中国农业大学学报 2007,12(3):85-88
Journal of China Agricultural University

一种新的关联规则抽样算法

秦如新 陈 静 冯一宁
(中国农业大学 理学院,北京 100083)

摘 要 针对目前经典的关联规则挖掘 Apriori 算法需对数据库多次扫描费时多计算量大,而抽样扫描会造成挖掘精确度下降等问题,采用控制样本频繁项目集的方法,利用频繁 1 项集进行抽样处理,对关联规则挖掘的抽样操作和精度控制进行研究,提出了基于抽样操作的关联规则挖掘算法——HAC 算法。理论分析及性能试验结果表明:HAC 算法能够有效缩减数据库规模,至少少扫描数据库 1 次,提高了关联规则挖掘的效率,同时其计算精度不受影响。

关键词 关联规则;抽样;准则系数;Apriori 算法;HAC 算法
中图分类号 TP 311 **文章编号** 1007-4333(2007)03-0085-04 **文献标识码** A

A new sampling algorithm for association rule

Qin Ruxin, Chen Jing, Feng Yining
(College of Science, China Agricultural University, Beijing 100083, China)

Abstract In order to reduce the long time spent for scanning the database by using Apriori algorithm, which may descend the mining accuracy, the research on the sample operation and precision control with the help of frequent itemset, especially, the frequent 1-itemset is presented in this paper. The HAC algorithm based on sampling was designed. The results in theory and capability experiment indicated that HAC algorithm could decrease the scanning times by at least once, promote the efficiency of mining and improve the computation precision.

Key words association rule; sampling; guide coefficient; Apriori algorithm; HAC algorithm

关联规则挖掘就是从大量的数据中挖掘出描述数据之间相互联系的有价值的知识,是数据挖掘的重要内容。关联规则挖掘需要对数据库进行扫描,当数据库较大时,扫描过程代价昂贵,挖掘关联规则的效率通常很低,包括最经典的 Apriori 算法[1-3]。为了提高挖掘效率通常采用缩减交易数据库的方法进行挖掘,即在一个较小的交易数据库中挖掘关联规则。通常的缩减数据库方法是在整个数据库中随机抽样得到一个较小的交易数据库,但在小交易数据库中得到的关联规则质量无法保证。

对于关联规则的抽样挖掘问题,国外研究[4-6]并未给出有精度保证的可操作的抽样挖掘算法。文献[4]中提出的 FAST 算法是一种基于抽样的关联规则挖掘方法,该算法在剪切边缘交易的基础上对数据库进行随机抽样得到样本,然后在样本上进行关联规则挖掘。FAST 算法能够在一定程度上保证关联规则的质量,但在抽样过程中需要求解一系列实际上是无法解决的优化问题(NP 难问题)。EA 算法[5-6]利用染色减半的抽样方法对数据库进行处理,能够在给定样本大小的情况下得到较 FAST 算法更为精确的结果,但是运行成本比 FAST 算法还要高。这些算法虽有精度保证,但都需要解决一系列 NP 难问题,应用在大型数据库中效率不高。近年来,国内对抽样关联规则进行了研究[7-11],且都是直接将抽样技术应用到关联规则挖掘中,而没有给出控制样本大小的标准,得到的关联规则质量也不能保证。

笔者提出基于抽样操作的关联规则挖掘算法

收稿日期:2006-10-25
基金项目:国家自然科学基金资助项目(10371131;60573158)
作者简介:秦如新,博士研究生,E-mail:qinruxin99@163.com;陈静,副教授,通讯作者,主要从事最优化方法、支持向量机研究,E-mail:jing_quchen@163.com

