基于规模压缩的关联规则数据挖掘算法研究

何 丽
(重庆师范大学数学与计算机科学学院 重庆 400047)

摘 要 基于关联规则的数据挖掘算法是人工智能和数据库研究的热点之一。本文提出的关联规则算法通过压缩规模,及时删除数据库中无用的事务记录,减少了事务数据的数量,提高了算法的执行效率。本算法能够生成较小规模的频繁候选集,有效减少了生成的候选集的规模,实现方便,在很大程度上也提高了效率。

关键词 关联规则,数据挖掘,规模压缩,算法

Research on Algorithms of Association Rules Data Mining Based on Reduction Scale

HE Li
(School of Mathematicas and Computer Science ,Chongqing Normal University, Chongqing 400047)

Abstract Data mining based on association rules is one of the most active and new research in the field of artificial intelligence and database. Association rules are an important aspect of research of data mining. The new algorithms produces a more small amount of candidate patterns when it looks for the frequent patterns of the data base, using theory that the Parent pattern of the non_frequent pattern is non_freqent pattern. So it can reduce the scale of the candidate_frequent patterns greatly. It can be easily implemented, and can improve the efficiency greatly by deleting those useless records.

Keywords Association rules, Data mining, Reduction scale, Algorithms

1 引言

数据挖掘(Data Mining)[1~3]是从大型数据库或数据仓库中提取人们感兴趣的知识,这些知识是隐含的、事先未知的潜在有用信息。数据挖掘是目前国际上数据库和信息决策领域的最前沿研究方向之一,引起了学术界和工业界的广泛关注。研究的主要目标是发展有关的方法论、理论和工具,以支持从大量数据中提取有用的和让人感兴趣的知识和模式。数据挖掘的主要技术有关联规则、遗传算法、人工神经网络、决策树、最近邻技术规则、归纳和可视化等。其中关联规则的提取是数据挖掘技术研究的一个重要课题,并广泛应用于商业、保险等行业。

关联规则[1,4~6]的算法相当多,Agrawal 等人提出了 AIS,AprioriAprioriTid 和 AprioriHybrid,Park 等人提出了 DHP,Sabasere 等人提出了 PARTITION 以及 Toivonen 提出了抽样算法 Sampling 等。其中最有效和最有影响的算法为 Apriori、DHP、PARTITION。另外还有很多改进算法,绝大部分是经典算法 Apriori 的演绎和改进。本论文提出的规模压缩算法也是在 Apriori 算法的基础上进行的改进,以提高运行速度、减少运行时间和空间耗费。

2 关联规则

关联规则是表示数据库中一组对象之间某种关联关系的规则,关联规则可以用数学模型描述如下:设 $I=\{i_1,i_2,\cdots,i_n\}$是 n 个不同项目的集合,数据库 DB 是针对 I 的交易集合,每一笔交易 T 包含若干项目,且 T 由被称为 TID 的标识符唯一标识。关联规则表示为蕴涵式 $X=>Y$,其中,$X\subseteq I$,$Y\subseteq I$,$X\cap Y=\Phi$。交易集 DB 中的规则 $X=>Y$ 由置信度 C(Confidence)和支持度 S(Support)约束。置信度 C 定义为:DB 中含有 X 中的交易的 C%也含有 Y。支持度 S 定义为:包含 $X\cup Y$ 的交易占 DB 的 S%。置信度表示蕴含式的强度,支持度表示在规则中出现该型式的频度,具有高置信度和高支持度的关联规则称为强关联规则。在实际研究中,满足一定的置信度和支持度的关联规则才有意义。为此需要定义两个阈值:最小置信度(记为 Min_Conf)和最小支持度(记为 Min_Sup)。如果项目集 $X\subseteq I$,$\sup(X=>Y)\geqslant$Min_Sup,则称 X 是频繁项集;如果 $\mathrm{conf}(X=>Y)\geqslant$Min_Conf,则称规则 $X=>Y$ 成立。关联规则的挖掘就是在事务数据库 DB 中找出具有用户给定的最小支持度 Min_Sup 和最小置信度 Min_Conf 的关联规则。

首先描述产生有代表性的关联规则的基本原理。

一个规则 $r:X\Rightarrow Y$ 的覆盖集,用符号表示为 $C(r)$,是指可以由规则 r 推导出来的关联规则集。也就是:$C(r:X\Rightarrow Y)=\{X\cup U\Rightarrow V|U,V\subseteq Y,U\cap V=\phi,V\neq\phi\}$

覆盖操作的一个重要属性:如果一个关联规则的支持度为 s、置信度为 c,那么每一个规则 $r'\in C(r)$ 的支持度至少为 s,置信度度至少为 c。覆盖运算符的这个性质表明它是关联规则的一个很好的推导运算符。

