第 35 卷 第 7 期 计 算 机 工 程 2009 年 4 月
Vol.35 No.7 Computer Engineering April 2009
·软件技术与数据库· 文章编号：1000—3428(2009)07—0068—02 文献标识码：A 中图分类号：TP311

基于前缀树的模糊关联规则挖掘算法

程 江[1]，易云飞[1,3]，林建辉[2]，余启港[1]
(1. 中南民族大学计算机科学学院，武汉 430074；2. 武汉大学电信工程学院，武汉 430079；3. 河池学院计算机与信息科学系，宜州 546300)

摘 要：针对布尔型关联规则不能表达挖掘对象中模糊信息的关联性，给出一系列有关模糊关联规则的定义，并提出一种基于前缀树的模糊关联规则挖掘算法。通过构造前缀树来压缩存储模糊模式候选集和频繁集，有效地节约了内存开销，且只需扫描数据库 2 遍。实验结果表明，该算法是有效的。

关键词：数据挖掘；关联规则；前缀树；模糊模式

Mining Algorithm for Fuzzy Association Rules Based on Prefix Tree

CHENG Jiang[1], YI Yun-fei[1,3], LIN Jian-hui[2], YU Qi-gang[1]
(1. School of Computer Science, South Central University for Nationality, Wuhan 430074; 2. School of Electronic and Information Engineering, Wuhan University, Wuhan 430079; 3. Department of Computer and Information Science, Hechi University, Yizhou 546300)

【Abstract】In view of that the boolean association rules can not express the association of fuzzy data, this paper gives a series of definitions of fuzzy association rules and proposes a mining algorithm based on prefix tree for fuzzy association rules. The algorithm can store fuzzy pattern candidate sets and frequent sets compressibly by constructing prefix tree, which effectively saves the memory cost, besides it only scans database twice. The efficiency of the algorithm is verified by the experiment.

【Key words】data mining; association rule; prefix tree; fuzzy pattern

关联规则挖掘作为数据挖掘技术的一种重要方法，目前被广泛应用于商务决策、信息安全等领域。然而，数据挖掘通常只能对离散值进行处理，在数据预处理中要将连续属性域划分为若干离散区间，这就导致了所谓的“尖锐边界”问题[1]。为了解决这一问题，国内外一些学者提出了结合模糊集理论的模糊数据挖掘技术，并取得了较好的效果。

1 模糊关联规则及相关概念

设论域 U 上，映射 $\mu_A:U\to[0,1],\forall X\in U$，$\mu_A(X)$ 为 A 上的隶属度。对 A,B 2 个模糊集，有

$$(A\cap B)(U)=\mu_A(X)\wedge\mu_B(X)=\min\{\mu_A(X),\mu_B(X)\}$$

对于 $\forall\lambda\in[0,1]$，定义 $A_\lambda=\{\mu_A(X)\geqslant\lambda\}$ 为模糊集 A 的 λ 截集；若 $A_\lambda=\{\mu_A(X)>\lambda\}$，则为 λ 强截集。

令数据库 $D=\{t_1,t_2,\cdots,t_k,\cdots,t_n\},I=\{i_1,i_2,\cdots,i_m\}$ 为 D 中属性集，要发现的模糊关联规则是形如以下的蕴涵式：

$$<X,A>\Rightarrow<Y,B>$$

其中，$X=\{x_1,x_2,\cdots,x_n\},Y=\{y_1,y_2,\cdots,y_n\}$，且 $X\cap Y=\varnothing$，$A=\{f_{x1},f_{x2},\cdots,f_{xp}\}$，$B=\{f_{y1},f_{y2},\cdots,f_{yq}\}$ 分别是与 X,Y 中属性相应的模糊集集合。序偶集<X, A>表示 X 中的属性取 A 中的相应值。

定义 1 序偶集<X, A>表示“项集-模糊集”，X 是属性 x_i 的集合，A 是模糊集 a_j 的集合，则支持度：

$$sup(<X,A>)=\sum_{\in D}(\wedge_{x_j}\in X\{\alpha_{a_j}(d_i[x_j])\}/|D|$$

其中，$\alpha_{a_j}(d_i[x_j])=\mu_{a_j}\in A(d_i[x_j])$；$d_i[x_j]$ 为第 i 个事务中 x_j 的值。若 sup (<X,A>)$\geqslant minsup$，则称为频繁序偶集；若 X 包含 k 个属性，则 $<X,A>$ 是频繁 k-序偶集。

定义 2 频繁集 $<Z,C>$ 生成规则 $<X,A>\Rightarrow<Y,B>$：if X is A then Y is B，其中 $X\subset Z,Y=Z-X,A\subset C,B=C-A$。则置信度 $conf(<X,A>\Rightarrow<Y,B>)=sup<X\cup Y,A\cup B>/sup<X,A>$。若 $<X,A>,<Y,B>$ 都是频繁序偶集，且 $conf(<X,A>\Rightarrow<Y,B>)\geqslant minconf$，则称 $<X,A>\Rightarrow<Y,B>$ 是强模糊规则。