——层次二分算法(HAC),利用频繁1项集对数据库进行抽样,以期提高挖掘算法的效率,使其具有可操作性的同时精度不受影响。

1 关联规则

设 $I=\{i_1, i_2, \cdots, i_m\}$ 是项目的集合。设事务数据 D 是事务的集合,其中每个事务 T 是项目的集合,使得 $T \subseteq I$。每个事务都有1个标识符,称为TID。设 A 是一个项目集,事务 T 包含 A 当且仅当 $A \subseteq I$。关联规则可形式化表示为 $A \Rightarrow B$ 的蕴涵式,其中 $A \subset I, B \subset I$ 并且 $A \cap B = \phi$。关联规则 $A \Rightarrow B$ 在事务集 D 中出现,具有支持度 s,其中 s 是 D 中事务包含 $A \cup B$ 的比例,概率为 $P(A \cup B)$。关联规则 $A \Rightarrow B$ 在事务集 D 中具有置信度 c,如果 D 中包含 A 的事务的同时也包含 B 的比例为 c,这实际上是条件概率 $P(B|A)$,也就是,$\text{support}(A \Rightarrow B) = P(A \cup B)$,$\text{confidence}(A \Rightarrow B) = P(B|A)$。

要挖掘有效的关联规则,必须给定最小支持度($s_{\min}$)和最小置信度($c_{\min}$)。关联规则的挖掘问题就是在 D 中寻找所有支持度和置信度超过 $s_{\min}$ 和 $c_{\min}$ 的关联规则,即要寻找满足 $\text{support}(A \Rightarrow B) \geq s_{\min}$ 和 $\text{confidence}(A \Rightarrow B) \geq c_{\min}$ 的规则 $A \Rightarrow B$。因此,挖掘有效的关联规则可分为2个子问题:

1)事务数据库中所有频繁项目集的挖掘;

2)频繁项目集中所有大于最小置信度的关联规则的获得。

相对于1)来说,2)比较容易,目前大多数研究主要集中在1)。关联规则描述虽然简单,但计算量很大。假设数据库含 m 个项目,就有 2^m 个子集可能是频繁子集,可以证明要找出的某一频繁项目集是一个 NP 难问题。当 m 较大时,要穷尽搜索每个子集几乎是不可能的。同时,处理数据库中存储的大量记录要求繁重的磁盘 I/O 操作,随着数据库规模的不断增大,数据属性向高维发展,对数据库直接进行挖掘很难适应大规模、可扩展(scalability)的挖掘需要。因此,可以用抽样方法缩小数据库的规模。

2 Apriori 算法

挖掘关联规则最经典的算法是 Apriori 算法,该算法通过连接和剪枝实现频繁项集挖掘。利用频繁项集性质的先验知识,用逐层搜索的迭代方法获得频繁项目集,其中 k 项集用于探索 $k+1$ 项集。首先找到频繁1项集,记为 L_1,L_1 用于找频繁2项集 L_2,如此下去,直到不能找到频繁 k 项集。

Apriori 算法具有如下性质:频繁项集的所有非空子集都必须是频繁的。这是因为根据定义,假设项集 I 不满足最小支持度 $s_{\min}$,则不是频繁的,如果把项 A 添加到 I,则结果项集($I \cup A$)不可能比 I 更频繁出现。因此,结果项集也不是频繁的。利用 Apriori 算法的性质,通过连接和剪枝可实现频繁项集的挖掘:

1)连接:对 L_{k-1} 中的每个元素执行连接,得到了 L_k 的候选集合 C_k;

2)剪枝:C_k 是 L_k 的超集,即它的成员中也有不是频繁的。首先,根据 Apriori 性质,缩小 C_k 的范围,然后扫描数据库,确定 C_k 中每个候选的计数,从而确定 L_k。下面给出 Apriori 算法的伪代码:

```
输入:事务数据库 D;最小支持度 s_min
输出:D 中的频繁项集 L
L_1 = {frequent 1 - itemset};
For (k = 2; L_{k-1} ≠ φ; k + +) do begin
  C_k = apriori-gen; L_{k-1} //生成所有长度为 k 的候选集
  For each transactions t ∈D
    C_t = subset(C_i, t); //t 中包含的候选集
    For each candidates c ∈C_i, do
    c. count + +;
  end
    L_k = {c ∈C_k | c. count > min sup port}
end
```