使用覆盖运算符,可以把有最小支持度 s、最小置信度 c 的有代表性的关联规则表示如下:$RAR(s,c)=\{r\in AR(s,c)|r'\in AR(s,c),r\neq r',$且 $r\in C(r')\}$。即,有代表性的关联规则是指覆盖了所有的关联规则并且有这组规则可以产生所有的关联规则的最小关联规则集。找到了有代表性的关联规则之后,用户就可以以一定的兴趣度和置信度查询由这组关联规

何 丽 讲师,研究方向:数据挖掘、计算机应用等。

则覆盖的规则。

假设规则 $X\Rightarrow Y$ 的长度是 $X\cup Y$ 的项的数目。RAR 具有下面的重要属性：

(1)假设 $r:X\Rightarrow Y$ 和 $r':X'\Rightarrow Y'$ 是两个不同的关联规则：

①如果 r 比 r' 长，那么 $r\notin C(r')$。

②如果 r 比 r' 短，那么 $r\in C(r')$ 当且仅当 $X\cup Y\subset X'\cup Y'$ 且 $X\supseteq X'$。

③如果 r 和 r' 的长度相同，那么 $r\in C(r')$ 当且仅当 $X\cup Y=X'\cup Y'$ 且 $X\supset X'$。

(2)假设 $r:X\Rightarrow Z\backslash X\in AR(s,c)$ 且 $\max\ Sup=\max(\{\sup(Z')|Z\subset Z'\subseteq X\}\cup\{0\})$，那么如果下面的两个条件成立，就有 $r\in RAR(s,c)$：

①$\max\ Sup<s$ 或 $\max\ Sup/sup(X)<c$。

②X'，$\phi\subset X'\subset X$ 使得 $X'\Rightarrow Z\backslash X'\in AR(s,c)$。

第一个条件保证 r 不在长度大于 r 的任意关联规则的覆盖集里，第二个条件保证长度等于 r 的任意关联规则的覆盖集里。

(3)假设 $\phi\neq X\subset Z\subset Z'\subseteq I$ 且 $\sup(Z)=\sup(Z')$，那么，没有规则 $r:X\Rightarrow Z\backslash X\in AR(s,c)$ 使得 $r\in RAR(s,c)$。

3 Apriori 算法描述

3.1 基于 Hash 表技术

利用 Hash 表技术可以帮助有效减少候选 k-项集 Ck($k>1$)所占用的空间。在扫描交易数据库以便从候选 1-项集 C1 中产生频繁 1-项集 L-时，就可以为每个交易记录产生所有的 2-项集并将它们 Hash 到 Hash 表的不同栏目中，且增加相应栏目的技术。如果 Hash 表中一个存放 2-项集的栏目技术低于最小支持频度，则表示相应 2-项集为非频繁项集而被移出候选集。利用这样 Hash 表技术可以帮助有效减少需要检查的候选项集。

3.2 数据划分

可以利用数据划分技术挖掘频繁项集而只需扫描整个数据库两次。这主要包括两个主要处理阶段。第一阶段，算法将交易数据库 DB 分为 n 个互不相交的部分；若数据库 DB 中的最小支持阈值为 min_sup，那么每个部分所对应的最小支持频度阈值为：min_sup×number_of_transaction_of_partition。

对于每个划分(部分)，挖掘其中所有的频繁项集；它们被称为是局部频繁项集，可以利用一个特别的数据结构记录包含这些频繁项集的交易记录的 TID，以便使得在一次数据库扫描中就能够发现所有的局部频繁 k-项集 $k=1,2,\cdots$。

就整个数据库 DB 而言，一个局部频繁项集不一定就是全局频繁项集；但是任何全局频繁项集一定会出现从所有划分所获得的这些局部频繁项集中。这一点很容易反证获得。因此可以将从 n 个划分中所挖掘出的局部频繁项集作为整个数据库 DB 中频繁项集的候选项集；而在第二阶段中再次扫描整个数据库以获得所有候选项集的支持频度，以便最终确定全局频繁的项集。

3.3 采样技术

所谓采样技术就是对给定数据集的一个子集进行挖掘。采样方法的核心是随机从数据集 DB 中采集 S 样本集，然后搜索 S 中频繁项集而不是 DB 中的。这样就以效率换取准确性。因此有可能漏掉一些全局频繁项集。为减少这种可能性，这里利用了一个比最小支持阈值要小的支持阈值来挖掘局部频繁项集。采样方法在对效率要求较高的应用场合是极具意义和重要的，尤其是在需要频繁进行这种密集计算的应用场合。

3.4 动态项集计数

动态项集计数是在扫描的不同时刻添加候选项集。动态项集计数是在对数据库进行划分挖掘时提出划分的。各数据块就被记上开始标志。在这一变化中，在任一开始点均可加入新的候选项集；在每次扫描数据库之前就已经决定了候选项集。这项技术是动态的，因为它是要估计至今所计数的所有项集的支持度；若一个项集的所有子集均被估计是频繁的，那就增加一个新的候选项集。这样所获得的算法需要进行两次扫描。

