数据挖掘中常用关联规则挖掘算法

王爱平,王占凤,陶嗣干,燕飞飞
(安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

摘 要:文中首先介绍了数据挖掘中关联规则的经典算法——Apriori算法。再从宽度、深度、划分、采样、增量式更新等几个角度对关联规则挖掘进行了分类讨论。然后运用文献查询和比较分析的方法对常见的关联规则挖掘算法进行了概述,主要包括FP-growth算法、DHP算法、Partition算法、FUP算法、CD算法等算法。最后对关联规则挖掘的发展远景进行了展望。

关键词:数据挖掘;关联规则;频繁项集;挖掘算法
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2010)04-0105-04

Common Algorithms of Association Rules Mining in Data Mining

WANG Ai-ping, WANG Zhan-feng, TAO Si-gan, YAN Fei-fei
(Ministry of Education Key Laboratory of Intelligent Computing & Signal Processing, Anhui University, Hefei 230039, China)

Abstract: First introduces the classical algorithm of association rule mining - Apriori. Then classified discusses the association rule mining from several angles such as width, depth, partition, sampling and incremental updating. It summarizes the commons algorithms of association rule mining through querying documents and comparative analysis. It mainly includes FP-Growth algorithm, DHP algorithm, Partition algorithm, FUP algorithm, CD algorithm and so on. At last prospect the association rule mining.

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

0 引 言

数据挖掘(Data Mining),又称数据库中的知识发现(Knowledge Discovery in Database),是从大量的、不完整的、有噪声的、模糊的、随机的大型数据中提取隐含在其中的、人们事先未知的、具有潜在价值的信息和知识的过程[1]。简单的说,数据挖掘就是从大量数据中提取或"挖掘"出人们有用的知识。面对当前"海量数据,微量信息"的现状,数据挖掘的重要研究分支——关联规则,作为一种高级和智能的数据处理和分析技术的研究正方兴未艾。

通过关联规则挖掘,可以得到隐含于海量数据中具有潜在价值的有用信息。关联规则的目标是以有效的方式提取最有趣的模式。

迄今为止已提出了许多高效的关联规则挖掘算法,其中以Agawal提出的Apriori算法[2]最为著名,大多数挖掘算法都是建立在Apriori算法基础之上,但是Apriori算法无论在时间效率还是空间伸缩性上都面临着挑战,因此研究人员探索出很多新的挖掘方法,并拓展了关联规则概念及应用范围。

1 关联规则的基本概念

设集合 $I=\{i_1,i_2,\cdots,i_m\}$,其中 $i_k(k=1,2,\cdots,m)$ 表示项。如果 $X\subset I$,集合 X 被称为项集。当 $|X|=k$,则 X 被称为 k-项集。事务二元组 $T=(\text{tid},X)$,tid是事务唯一的标识符称为事务号。数据集 $D=\{t_1,t_2,\cdots,t_n\}$ 是由 $t_1,t_2,\cdots,t_n$ 事务组成的集合。

关联规则可以描述为:形如A⇒B的蕴涵式,其中 $A\subset I,B\subset I$,并且 $A\cap B=\varnothing$。项集 X 的支持度 s 是 D 中包含 X 的事务数占所有事务数的百分比,记为 $s(X)=p(X)=\frac{\sup(X)}{|D|}$。项集 X 的置信度 c 是 D 中同时包含 $X\cup Y$ 的事务数占包含 X 的所有事务数的百分比,记为 $c(X)=P(X|Y)=\frac{\sup(X\cup Y)}{\sup(X)}$。至于

收稿日期:2009-08-15;**修回日期**:2009-11-21
基金项目:国家自然科学基金项目(60472065)
作者简介:王爱平(1956-),女,甘肃庆阳人,教授,从事计算机教学与研究。

最小支持度 minsup 和最小置信度 minconf 都是由用户所给定，如果项集 X 的 sup(X)≥minsup，那么项集 X 被称为频繁项集，其中生成的关联规则中所有支持度和置信度都不小于 minsup 和 minconf 的被称为强关联规则。

关联规则的支持度表示在整个数据库中的重要性，而置信度则反映其可靠程度。只有支持度和置信度均为较高的关联规则才是用户感兴趣的、有用的关联规则。

2 关联规则的种类

根据不同的标准，关联规则可以用很多不同的方法分成若干类型[1]，根据挖掘模式的完全性可以把关联规则分为闭频繁项集、挖掘频繁项集的完全性、极大频繁项集和被约束的频繁项集。根据规则涉及的数据的层和维可以把关联规则分为单层关联规则、多层关联规则、单维关联规则和多维关联规则的挖掘。根据规则所处理的值的类型可以把关联规则分为挖掘布尔型关联规则和量化关联规则。根据所挖掘的规则类型可以把关联规则分为关联规则和相关规则挖掘。根据所挖掘的模式类型可以把关联规则分为频繁项集挖掘、序列模式挖掘、结构模式挖掘等。根据所挖掘的约束类型可以把关联规则分为知识类型约束、数据约束、维/层约束、兴趣度约束、规则约束。

