正负关联规则挖掘算法研究*)

朱玉全[1] 陈 耿[2] 杨鹤标[1]

(江苏大学计算机科学与通信工程学院 镇江 212013)[1] (东南大学计算机科学与工程系 南京 210096)[2]

摘 要 本文提出了一种快速有效的正、负关联规则挖掘算法 MPNAR。另外,针对关联规则挖掘算法中支持数计算的复杂性,提出了一种基于二进制形式的支持数计算方法。实验结果表明算法 MPNAR 是有效和可行的。

关键词 数据挖掘,频繁项目集,负关联规则

Research on Algorithm for Mining Positive and Negative Association Rules

ZHU Yu-Quan[1] CHEN Geng[2] YANG He-Biao[1]

(School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013)[1]

(Department of Computer and Engineering, Southeast University, Nanjing 210096)[2]

Abstract In this paper, a fast and efficient algorithm MPNAR is presented to discover positive and negative association rules. Meanwhile, a method based on binary format is proposed to calculate the support of candidate ferquent itemsets. The experiments show that algorithm MPNAR is efficient and feasible.

Keywords Data mining, Frequent itemsets, Negative association rules

1 引言

自从 Rakesh Agrawal[1] 等人首次提出关联规则挖掘这个研究课题以来,已提出了许多相应的算法[2~8],这些算法仅能用来发现强模式,即那些具有高频率和强相关的显式模式。实际上数据库中还存在许多采用这些挖掘技术所不能发现的隐式模式,这些重要的隐式模式之一便是负关联规则,它具有低频率、强相关的性质,表现了数据项目集间的不易直接觉察到的强相关性质,这种隐式规则告诉我们哪些数据项目较少地一起发生,但它们之间有着相当强的相关性,包含了非常有价值的信息,因而发现负关联规则具有十分重要的意义。负关联规则的一个简单应用例子是超市中的商品摆设问题,显然,对于两个高频率商品 X 和 Y,如果 X 很少跟 Y 同时发生,Y 就应尽量摆放在远离 X 的货架上。

目前可用于负关联规则挖掘的算法并不多,仅有的几种算法也存在一定的局限性。如:文[9]提出了一种新的负关联规则挖掘思想,该算法需要用户事先给定一个定理好的层次分类结构,其实这是很难做到的,也是不切合实际的;文[10]提出了一种基于兴趣度的正、负关联规则挖掘算法,该算法在生成候选项目集时采用了类 Apriori 算法,这样可能会丢失十分有用的非频繁项目集;文[11]提出了一种基于支持度、置信度、关联系数的正、负关联规则挖掘算法,算法的执行过程中需要不断地调整关联系数,并且算法本身也不能发现所有负关联规则。针对如上不足,本文提出了一种快速有效的正、负关联规则挖掘算法 MPNAR(Mining Positive and Negative Association Rules),该算法可以发现事务数据库 D 中所有正负关联规则。另外,针对 Aprior 算法中支持数计算的复杂性,本文提出了一种基于二进制形式的支持数计算方法,该方法只需进行一些"或"逻辑运算操作,将该方法用于正、负关联规则挖掘中支持度的计算,可以进一步提高算法的执行效率。

2 问题描述

设 $I=\{i_1, i_2, \cdots, i_m\}$ 是 m 个不同项目的集合。给定一个事务数据库 D,其中每一个事务 T 是 I 中一组项目的集合,即 $T\subseteq I$。如果对于 I 中的一个子集 X,有 $X\subseteq T$,我们就说一个事务 T 包含 X。一条负关联规则(negative association rule)就是一个形如 $\neg X\Rightarrow Y$(或 $X\Rightarrow\neg Y$,或 $\neg X\Rightarrow\neg Y$)的蕴涵式,其中 $X, Y\subseteq T$,而且 $X\cap Y=\Phi$。

