计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(4):194−198 [doi: 10.15888/j.cnki.csa.006856]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于分类修剪的关联分类算法改进①

秦晨普, 张云华

(浙江理工大学 信息学院, 杭州 310018)
通讯作者: 秦晨普, E-mail: 1095247329@qq.com

摘　要: 针对现有关联分类算法资源消耗大、规则剪枝难、分类模型复杂的缺陷, 提出了一种基于分类修剪的关联分类算法改进方案 ACCP. 根据分类属性值的不同对分类规则前项进行分块挖掘, 并对频繁项集挖掘过程和规则修剪进行了改进, 有效提高了分类准确率和算法运行效率. 实验结果表明, 此算法改进方案相比传统 CBA 算法和 C4.5 决策树算法有着更高的分类准确率, 取得了较好的应用效果.

关键词: 关联分类; 分类修剪; 事先剪枝; ACCP

引用格式: 秦晨普,张云华.基于分类修剪的关联分类算法改进.计算机系统应用,2019,28(4):194–198. http://www.c-s-a.org.cn/1003-3254/6856.html

Improved Association Classification Algorithm Based on Classification Pruning

QIN Chen-Pu, ZHANG Yun-Hua

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Aiming at the shortcomings of the existing association classification algorithm, such as large resource consumption, difficult rule pruning, and complex classification model, an improved classification scheme ACCP based on classification and pruning is proposed. The algorithm mines the fore items of classification rules respectively according to the different classification attribute values, and improves the frequent item set mining process and rule pruning, which effectively improves the classification accuracy and algorithm operation efficiency. The experimental results show that the improved algorithm has higher classification accuracy than traditional CBA algorithm and C4.5 decision tree algorithm, and has achieved satisfied application results.

Key words: association classification; classified pruning; pre-pruning; ACCP

1998 年, 新加坡国立大学的 Liu Bing 教授提出了一种将关联规则挖掘和分类技术结合在一起的分类算法——关联分类算法 (Classification-Based Association, CBA)[1], 因其较好的结合了关联规则挖掘和传统分类算法的优点, 受到了研究者的广泛关注. 实践证明, 关联分类算法相较于决策树、朴素贝叶斯、SVM 支持向量机等传统的分类算法具有更优的分类性能且分类模型更易理解. 另一方面, 关联分类算法是基于传统 Apriori 算法挖掘事务项之间的关联来产生分类规则, 也因此也不可避免的继承了关联规则挖掘算法需要多次扫描数据库、I/O 负载较大的缺点, 算法效率不是很理想. 之后的研究者又先后提出了关联决策树 (Association based Decision Tree, ADT) 方法[2]、基于多关联规则的分类算法 (Classification based on Mutiple Association Rules, CMAR)[3]、等, 虽说在算法性能的提升上取得了一定的成果, 但也或多或少的存在着算法鲁棒性差、冗余节点多的问题.

在现有 CBA 关联分类算法的基础上, 本文提出了一种基于分类修剪的关联分类算法改进方案 ACCP, 在分类关联规则的挖掘过程中基于分类标识对事务数

① 收稿时间: 2018-09-23; 修改时间: 2018-11-12; 采用时间: 2018-11-22; csa 在线出版时间: 2019-03-28

据集进行分类修剪, 并加入了基于最大频繁项集的事先剪枝, 避免了无法生成规则的无效连接操作, 有效提高了规则挖掘效率. 同时, 借鉴已有的研究成果在构造分类器的过程中利用改进后的数据覆盖法对分类规则进行修剪, 提高分类准确率.

1 概念描述

关联分类实质上就是基于关联规则挖掘的分类技术, 它既反映了知识的应用特点——分类或预测, 又体现了知识内在的关联特性[4]. 设 D 是一个包含着 n 条记录的事务数据集, $I=\{i_1,i_2,i_3,\cdots,i_m\}$是全体事务项的集合, I 的子集一般称为项集, 根据子集中事务项的个数依次可称为 1-项集, 2-项集, …, k-项集. 数据库中每条事务记录 $t_i(i=1,2,3,\cdots,n)$ 均对应着 I 的一个子集, 且具有唯一标识符 TID. $\|D\|$表示数据集 D 中包含的事务数量.

