

# 一种改进的正负关联规则挖掘算法

陈宁军<sup>1,2</sup> 高志年<sup>1</sup>

(南京陆军指挥学院作战实验中心 南京 210045)<sup>1</sup> (南京陆军指挥学院研究生三队 南京 210045)<sup>2</sup>

**摘 要** 针对传统正负关联规则挖掘算法需要多次扫描数据库并且生成大量候选频繁项集的问题,在对比目前相关研究成果的基础上,提出了一种改进的正负关联规则挖掘算法,它通过两次数据扫描完成对正负关联规则的挖掘,对最大频繁项集的挖掘算法做了改进,有效提高了算法效率,同时对置信度标准做了改进。基于某真实事务集的实验表明,算法提高了规则挖掘的质量和有效性。

**关键词** 正负关联规则,关联规则挖掘,最大频繁项集,置信差

中图法分类号 TP391 文献标识码 A

## Improved Positive and Negative Association Rules Mining Algorithm

CHEN Ning-jun<sup>1,2</sup> GAO Zhi-nian<sup>1</sup>

(Dept. of Operation Research Center, ACC, Nanjing 210045, China)<sup>1</sup> (Team 3 of Post Graduated, ACC, Nanjing 210045, China)<sup>2</sup>

**Abstract** Aiming at the problems of the traditional positive and negative association rules, such as multi-scanning database and generating large candidate frequent itemset, on the basis of comparing recent research on association rules mining, put forward an improved positive and negative association rules mining algorithm, which can accomplish mining positive and negative association rules by scanning Dataset twice; and improve the algorithm of mining the maximal frequent itemset, which result in upgrading the efficiency of the algorithm; besides, improve the standard of confidence level, in order to increase the quality of association rules mining.

**Keywords** Positive and negative association rules, Association rules mining, Maximal frequent itemset, D-value of confidence level

关联规则挖掘就是从大量的数据中挖掘出有价值的描述数据项之间相互联系的有关知识。随着收集和存储在数据库中的数据规模越来越大,人们对从这些数据中挖掘相应的关联知识越来越有兴趣。例如,从大量的商业交易记录中发现有价值的关联知识就可以帮助进行商品目录的设计、交叉营销或帮助进行其它有关的商业决策。挖掘关联知识的一个典型应用实例就是市场购物分析。根据被放到一个购物篮中的内容记录数据而发现不同商品之间存在的关联知识,无疑将会帮助商家分析客户的购买习惯。发现常在一起购买的商品(关联知识)将帮助商家制定有针对性的市场营销策略。如顾客在购买牛奶时会有效地帮助商家进行有针对性的促销,以及进行合适的货架商品摆放,比如可以将牛奶和面包放在相近的地方或许会促进这两个商品的销售。

### 1 相关研究

关联规则作为一个研究领域仅仅是最近几年的事情,但是在出现后的几年里已经成为数据库界广泛研究的热点。关联规则挖掘问题最早是由 Agrawal 等人在 1993 年于文献[1]中首次提出,随后,诸多的研究人员对其进行了大量研究。这些研究主要在于解决 3 方面的问题:(1)怎样提高关联规则挖

掘效率,怎样减少关联规则存储空间;(2)怎样在稠密的、海量的数据集下进行关联规则挖掘;(3)怎样挖掘有价值的关联规则。

发现最大频繁项目集是关联规则挖掘中的一个重要研究方向,在国外主要有 A. Savasere 等人提出的 Partition 算法<sup>[2]</sup>, Zaki 提出的 Eclat 算法<sup>[3,4]</sup>, Christian Borgelt 提出的 Relim 算法<sup>[5]</sup>, Christian Hidber 1999 年提出的在线挖掘关联规则的 Carma 算法<sup>[6]</sup>, Gunopulos 等人提出的 ALL-MFS<sup>[7]</sup> 算法, Bayardo 等人提出的 Max-Miner 算法<sup>[8]</sup>, Lin 等人提出的 Pincer-Search 算法<sup>[9]</sup>, 以及 Burdick 等人提出的 Mafia 算法<sup>[10]</sup>等;在国内主要有路松峰等人提出的算法 DMFI<sup>[11]</sup>, 宋余庆等人提出的算法 DMFIA<sup>[12]</sup>等。