给定支持度 s 和置信度 c。如果 D 中有 $(100\times s)\%$ 的事务包含 Y 但不包含 X,则负关联规则 $\neg X\Rightarrow Y$ 的支持度为 s,记 $\mathrm{supp}(\neg XY)=s$。如果不包含 X 的事务中有 $(100\times c)\%$ 的事务包含 Y,则负关联规则 $\neg X\Rightarrow Y$ 在 D 中的置信度为 c,记 $\mathrm{conf}(\neg XY)=c$。同样可以定义负关联规则 $X\Rightarrow\neg Y$、$\neg X\Rightarrow\neg Y$ 的支持度和置信度。

负关联规则挖掘就是在 D 中筛选出所有具有用户指定的最小支持度 minsup 和最小置信度 minconf 的负关联规则 $\neg X\Rightarrow Y$(或 $X\Rightarrow\neg Y$,或 $\neg X\Rightarrow\neg Y$),即这些负关联规则的支持度和置信度分别不小于最小支持度 minsup 和最小置信度 minconf,且 X 和 Y 均为频繁项目集。

由于篇幅所限,本文仅讨论负关联规则 $\neg X\Rightarrow Y$ 的挖掘问题,其基本思想亦适用于负关联规则 $X\Rightarrow\neg Y$ 的挖掘。另外,由于负关联规则 $\neg X\Rightarrow\neg Y$ 的挖掘比较简单,在此就不作讨论了。

3 负关联规则的挖掘算法

对于负关联规则 $\neg X\Rightarrow Y$ 挖掘而言,由于 $X\cup Y$ 是非频繁项目集,而数据库 D 中的非频繁项目集的数量是指数级

*)本文得到国家自然科学基金(60572112)和江苏大学科研启动基金(04KJD001)资助。朱玉全 博士,副教授,主要研究方向为复杂信息系统集成、知识发现和入侵检测等。

的，因此负关联规则的搜索空间比正关联规则的搜索空间要大得多，例如，假设某百货商场有1000种商品，事务数据库D中频繁项目集的个数大约为2^{50}，则D中的非频繁项目集个数高达($2^{1000}-2^{50}$)，接近于2^{1000}，因此想要求出所有这些非频繁项目集的支持数是不现实的，也是无意义的。其实，对于此问题，尽管$X\cup Y$是非频繁项目集，但X和Y均为频繁项目集，因此，我们可以将负关联规则挖掘分解成以下两个子问题：① 求出事务数据库D中所有的频繁项目集；② 根据第一步所得的频繁项目集确定所有负关联规则。对于第一个子问题，已有许多快速的算法可直接使用，如算法Apriori[1]、FP-growth[6]等，因此，第二个子问题是负关联规则挖掘的核心问题，我们将重点讨论此子问题。

定义1 候选负关联规则和候选非频繁项目集。如果$\neg X\Rightarrow Y$有可能成为负关联规则，则称$\neg X\Rightarrow Y$为候选负关联规则，$X\cup Y$为候选非频繁项目集。

定理1 设$\neg X\Rightarrow Y$为候选负关联规则，则其支持度$\mathrm{supp}(\neg XY)=\mathrm{supp}(Y)-\mathrm{supp}(X\cup Y)$，置信度$\mathrm{conf}(\neg XY)=(\mathrm{supp}(X)-\mathrm{supp}(X\cup Y))/(1-\mathrm{supp}(X))$。

证明：由于$\mathrm{supp}(Y)=\mathrm{supp}(\neg XY)+\mathrm{supp}(X\cup Y)$，因此$\mathrm{supp}(\neg XY)=\mathrm{supp}(Y)-\mathrm{supp}(X\cup Y)$。$\mathrm{conf}(\neg XY)=\mathrm{supp}(\neg XY)/\mathrm{supp}(\neg X)=(\mathrm{supp}(X)-\mathrm{supp}(X\cup Y))/(1-\mathrm{supp}(X))$。定理1得证。

定理1表明，确定负关联规则的关键技术是如何快速有效地计算候选非频繁项目集$X\cup Y$在事务数据库D中的支持数(度)。

定理2 设事务数据库D中的频繁项目集的个数为m，则候选负关联规则或候选非频繁项目集的个数远低于$m*m$。

证明：对于候选负关联规则$\neg X\Rightarrow Y$而言，X和Y均为频繁项目集，且$X\cap Y=\Phi$。由于事务数据库D中的频繁项目集的个数为m，因此候选负关联规则的前件最多有m种可能，因为$X\cap Y=\Phi$，所以后件的可能数远低于m。故候选负关联规则或候选非频繁项目集的个数远低于$m*m$。定理2得证。