定义 1. 项集 X 的支持度: 数据集中包含项集 X 的事务出现的频率, 记为

$$support(x)=\frac{|\{T|X\subseteq T,T\subseteq D\}|}{|D|}\times 100\% \qquad (1)$$

其中, $|\{T|X\subseteq T,T\subseteq D\}|$代表的是事务数据集 D 中包含项集 X 的事务总数, 即项集的支持数.

定义 2. 频繁项集: 若项集的支持度超过或等于人为设定的最小支持度阈值 $minSupp$, 则称此项集为频繁项集.

定义 3. 最大频繁项集: 如果一个频繁项集的任一直接超集都是非频繁项集, 那么就称这个频繁项集为最大频繁项集.

定义 4. 规则置信度: 假设数据集中关联规则 $X\Rightarrow Y$ 成立, 则其置信度是指包含项集 X 的事务同时包含项集 Y 的概率, 其表述的是规则的可靠性, 表达式为:

$$conf(X\Rightarrow Y)=\frac{|\{T|X\cup Y\subseteq T,T\subseteq D\}|}{|\{T|X\subseteq T,T\subseteq D\}|}\times 100\% \qquad (2)$$

定理 1. 项目集空间理论: 频繁项集的子集仍是频繁项集, 非频繁项集的超集是非频繁项集[5].

2 基于分类修剪的关联分类模型 ACCP

2.1 基于分类标识的规则挖掘

之前的研究人员所运用的基于分类标识的规则后项约束, 大多先由频繁 k–1 项集的集合 L_{k-1} 自连接生成候选 k 项集的集合 C_k, 再对包含分类标识的候选 k 项集进行基于最小支持度阈值 $minSupp$ 的剪枝操作. 实际上, 当频繁 k–1 项集 I_1 作为规则前项只出现在分类标识为 C_i 的事务中时, 那么对分类标识不等于 C_i 的候选 k 项集$\{I_1, C_{i+1}\}$进行支持度计数就显得没有必要, 本文基于此思想对分类关联规则的挖掘过程进行了改进.

将事务数据集 D 根据分类属性值的不同, 划分为多个事务子集$\{D_1, D_2, D_3, \cdots, D_n\}$, 其中 n 为分类属性值的个数, 每个事务子集中挖掘得到的规则项集具有统一的分类标识. 对每个子集进行单独的分类关联规则挖掘, 在分类标识为 C_i 的事务子集中, 项集$\{I_i\}$的支持数和事务总集中包含分类标识 C_i 的规则项集$\{I_i, C_i\}$支持数一致, 只要根据项集 I_i 的支持数进行连接剪枝即可, 从而大幅的降低了每次扫描数据库时的数据维度, 避免了无法生成规则的项集的产生, 减少了候选项集的数量.

2.2 基于最大频繁项集的事先剪枝

原始的关联规则挖掘过程有两次剪枝操作, 第一次是在 L_{k-1} 自连接之后, 根据 Apriori 算法性质 (项目集空间理论) 剪除非频繁项集, 第二次是由候选项集 C_k 生成 L_k 时, 通过计算项集支持度剪除非频繁项集. 本文将在此基础上加入一次基于最大频繁项集的事先剪枝, 即在自连接之前利用项目集空间理论, 提前判断出频繁 k–1 项集的集合 L_{k-1} 中的某些最大频繁项集, 将其进行剪除, 从而省去了它们的连接操作, 进一步减少了候选项集数, 提高了分类关联规则的挖掘效率.

根据项目集空间理论, 频繁 k-项集的所有子集均为频繁项集. 由此可得, 每个频繁 k-项集可抽取出 k 个频繁 k–1 项子集, 则包含这 k 个频繁 k–1 项集的集合 L_{k-1} 当中每个事务项出现的次数必然大于等于 k–1. 下面用一个简单的例子说明这个原理.

已知 4-项集{a, b, c, d}为频繁项集, 其有 4 个频繁 3-项子集, 分别为{a, b, c}、{a, b, d}、{a, c, d}、{b, c, d}, 则包含项集{a, b, c}、{a, b, d}、{a, c, d}、{b, c, d}的集合 L_3 中, 事务项 a、b、c、d 出现的次数都至少为 3. 反之, 若 a、b、c、d 任意一个事务项在 L_3 中出现次数小于 3, 则 4 个 3-项集中至少有一个不包含于 L_3 即不是频繁项集, 由此可得{a, b, c, d}亦不是频繁项集. 推而广之, 不难得出:

定理 2. 频繁 k–1 项集中存在事务项在集合 L_{k-1} 中出现次数小于 k–1 次是此频繁项集为最大频繁项集的

充分条件.