3 关联规则挖掘算法

3.1 经典的关联规则挖掘算法

1994 年 Agrawal 提出的 Apriori 算法是挖掘完全频繁项集中最具有影响力的算法。算法有两个关键的步骤：一是发现所有的频繁项集；二是生成强关联规则。

发现频繁项集是关联规则挖掘中的关键步骤。在 Apriori 算法中还利用了“频繁项集的子集是频繁项集，非频繁项集的超集是非频繁项集”这一个性质有效的对频繁项集进行修剪。

算法核心思想：

给定一个数据库，第一次扫描数据库，搜索出所有支持度大于等于最小支持度的项集组成频繁 1 - 项集即为 L_1，由 L_1 连接得到候选 1 - 项集 C_1；

第二次扫描数据库，搜索出 C_1 中所有支持度大于等于最小支持度的项集组成频繁 2 - 项集即为 L_2，由 L_2 连接得到候选 2 - 项集 C_2；

同理第 k 次扫描数据库，搜索出 C_{k-1} 中所有支持度大于等于最小支持度的项集组成频繁 k - 项集即为 L_k，由 L_k 连接得到候选 k - 项集 C_k，直到没有新的候选集产生为止。

Apriori 算法需扫描数据库的次数等于最大频繁项集的项数。Apriori 算法有两个致命的性能瓶颈：产生的候选集过大(尤其是 2 - 项集)，算法必须耗费大量的时间处理候选项集；多次扫描数据库，需要很大的 1/0 负载，在时间、空间上都需要付出很大的代价。

3.2 常用的关联规则挖掘算法

目前常见的关联规则挖掘算法大致可分为宽度优先算法、深度优先算法、数据集划分算法、采样算法、增量式更新算法等。下面对一些常用算法做简单的介绍。

3.2.1 宽度优先算法

宽度优先算法又称为分层算法，包括由 Agrawal 等人提出的 Apriori、AprioriTid[3]和 AprioriHybrid[4]算法，Park 等人提出的 DHP 算法[5]等等。

Apriori 算法也是宽度优先算法，AprioriTid 算法是在 Apriori 算法的基础上演化而来的。该算法第一趟扫描数据库时采用 Apriori 算法，当再次扫描时不再是扫描整个数据库，而只是扫描上次生成的候选项集，扫描的同时还会计算出频繁项集的支持度，以减少扫描数据库的时间来提高算法的效率。Apriori 算法和 AprioriTid 算法的融合产生了 AprioriHybrid 算法，初始扫描数据库时使用 Apriori 算法，当生成的候选项集大小可以存放到内存中进行处理时再转向 AprioriTid 算法，直到找出所有的频繁项集。DHP 算法采用哈希(Hash)表技术对数据集和候选项集进行修剪来降低算法的时间和空间的开销。它利用哈希表在计算(k - 1) - 项集时先粗略计算出 k - 项集的支持度，排除无意义的候选 k - 项集来减少候选 k - 项集的数量，尤其是对候选 2 - 项集的数量控制特别突出。总的来说，宽度优先算法的不足之处还是在于需要生成大量候选项集，需要多次扫描数据库。

3.2.2 深度优先算法

深度优先算法中常见的算法有 FP - growth 算法[6]、OP 算法[7]、TreeProjection 算法[8]等。

FP - growth 算法是深度优先算法中最新最高效的且从本质上不同于 Apriori 算法的经典算法。基本思想是：采取分而治之的策略，首先在保留项集关联信息的前提下，将数据库压缩到一棵频繁模式树(FP - tree)中；然后将这种压缩后的 FP - tree 分成一些条件数据库并分别挖掘每个数据库。在算法中有两个关键步骤：一是生成频繁模式树 FP - tree；二是在频繁模式树 FP - tree 上挖掘频繁项集。

与 Apriori 算法相比，FP - growth 算法具有以下优

点:FP-growth算法只需扫描数据库两次,避免多次扫描数据库;不需要产生庞大的候选项集,在挖掘过程中大大减少了搜索空间,在时间效率、空间效率上都有一个量级的提高。但它的应用难点在于处理很大的且很稀疏的数据库时,在挖掘处理、递归运算中都需要相当大的空间。