定理2表明，尽管数据库D中的非频繁项目集的数量是指数级的，但候选负关联规则或候选非频繁项目集的个数还是较少的，在最坏情况下其数量级是平方级，而在算法Apriori中候选频繁项目集个数至少为m，一般情况下，其数量级亦达到平方级，甚至更高，因此求出所有负关联规则是可行的。如：某百货商场有1000种商品，事务数据库D中频繁项目集的个数大约为2^{50}，则D中的候选负关联规则或候选非频繁项目集的个数远低于$2^{50}*2^{50}=2^{100}$，仅占整个项目集的很小一部分($2^{100}/2^{1000}=1/2^{900}$)。

定理3 设$X\Rightarrow\neg Y$为负关联规则，对于任意频繁项目集$Z\supseteq Y$，则$X\Rightarrow\neg Z$为负关联规则。

证明：由于$Z\supseteq Y$，因此$\mathrm{supp}(Z)\leqslant\mathrm{supp}(Y)$，$\mathrm{supp}(X\cup Z)\leqslant\mathrm{supp}(X\cup Y)$。因为$X\Rightarrow\neg Y$为负关联规则，所以$\mathrm{supp}(X\neg Y)\geqslant\mathrm{minsup}$，$\mathrm{conf}(X\neg Y)\geqslant\mathrm{minconf}$，而$\mathrm{supp}(X\neg Z)=\mathrm{supp}(X)-\mathrm{supp}(X\cup Z)\geqslant\mathrm{supp}(X)-\mathrm{supp}(X\neg Y)=\mathrm{supp}(X\cup Y)\geqslant\mathrm{minsup}$，$\mathrm{conf}(X\neg Z)=\mathrm{supp}(X\neg Z)/\mathrm{supp}(X)=\mathrm{supp}(X)-\mathrm{supp}(X\cup Z))/\mathrm{supp}(X)\geqslant\mathrm{supp}(X)-\mathrm{supp}(X\cup Y))/\mathrm{supp}(X)=\mathrm{conf}(X\neg Y)\geqslant\mathrm{minconf}$，即$\mathrm{supp}(X\neg Z)\geqslant\mathrm{minsup}$，$\mathrm{conf}(X\neg Z)\geqslant\mathrm{minconf}$，故$X\Rightarrow\neg Z$为负关联规则。定理3得证。

定理3表明，如果$X\Rightarrow\neg Y$为负关联规则，那么就没有必要计算$X\cup Z(Z\supseteq Y)$的支持数，这样可以删除很多候选非频繁项目集。

由于在负关联规则挖掘算法中，每次循环所产生的候选项目集的维数是不一样的，因此如何计算项目集的支持数是负关联规则挖掘算法中的一个非常重要的问题。下面将提出一种全新的项目集支持度计算方法，该方法只需进行一些"或"逻辑运算操作，显著降低了算法的执行时间。

定义2 给定$I=\{i_1,i_2,\cdots,i_m\}$，事务数据库D，事务$T\in D$(项目集X)，且$T\subseteq I(X\subseteq I)$。我们称$f:T\times I(X\times I)\to b_1b_2\cdots b_{m-1}b_m$为事务-二进制数关系(项目集-二进制数关系)，称$b=b_1b_2\cdots b_{m-1}b_m$为事务$T$(项目集$X$)所对应的二进制数，记为$B_t(B_x)$，其中$b_j\in\{0,1\}$，且如果$i_j\in T(i_j\in X)$，则$b_j=1$，否则，$b_j=0$，$j=1,2,\cdots,m$。

例1 给定一事务数据库D，$I=\{1,2,3,4,5,6,7,8\}$，$T=\{1,3,4,5,7,8\}$，则事务T所对应的二进制数B_t为10111011。设某项目集$X=\{2,3,4,7,8\}$，则项目集X所对应的二进制数B_x为01110011。

