同时，对扫描过程中发现的其他 k-项集（如第二个 k-项集等），也对其做同样计数处理，那么算法就不必再为计算第二个 k-项集去扫描 T 了，因为其他 k-项集在 T 中出现的次数随着第一个 k-项集次数的计算而得到了相应的计算，从而将对事务数据库 T 的 m 次扫描降到了 1 次。

推广开来，要寻找事务数据库 T 中的频繁项集，需要对 T 中的每个事务的项进行组合（非空，非一项子集，下同），并按照上述方法一次扫描 T 以计算所有的项集组合在 T 中出现的次数。

综上所述，S3CA 算法的基本思想是：扫描事务数据库 $T=\{t_1,t_2,\cdots,t_n\}$，同时对每个事务的项进行本事务内部的组合，得到 x-项集，并对扫描过程中出现的相同项集进行次数累加，扫描结束后，就得到事务数据库 T 中所有项集出现的次数。这是挖掘频繁项集时需要的重要数据。

因为一次扫描就得到了所有有意义项集的出现次数，不需要重复的扫描数据库，不产生候选项集。得到 T 中每一事务的组合及其次数之后，项集中数据项最多且满足最小支持度的项集即为频繁项集。

3.3 S3CA 算法举例

假设有一事务数据库 T（如表 1 所示），数据库中有 10 个事务，事务中的项按字典次序存放，约定最小事务的支持计数为 2，即最小支持度阈值 min_sub=2/10=20%。现在要求用 S3CA 算法来寻找 T 中的频繁项集。

3.3.1 扫描，组合，计数

(1)扫描 T_{100}，对 T_{100} 中的项通过组合构建项集，并记录项集的出现次数 1。项集的组成及对应次数见表 2。

(2)扫描 T_{200}，对 T_{200} 中的项通过组合构建项集。这些项集若在表二中不存在，则称其为新项集，新项集需要在表二的“项集”栏目中添加，并记录其出现次数 1。若 T_{200} 中中产生的项集在 T_{100} 已经产生，则将该项集的次数进行加 1 操作，操作后的结果见表 3。

表 1 事务数据库

Tid	项的 ID 列表
T_{100}	I_1,I_2,I_3,I_5
T_{200}	I_1,I_2,I_3
T_{300}	I_1,I_3
T_{400}	I_2,I_3
T_{500}	I_1,I_2,I_4
T_{600}	I_2,I_3
T_{700}	I_2,I_4
T_{800}	I_1,I_2,I_5
T_{900}	I_1,I_2,I_4
T_{1000}	I_2,I_3,I_4,I_5

表 2 T_{100} 中组合项、次数

项集	次数
I_1,I_2	1
I_1,I_3	1
I_1,I_5	1
I_2,I_3	1
I_2,I_5	1
I_3,I_5	1
I_1,I_2,I_3	1
I_1,I_2,I_5	1
I_2,I_3,I_5	1
I_1,I_3,I_5	1
I_1,I_2,I_3,I_5	1

表 3 T_{200} 中组合项集、次数

项集	次数
I_1,I_2	1+1
I_1,I_3	1+1
I_1,I_5	1
I_2,I_3	1+1
I_2,I_5	1
I_3,I_5	1
I_1,I_2,I_3	1+1
I_1,I_2,I_5	1
I_2,I_3,I_5	1
I_1,I_3,I_5	1
I_1,I_2,I_3,I_5	1

(3)依此类推，扫描事务数据库 T 中的其余事务，并分类整理，结果见表 4。

表 4 T 中的项集组合及出现次数

序号	2-项集	次数	序号	3-项集	次数	序号	4-项集	次数
1	I_1,I_2	5	11	I_1,I_2,I_3	2	19	I_1,I_2,I_3,I_5	1
2	I_1,I_3	3	12	I_1,I_2,I_5	2	20	I_2,I_3,I_4,I_5	1
3	I_1,I_4	2	13	I_1,I_3,I_5	1			
4	I_1,I_5	2	14	I_2,I_3,I_5	2			
5	I_2,I_3	5	15	I_1,I_2,I_4	2			
6	I_2,I_4	4	16	I_2,I_3,I_4	1			
7	I_2,I_5	3	17	I_3,I_4,I_5	1			
8	I_3,I_4	1	18	I_2,I_4,I_5	1			
9	I_3,I_5	2						
10	I_4,I_5	1						