对最大频繁项集事先剪枝的具体实现步骤如下:

(1) 计算频繁 k–1 项集的集合 L_{k-1} 中每个事务项出现的次数, 用 $L_{k-1}(p)$ 表示;

(2) 记录下出现次数小于 k–1 的事务项, 记作 $P=\{p||L_{k-1}(p)|<k-1\}$;

(3) 将 L_{k-1} 中包含有 P 中任一元素的频繁项集删除, 记为 L_{k-1}';

(4) L_{k-1}'自连接, 生成候选 k-项集的集合 C_k.

2.3 实例分析

我们以表 1 所示的事务数据集为例, 简单阐述一下改进后的关联分类算法的分类关联规则挖掘过程.

表 1 数据库示例

TID	事务	类别属性
001	a, b, c	A_1
002	a. b, c, e	A_1
003	a, b, d	A_2
004	a, c	A_1
005	b, d, e	A_2
006	a, c, d, e	A_2
007	a, b, d	A_1
008	a, d, e	A_2
009	b, c, d	A_1
010	d, e	A_2

由表 1 可得, 事务数据集 D 有两个类别属性值 A_1, A_2, 可将事务数据集分为表 2, 表 3 所示的事务子集 D_1, D_2.

表 2 分类得到的事务子集 D_1

TID	事务	类别属性
001	a, b, c	A_1
002	a. b, c, e	A_1
003	a, c	A_1
004	a, b, d	A_1
005	b, c, d	A_1

表 3 分类得到的事务子集 D_2

TID	事务	类别属性
001	a, b, d	A_2
002	b, d, e	A_2
003	a, c, d, e	A_2
004	a, d, e	A_2
005	d, e	A_2

C_k 表示候选 k-项集的集合, L_k 表示频繁 k-项集的集合, R_i 表示事务子集 D_i 中挖掘出的分类规则集, 假定最小支持数 $minSupp$ 为 2, 首先对事务子集 D_1 进行规则挖掘. 如图 1 所示, 第一次扫描事务子集 D_1 后得到候选 1-项集的集合 C_1 及其中各项集所对应的支持数, 将 C_1 中支持数小于最小支持数 $minSupp$ 的项集剪除便得到频繁 1-项集的集合 L_1, 图 1 左上表便是 C_1 到 L_1 的剪枝过程, 其中边框为虚线的项集即为被剪枝的非频繁项集. 将 L_1 自连接可得到候选 2-项集的集合 C_2, 基于最小支持数 $minSupp$ 剪枝后得到频繁 2-项集的集合 $L_2=\{\{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}\}$.

