文章编号:1671-9352(2007)09-0088-03

基于序列模式的正负关联规则研究

郭跃斌,翟延富,董祥军*,杨越越,李刚
(山东轻工业学院 信息科学与技术学院,山东 济南 250353)

摘要:序列模式可预测企业的发展方向,负关联规则可展现不良因素的根源,序列模式的正负关联规则为企业决策提供更全面的信息.将序列模式和负关联规则的挖掘算法相结合,利用项集间的相关性,挖掘出序列模式的正负关联规则.

关键词:序列模式;正关联规则;负关联规则;相关性
中图分类号:TP311 **文献标志码**:A

Positive and negative association rules based on sequential patterns

GUO Yue-bin, ZHAI Yan-fu, DONG Xiang-jun*, YANG Yue-yue, LI Gang
(Department of Computer Science and Technology, Shandong Institute of Light Industry, Jinan 250353, Shandong, China)

Abstract: Sequential patterns can predict company developing trends. The negative association rules can show the origin of bad factors, and the positive and negative association rules based on sequential patterns can offer more comprehensive information. The mining of the positive and negative association rules based on sequential patterns was given by combining the sequential patterns algorithm and the negative association rules algorithm and utilizing the correlation between item sets.

Key words: sequential pattern; the positive association rules; the negative association rules; correlation

0 引言

序列模式的关联规则主要描述数据间的前后或因果关系,要求各事件按时间次序登记[1].对企业来说,序列模式可预测顾客的购买行为,促进销售量.比如 $A \Rightarrow B$,这条规则是指顾客在购买了商品 A 之后,往往会接着买商品 B.然而有时还会出现以下情况,顾客在购买了商品 A 之后,往往不会买商品 C,这条规则记为 $A \Rightarrow \neg C$,这就是序列模式的负关联规则.在企业制定决策时,序列模式的负关联规则对于如何减少负面因素,最大限度的增加正面效益尤为重要.

1 相关概念

1.1 序列模式关联规则

序列模式把数据之间的关联性与时间联系起来,为了发现序列模式,不仅需要知道事件是否发生,而且要确定各事件发生的先后顺序.基本概念如下:设有2个序列 $a < a_l, a_2, \cdots a_n >$ 和 $b < b_l, b_2, \cdots b_m >$,如果存在整数 $i_1 < i_2 \cdots < i_n$,且 a_1 包含于 b_{i1},a_2 包含于 b_{i2},…,a_n 包含于 b_{in},则称序列 a 包含于序列 b 中.在一个序列集中,如果序列 S 不包含于任何其它序列中,则称序列 S 为最大的序列.如果一个序列 S 包含于一个客户序列中,则称该客

收稿日期:2007-04-18
基金项目:山东省优秀中青年科学家奖励基金资助项目(2006BS01017);山东省教育厅科技计划资助项目(J06N06)
作者简介:郭跃斌(1978-),女,硕士研究生,主要研究方向:数据挖掘. Email:gyb-07@163.com
*通讯作者: Email:D-XJ@163.com

户支持序列 S. 一个具体序列的支持定义为那一部分支持该序列的客户总数. 给定一个由客户交易组成的数据库 D,挖掘序列模式的问题就是从那些具有客户指定最小支持度的序列中找出最大序列. 而每个这样的最大序列就代表了一个序列模式.

找出所有的序列模式分 5 个具体阶段来实现:(1)排序阶段,(2)大项集阶段,(3)转换阶段,(4)序列阶段,(5)选最大阶段[2].

首先将事务数据库转换成按时间排序的事务序列. 然后找出所有大项集组成的集合 L. 接着利用已知的大项集的集合来找到所需的序列. 我们从一个由大序列组成的种子集开始,利用这个种子集,可以产生新的潜在的大序列. 在遍历数据的过程中,我们计算出这些候选序列的支持度,这样在一次遍历的最后,就可以决定哪些候选序列是真正的大序列,这些序列构成下一次遍历的种子集. 最终找到最长的序列模式. 在第一次遍历前,所有在大项集阶段得到的具有最小支持度的大 1-序列组成了种子集.

1.2 负关联规则

设 X 表示一项集,则$\neg X$ 被称为负项集[3],它表示一个事务中不存在项集 X,并把 $A\Rightarrow\neg B$, $\neg A\Rightarrow B$, $\neg A\Rightarrow\neg B$ 称为负关联规则. $A\Rightarrow B$ 相应地称为正关联规则.

例如:设 A 表示购买了茶叶的顾客,B 表示购买了咖啡的顾客. 关联规则"$A\Rightarrow\neg B$(support = 20%,confidence = 70%)说明在所有的顾客事务中,有 20%的顾客购买了茶叶而没有买咖啡,其支持度 support = 20%,而购买了茶叶的顾客中有 70%的顾客没有买咖啡,其置信度 confidence = 70%".

