改进的关联规则算法及其应用

Improved Association Rule Algorithm and Its Application

张毅驰　（苏州大学 文正学院 江苏 苏州 215104）
朱巧明　（苏州大学 计算机科学与技术学院 江苏 苏州 215006）

摘要：本文根据数据挖掘中关联规则的性质以及高校成绩管理数据库的自身特点，在经典关联规则算法 Apriori 算法的基础上提出了一种改进的算法 A++算法，并利用该算法对学生成绩管理数据库进行了关联规则挖掘，得到了隐含在数据库中的有用信息。

关键词：数据挖掘 关联规则 Apriori 算法 A++算法 成绩管理数据库

1　关联规则挖掘

1.1　关联规则概述

在数据挖掘的模式中，关联规则挖掘是比较重要的一种。所谓关联规则挖掘是寻找数据项中的有趣联系，决定哪些事情将一起发生。例如：超市中客户在购买 A 的同时经常会购买 B，即 A = > B，这就是关联规则。

关联规则挖掘最早是由 Agrawal 等人于 1993 年提出的[2]，其形式化的描述如下：

设 $I=\{i_1,i_2,i_3,\cdots,i_m\}$ 是由 m 个不同的数据项目组成的集合，其中的元素称为项（item），项的集合称为项集，包含 k 个项的项集称为 k 项集，给定一个事务 D，即事务数据库，其中的每一个事务 T 是数据项 I 的一个子集，即 T　I，T 有一个唯一的标识符 TID；当且仅当 X　T 时，称事务 T 包含项集 X；那么关联规则就是形如“X = > Y”的蕴涵式；其中 X　I，Y　I，$X\cap Y=$　。关联规则 X = > Y 在事务数据库中成立，具有支持度 s 和具有置信度 c。

其中支持度 s 表示 D 中有 s% 的事务包含 $X\cup Y$，描述为：

support（X = > Y）= 同时包含 X 和 Y 的事务数 ÷ 总事务数

置信度 c 表示 D 中包含 X 的事务中有 c% 的事务同时也包含 Y，描述为：

confidence（X = > Y）= 同时包含 X 和 Y 的事务数 ÷ 包含 X 的事务数

关联规则挖掘就是在用户给定“最小支持度（MinSup）”和“最小置信度（MinConf）”的基础上对事务数据库进行挖掘，以找到满足 MinSup、MinConf 的规则为人们所用。

1.2　经典关联规则算法描述——Apriori 算法

继关联规则挖掘问题提出后不久，Agrawal 等人于 1994 年提出了该挖掘的核心算法——Apriori 算法[3]，该算法堪称关联规则挖掘的经典算法，其核心思想如下：

（1）找出存在于事务数据库中的所有频繁项集（frequent itemset）。所谓频繁项集是指那些支持度（support）大于等于最小支持度（MinSup）的项集。

（2）利用频繁项集生成关联规则。

两步中较难实现的是第 1 步，因该步的实现需要多次扫描整个事务数据库，这需要消耗很大的时间和空间。因此这是制约 Apriori 算法运行效率的关键所在。下面就第 1 步的实现算法简单描述如下：

该算法使用了递推的方法。首先产生频繁 1 项集的集合 L_1，然后产生频繁 2 项集的集合 L_2，直到有某个 r 值使得 L_r 为空，这时算法停止。

这里在第 k 次循环中，过程先产生候选 k 项集的集合 C_k，然后扫描数据库以计算每个项集的支持度，保留支持度大于等于最小支持度的项集产生频繁 k 项集 L_k。

其中 C_k 的产生需要经过两步来完成：

① 连接。Ck 中的每一个项集是由 Lk－1 中的两个

项集连接产生的，这两个项集需满足如下要求：项集中前 k－2 项均相同，只有最后一项（k－1 项）不同。

② 剪枝。因一个频繁项集的任一非空子集都是频繁项集，因此在进行了第一步连接后，需删除那些具有非频繁子集的项集，即所谓剪枝。

1.3 关联规则算法优化研究现状

综上分析 Apriori 算法是一种宽度优先的算法，该算法通过多次扫描数据库 D 来发现所有频繁项集。数据库 D 一般是一种海量数据，程序通过对海量数据的多次扫描需要消耗大量的时间和空间，随着数据库中数据的海量增长，这种消耗是呈指数级增长的。因此这也就是 Apriori 算法的瓶颈所在。为此后人提出了许多优化算法用以改进 Apriori 的性能。其中主要包括：

1995 年 Park 等人提出的基于 hash 的算法——Dynamic Hashing and Pruning（DHP）算法[4][5]。该算法通过引入 hash 技术来提高生成频繁 2 项集的效率从而提高整个算法的运行效率。