Apriori 算法是最经典的关联规则算法,但计算效率不高。寻找每一个频繁 k 项集都需要对数据库扫描1次,共需要扫描 k 次。因此,当数据库或者 k 太大时,算法耗时太多以至于无法完成。

3 层次二分算法(HAC)

本研究中抽样关联规则挖掘方法称为层次二分算法:在满足一定条件的情况下对得到的样本数据库进行抽样得到新样本,使得新样本的势(元素个数)是原来样本势的1/2,迭代产生最后的样本,并在此样本上利用 Apriori 算法挖掘关联规则。根据挖掘关联规则的目的,每次抽样得到的理想结果应该是,对交易数据库平均划分的同时频繁项目集也得到平均划分。因为所有频繁项目集都是由频繁1项目集生成的,只要频繁1项目集能够在样本中得到平均划分就足够了。这种理想结果在随机抽样条

件下是很难达到的。本研究根据频繁 1 项目集进行抽样,并在一定程度上保证现有样本与新样本在频繁 1 项目集上的一致性,按照精度要求给出每次抽样的实施方案(下面算法中 4))。本研究的处理方法远比随机抽样方法更有依据,并且在每次抽样缩减后给出一个准则系数,准则系数确定算法的终止准则。准则系数正是利用频繁 1 项目集构造的。下面先给出算法,然后给出准则系数。为叙述方便将用 D^j 和 L_k^j 分别表示第 j 次抽样后的样本和频繁 k 项集。

层次二分算法:

1) 扫描交易数据库 D 得到 L_1^0,记 $L_1^0 = \{i_1, i_2, \cdots, i_p\}$,按照其支持度递减排列 L_1^0 中的项目,仍记为 $\{i_1, i_2, \cdots, i_p\}$。

2) 第 1 次抽样后得 D^1,使得扫描 $|D^1| = \frac{|D|}{2}$,D^1 按 1) 中的支持度计算得 L_1^1,计算准则系数 e。若 $e > e_0$,则对 D^1 进行第 2 次抽样;否则,仍然使用 D 进行关联规则挖掘。

3) 第 j 次抽样后得 D^j,使得扫描 $|D^j| = \frac{|D^{j-1}|}{2}$,对 D^j 按 1) 的支持度计算得 L_1^j,计算准则系数 e。若 $e > e_0$,则对 D^j 进行第 $j+1$ 次抽样;否则,用 D^j 进行关联规则挖掘。

4) 第 j 次抽样:在 D^{j-1} 中,记 $L_1^j = \{i_1^{j-1}, i_2^{j-1}, \cdots, i_l^{j-1}\}$,按照支持度递减排列后仍记为 $i_1^{j-1}, i_2^{j-1}, \cdots, i_l^{j-1}$。将 D^{j-1} 中含有 i_1^{j-1} 的交易集合记为 E_1^j,在 E_1^j 中随机抽取势为 $\frac{|E_1^j|}{2}$ 的样本记为 F_1^j,记 $D^{j-1}/E_1^j = D_1^j, \cdots$,同理在 D_{q-1}^j 中利用 i_1^{j-1} 得 F_q^j,其中 $D_{q-1}^j = D^{j-1} \setminus E_1^j \setminus E_2^j \setminus \cdots \setminus E_{q-1}^j$。令 $D_q^j = D^{j-1} \setminus E_1^j \setminus E_2^j \setminus \cdots \setminus E_q^j$,在 D_q^j 中随机抽取样本 R_q^j,使得 $|R_q^j| = \frac{|D_q^j|}{2}$,则 $D^j = F_1^j + F_2^j + \cdots + F_l^j + R_q^j$,其中 $q < 1$。