1997 年 Brin<sup>[13]</sup>等第一次提出了在两个频繁项集之间存在一种负相关关系的概念,并提出用卡方值来判断两个项集是否相互独立,如果不是相互独立,就需要一种度量来确定两个项集之间是否正相关或负相关。负关联规则挖掘是关联规则挖掘中一个较新的研究方向,目前国内外对负关联规则的研究还比较少,主要的研究成果包括: Savasere 等人的 SNA (strong negative associations) 算法<sup>[14]</sup>; Wu, X. 等人提出的 PNR (positive and negative association rules) 算法<sup>[15]</sup>和 Wei-

到稿日期:2011-05-29 返修日期:2011-07-20 本文受中国博士后科学基金(200902695)资助。

陈宁军(1980-),男,博士生,讲师,主要研究方向为军事仿真、模式识别, E-mail: cnj\_nj@163.com; 高志年(1957-),男,博士,教授,主要研究方向为军事仿真、多模型建模。

Guang Teng 等提出的 SRM(substitution rules mining)算法<sup>[16]</sup>。

## 2 算法思想及设计

本文提出了一种改进的正负关联规则挖掘算法,称为 ACPNAR 算法。其主要求解思想是:(1)计算包括负项的项集支持度;(2)生成包括正负项目的最大频繁项集;(3)将置信度标准改进为置信差标准,并根据这一新标准生成最终的正负关联规则。

### 2.1 包括正负项的项集支持度计算方法

包含正负项目的项集支持度,可以由只包含正项目的项集支持度导出,由此我们得到了计算包括正负项的项目集支持度计算方法。文献[4]中给出了正、负项之间的关系:

**定理 1** 设  $A, B \subset D, A \cap B = \Phi$ , 则:

$$S(\neg A) = 1 - S(A);$$

$$S(A \cup \neg B) = S(A) - S(A \cap B);$$

$$S(\neg A \cup B) = S(B) - S(A \cap B);$$

$$S(\neg A \cup \neg B) = 1 - S(A) - S(B) + S(A \cap B).$$

在此基础上,不难得出计算包含负项目的项集支持度的方法。

**推论 1** 设  $I = \{A, B, \neg C\}$ , 则  $S(I) = S(A \cup B) - S(A \cup B \cup C)$ 。

根据定理 1 及其推论,我们很容易设计出求解包含正负项的项集支持度的正负关联规则挖掘算法。

据此,我们采用传统 Apriori 算法的思想,设计出了一种能同时挖掘正负关联规则的算法,并对最大频繁项集的生成算法做了改进。

### 2.2 对最大频繁项集挖掘算法的改进

Apriori 算法作为经典的挖掘事件关联规则的算法,一直以来受到极大的关注,但是 Apriori 算法具有两个缺点,即,1)该算法在计算的过程中需要产生大量的候选项目集;2)Apriori 算法需要对数据库进行多次扫描,并通过模式匹配检查候选项目集,如果数据库的容量很大或要进行匹配的模式很长时,算法的效率会大大降低。

**定理 2** 频繁项集的所有非空子集都必须也是频繁的。

如果项集  $I$  不满足最小支持度阈值  $min\_sup$ , 则  $I$  不是频繁的,即  $P(I) < min\_sup$ 。如果项  $A$  添加到  $I$ , 则结果项集(即  $I \cup A$ )不可能比  $I$  更频繁出现。因此  $I \cup A$  也不是频繁的,即  $P(I \cup A) < min\_sup$ 。

**推论 2** 如果  $k$ -项集  $T_k$  的任意一个子集  $(k-1)$ -项集不是频繁的,则  $T_k$  也不是频繁的。

由定理 2 很容易看出推论是正确的,在这里不加以证明。

**推论 3**  $T_k$  是  $k$ -项集,如果所有  $(k-1)$ -项集集合  $L_{k-1}$  中包含  $T_k$  的  $(k-1)$ -项子项集的个数小于  $k$ , 则  $T_k$  不是频繁集。

证明:很明显,数据项集  $C_{k-1}$  的  $(k-1)$ -项子项集的个数为  $k$ , 如果频繁集  $L_{k-1}$  中包含  $T_k$  的  $(k-1)$ -项子项集的个数小于  $k$ , 则存在  $T_k$  的  $(k-1)$ -项子项集不是频繁集,由推论 2 知  $T_k$  不是频繁集。

在对 Apriori 算法改进的基础上,我们设计了这种新的求取包含负项的频繁项集生成算法,该算法主要把求取频繁集的过程分为两部分:(1)求取频繁 2-项集;(2)求取频繁  $k$ -项集