同年 Savasere 等人提出的基于划分的算法[6]。该算法把数据库从逻辑上划分成几个互不相交的块，算法分别对每个分块进行频繁项集的搜索，然后把产生的频繁项集合并生成所有可能的频繁项集，并计算这些项集的支持度。该算法的优点在于只需两次扫描整个事务数据库从而提高了算法的效率。

1996 年 Toivonen 提出的基于采样的算法[7]。该算法先对数据库进行采样，其次对样本数据库进行挖掘从中得到相应规则，最后将这些规则带到数据库中进行验证。该算法显著提高了算法的运行效率，但有时会使产生的结果不精确。

另外还有 Brin 等人提出的动态项集计数算法[8]，以及我国学者近年来提出的基于栈变换的算法[9]；关联规则的矩阵算法[10]；基于事务压缩和项目压缩的算法[11]等等，这些算法都在不同方面对关联规则算法尤其是其中的经典 Aprior 算法进行了优化。

1.4 一种改进的 Apriori 算法——A＋＋算法

上节介绍了关于关联规则算法优化的国内外研究现状，本节作者将根据高校教学管理系统的实际，给出一个新的改进算法，在此命名为 A＋＋算法。

1.4.1 A＋＋算法改进的基本思想

（1）将整个数据库一次性读入一个二维数组，避免多次扫描数据库，提高程序运行的效率。根据高校成绩管理系统的实际，其数据库的规模不是非常巨大，一般来说字段数不会超过 200 个，记录数不会超过 30,000 条。另 Apriori 算法处理的都是布尔型数据，该类型的数据存储最多 1 个字节，则该数据库占用空间为 200×30,000×1＝6,000,000 字节＝5.7M≈6M，目前计算机的内存可以承受。

（2）及时删除“无用”事务。基于 Apriori 算法的两个性质[12]：

① 对于已知规模的数据库 D，任意一个项集 I 的支持度与规模小于 I 的事务无关。因此可以删除规模小于 I 的事务。

② 当一个事务不包含长度为 k 的频繁项集，则必然不包含长度为 k＋1 的频繁项集。因此可以在生成 k＋1 频繁项集之前先删除这样的事务，以减少下次扫描数组（数据库）的时间。

因数组的删除操作耗时较多，这里对删除事务的操作并非真正删除数组元素本身，而是做一个标记表示该记录已被删除。在此本算法将二维数组每行的第 0 个元素作为标记位（“0”表示存在，“1”表示已被删除）。

1.4.2 A＋＋算法描述

基于以上两点基本思想，A＋＋算法描述如下：

（1）生成频繁项集 L，由以下五步完成：

第一步：根据数据库 D 的大小建立相应大小的二维数组 DA。将数据库中的内容一次性全部读入数组中，置所有标志位为“0”，完成初始化工作。

第二步：扫描 DA，找出所有 C_1 的支持度，同时标志 DA 中所有项集长度小于等于 1 的事务，根据用户给定的最小支持度（MinSup）确定 L_1。

第三步：对 L_k 进行连接操作，生成 C_{k+1}。（k＝1,2,3,…）

第四步：扫描 DA 算出 C_{k+1} 的支持度，同时标志 DA 中所有项集长度小于等于 k＋1 的事务；标志 DA 中所有不包含 C_{k+1} 的事务，根据用户给定的最小支持度（MinSup）成生 L_{k+1}。

第五步：k＝k＋1，转到第三步，直到 L_k 为空集时终结。

（2）生成关联规则

根据频繁项集的性质：频繁项集的所有子集均是频繁项集，生成关联规则可以如下完成。

定义 L_{max} 为最高频繁项集的集合，即包含项数最多的频繁项集的集合。

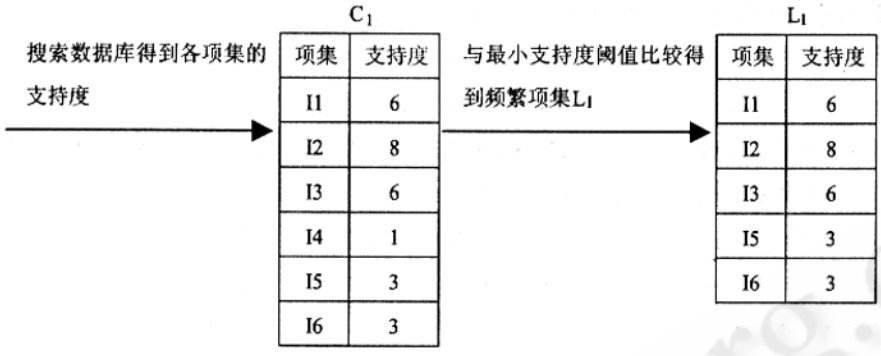

图 1　生成 C_1，扫描得到 L_1

从 L_{max} 开始依次递减直到 L_2 为止执行循环操作。这里第 k 次循环（$2 \leqslant k \leqslant max$），系统对 L_k 的每个元素 l_k 到 L_{k-1} 中找子集 $l(k-1)$，如果找到子集，并且 support(l_k)/support($l_{(k-1)}$) ≥MinConf，则输出该规则。