定义3 二进制数-项目集关系。给定$I=\{i_1,i_2,\cdots,i_m\}$，事务数据库D，m位二进制数b。我们称$f:b\times I\to X_b$为二进制数-项目集关系，并称X_b为二进制数b所对应的项目集，其中，X_b为项目集，如果$b_j=1$，则$i_j\in X_b$，否则，$i_j\notin X_b$，$j=1,2,\cdots,m$。

例2 给定一事务数据库D，$I=\{1,2,3,4,5,6,7,8\}$，$b=01101101$，则二进制数b所对应的项目集$X_b=\{2,3,5,6,8\}$。

定理4 给定$I=\{i_1,i_2,\cdots,i_m\}$及m位二进制数b。如果b的第$j_1,j_2,\cdots,j_k$位位值为1，那么$X_b=\{i_{j1},i_{j2},\cdots,i_{jk}\}$。

证明：根据定义3，定理4显然成立。

定理5 给定$I=\{i_1,i_2,\cdots,i_m\}$，事务数据库D中的事务T，项目集c。如果B_t or $B_c=B_t$，则T支持c。反之亦然。

证明：假设B_t的第$j_1,j_2,\cdots,j_m$位位值为1，其它位位值为0，由于B_t or $B_c=B_t$，因此，B_c的第$j_1,j_2,\cdots,j_m$位位值为1或0，其余位位值为0(否则或值后的值为1)，根据定理4，$T\supseteq c$。反之，如果T支持c，显然有B_t or $B_c=B_t$成立。定理5得证。

综上所述，我们可以得到负关联规则挖掘算法。其中，候选频繁k-项目集生成算法类同于算法Apriori中的候选频繁k-项目集生成算法。设$AL_{k-2}=\{c\mid c\in\cup pL_p\ \text{and}\ p<k-1\}$($AL_0=\Phi$，$k\geqslant 2$)，NAS为负关联规则的集合，第$k$步的候选负关联规则或候选非频繁项目集$\neg X\Rightarrow Y(X\cup Y)$的形成方法如下：

(1)$X\in L_{k-1}$，$Y\in L_{k-1}$，$X\cap Y=\Phi$；

(2)$X\in AL_{k-2}$，$Y\in L_{k-1}$，$X\cap Y=\Phi$；

(3)$X\in L_{k-1}$，$Y\in AL_{k-2}$，$X\cap Y=\Phi$。

算法：负关联规则挖掘算法MPNAR。

输入：事务数据库D；最小支持度阈值minsup；最小置信度阈值minconf。

输出：事务数据库D中的频繁项目集L；事务数据库D中的负关联规则NAS。

方法：

(1)$L_1=\{\text{frequent 1-itemsets}\}$；//生成频繁1-项目集
(2)$\mathrm{NAS}=\Phi$；//存放负关联规则
(3)$AL_0=\Phi$；
(4)$\mathrm{KIS}=\Phi$；//见语句(11)
(5)for($k=2;L_{k-1}\neq\Phi;k++$) do begin
(6)$PC_k=$ Candidate-frequent-gen(L_{k-1})[1]；//生成候选频繁k-项目集
(7)$NC_k=$ Candidate-NF-gen(AL_{k-2},L_{k-1})；// NC_k为第k步生成的候选非频繁项目集，其维数并不一定为k，并表明候选负关联规则的前件和后件

(8)C_k= $PC_k \cup NC_k$-KIS;// C_k 为待求支持数的所有候选频繁和非频繁项目集,KIS 表示 C_k 中支持数已知的项目集集合
(9)调用 Calculate-count(C_k);//计算 C_k 中各项目的支持数
(10)L_k={$c \in PC_k$| c. count/|D|≥ minsup};//count 为支持数
(11)KIS={c | $c \in$ (KIS$\cup NC_k$)and |c|>k};// KIS 为支持数已求且其维数大于 k 的项目集,请参见语句(7)、语句(8)
(12)NAS=NAS$\cup$ {$\neg X \Rightarrow Y$|{$X \cup Y$}$\in NC_k$ and $X \cap Y=\Phi$ and {$X \cup Y$}. count/|D|<minsup and supp($\neg XY$)≥minsup and conf($\neg XY$)≥minconf };
(13)end;
(14)$L=\cup_k L_k$;