( $k > 2$ )。

由于数据挖掘面临的是海量数据,因此在许多情况下会产生大量的候选项集,尤其是候选 2-项集,为了计算候选 2-项集的支持度,每一个候选项集需要扫描一次数据库,从而严重影响挖掘的效率。ACPNAR 算法第一步解决在不产生候选 2-项集的情况下直接产生频繁 2-项集。算法如下:

初始项目集 ITEM, ITEM 包括所有的项目:  $ITEM = \{item1, item2, \dots\}$ , 定义一个 team 结构数组,每一个 team 结构对应一个项目,每个 team 中包括包含该项目的所有事务序号(TID)和一个 count 变量(存储这个 team 中包括的 TID 的数量),设定最小支持度  $minsup = a\%$ , 事务总数为  $K$ 。

(1)生成 team 结构数组。

对 item1, 搜索数据库求取包含它们的 TID, 生成对应的 team1;

对 item2, 搜索数据库求取包含它们的 TID, 生成对应的 team2;

.....

对于所有 item 生成对应的 team。

(2)通过查找 team 中每一个结构的 count 变量,去掉事务数小于  $K * a\%$  的项目。

(3)对任意两个 team 中的事务做交集运算,  $team1 \cap team2, team1 \cap team3, \dots$ , 计算求交集后的集合中 TID 的数量,如果  $team_i \cap team_j$  数量大于  $K * a\%$ , 则生成新的项集  $\{i, j\}$ ; 如果  $team_i \cap team_j$  数量小于  $K * a\%$ , 则舍去。

(4)整理所有保留 2-项集,即为求得全部频繁 2-项集。

ACPNAR 算法在求取频繁  $k$ -项集( $k > 2$ )时,采用如下方法:定义一个 group 结构数组,在生成频繁 2-项集之后,每一个频繁 2-项集对应了 group 数组中的一个结构,每个 group 结构保存了包括该项目的所有事务的 TID;对于下面产生的频繁 2-项集、频繁 3-项集....., 每组候选项集分别对应了 group 数组中的结构,每个 group 结构保存包括该候选项集的所有事务的 TID。对于候选项集  $K$  来说,  $K$  的 group 由那些包含  $K$  的事务的 TID 组成,用  $K.group$  表示,  $K$  的支持度等于  $K.group$  中的事务个数,用  $|K.group|$  表示。1-项集的 group 通过搜索事务数据集得到,候选  $k$ -项集的 group 由产生该候选  $k$ -项集的那两个  $(k-1)$ -项集的 group 求交集得到。算法的描述如下:

- 1) for all items  $i \in ITEM$
- 2) create a  $team_i$  for  $i$ ;
- 3) if ( $team_i.count < K * a\%$ )
- 4) delete item  $i$  from ITEM;
- 5) for all item  $i, j \in ITEM$
- 6) if ( $team_i \cap team_j \geq K * a\%$ )
- 7) itemsets =  $\{i, j\}$ ;
- 8) create  $L_2$ ;
- 9) for ( $k=3; L'_{k-1} \neq \Phi; k++$ ) {
- 10)  $L_k = \Phi; L'_k = \Phi$
- 11) for all itemsets  $l_1 \in L'_{k-1}$
- 12) for all itemsets  $l_2 \in L'_{k-1}$
- 13) if  $l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \dots \wedge l_1[k-1] < l_2[k-1]$  then
- 14)  $C'.itemsets = l_1[1]l_1[2] \dots l_1[k-1]l_2[k-1]$
- 15)  $C'.group = l_1.group \cap l_2.group$
- 16)  $C'.count = |C'.group|$
- 17) if ( $C'.count \geq minsup$ ) then

```

18)  $L_k' = L_k' \cup \{C'\}$ 
19)  $C.itemsets = C'.itemsets$ 
20)  $C.count = C'.count$ 
21)  $L_k = L_k \cup \{C\}$ 
22) }
23) return  $L_k$ ;

```

在 ACPNAR 算法中,1)~8)生成频繁 2-项集,通过建立 team 结构数组,在不产生候选 2-项集的情况下可以直接生成频繁 2-项集;9)~23)产生频繁  $k$ -项集,只搜索一次事务集,每次只需要搜索 group 就可以得到候选集的支持度,大大缩短了检索时间。