5) 在最后的样本 D^0 上利用 Apriori 算法挖掘关联规则。

2) 和 3) 给出抽样处理的整个过程,在准则系数一直满足的条件下可以连续抽样处理,使得交易数据库规模不断降低,最后得到一个较小的数据库,在此基础上可以使用任何方法进行关联规则挖掘。4) 给出了每一次抽样的具体操作方法,在 1 次抽样中,首先将 D^{j-1} 中含有最频繁的 i_1^{j-1} 的集合 E_1^j 随机抽取 1 个集合 F_1^j,其样本势为集合 E_1^j 的样本势的 1/2,也就是优先保证最频繁的项 i_1^{j-1} 能够在抽样中得到平均划分;然后考虑 E_1^j 剩余部分使得 i_2^{j-1} 能够在抽样中尽量平均划分,依此类推。迭代次数越多即 q 越大时,结果就越准确。当然在效率要求较高时 q 取较小的值。当 $q = 1$ 时,只是利用 i_1^{j-1} 进行抽样。

HAC 算法中的准则系数定义为:假设第 $j-1$ 次抽样得到 D^{j-1} 的频繁 1 项目集 $L_1^{j-1} = \{i_1, i_2, \cdots, i_p\}$,第 j 次抽样得 D^j 的频繁 1 项目集 $L_j^j = \{i_{q_1}, i_{q_2}, \cdots, i_{q_s}\}$,则 $e = \frac{|L_1^j \cap L_1^{j-1}|}{|L_1^{j-1}|}$ 称为准则系数。

准则系数表征样本的频繁 1 项目集与原数据库的比例,$e = 1$ 表示样本能完全代替原数据库,这是抽样的理想化结果,通常准则系数小于 1。

该算法首先需要给出准则系数的阈值 e_0,在每次抽样结束后都要计算准则系数 e 并验证是否满足 $e > e_0$。

4　HAC 算法的有效性

数据库抽样操作的合理性和有效性是毫无疑问的[1,4,7-11]。抽样是大型数据库操作最常用的方法,只要在数据库中进行了抽样操作,相同条件下的数据挖掘都会提高速度。本研究的 HAC 算法同样能提高运算速度。理论分析如下:

在大型数据库中进行关联规则挖掘时,对数据库的扫描占用了大部分时间,因此扫描数据库的次数就是评价关联规则挖掘算法效率的主要指标。

当对数据库直接进行关联规则挖掘时,Apriori 算法得到频繁 k 项目集需要进行 $k+1$ 次扫描。当采用 HAC 算法时,扫描数据库次数的上界为 $3 + (k+1)/2$。显然,$k \geq 5$ 时,有 $3 + (k+1)/2 \leq (k+1)$,也就是 HAC 算法至少比 Apriori 算法少扫描数据库 $k+1-3-(k+1)/2 = (k+1)/2 - 3$ 次,而在巨型数据库中通常 $k > 5$。所以,在巨型数据库中应用 HAC 算法将有效的减少对数据库的扫描次数,即使在 $q = 1$ 的情况下,采用 HAC 算法也能够减少扫描数据库次数。而相对于巨型数据库中采用随机抽样的方法,HAC 算法无疑将极大地提高关联规则的挖掘效率。

本研究采用合成数据对 HAC 算法与 Apriori 算法的精度进行比较,获得合成数据的方法与文献[12]类似。虽然抽样方法适合于对巨大数据库的操

作，但方便起见本研究的合成数据包括10 000个项目；最小支持度设为2%；参数 e_0 的取值分别为0.1，0.2，…，1.0，在这10个参数下，分别比较HAC算法与Apriori算法关于频繁7项集的差异度 d，$d = \mathrm{card}(H \cap A)/\mathrm{card}(A)$，其中 $\mathrm{card}(H \cap A)$ 为HAC算法与Apriori算法分别得到的频繁7项集的交集的势，$\mathrm{card}(A)$ 是由HAC算法得到的频繁7项集的势，试验结果见图1。当频繁项目集的项数小于6时，HAC算法较Apriori算法并无效率优势。可以看出，当 $e_0 = 0.9$ 或1.0时，HAC算法与Apriori算法得到的频繁7项集是相同的，这是因为在实际运算中，抽样操作并未进行。当 e_0 为0.5～0.8时，可以看到HAC算法与Apriori算法得到的频繁7项集的差异是有限的，特别当 $e_0 = 0.8$ 时。但是HAC算法至少少扫描数据库1次，也就是说，在效率提高的基础上计算结果的精度并没有降低。