Procedure Candidate-NF-gen(AL_{k-2}, L_{k-1})

/* 候选非频繁项目集生成方法 */
(15)$NC_k=\Phi$;
(16)for each $X \in L_{k-1}$ do
(17)for each $Y \in (L_{k-1} \cup AL_{k-2})$and $X \cap Y=\Phi$ do
(18)NC_k= $NC_k \cup \{X \cup Y\}$;
(19)for each $X \in AL_{k-2}$ do
(20)for each $Y \in (L_{k-1} \cup AL_{k-2})$and $X \cap Y=\Phi$ do
(21)NC_k= $NC_k \cup \{X \cup Y\}$;

Procedure Calculate-count(C_k)

/* 计算支持数 */
begin
(22)for each transaction $T \in D$ do
(23)for each $c \in C_k$ do
(24)if B_t or B_c= B_t then
(25)c. count++;//定理 5
end

定理 6 算法 MPNAR 是正确的、完备的。

证明:一方面,由语句(12)可知,算法 MPNAR 所求的关联规则均为负关联规则$\neg X \Rightarrow Y$;另一方面,对于任意负关联规则$\neg X \Rightarrow Y$,设频繁项目集 X、Y 的维数分别为 p、q,分下列三种情况讨论:① $p>q$,则语句(5)执行到 $k=(p+1)$后,$X \in L_p$,$Y \in AL_{p-1}$,有{$X \cup Y$}$\in NC_{(p+1)}$,能执行到语句(12),生成负关联规则$\neg X \Rightarrow Y$;② $p=q$,则语句(5)执行到 $k=(p+1)$后,$X \in L_p$,$Y \in L_p$,有{$X \cup Y$}$\in NC(p+1)$,能执行到语句(12),生成负关联规则$\neg X \Rightarrow Y$;③ $p<q$,则语句(5)执行到 $k=(q+1)$后,$X \in AL_{q-1}$,$Y \in L_q$,能执行到语句(12),生成负关联规则$\neg X \Rightarrow Y$。由①、②、③可知算法 MNAR 能求出负关联规则$\neg X \Rightarrow Y$。综上所述,算法 MPNAR 是正确的、完备的。定理 6 得证。

4 算法分析及比较

算法 MPNAR 扫描事务数据库 D 的次数同算法 Apriori 是一样的。设频繁项目集的个数为 m,频繁项目集的平均长度为 al,则候选非频繁项目集的个数不会超过 $m*(m-2^{al}+1)$,其数量级为平方级,而算法 Apriori 中候选频繁项目集个数至少为 m,一般情况下,其数量级亦达到平方级,甚至更高,另外,算法 MPNAR 采用了更为有效和快速的项目集支持度计算方法,因此,从理论上讲,算法 MPNAR 是可行和有效的。

为了进一步验证算法的有效性、可行性和可扩展性,我们用 Vc++6.0 在内存 128M、CPU 为 Pentium III-733MHz、操作系统为 Windows 2000(professional)的机上实现了挖掘算法 MPNAR、正关联规则挖掘算法 Apriori[1],置信度 minconf=85%,使用了与文[2]同样的生成程序来合成所需要的测试数据,测试数据库的有关参数与文[2]相同:|D|表示事务数据记录的数目;|T|表示事务数据记录的平均长度;|I|表示最大频繁项目集的平均长度;|L|表示最大频繁项目集的数目;N 表示事务项目集的个数,而一个测试数据库实例记为 $Tx.Iy.DmK$。在我们的实验中,N=1000,|L|=2000,|D_1|=10k,|D_2|=50k,|T|=20,|I|=10,实验结果如图 1、2 所示。图 1、2 表明算法 MPNAR 与算法 Apriori 的执行时间(包括计算频繁项目集和关联规则的执行时间)是处于同等数量级的,算法 MPNAR 是有效和可行的。