说明：通过对事务数据库 T 的一次扫描，获得了 T 中所有的项集及次数，这是我们挖掘频繁项集的重要数据。

3.3.2 求频繁项集

按照算法的规定“项集中数据项最多且满足最小支持度的项集即为频繁项集”，表四中有 4-项集，但是它们在 T 中的出现次数均为 1，不满足最小支持度 20%这个约定条件；其次 3-项集有$\{I_1,I_2,I_3\}$，$\{I_1,I_2,I_5\}$，$\{I_2,I_3,I_5\}$，$\{I_1,I_2,I_4\}$满足最小支持度 20%，且在排除不符合条件的 4-项集后，它们的数据项最多，所以符合要求。2-项集不满足“项集中数据项最多”这个条件，所以不予考虑。

所以 T 中的频繁项集是$\{I_1,I_2,I_3\}$，$\{I_1,I_2,I_5\}$，$\{I_2,I_3,I_5\}$，$\{I_1,I_2,I_4\}$。

按照 Apriori 算法的性质：频繁项集的所有非空子集都必须是频繁的[7]。本算法规定“项集中数据项最多且满足最小支持度的项集即为频繁项集”，不会出现数据项最多且满足最小支持度的项集其非空子集出现不频繁的情况。

这是因为，本算法在进行项集的组合的时候，有这样一条

（下转第 226 页）

gent必须从各个分项评价 Agent 那儿获取单项评定结论，所以在系统的运行中，进程间的通信是必须解决的问题。

在模块通信设计中，本研究提出采用黑板通信方式和用KQML来表示模块间的通信原语。具体在本系统中，首先定义了两种主要的通信原语，分别为咨询型和应答型，相应的关键字为"Ask"和"Reply"。然后规范了系统中各 Agent 的交互内容信息，并为每个功能 Agent 提炼出最主要的通信内容类型。

```
Ask
    :sender PriceAgent
    :receiver InfoAgent
    :content  BasePrice
    :sub-item NULL
    :value NULL
```

```
Reply
    :sender InfoAgent
    :receiver PriceAgent
    :content  BasePrice
    :sub-item NULL
    :value $1000.00
```

图 5　Agent 间的通信原语

图 5 是报价 Agent 和信息 Agent 的通信原语的实例，系统中报价 Agent 的功能相当于评标委员会中报价评定专家的作用。在报价评定期间，报价 Agent 需要知道的信息内容主要有投标单位数目、标底价、各标书的投标价以及评标策略等。然而这些信息在系统启动时，由用户输入到信息 Agent 中，所以报价 Agent 需要向信息 Agent 咨询标底信息和投标信息，后者要根据请求不同内容进行相应的应答。示图给出了报价 Agent 向信息 Agent 询问标底数值的通信原语结构和后者的应答原语结构，同理可以实现其它 Agent 和它们的通信原语。

结论与展望　本研究选择以工程项目评标决策问题为研究对象，应用 Multi-Agent 理论设计和实现了智能评标系统，不仅拓宽了智能技术的应用范围，同时也是对 Agent 自身理论的延伸。由于招投标的范围广泛，我们现在研究主要针对工程项目招投标中的评标，所以今后的研究可以在此基础上拓宽，使得系统功能更加强大。

参 考 文 献

1　鲁耀斌，朱志伟，等. 基于 Web 的招标投标系统的设计与实现[J]. 计算机工程与应用，2001(2)：41～44
2　刘艳，李伟. 评标智能决策支持系统的研究与实现. 施工技术[J]，2004，33(12)：45～46
3　夏恩君，苏广领. 工程项目评标优化决策模型研究[J]. 中国软科学，2005，1：133～138
4　王芳，方东升. 建筑工程投标报价估测模型研究[C]. 长沙铁道学院学报，1999，(6)：80～85
5　鲁耀斌，张金隆. 投标标书评价模型的研究．华中理工大学学报[C]，1999(2)：52～54
6　Ouelhadja D, Petrovica S, et al. Inter-agent cooperation and communication for agent-based robust [J]. dynamic scheduling in steel production Advanced Engineering Informatics, 2004(18): 161～172
7　Nahm Y E, Ishikawa H. A hybrid multi-agent system architecture for enterprise integration using computer networks [J]. Robotics and Computer-Integrated Manufacturing, 2005(21): 217～234
8　阮连法，温海珍，汪尤升. 建设工程招标评标的模糊方法研究. 浙江大学学报(理学版)，2002，29(2)：233～240

(上接第 197 页)