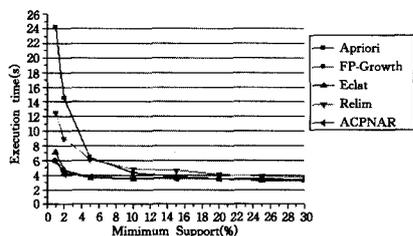
### 2.3 对最小置信度标准的改进

Confidence( $A \Rightarrow B$ )描述的是在项集  $A$  存在的情况下项集  $B$  出现的概率  $P(B|A)$ 。 $P(B|A)$  的大小与  $A$ 、 $B$  本身在整个数据集中的出现概率  $P(A)$ 、 $P(B)$  及  $A$  和  $B$  之间的相关程度都有关系。这种传统的最小置信度标准其实不能很好地反应  $A$  和  $B$  之间的相关性。当  $P(B)$  较大时,即使  $A$  和  $B$  之间的相关性较小,置信度仍然会比较大,而当  $B$  出现很少时,即使  $A$ 、 $B$  是强相关的,置信度依然会很小。

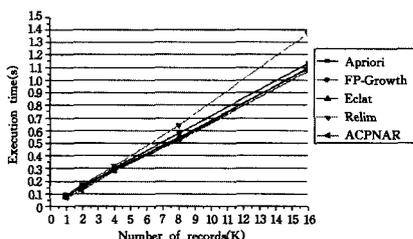
据此,我们提出用置信差代替置信度作为衡量标准。置信差可定义为: $SubC = P(B|A) - P(B|\neg A)$ 。其取值范围为  $[-1, 1]$ ,当  $SubC = 0$  时,两项相互独立,不生成规则;当  $SubC > 0$  时,生成正规规则;当  $SubC < 0$  时,生成负规则;且  $|SubC|$  越接近 1, $A$ 、 $B$  相关性越强。依照这一新的置信差标准,不难从上文生成的包含负项的最大频繁项集中生成相应的正负关联规则。

### 3 算法性能测试

以某系统的真实事务集作为实验数据集,选取的事务集中包含 48842 条事务,每条事务长度为 14 个项目左右。分别对 Apriori、FP-Growth、Eclat、Relim 和 ACPNAR 算法进行性能分析,测试结果如图 1 所示。



(a) 事务数为 48842 条,支持度变化情况



(b) 支持度为 20%,事务数变化情况

图 1 Apriori、FP-Growth、Eclat、Relim 和 ACPNAR 算法性能测试

随着支持度的变化,Apriori 算法的性能最差,特别在支持度小于 5% 时,所需要的求取频繁集的时间远远高于其它算法,在支持度大于 10% 以后,与其它算法的求取时间接近,趋近于定值;Relim 算法性能在支持度小于 5% 时仅高于

Apriori 算法,但在支持度大于 5% 后,其所需求取频繁集的时间最多,高于其它算法;FP-Growth 和 Eclat 算法性能相近,ACPNAR 算法略高于 FP-Growth 和 Eclat 算法。随着事务集数量的变化,Relim 算法所需要求取频繁集的时间最多,特别当事务集增大后,Relim 所耗费的时间增量更大,越来越高于其它算法;Apriori 算法仅好于 Relim 算法;ACPNAR 算法所需时间最少,特别当事务集数量越来越大时,ACPNAR 算法的性能优势更加明显,所需时间更少。

**结束语** 本文提出了一种新的求取事件关联规则的 ACPNAR 算法。ACPNAR 算法能同时完成对正负关联规则的挖掘,并把求解最大频繁项集分为两步,第一步直接生成频繁 2-项集,避免了生成多个无用候选集;第二步通过建立 group 数组,只需要扫描一次数据库就可以求出频繁集,不需要多次扫描数据库;同时提出了用置信差代替置信度作为规则衡量标准,提高了规则的有效性。通过性能对比,ACPNAR 相比 Apriori、Relim、Eclat 和 FP-Growth 等算法需要更少的求取频繁集时间,性能更高。下一步我们将继续研究如何进一步缩减挖掘出关联规则的数量,如何过滤出实用的关联规则,以提高规则的有效性。