3.5 序列模式

序列模式就是发现随时间变化的交易序列(模式)。模式分析的目的就是挖掘以发生时间顺序为主的项集发生序列。这种研究在实际中较少。

4 压缩规模的关联规则算法

数据仓库的工作范围和成本常常是巨大的，而且时间消耗是很多的。信息技术部门必须针对所有的用户并以整个企业的眼光对待任何一次决策分析。这样就形成了代价很高、时间较长的大项目。采用规模压缩挖掘算法能及时删除数据库中无用的事务记录，减少了事务数据的数量，提高了算法的执行效率。

在关联规则系统中，规则本身是“如果条件怎么样、怎么样，那么结果或情况就怎么样”的简单形式，可以表示为“$A\Rightarrow B$”关联规则，它包括两个部分：左部 A 称为前件，右部 B 称为后件。前件可以包括一个或多个条件，在某个给定的正确率中。要使后件为真，前件中的所有条件必须同时为真。后件一般只包含一种情况，而不是多种情况。

Apriori 是一种广度优先算法，通过对数据库 DB 的多趟扫描来发现所有的频繁项目集，在每一趟扫描中只考虑具有同一长度 k(即项目集中所含项目的个数)的所有 k-项目集。在第一趟扫描中，Apriori 算法计算数据库 DB 中所有单个项目的覆盖度，生成所有长度为 1 的频繁项目集。在后续的每一趟扫描中，首先以前一趟中所发现的所有频繁项目集为基础，生成所有新的候选项目集(CandidateItemsets)，即潜在的频繁项目集，然后扫描数据库 DB，计算这些候选项目集的支持度，最后确定候选项目集中哪一些真正成为频繁项目集。重复上述过程直到再也发现不了新的频繁项目集。算法高效的关键在于生成较小的候选项目集，也就是尽可能不生成和计算那些不可能成为频繁项目集的候选项目集。它利用了这样一个基本性质：即一个频繁项目集的任一子集必定也是频繁项目集。

Apriori 算法就是根据有关频繁项集特性的先验知识(prior knowledge)而命名的。该算法利用了一个层次顺序搜索的循环方法来完成频繁项集的挖掘工作；这一循环方法就是利用 k-项集来产生(k+1)-项集。具体做法就是：首先找出频繁 1-项集，记为 L1；然后利用 L1 来挖掘 L2，即频繁 2-项集；不断如此循环下去直到无法发现更多的频繁 k-项集为止。每挖掘一层 Lk 就需要扫描整个数据库一遍。

在从数据库 DB 中挖掘出所有的频繁项集后，就可以较为容易获得相应的关联规则。也就是要产生满足最小支持度和最小信任度的强关联规则。

具体产生关联规则的操作说明如下：

(1)对于每个频繁项集 1，产生 l 的所有非空子集；

(2)对于每个 1 的非空子集 s，若 $\frac{\sup\ port\_count(1)}{\sup\ potr\_count(s)} \geqslant$ min_$conf$；则产生一个关联规则"$S \Rightarrow (l-S)$；其中 min_conf 为最小置信度阈值。

由于规则是通过频繁项集直接产生的，因此关联规则所涉及的所有项集均满足最小覆盖度阈值。

4.1 规模压缩算法

Apriori 是一种宽度优先算法，通过对数据库 DB 的多趟扫描来发现所有的频繁项目集。由于数据库 DB 一般是海量存储，因此对数据库 DB 多趟扫描将耗费大量时间和内存空间。随着数据库海量增大，这也就成为 Apriori 算法的瓶颈。规模压缩算法基于两种情况：

(1)对于已知规模的事务数据库 DB，任意一个项集 I 的出现频繁度与规模小于 $|I|$ 的事务无关，所以可以删除规模小于 $|I|$ 的事务记录。

(2)由于不包含任何 k-项集的事务不可能包含任何一个(K+1)-频繁项集，因此在生成(K+1)-频繁项集之前对这样的事务记录进行删除操作，以便来减少下次扫描事务数据库的记录数。考察在关联规则中的 JOIN(连接操作)运算，对于由(k−1)项集生成 K 项集，则仅对最后一位进行表记录添加和表记录删除操作，产生新的候选频繁项，并及时对表中项进行覆盖度判断。

在频繁数据项产生时，对于不能满足的项马上删除，而不是在生成数据项后进行统计、删除，从而减少了空间耗费。用 DB 表示数据库，用 $|T|$ 表示数据库中事务 T 的规模，即 T 中项的个数。用 $|d|$ 表示项集 d 的长度，其过程描述如下：