规定——"对 T 中的每个事务的项进行非空组合"，换言之，本算法项集组合是以每个事务为单位来进行的(单事务项集组合)，没有在事务之间进行项的组合，高数据项的频繁项集 I 是在事务数据库中真实存在，假如 I 满足最小事务的支持计数，它们其子集将更能满足最小事务的支持计数，因为 I 的子集出现的频率通常比 I 高，但是最低限度能保证 I 的水平。而 Apriori 算法候选项的组合是以整个事务数据库为基础来进行的组合，这肯定会产生一些毫无意义的项集，因此产生 Apriori 算法中"频繁"项集的子集存在不频繁的可能。

认 $\{I_1, I_2, I_3\}$ 为例，其子集有：$I_1=\{I_1, I_2\}$，$I_2=\{I_1, I_3,\}$，$I_3=\{I_2, I_3\}$，从表 4 中，可以查出项集 I_1，I_2，I_2 的计数分别为 5，3，5，均满足最小支持度 20%这个要求，符合上面的结论。

所以项集中数据项最多且满足支持度的项集即为频繁项集。

4　S3CA 与 FP-growth 算法的运行效率比较

使用上述事务数据库，分别采用 FP-growth 算法和 S3CA 算法，图 1 是实验的结果比较。

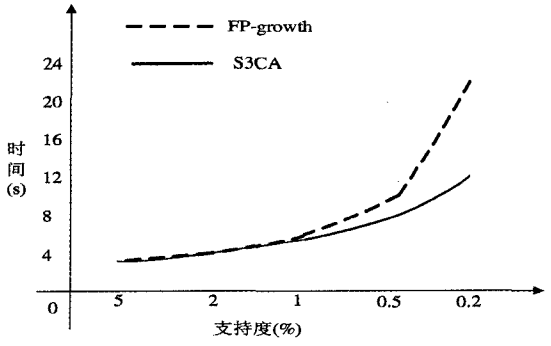

图 1　S3CA 与 FP-growth 算法的效率比较

从图中可以看出，两种算法在支持度比较大的情况下，算法的运行效率差别比较小，但是当支持度比较小的时候，S3CA 的运行效率明显比 FP-growth 算法高。

结论　S3CA 算法在一次扫描的过程中进行单事务内部项集组合，相同项集次数的累加。项集组合的范围仅局限于事务内部，而不是事务与事务之间，减少了项集的数量以及对系统资源的消耗，提高了频繁项集挖掘的效率。最后根据扫描组合累加的结果，依照"项集中数据项最多且满足最小支持度的项集即为频繁项集"的原则，通过比较项集的数据项数量及支持度，最终挖掘频繁项集。

与 Apriori 算法比较起来，S3CA 算法只扫描一次事务数据库，不产生候选项，单事务内部的项集组合等思想，大大提高了单维布尔关联规则频繁项集的挖掘效率。

本算法最大的缺陷是：若事务中的项过多，将会产生非常多的非空频繁子集，这将极大地消耗系统内存，所以本算法的进一步优化是下一步的研究方向。

参 考 文 献

1　Park J S, Chen M S, Yu P S. An Effective Hash Based Algorithm for Mining Association Rules [R]. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, CA, 1995. 175～186
2　Agrawa lR, Imielinski T, Swami A. Mining association rules between sets of items in large databases (C). In: Buneman P, Jajodia S, eds. Proc. of the ACM SIGMOD Conf. on Management of Data (SIGMOD'93). New York: ACM Press, 1993. 207～216
3　Agrawa lR, Srikant R. Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C, eds. Proc. of the 20th Int'l Conf. on Very Large Data Bases. Santiago: Morgan Kaufmann, 1994. 478～499
4　Savasere A, Omiecianski E, Navathe S. An efficient algorithm forminin gas sociation rules in large databases
5　Toivonen H. Sampling Large Databases for Association Rules. In: Proc of the 22nd Int'l Conf. on Very Large Data Bases. Mumbai, India, 1996. 134～145
6　Brin S, Motwani R, Ullman J D, Tsur S. Dynamic itemset counting and implication rules for market basket data. In: ACM SIGMOD International Conference on the Management of Data, May 1997. 255～264
7　Han Jiawei, Kamber M. Data Mining Concepts and Techniques, 2001. 225～244
8　黄传明. 一种基于散列技术和事务压缩的关联规则挖掘算法[J] 计算机工程，2003，29(22)
9　周焕银，张永，蔺鹏. 一种不产生候选项挖掘频繁项集的新算法[J]. 计算机工程与应用，2004，15(184)