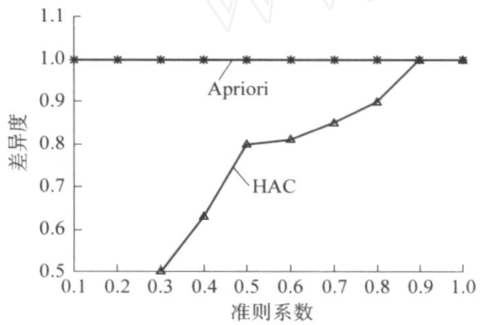

图1 HAC算法与Apriori算法的频繁7项集差异度比较

Fig. 1 Comparison of the frequent 7-item-set discrepancy between HAC and Apriori algorithms

5 结束语

本研究提出了一种有精度保证的关联规则抽样方法——HAC算法，用户可以根据需要，通过选择迭代次数 q 值权衡效率和准确性。理论分析表明，与Apriori算法相比，HAC算法扫描数据库的次数明显减少，对于巨型数据库其挖掘效率有较大的提高。与文献[4]-[6]的抽样算法相比，HAC算法无需解决NP难问题。与文献[7-11]的抽样算法相比，HAC算法可以根据精度要求确定样本大小。HAC算法中阈值 e_0 的选取将影响到算法的迭代次数和关联规则的精度，所以阈值 e_0 的选择有待进一步研究。

参 考 文 献

[1] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large database[C]//In Proceeding of the ACM SIGMOD International Conference on Management of Data. Washington: ACM Press NY, 1993: 207-216

[2] Tan Pangning, Steinbach M, Kumar V. 数据挖掘导论[M]. 范明，范宏建，译. 北京：人民邮电出版社，2006:241-327

[3] 林杰斌，刘明德，陈湘. 数据挖掘与OLAP理论与实务[M]. 北京：清华大学出版社，2003：172-243

[4] Chen Bin, Haas P, Sdcheuemann P. A new two-phase sampling based algorithm for discovering association rules[C]//Proceeding of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Andreas:Association for Computing Machinery Press Room, 2002:462-468

[5] Chazelle B. The discrepancy method[M]. Cambridge United Kingdom: Cambridge University Press, 2000: 340-392

[6] Broennimann H, Chen Bin, Dash M, et al. Efficient data-reduction methods for on-line association rule discovery[C]//National Science Foundation Workshop on Next Generation Data Mining. Camden Yards: Baltimore Marriott Press, 2002:190-208

[7] 王星. 关联规则的序贯抽样算法比较研究[J]. 计算机工程与应用，2005，41(1):27-30

[8] 李宏，陈松乔，杜剑峰，等. 基于抽样的分布式约束性关联规则挖掘算法研究[J]. 计算机科学，2006，33(7):190-195

[9] 陆如松，闪四清. 基于抽样策略的关联规则挖掘算法[J]. 大众科技，2006(2):52-53

[10] 王星. 对一个关联规则序贯算法的改进与效率分析[J]. 统计与决策，2005(6):8-10

[11] 李梅花，王黎明，许红涛. 利用抽样技术和元学习的分布式关联规则挖掘算法[J]. 计算机应用，2006，26(4):872-877

[12] Agrawal R, Srikant S. Fast algorithms for mining association rules[C]//Proceedings of the 20th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Press, 1994: 487-499