对于负关联规则的支持度与置信度,可以利用正关联规则的相关信息计算[4]:

支持度计算方法

设 $A,B\subset I,A\cap B=\Phi$,则有:

(1) $\mathrm{supp}(\neg A)=1-\mathrm{supp}(A)$.

(2) $\mathrm{supp}(A\cup\neg B)=\mathrm{supp}(A)-\mathrm{supp}(A\cup B)$;

(3) $\mathrm{supp}(\neg A\cup B)=\mathrm{supp}(B)-\mathrm{supp}(A\cup B)$;

(4) $\mathrm{supp}(\neg A\cup\neg B)=1-\mathrm{supp}(A)-\mathrm{supp}(B)+\mathrm{supp}(A\cup B)$.

置信度计算方法

设 $A,B\subset I,A\cap B=\Phi$,则有:

(1) $\mathrm{conf}(A\Rightarrow\neg B)=\dfrac{\mathrm{supp}(A)-\mathrm{supp}(A\cup B)}{\mathrm{supp}(A)}=1-\mathrm{conf}(A\Rightarrow B)$;

(2) $\mathrm{conf}(\neg A\Rightarrow B)=\dfrac{\mathrm{supp}(B)-\mathrm{supp}(A\cup B)}{1-\mathrm{supp}(A)}$;

(3) $\mathrm{conf}(\neg A\Rightarrow\neg B)=\dfrac{1-\mathrm{supp}(A)-\mathrm{supp}(B)+\mathrm{supp}(A\cup B)}{1-\mathrm{supp}(A)}=1-\mathrm{conf}(\neg A\Rightarrow B)$.

在同时研究正负关联规则后,会发现有的规则是相互矛盾的.

例如:设 A 表示购买了茶叶的顾客,B 表示购买了咖啡的顾客,则 $\mathrm{supp}(A)=50\%$, $\mathrm{supp}(B)=50\%$, $\mathrm{supp}(A\cup B)=20\%$, $\mathrm{conf}(A\Rightarrow B)=40\%$.

由以上方法可得:

$\mathrm{supp}(\neg A)=50\%$, $\mathrm{supp}(\neg B)=50\%$, $\mathrm{supp}(A\cup\neg B)=30\%$, $\mathrm{supp}(\neg A\cup B)=30\%$, $\mathrm{supp}(\neg A\cup\neg B)=20\%$.

$\mathrm{conf}(A\Rightarrow\neg B)=60\%$, $\mathrm{conf}(\neg A\Rightarrow B)=60\%$, $\mathrm{conf}(\neg A\Rightarrow\neg B)=40\%$

假定最小支持度 $\mathrm{supp}_{\min}=20\%$,最小置信度 $\mathrm{conf}_{\min}=40\%$. 则 4 个规则 $A\Rightarrow B$, $A\Rightarrow\neg B$, $\neg A\Rightarrow B$, $\neg A\Rightarrow\neg B$ 都是有效规则. 他们分别表示①顾客购买茶叶也会购买咖啡;②顾客购买茶叶不会购买咖啡;③顾客不买茶叶会购买咖啡;④顾客不够买茶叶也不会购买咖啡. 但①④与②③互相矛盾.

为解决此矛盾,可通过判断项集间的相关性来挖掘出频繁项集中的正、负关联规则,并检测和删除相互矛盾的规则. 相关性的定义如下[5]:

设 A,B 为 2 个项集,它们的相关性表示为

$$\mathrm{corr}_{A,B}=\frac{\mathrm{supp}(A\cup B)}{\mathrm{supp}(A)\mathrm{supp}(B)}$$

$\mathrm{corr}_{A,B}$的值有以下 3 种情况:

(1) 如果 $\mathrm{corr}_{A,B}>1$, 那么 A 和 B 正相关,事件 A 出现的越多,事件 B 出现的也越多;

(2) 如果 $\mathrm{corr}_{A,B}=1$, 那么 A 和 B 相互独立,事件 B 的出现与事件 A 无关;

(3) 如果 $\mathrm{corr}_{A,B}<1$,那么 A 和 B 负相关,事件 A 出现的越多,事件 B 出现的越少.

2 序列模式的正负关联规则挖掘算法分析

下面以商场顾客的购买行为作为挖掘对象来阐述算法的实现.

2.1 排序阶段

数据库以顾客号为主键,交易时间为次键进行

排序. 如表 1 所示(其中用数字来代表商品).

表 1　以顾客号(Cust_id)及交易时间(Tran_time)排序的数据库

Table 1　The database ordered by Cust_id and Tran_time

顾客号(Cust_id)	交易时间(Tran_time)	物品(Item)
1	2002-01-25	1
1	2002-01-30	6
2	2002-01-10	7,8
2	2002-01-15	1
2	2002-01-20	2,4,5
3	2002-01-25	2
3	2002-01-30	3
4	2002-01-12	1
4	2002-01-23	2,5
4	2002-01-30	6
5	2002-01-12	3
5	2002-01-20	5
5	2002-01-30	6

2.2　大项集阶段

在此阶段找出所有大项集组成的集合 L. 同时得到所有的大 1-序列组成的集合. 假定最小支持度计数为 2,由表 1 得到的大项集如表 2 所示(其中每个大项集映射为一个数字).