### 参考文献

- [1] Agrawal R, Imielinski T, Swami A N. Database mining: A performance perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6): 914-925
- [2] Park J S, Chen M-S, Yu P S. An effective hash-based algorithm for mining association rules[C] // Proc 1995 ACM-SIGMOD Int. Conf Management of Data (SIGMOD'95). San Jose: CA, May 1995: 175-186
- [3] Zaki M J, Parthasarathy S, Ogihara M, et al. New Algorithms for Fast Discovery of Association Rules[C] // Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 1997
- [4] Zaki M J. Scalable Algorithms for Association Mining[J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 12
- [5] 刘喜苹, 刘彩苹, 谭义红. 一个新的不需要候选集的挖掘关联规则算法-Relim 算法的研究[J]. 计算技术与自动化, 2006, 25(2): 81-84
- [6] Hidber C. Online Association Rule Mining[C] // Proceedings of ACM SIGMOD International conference on Management of Data, 1999: 145-156
- [7] Gunopulos D, Mannila H, Saluja S. Discovering all most specific sentence by randomized algorithms[C] // 6th International Conference in Database Theory, 1997: 215-229
- [8] Bayardo R. Efficiently mining long patterns from databases[C] // Proceedings of the ACM SIGMOD International Conference on Management of Data, New York: ACM Press, 1998: 85-93
- [9] Lin Dao-I, Kedem Z M. Pincer-Search: a new algorithm for discovering the maximum frequent set[C] // Proceedings of the 6th European Conference on Extending Database Technology, Heidelberg: Springer-Verlag, 1998: 105-119
- [10] Burdick D, Calimlim M, Gehrke J. Mafia: A maximal frequent itemset algorithm for transactional database[C] // International Conference on Data Engineering, 2001
- [11] 路松峰, 卢正鼎. 快速开采最大频繁项目集[J]. 软件学报, 2001, 12(2): 293-297

(下转第 212 页)

称为对  $H$  的外粗化运算。

例 5 对图 4 的粒结构中层次 Level2 中的粒  $g_5, g_6, g_7$  无法直接进行粗化运算,对它们进行内粗化和外粗化运算有:

$$\underline{\sigma}(\{g_5, g_6, g_7\}) = g_2$$
$$\bar{\sigma}(\{g_5, g_6, g_7\}) = \{g_1, g_2\}$$

下面的定理反映了内外粗化运算的性质。

定理 3 设  $l_1, l_2$  为两个层次,  $l_1$  的粒度比  $l_2$  粗糙,  $A \subseteq l_1, B \subseteq l_2$ , 那么,

$$\omega(A) \subseteq B \Leftrightarrow A \subseteq \underline{\sigma}(B)$$

$$B \subseteq \omega(A) \Leftrightarrow \bar{\sigma}(B) \subseteq A$$

定理 4 设  $l_1, l_2$  为两个层次,  $l_1$  的粒度比  $l_2$  粗糙,  $\forall A, B \subseteq l_2, \omega$  是从  $l_1$  到  $l_2$  的细化运算,  $\underline{\sigma}, \bar{\sigma}$  分别是  $l_2$  到  $l_1$  的内外粗化运算, 则,

$$\underline{\sigma}(\emptyset) = \bar{\sigma}(\emptyset) = \emptyset$$

$$\underline{\sigma}(l_2) = \bar{\sigma}(l_2) = l_1$$

$$\underline{\sigma}(A) \subseteq \bar{\sigma}(A)$$

$$\underline{\sigma}(A) = (\bar{\sigma}(A^c))^c$$

$$\bar{\sigma}(A) = (\underline{\sigma}(A^c))^c$$

$$\underline{\sigma}(\omega(A)) = \bar{\sigma}(\omega(A)) = A$$

$$\omega(\underline{\sigma}(A)) \subseteq A$$

$$A \subseteq \omega(\bar{\sigma}(A))$$

$$A \subseteq B \Rightarrow \underline{\sigma}(A) \subseteq \underline{\sigma}(B)$$

$$A \subseteq B \Rightarrow \bar{\sigma}(A) \subseteq \bar{\sigma}(B)$$

$$\underline{\sigma}(A \cap B) = \underline{\sigma}(A) \cap \underline{\sigma}(B)$$

$$\bar{\sigma}(A \cap B) \subseteq \bar{\sigma}(A) \cap \bar{\sigma}(B)$$

$$\underline{\sigma}(A) \cup \underline{\sigma}(B) \subseteq \underline{\sigma}(A \cup B)$$

$$\bar{\sigma}(A \cup B) = \bar{\sigma}(A) \cup \bar{\sigma}(B)$$

结束语 粒计算的三元论模型以粒结构为基础,包括 3 个部分,即哲学思想(结构化思维)、方法论(结构化问题求解)、计算模式(结构化信息处理),将其将现有粒计算研究成果的共性抽象出来,为问题求解提供了统一的方法论。粒计算研究的对象所具有的结构称为粒结构,其组成元素包括粒、层次及分层结构。多层次和多视角是粒结构的核心内容。本文基于图定义了图上的粒、层次,然后基于包含关系定义了粒结构。在基于图的粒结构的基础上,给出了粒结构中的细化、粗化运算符。粗化运算符能够实现从细粒度到粗粒度的转换,将细粒度的层次转换为粗粒度的层次,或者将细粒度层次中的粒转换为粗粒度层次中的粒。细化运算符实现从粗粒度到细粒度的转换,将粗粒度层次转换为细粒度层次,或者将粗粒度层次中的粒转换为细粒度层次中的粒。通过两个运算符,我们可以在不同的粒度间自由转换,从而能够模仿人类问题的求解方式来更好地处理问题。