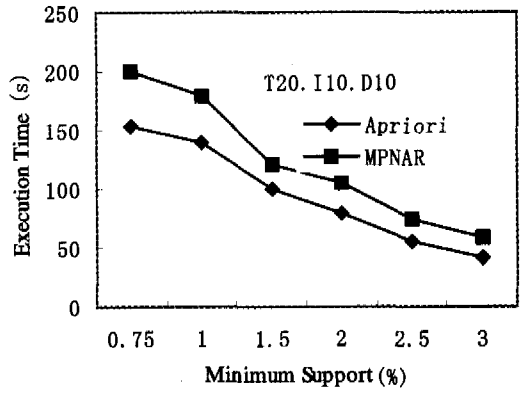

图 1 算法执行时间(|T|=10K)

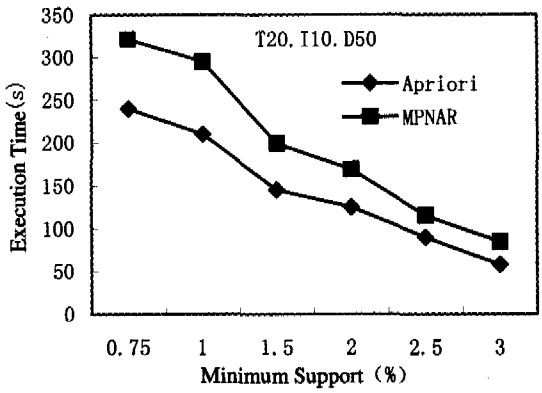

图 2 算法执行时间(|T|=50K)

结束语 自从 Rakesh Agrawa 等人首次提出关联规则挖掘这个研究课题以来,已提出了许多关联规则挖掘算法,然而可用于负关联规则挖掘的算法却很少。本文提出了一种有效的、新颖的负关联规则挖掘算法,该算法可以发现事务数据库中所有的负关联规则。实验表明算法是有效和可行的。另外,本文仅讨论了负关联规则的挖掘问题,在今后的研究工作中我们将在如下两个方面进行研究:① 将 Apriori 算法的改进算法用于负关联规则的挖掘;② 对其更新问题进行深入研究。

参 考 文 献

1 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases [C]. In: Proc. of ACM SIGMOD Int. Conf. Management of Date, Washington D C, 1993. 207~216
2 Agrawal R, Srikant R. Fast algorithm for mining association rules [C]. In: Proc. 20th Int. Conf. on VLDB, Santiago, Chile, 1994. 487~499
3 Han J, Kamber. MData Mining: Concepts and Techniques [M]. Beijing: High Education Press, 2001
4 Goethals B. Survey on frequent pattern mining [R]. Helsinki Institute for Information Technology: [Technical Report], 2003
5 王晓峰,王天然,赵越. 一种自顶向下挖掘长频繁项的有效方法[J]. 计算机研究与发展,2004,41(1):148~155
6 Han J, Jian P, et al. Mining Frequent Patterns without Candidate Generation [C]. In: Proc. of 2000 ACM SIGMOD Intl. Conf. on Management of Data, Dallas, TX, 2000. 1~12
7 朱玉全,孙志挥,季小俊. 基于频繁模式树的关联规则增量式更新算法[J]. 计算机学报,2003,26(1):91~96
8 朱玉全,孙志挥,赵传申. 快速更新频繁项集[J]. 计算机研究与发展,2003,40(1):94~99
9 Savasere A, Omiecinski E, Navathe S. Mining for strong negative rules for statistically dependent items [C]. In: Proc. of ICDM, 2002. 442~449
10 Wu Xin-dong, Zhang Cheng-qi, Zhang Shi-chao. Mining both positive and negative association rules [C]. In: Proc. of the 19th intl conf. on machine learning. San Mateo: Morgan Kaufmann Publishers, 2002. 658~665
11 Antonie M-L, Zaiane O R. Mining Positive and Negative Association Rules: An Approach for Confined Rules [C]. In: Proc. of 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases(PKDD 04), Springer Verlag LNCS 3202, Pisa, Italy, Sept. 2004. 27~38