常用的模糊关联规则挖掘算法都是基于 Apriori 算法思想[1-2]。该类算法主要采用候选项集生成-筛选方法，从而必须耗费大量时间和空间来处理规模巨大的候选项集，同时在对候选项集的筛选过程中要对数据库进行多次扫描。而本文提出一种基于前缀树的模糊关联规则挖掘算法(FARMBPT)，利用前缀树来压缩存储模糊模式集，节约了内存开销，且在整个挖掘过程中只需扫描数据库 2 遍，从而提高了模糊关联规则挖掘算法的性能。

2 FARMBPT算法

FARMBPT 算法基于构造前缀树去存储模糊模式集，这种前缀树称为模糊模式前缀树。

2.1 模糊模式前缀树

在 FARMBPT 算法中，模糊模式前缀树记录的是事务数据库中每个事务的模糊频繁项。模糊模式在模糊模式前缀树中依字典排序[3]，包含在模糊模式前缀树中的模糊 k-模式的数目等于前缀树中 $L[k]$所在层次的模糊概念的数目，而相应

基金项目：国家自然科学基金资助项目(60603008)；广西教育科学课题基金资助项目(2006A-E004)

作者简介：程 江(1983－)，男，硕士研究生，主研方向：数据挖掘，信息安全；易云飞，助教、硕士研究生；林建辉，副教授、博士研究生；余启港，副教授

收稿日期：2008-08-29 **E-mail：**jiang616@gmail.com

的模糊 *k*-模式则可以由从根节点到 *L*[*k*]所在层次的模糊概念连接生成，且模糊 *k*-模式的支持度和模糊模式的第 *k* 个模糊概念存放在一起。在模糊模式前缀树中，每个节点由 4 个域组成：节点名称，从根节点到该节点所连接成模糊模式的支持度，子女节点链和父节点指针。图 1 是有 4 个频繁项的完全前缀树，表示 4 个频繁项所构成的所有可能模式。

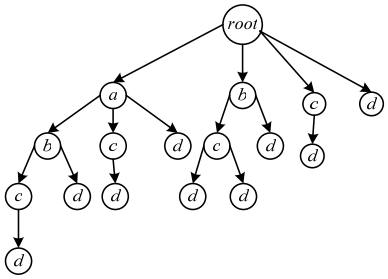

图 1　频繁项 *a, b, c, d* 的完全前缀树

在模糊模式前缀树中，如果 $\alpha.i_k$ *-prefix* 包含的模式是频繁的，则 $\alpha.i_k.i_{k+1}$ *-prefix* 包含的模式才可能是频繁的。如果 $\alpha.i_k$ *-prefix* 包含的模式不频繁，则 $\alpha.i_k.i_{k+1}$ *-prefix* 包含的模式一定不频繁。其中，i_k 表示单个前缀项；α 表示通用前缀项。

该性质提供了一个很好的挖掘策略：如果某节点上的支持度小于支持度阈值 *minsup*，即该节点表示的模式不频繁，则以该节点为根的整个子树不需考虑。因此，利用该性质可以显著地缩小搜索空间，有效地避免组合爆炸，极大地提高挖掘效率。

2.2　FARMBPT算法流程

FARBPT 算法主要有 3 步：(1)扫描数据库一遍得到各项目的支持度，根据最小支持度 *minsup* 得到频繁 1-项目集，并依字典顺序排成频繁 1-项目表 L_1；(2)根据表 L_1 中的频繁 1-项目集创建和初始化模糊模式前缀树，再扫描数据库一遍，计算由表 L_1 中频繁 1-项目连接生成的候选项集的支持度，并插入到模糊模式前缀树中；(3)根据模糊模式前缀树的有关性质，进行深度优先遍历，挖掘产生强模糊关联规则。具体算法如下：