表 1　演示事务数据库

TID	I1	I2	I3	I4	I5	I6
T001	1	1			1	1
T002		1				
T003		1	1			1
T004	1	1		1		
T005	1		1		1	
T006		1	1			
T007	1		1			
T008	1	1	1		1	
T009	1	1	1			
T010		1				1

1.4.3　A + +算法与经典 Apriori 算法的比较

利用表 1 将 A + +算法与经典 Apriori 算法在扫描数据库规模上进行比较。这里不考虑 A + +算法一次性将数据库读入内存的因素，只就“及时删除‘无用’事务”与经典算法进行比较。设定最小支持度为 20%，对于表中 10 个事务，则最小支持事务数为 2 个事务。

算法执行过程如下：

（1）扫描数据库，得到 C_1 及其支持度。筛选出支持度≥2 的项集得到 L_1（此处删除了 I4），删除数据库中交易项不大于 1 的 TID（此处删除了 T002）。如图 1 所示。

（2）用 L_1 的连结产生 C_2，扫描数据库得到 C_2 中各元素的支持度。扫描同时删除交易项不大于 2 的 TID（此处删除了 T006、T007、T010）以及交易项中不包含任何 C_2 元素的 TID（此处没有）。筛选出支持度≥2 的项集得到 L_2（此处删除了 I116、I316、I516）如图 2 所示。

限于篇幅原因，这里不再继续演示，直接列出演示结果于表 2 中。由该表可见，删除事务是 A + +算法提高效率的关键之一。当为确定候选 k 项集中元素的支持度而扫描数据库的同时删除了数据库中事务项不大于 k 的事务和事务中不包含任何 C_k 元素的事务。显而易见，通过多轮扫描，数据库中的大量事务被删除从而节省了扫描时间。而 Apriori 算法每次都要彻底扫描整个原始数据库 D，这要耗费大量的时间。

2　关联规则挖掘在成绩管理系统中的应用

2.1　问题的提出——课程间相关联系的挖掘

目前大多高校中教学计划的制定工作一般都是由一些富有多年本专业教学经验的专家来完成的，他们在制定教学计划时大多根据自己多年的教学经验以及以往的教学计划来编排课程的前后顺序。这种主要凭借主观经验的制定方法因缺少客观事实根据多少会发生一些偏差。通过关联规则在成绩管理数据库中的应用，可以挖掘出一些隐含其中的课程间相关联系，为高校的教学计划的编排及修订提供参考。这里以某高校信息与计算科学专业历届毕业生的所有成绩为数据源，挖掘该专业的课程间相关联系。

由L_1得到C_2，并扫描数据库得到各项集的支持度

C_2

项集	支持度	项集	支持度
I1、I2	4	I2、I5	2
I1、I3	4	I2、I6	3
I1、I5	3	I3、I5	2
I1、I6	1	I3、I6	1
I2、I3	4	I5、I6	1

与最小支持度阈值比较得到频繁项集L_2

L_2

项集	支持度
I1、I2	4
I1、I3	4
I1、I5	3
I2、I3	4
I2、I5	2
I2、I6	3
I3、I5	2

图 2　生成C_2，扫描得到L_2

表 2　Apriori 算法与 A++算法扫描数据库规模比较

K	Apriori 算法扫描数据库规模	A++算法扫描数据库规模
1	10	10
2	10	9
3	10	6
4	10	2

2.2 数据的预处理

在做数据挖掘之前本文对成绩管理数据库做了如下预处理：

(1) 成绩缺、多的处理。成绩的缺少主要是因为休学、退学等原因造成的。对于休学、退学的学生，只考虑他在休学、退学前的成绩。成绩的多出主要是因为补考、重新学习、重复选课等原因造成的。对于补考、重新学习的成绩，不予考虑，只考虑正常考试成绩；对于重复选课的，只考虑他第一次选修该课程的成绩。

(2) 数据的转换与简化。将以事务先后顺序形式存储的成绩数据库转换为电子表格的形式，即一个学生只占一条记录。删除除成绩字段以外的其它字段。

(3) 数据离散化。为适应本算法的运行，需将连续性数据离散成布尔型数据。为此本文做了两步工作：第一：根据统计规律中的正态分部原理将原来的绝对分数(X)转换为相对分数(Z)，具体转换方法因篇幅原因不再赘述；第二：确定“高分”标准离散数据，规定Z≥1离散为1，其余为0。