表 2　大项集

Table 2　Large itemsets

Itemsets	Support	Mapped To
(1)	3	1
(2)	3	2
(3)	2	3
(5)	3	4
(2,5)	2	5
(6)	3	6

2.3　转换阶段

将客户序列转换成大项集序列. 如果一条交易不包含任何大项集,则此交易不被保留. 如表 3 所示.

表 3　转换后的数据库

Table 3　Transformed database

Cust_id	Sequence	Mapping
1	{(1)}{(6)}	{1}{6}
2	{(1)}{(2),(5),(2,5)}	{1}{2,4,5}
3	{(2)}{(3)}	{2},{3}
4	{(1)}{(2),(5),(2,5)}{(6)}	{1}{2,4,5}{6}
5	{(3)}{(5)}{(6)}	{3}{4}{6}

2.4　序列阶段

利用已知的大项集来找到所需的序列. 算法如下:

L_1 = frequent 1-sequences;

for($k=2; L_{k-1}\neq 0; k++$) do begin

C_k = apriori_ gen(L_{k-1}, min_sup);//new candidates generated from L_{k-1}

For each customer-sequence $t\in D$ do begin

C_t = subset(C_k, t); //get the subsets of t that are candidates

For each candidate $c\in C_t$ do

C.count++;

end

$L_k = \{C\in C_k \mid c.\text{count} >= \text{min\_sup}\}$

end

Answer = $U_k L_k$;

function apriori_gen(L_{k-1}; min_sup)

insert into C_k

select p.item1, p.item2, …, p.item$_{k-1}$, q.item$_{k-1}$ from L_{k-1} p, L_{k-1} q

where p.item$_1$ = q.item$_1$ and p.item$_2$ = q.item$_2$ … and p.item$_{k-1}$ < q.item$_{k-1}$;

forall itemsets $C\in C_k$ do

forall($k-1$)-subsets s of c do

if ($s\notin L_{k-1}$) then

delete c from C_k

return C_k

其中 L_k 表示所有 k-序列组成的集合, C_k 表示候选 k-序列组成的集合.

2.5　选最大阶段

定义最长序列的长度为 n,用以下算法找出最长序列:

for ($k=n; k>1; k$--)do

foreach k-sequence s_k do

delete from s all subsequence of s_k

此例可得长度为 2 的序列{(1)}{(6)}和{(1)}{(2,5)}. 因此(1)⇒(6),(1)⇒(2,5)为本例挖掘的序列模式的正关联规则.

利用表 2 中的大项集作为输入,挖掘序列模式的负关联规则.

由于此时挖掘的是序列模式的负关联,所以在考虑相关性时,应将时间因素加入. 在相关性的表达式中,supp($A\cup B$)是指 A 发生后 B 再发生的支持度.

由表 1 和表 2 可得相关性小于 1 的项集为(1)和(3),由 1.2 节关于计算负关联　(下转第 95 页)

(上接第 90 页) 规则支持度的方法可知(1)⇒¬(3)的支持度计数为 3. 所以(1)⇒¬(3)为序列模式的负关联规则.

此阶段算法在正关联规则算法的基础上添加了部分计算和判断语句,两者具有相同的时间复杂度.

3 试验

为了验证上述算法的有效性, 在合成数据上做了一个试验, 实验是在 AMD athlon64 3 000 MHz, 512 RAM, WIN 2000, VisualBasic 环境下进行的. 共有 200 个事务的实验数据. 设最小支持度为 25, 序列长度为 2, 共挖掘出 152 个正关联规则和 87 个负关联规则.

4 结语

本文给出了序列模式和负关联规则的基本概念和挖掘方法,并将两者结合用来挖掘序列模式的负关联规则. 此算法在进行相关性检查时,应注意事件发生的时间次序. 最后以商场顾客的购买行为为例,证明该算法是可行有效的.

参考文献:

[1] Margaret H Dunham.数据挖掘教程[M]. 郭崇慧,田凤占,靳晓明等译.北京:清华大学出版社,2003.

[2] 毛国军,段立娟,王实,等.数据挖掘原理与算法[M].北京:清华大学出版社,2005.

[3] WU Xindong, ZHANG Chengqi, ZHANG Shichao. Mining both positive and negative association rules[C]// Proceedings of the 19th International Conference on Machine Learning (KML-2002). San Francisco: Morgan Kaufmann Publishers, 2002: 658-665

[4] 董祥军,王淑静,宋瀚涛, 等. 负关联规则的研究[J]. 北京理工大学学报,2004,24(11):978-981.

[5] BRIN S, MOTWANI R, SILVERSTEIN C. Beyond market: Generalizing association rules to correlations[C]// Processing of the ACM SIGMOD Conference 1997. New York: ACM Press, 1997: 265-276.

(编辑:孙培芹)