3.2.3 数据集划分算法

数据集划分算法包括Savasere等人提出的Partition算法[9],Brin等人提出的DIC算法[10]等。Partition算法是从逻辑上将整个数据库划分成几个相互独立的可以存放在内存中进行处理的数据块,节省访问外存时I/O的开销。它单独考虑每个逻辑块生成相应的频集,然后利用"频繁项集至少在一个分区中是频繁的"这一性质把所有逻辑块生成的所有频集合并生成所有可能的全局候选项集,最后再次扫描数据库计算项集的支持度进行全局计数。整个过程只需对数据库进行两次扫描,但是产生的候选项集数量比较大。DIC算法同样采取数据库划分的思想,将数据库划分为若干个分区并在每个分区的开始部分做标记,在扫描数据库过程中可以在各个分区的标记点添加候选项集,在计算项集时并行计算可能为频集的支持度。算法扫描数据库的次数基本上是少于最大频集的项数。在数据块划分恰到好处时只需通过两次扫描数据库就能找出所有的频繁项集。

在基于划分的算法中主要瓶颈是算法执行的时间,同时产生的频繁项集的精度也不是很高。但是该类型的算法具有高度的并行性,只需扫描两次数据库,大大减少了I/O操作从而提高了算法效率。

3.2.4 采样算法

采样算法包括由Park等人提出的可调精度的挖掘算法[11]、Toivonen提出的Sampling[12]算法等。Sampling算法是从数据库 D 中随机抽取一个可以调入内存的数据库子集 D',然后求出数据库子集 D' 中可能在数据库 D 中成立的所有规则,再用数据库 D 中剩余部分($D-D'$)来验证结果的正确性。它适用于挖掘准确性不太高而挖掘效率较高的环境。采样算法很大程度上减少了扫描数据库的时间开销,但它最大的缺点就是可能产生数据扭曲导致结果不精确。当选取的随机样本不能代表整个数据库的分布形式时,就有可能丢失一些全局频繁项集导致结果不精确。如果频繁项集包含了数据库 D 中的所有频繁项集,则只需要扫描一次 D。否则,为了减少这个问题带来的影响,可以使用更小的支持度阈值在随机样本上做第二次扫描数据库再次产生频繁项集,找出在第一次扫描中遗漏的频繁项集。通过对数据库多次扫描来减少频繁项集的遗漏。对于数据扭曲现象,Lin和Dunham在文献[13]中讨论了反扭曲(Anti-skew)算法来挖掘关联规则,可以使得扫描数据集的次数少于2次。

3.2.5 增量式更新算法

增量式更新算法是利用已挖掘的关联规则在变化了的数据库或参数上发现新的关联规则、删除过时的关联规则来维护数据集更新的问题。目前大多数的增量式更新算法都是以Apriori算法为核心进行的改进与演化,包括D. W. Cheung等人提出的FUP和FUP_2算法[14],冯玉才等人提出的IUA和PIUA算法[15]、高峰等人提出的IUAR算法[16]等等。

FUP算法是Apriori算法的改进,也是解决增量更新问题的一种经典算法。FUP算法主要是针对在最小支持度和最小置信度不变的情况下,数据库DB被添加、删除或修改时,如何生成更新后的数据库的关联规则。它利用已挖掘得到的频繁项集信息来避免重复计算频繁项集支持数的时间开销来提高算法效率。FUP算法不足之处:算法在处理规模巨大的候选项集时耗费大量时间;对候选项集进行模式匹配时需要多次重复扫描数据库代价很大。

FUP_2算法同时考虑到增加数据库和修改、删除数据库的情况,比较适用于大量的增加数据库和少量的删除数据库的情况。

IUA、PIUA算法都是主要考虑在最小支持度和最小置信度发生变化而数据库DB不变时,如何生成DB中的关联规则。

IUAR算法主要考虑在最小支持度和最小置信度和数据库DB同时发生变化时,如何生成更新后的关联规则。

3.2.6 并行挖掘算法

并行算法是利用同时执行的诸进程的集合相互作用和协调完成对给定问题的求解。包括Agrawal等人提出的CD、DD、CaD算法[17],Park等人提出的PDM[18]算法,Cheung等人提出的DMA和FDM算法[19]等。

CD算法允许在空闲的处理器上进行并行冗余计算以减小通信量,速度几乎可以达到线性加速比的速度。但它的缺点是通信量和候选频繁项集都比较大。

DD算法通过把候选集划分到各个处理器来克服CD算法的缺陷,然而DD算法由于数据移动方案效率较低导致通信负载较大、处理器间的交互模式易导致处理器处于空闲状态、每一笔交易记录都根据多个哈希树进行处理导致冗余计算等缺点[20]。

CaD算法试图通过划分数据库和候选集的办法来减少处理器之间的数据依赖性,使每个处理器可以独

立地进行计算。但它在划分候选集时要对整个的事务数据库进行划分并分配到每一个处理器节点中,从而消耗了大量的时间用于通信。

PDM 算法类似于 CD 算法,所有处理器含有相同的杂凑表和候选集。并行候选集生成的过程是通过每个处理器生成一个候选子项集,然后交换所有处理器上的子项集生成全局候选集来实现。但是 PDM 算法对非大项集的项目和事务的物理剪枝要涉及大量磁盘 I/O 操作。