## 参考文献

- [1] Bargiela A, Pedrycz W. Granular computing: an introduction [M]. Kluwer Academic Publishers, Boston, 2002
- [2] Chen G, Zhong N, Yao Y Y. Hypergraph model of granular computing [C]// Proceedings of the 2008 IEEE International Conference on Granular Computing, 2008; 80-85
- [3] Giunchiglia F, Walsh T. A theory of abstraction [J]. Artificial Intelligence, 1992, 56: 323-390
- [4] Hobbs J R. Granularity [C]// Proceedings of the 9th International Joint Conference on Artificial Intelligence, 1985; 432-435
- [5] Hornsby K. Temporal zooming [J]. Transactions in GIS, 2001, 5: 255-272
- [6] McCalla G, Greer J, Barrie J, et al. Granularity hierarchies [J]. Computers and Mathematics with Applications, 1992, 23: 363-375
- [7] 苗多谦, 王国胤, 刘清, 等. 粒计算: 过去现在与展望 [M]. 北京: 科学出版社, 2007; 1-20
- [8] 苗多谦, 范世栋. 知识的粒度计算及其应用 [J]. 系统工程理论与实践, 2002, 22(1): 49-57
- [9] Pawlak Z. Rough sets [J]. International Journal of Computer and Information Sciences, 1982, 11: 341-356
- [10] Shafer G. A Mathematical Theory of Evidence [M]. Princeton University Press, Princeton, 1976
- [11] Yager R R, Filev D. Operations for granular computing: mixing words with numbers [C]// Proceedings of IEEE International Conference on Fuzzy Systems, 1998; 123-128
- [12] Yao Y Y. A partition model of granular computing [C]// LNCS Transactions on Rough Sets, 1, LNCS 3100, 2004; 232-253
- [13] Yao Y Y, Zhong N. Granular computing [J]. Wiley Encyclopedia of Computer Science and Engineering, 2009, 3: 1446-1453
- [14] Yao Y Y. The art of granular computing [C]// Proceeding of the International Conference on Rough Sets and Emerging Intelligent Systems Paradigms, LNAI 4585, 2007; 101-112
- [15] Yao Y Y, Miao D Q, Zhan N, et al. Set-theoretic models of granular structures [C]// Proceedings of the 5th International Conference on Rough Set and Knowledge Technology, LNAI 6401, 2010; 94-101
- [16] Zadeh L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic [J]. Fuzzy Sets and Systems, 1997, 90; 111-127
- [17] 张铃, 张钊. 问题求解理论及应用——商空间粒度计算理论及应用(第 2 版) [M]. 北京: 清华大学出版社, 2007
- [18] 张燕平, 罗斌, 姚一豫, 等. 商空间与粒计算——结构化问题求解理论与方法 [M]. 北京: 科学出版社, 2010; 115-143
- [19] (上接第 193 页)
- [12] 宋余庆, 朱玉全, 孙志挥. 基于 FP-Tree 的最大频繁项目集挖掘及更新算法 [J]. 软件学报, 2003, 14(9): 1586-1592
- [13] Brin S, Motwani R, Silverstein C. Beyond Market Baskets; Generalizing Association Rules to Correlations [C]// Proceedings of the ACM SIGMOD Conference, 1997; 265-276
- [14] Savasere A, Omiecinski E, Navathe S. Mining for strong negative associations in a largedatabase of customer transactions [C]// Proceedings of International Conference on DataEngineering, February 1998
- [15] Wu Xin-dong, Zhang Cheng-qi, Zhang Shi-chao. Mining both positive and negative association rules [C]// Proceedings of the 19th international conference on machine learning, SanMateo: Morgan Kaufmann Publishers, 2002; 658-665
- [16] Teng Wei-guang, Hsieh M-J, Chen M-S. On the mining of substitution rules for statistically dependent items, Data Mining, 2002. ICDM 2002 [C]// Proceedings, 2002 IEEE International Conference, Dec 2002
- [17] 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