2.3 关联规则挖掘

通过上述对数据的预处理，最终得到了适合本算法的数据源，该数据源由184条学生记录40门课程成绩组成。设置最小支持度为7.5%，最小置信度为60%，用A++算法挖掘得到64条关联规则，现选取部分规则如表3所示。

2.4 结果评价

根据上述关联规则的挖掘，我们可以作如下解释：

从第1、2条规则可以看出，毕业论文写得好的同学毕业设计也做得好，同时毕业设计做得好的同学毕业论文也写得好，也就是说这两门课程具有相互促进的作用。

从第3、4条规则可以看出，汇编语言的学习有助于促进C语言程序设计的学习，与此同时C语言程序设计的学习又有助于计算方法课程的学习，可见这3门课程有一个前后呼应的关系。学生先通过低级程序语言的学习，学习一些简单的编程方法，进而转向高级程序语言的学习，学习更复杂的编程方法，最终抽象为纯数学计算方法的学习。从这两条规则可以看到教学计划中隐含着这样一条课程链。

表 3　部分关联规则

序号	规则	支持度(%)	置信度(%)
1	毕业论文=>毕业设计	14.7	93.1
2	毕业设计=>毕业论文	14.7	96.4
3	汇编语言=>C语言程序设计	8.2	65.2
4	C语言程序设计=>计算方法	8.2	62.5
5	汇编语言=>计算机原理	7.6	60.9
6	计算机原理=>计算方法	8.2	65.2

从第5、6条规则我们又找到了一条课程链，汇编语言 = > 计算机原理 = > 计算方法，更巧的是这条链的起点和终点和第3、4条规则的起点和终点相同，只是走的路径不同，第3、4条规则走的是偏软件的路线，第5、6条规则走的是偏硬件的路线。反映出这几门课程需要学生通过计算机软硬件两方面的学习才能全面学好。

这两条课程链组成了一个课程网。在得出的64条规则中还隐含着很多这种课程链，各课程链之间又组成了多个课程网。纵观64条规则，通过课程链和课程网之间的相互关系，可以大致了解隐含在成绩数据库中的该专业的课程结构。我们还可以重新调整最小支持度和最小置信度，得到更多的规则，找出更多的课程结构网络图，从而为高校教学计划的编排提供参考。

3 结束语

本文通过A + +算法提高了关联规则挖掘的运行效率，并且将该算法应用到成绩管理数据库的挖掘中，得到了课程之间的相关联系，为高校的教学计划的编排及修订提供了参考。该算法还可以被应用到高校教学管理的其它方面，从中挖掘出隐含其中的有用信息，为高校的教学管理服务。

参考文献

1 唐常杰、杨富华、杨璐，数据采掘的基本方法及其与专家系统的差异[J]，计算机应用，1999，19(3)：17 -20.

2 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases [C]. Proceedings of the ACM SIGMOD Conference on Management of Data. New York ACM, 1993:207 -216.

3 Agrawal R, SriKant R. Fast algorithms for mining association rules [C]. Proceedings of the 20th International Conference on Very Large Database. [s. l.]: Morgan Kaufman Pub Inc, 1994:487 -499.

4 Park J S, Chen M S, Yu P S. An effective hash - based algorithm for mining association rules [C]. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York ACM, 1995:175 -186.

5 Park J S, Chen M S, Yu P S. Efficient parallel data mining of association rules [C]. Proceedings of the 4th International Conference on Information and Knowledge Management New York ACM. 1995:31 -36.

6 Savasere A, Omiecinski E, Navathe S. An efficient algorithm for mining association rules in large databases [C]. Proceedings of the 21st International Conference on Very Large Database. New York ACM, 1995:432 -443.

7 Toivonen H. Sampling large databases for association rules[C]. Proceedings of the 22nd International Conference on Very Large Database. Bombay, India [s. n.], 1996:134 -145.

8 Brin S, Motwani R, Ullman J D, etc. Dynamic Itemset Counting and Implication Rules for Market Basked Data [C]. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York ACM, 1997:255 -264.

9 惠晓滨、张凤鸣、虞健飞，一种基于栈变换的高效关联规则算法[J]，计算机研究与发展，2003，40(2)：330 -335.

10 曾万聃、周绪波、戴勃、常桂然、李春平，关联规则挖掘的矩阵算法[J]，计算机工程，2006，32(02)：45 -47.

11 彭仪普、熊拥军，关联规则挖掘AprioriTid算法优化研究[J]，计算机工程，2006，32(05)：55 -57.

12 周艳山，数据挖掘中关联规则算法的研究及应用[D]，硕士，哈尔滨理工大学，2005.3：28.