关联规则挖掘算法中目前常用的还包括:基于约束的关联规则挖掘算法、多层关联规则挖掘算法、多维关联规则挖掘算法、加权支持度关联规则挖掘算法等算法。

4 结束语

文中对数据挖掘中关联规则挖掘进行了细致清晰的分类讨论,并对比较常用的几种挖掘算法进行了分析比对和总结。关联规则挖掘中多值关联规则挖掘、兴趣度度量和评估技术、增强关联规则的维护方法、增强关联规则挖掘算法与用户的交互性、关联规则挖掘如何应用到隐私保护和信息安全问题中都将是下一步研究的热点问题。

不过现有的改进算法远远不能满足人们对挖掘系统快速及时响应的需求,如何提高挖掘过程的效率、与用户进行交互生成可视化结果等等都是以后研究工作的重点和难点。

参考文献:

[1] Han Jiawei, Kamber M. 数据挖掘概念与技术[M]. 范 明, 孟小峰等, 译. 北京:机械工业出版社, 2001:149-176.

[2] Agrawal R, Imielinski T, Swami A. Mining Association Rules between Sets of Items in Large Databases[C]//Proceedings of the 1993 ACM SIGMOD Conference. Washington D C: [s. n.], 1993:207-216.

[3] Agrawal. R, Srikant R. Fast Algorithm for Mining Association Rules[C]//Proceedings of the 20th Very Large Data Bases (VLDB'94) Conference. Santiago, Chile: [s. n.], 1994:487-499.

[4] Agrawal R, Srikant R. Fast Algorithm for Mining Association Rules in Large Databases[R]. San Jose, CA: IBM Almaden Research Center, 1994.

[5] Park J S, Chen M S, Yu P S. An effective hash based algorithm for mining association rules[C]//In: Proc. 1995 ACM SIGMOD. San Jose, CA: [s. n.], 1995:175-186.

[6] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]//In Proceeding of the 2000 ACM SIGMOD Conference on Managerment of Data. Dallas, TX: [s. n.], 2000.

[7] Liu J, Pan Y, Wang K, et al. Mining frequent item sets by opportunisitc projection[C]//Proc. of the Eighth ACM SIGKDD Intl. Conf on Knowledge Discovery and Data Mining. Alberta, Canada: [s. n.], 2002:229-238.

[8] Agarwal R, Aggarwal C, Prased V V V. A tree projection algorithm for generation of frequent itemsets[J]. J. Parallel and Distributed Computing, 2001, 61(3):350-371.

[9] Savasere A, Omieeinski E, Navathe S. An efficient algorithm for mining association rules in large databases[C]//Proceedings of the 21st International Conference on Very Large Databases. Zurich, Switzerland: [s. n.], 1995:432-443.

[10] Brin S, Motwani R, Ullman J D. Dynamic Itemset counting and implication rules for market basket data[C]//Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data. Tucson, Arizona: [s. n.], 1997:255-264.

[11] Park J S, Chen M, Yu P S. Using a Hash-Based Method with Transaetion Trimming and Database Scan Reduction for Mining Association Rules[J]. IEEE Transaction on knowledge and data engineering, 1997, 9(5):813-825.

[12] Toivonen H. Sampling Large Databases for Association Rules [C]//Proceedings of 1996 International Conference on Very Large Databases (VLDB'96). Bumbay, India: Morgan Kaufmann, 1996:134-145.

[13] Lin J L, Dunhan M H. Mining association rules: Anti-skew algorithms[C]//Proceedings of the International Conference on Data Engingeering. Orlando, Florida: [s. n.], 1998.

[14] Cheung D W, Han J, Ng V T. A Fast Distributed Algorithm for Mining Assoeiation Rules[C]//Proceedings of 1996 International conference on Parallel and Distributed Information System. Miami Beaeh, FL: [s. n.], 1996:31-44.

[15] 丁祥武. 挖掘时态关联规则[J]. 武汉交通科技大学学报, 1999, 23(4):365-367.

[16] 高 峰, 谢剑英. 发现关联规则的增量式更新算法[J]. 计算机工程, 2000, 26(12):49-50.

[17] Agrawal R, Shafer J. Parallel Mining of Association Rules[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6):962-969.

[18] Park J S, Chen M S, Yu P S. Efficient Parallel Mining For Association Rules[C]//Proc. 4th International Conference Information and Knowledge management. Baltimore, MD: [s. n.], 1995:31-36.

[19] Cheung D W, Ng V, Fu W C. Effieient mining of association rules in distributed databases[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(1):910-921.

[20] 刘 颖. 关联规则并行算法在医药销售系统中的应用[D]. 重庆:重庆大学, 2004.