算法　从模糊数据集中挖掘强模糊关联规则

输入　模糊数据集 *D*、最小支持度 *minsup*、最小可信度 *minconf*

输出　强模糊关联规则集

```
//获得频繁 1-项目表 L1
(1)for each  Transaction Ti  in  FuzzyDataSet D{
(2)for each  FuzzyPattern FPj  in  FuzzyDataSet D{
(3)FPj.vote  += Ti [FPj]; }} //计算模糊数据集 D 中模糊模式 FPj
//的隶属度
(4)for each  FuzzyPattern FPi  in  FuzzyDataSet D{
(5)FPi.sup = FPi.vote /|D|; //判断是否频繁模糊模式，若是则
(6)if(FPi.sup > minsup) Insert(L1, FPi);} //将其添加到频繁 1-项目
//集表 L1 中根据频繁 1-项目表 L1 生成模糊模式前缀树
(7)CandidateItemSet CIS = GenerateCandiateItemSet(L1);
(8)FuzzyPrefixTree  FPT = CreateFuzzyPrefixTree(L1);
(9)for each  Transaction Ti  in  FuzzyDataSet D{
(10)for each  CandidateItem Cj  in  CandidateItemSet CIS{
(11)Cj.vote += GetCandidateItemFrq(Cj, Ti);}}
  //计算候选项 Cj 的隶属度
(12)for each CandidateItem Cj in CandidateItemSet CIS{
(13)Cj.sup = Cj.vote/|D|;
(14)GrowFuzzyPrefixTree(Cj, FPT);}  //将候选项 Cj 插入到模糊
//模式前缀树遍历模糊模式前缀树，生成强模糊关联规则集
(15)PrefixTreeNode root_node = FPT.root;
(16)for each PrefixTreeNode chld_node[i] in root_node{
(17)TraverseFuzzyPrefixTree (chld_node[i], minsup, minconf);}
```

其中，GenerateCandiateItemSet 子过程完成对表 *L* 中频繁 1-项目进行连接组合，产生所有的候选项集 *CIS*。在连接组合时，要避免同一属性的模糊模式组合在一起，因为这样的项集对于最后产生的关联规则没有实际意义，同时也减少了存储空间和降低了计算量。CreateFuzzyPrefixTree 子过程完成创建和初始化模糊模式前缀树(FPT)，即创建 *root* 根节点和将表 *L* 中频繁 1-项目插入到 FPT 中。GetCandidateItemFrq 子过程完成计算当前事务中某一候选项的模糊隶属度。GrowFuzzyPrefixTree 子过程完成将候选项插入到 FPT 中。TraverseFuzzyPrefixTree 子过程完成深度优先遍历 FPT，并根据最小支持度和最小可信度产生强模糊关联规则。

2.3　应用实例

作为示例，使用表 1 所示的模糊数据集对上述算法进行分析，该模糊数据集包含了与网络流量相关的 4 个属性：TCP 和 UDP 包在全部数据包中的比例 *Ptcp* 和 *Pudp*、网络中每秒的平均数据包数量 Avg.Packet/s 以及每秒平均数据位 Avg.Mb/s。且分别将 *Ptcp*, *Pudp*, Avg.Packet/s 和 Avg.Mb/s 分成了 high 和 low 2 种模糊模式。

表 1　模糊数据集

TID	*Ptcp*/(%)		*Pudp*/(%)		Avg.Packet/s		Avg.Mb/s	
	T.L	*T.H*	*U.L*	*U.H*	*AP.L*	*AP.H*	*AM.L*	*AM.H*
1	0.00	0.88	1.00	0.0	0.00	0.62	0	1.00
2	0.00	0.92	0.91	0.0	0.00	0.75	0	0.92
3	0.00	0.56	1.00	0.0	0.00	1.00	0	1.00
4	0.00	0.92	0.00	1.0	0.00	0.73	0	0.86
5	0.00	0.87	0.53	0.0	0.89	0.00	1	0.00
6	0.00	0.00	0.00	0.5	0.00	0.61	0	1.00
7	0.00	1.00	0.56	0.0	0.00	0.93	0	1.00
8	1.00	0.00	1.00	0.0	0.00	1.00	0	0.91
9	0.73	0.00	1.00	0.0	0.63	0.00	1	0.00
10	0.91	0.00	1.00	0.0	0.00	1.00	0	0.83

取 *minsup*=0.25，由表 1 可得到频繁模糊 1-项目集 L_1 = {*T.L, T.H, U.L, AP.H, AM.H*}，依据上述算法，可生成模糊模式前缀树，如图 2 所示。

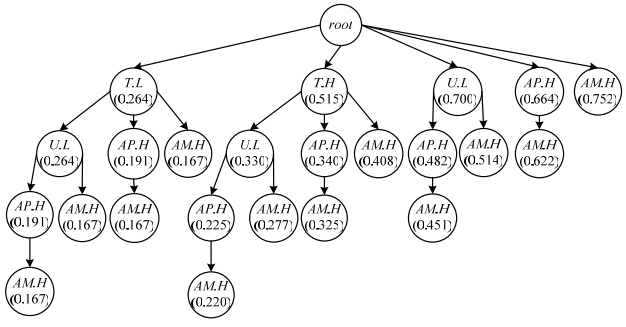

图 2　模糊模式前缀树

取 *minsup* = 0.25, *minconf* = 0.60 可挖掘出如下强模糊规则：

R_1: T.H ∧ U.L → AM.H　s = 0.286, c = 0.878

R_2: T.H ∧ AP.H → AM.H　s = 0.329, c = 0.971

…

这些规则在某种程度上揭示了属性之间的潜在关系，如规则 R_1 表示当网络流中 TCP 包数量为高，且 UDP 包为低，则每秒平均数据位为高。

3　算法实现与对比

为检测算法的性能，在配置为 AMD Athlon 64 X2，

（下转第 72 页）