(1)初始化，扫描事务数据库，从中找出所有的项集长度为 1 的频繁度，形成原始的频繁项集，生成临时表。

(2)对(k−1)-频繁项集进行连接操作，以生成 k-频繁项集以(k−1)-频繁项集为基础，(k−1)-项集的其它项集的最后一项依次添加进临时表和删除，进行判断，产生 k-频繁项集。

①把生成的候选项集中(长度为 k−1)第一个项集添加进临时表中。

②把项集中长度为 k1 且最后一项与添加进临时表中的项集最后一项不同的项添加进临时表中。生成了 k-项集，并计算其频繁度。若频繁度大于最小频繁度，则生成了第一个频繁项，保存该频繁项。

③删除临时表中最后一个记录。

④到下一个记录，若是到(k−1)-项集最后一个则转到⑤，否则转②。

⑤删除临时表中全部记录，把下一个项集添加到临时表中，转①。

⑥所有符合条件的项集生成。

(3)删除操作。删除事务数据库中所有的项集长度小于 k+1 的事务，以及删除事务数据库中所有的不包含任何 k-频繁项集的事务。

(4)依次重复执行(2)，直到表记录终结。

4.2 算法执行

Apriori 算法执行过程：

(1)算法的第一遍循环，数据库中每个(数据)项均为候选 1-项集 C1 中的元素。算法扫描一遍数据库 DB 以确定 C1 中各元素的覆盖频度。

(2)假设最小覆盖频度为 L2。这样就可以确定频繁 1-项集 L1。它由候选 1-项集 C1 中的元素组成。

(3)为发现频繁 2-项集 L2，算法利用 L1⊕L1 来产生一个候选 2-项集 C2，接下来就扫描数据库 DB，以获得候选 2-项集 C2 中各元素的覆盖度。

(4)由此可以确定频繁 2-项集 L2 的内容。它是由候选 2-项集 C2 中覆盖度不小于最小覆盖度的项组成。

(5)L2⊕L2 进行连接操作，获得的候选 3-项集 C3，为{(I1,I2,I3},(I1,I2,I5),(I1,I3,I5),(I2,I3,I4),(I2,I3,I5),(I2,I4,I5)}。根据 Apriori 性质"一个频繁项集的所有子集也是频繁的"。由此可以确定后四个项集不可能是频繁的，因此删除它们，从而减少了扫描次数。扫描数据库 DB，获得 C3 中各元素的覆盖度。

(6)得到 L3 后进行连接操作得到 C4，根据 Apriori 性质，可得出 C4=Φ。至此算法结束。

由此可见，Apriori 算法每次扫描都要彻底扫描整个原始数据库 DB，这要耗费大量的时间；而且进行连接操作而生成完整的候选项集后才进行覆盖度的计算。

结论 改进的 Apriori 算法效率较高并容易实现，与现在已有的 Apriori 算法相比较，大大减少了算法须对数据库的遍历次数和扫描记录数。

Apriori 算法的时间复杂度为 $O(\|C\| \times |DB|)$，$\|C\|$ 为潜在频繁项集的长度之和，$|DB|$ 为事务数据库规模。用 H_i 来表示第 i 次生成频繁项的长度，则 $\|C\| = \sum_{i=1}^{n} H_i$。

参 考 文 献

1 Pokajac D, Obradovic Z. Improved Spatial-Temporal Forecasting through Modeling of Spatial Residuals in Recent History. In: Proc. First SIAM Int'I Conf. on Data Mining SDM 2001. Chicago. USA, 2001

2 Roddick J F, Hornsby K, Spiliopoulou M. An Updated Temporal Soatial and Spatio-Temporal Data Mining and Knowledge Discovery Research Bibliography. In: Post-Workshop Proceedings of the International Workshop on Temporal. Spatial and Spatio-Temporal Data Mining TSDM2000 Springer Lecture Notes in Artificial Intelligence, 2001

3 Shekhar S, Huang Y, Wu W, Lu C T, Chawla S. What's Spatial About Spatial Data Mining: Three Case Studies. in Data Mining for Scientific Engineering Applications. In: Kumar V, Grossman R, Kamath C, Nambaru K, eds. Kluwer Academic, 2001

4 Ordonez C, Omiecinski E. Discovery association rules based on image content. In: Proceedings of the 1999 IEEE Forum on Research and Technology Advances in Digital Libraries, Baltimore, MD, May 1999. 38～49

5 Bloch, Isabelle. Fuzzy relative position between objects in image processing: A morphological approach. IEEE Transactions on Patten Analysis and Machine Intelligence. 1999, 21(7): 657～664

6 Han J, Koperski K, Stefanovic N. GeoMiner. A System Prototype for Spatial Data Mining[C]. In: Proc. ACM SIGMOD Conference on the Management of Data Tucson Arizona, 1997