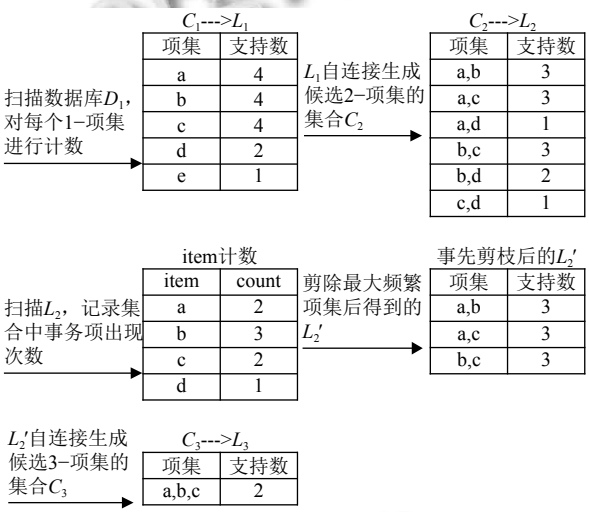

图 1 分类关联规则挖掘过程示例

接着对集合 L_2 进行遍历, 记录 L_2 中每个事务项出现的次数. 不难发现, 事务项 a 同时包含于频繁 2-项集 {a, b}、{a, c}, 即事务项 a 在 L_2 中出现了 2 次, 同理可得事务项 b 出现 3 次, 事务项 c 出现 2 次, 事务项 d 只出现了 1 次. 由于属性项目 d 出现的次数小于 L_2 中项集的项数, 由定理 2 可得 L_2 中所有包含项目 d 的项集均为最大频繁项集, 将其剪除后即得到最终的 L_2'. 由 L_2'自连接可得到候选 3-项集 C_3={{a, b, c}}并最终确定 L_3={{a, b, c}}, 因其无法再进行自连接, 频繁项集挖掘结束. 将最终生成的集合 L 中所有频繁项集加入分类标识 A_1 便得到分类关联规则集 R_1, 循环挖掘所有事务子集并将最后得到的分类规则集合并便得到整个数据库的分类关联规则集 R.

算法 1.

输入: 数据库 D, 最小支持度阈值 $minSupp$;
输出: 分类规则项集的集合 R;

Step 1. 根据分类标识的不同将事务数据集 D 分为事务子集 D_k={D_1, D_2, D_3, ……, D_n}, 其中 n 为分类属性的值的数量;
Step 2. for(i=1; D_i≠Ø; i++) do
Step 2.1. 扫描事务子集 D_i 得到规则前项的频繁 1-项集的集合 L_1;
Step 2.2. for(k=2; L_{k-1}≠Ø; k++) do
Step 2.2.1 扫描 L_{k-1}, 根据每个属性项目在其中出现的次数判断最大频繁项集并剪除, 得到 L_{k-1}';
Step 2.2.2. L_{k-1}'自连接生成候选 k-项集的集合 C_k;
Step 2.2.3. 依次扫描 D_i 中所有事务记录, 若包含 C_k 中的候选集, 则此候选集的支持数加 1;
Step 2.2.4. 对 C_k 进行基于最小支持数 *minSupp* 的剪枝得到 L_k
Step 2.3. 得到事务子集 D_i 中的分类规则集 R_i =∪L_k;
Step 3. 获得最终的分类关联规则集 R=∪R_i;

2.4 规则修剪

由于分类关联规则挖掘所得到的规则数量巨大, 在构造分类器时会占用大量内存空间, 并且会对分类准确率产生不利影响, 本文基于改进后的数据库覆盖方法对规则集进行规则修剪.

首先基于置信度、支持度从大到小以及规则项集维度从小到大的方式对分类规则进行优先级排序. 从优先级最高的规则依次进行考察, 遍历事务数据集记录下正确分类的比例并将此规则所能覆盖的所有事务样本删除, 直到没有剩余样本或已考察完所有规则. 最后删除分类性能较差的规则并多次执行以上步骤不断提高规则集的分类准确率[6]. 规则修剪的算法一般性描述如下:

算法 2.

输入: 初始规则集 R, 事务数据集 D, 最小置信度 *minConf*
输出: 改进后的关联分类器 C

Step 1. 新建临时数据集 X, 令其等于事务数据集 D;
Step 2. 扫描数据集 X, 计算 R 中每条未标记规则的置信度, 删除 R 中不满足最小置信度 *minConf* 要求的规则;
Step 3. 将分类规则集依次按照置信度、支持度从大到小以及规则项集维度从小到大的方式进行优先级排序;
Step 4. 选择最高优先级的规则进行考察, 记录下该规则在数据库 X 中正确分类的样本数与规则覆盖样本数之比, 并删除所有覆盖样本;
Step 5. 标记已遍历过的规则, 对剩余的样本循环执行 Step 2~Step 4, 直到数据库中没有样本或所有规则均已被遍历. 如果存在未被遍历的的规则, 则将该规则剪除; 如果存在所有规则均不能对其分类的样本, 则将此样本归为数据库中样本数量最多的类别即默认类别;
Step 6. 按照正确分类的样本数占覆盖样本数的百分比对规则进行排序, 剪除正确分类比为 0 或正确分类比不为 0 但排名最后的规则. 计算当前规则集对数据库 X 的整体分类正确率;
Step 7. 循环执行 Step 1~Step 6, 直到规则集对数据库的分类正确率不再提高为止, 得到最终的分类规则集 R.

3 实验与结果分析

3.1 实验环境

本次实验的实验环境如下: Intel(R)Core(TM) i5-2450M CPU @2.50 GHz 处理器; 8 G 内存; 120 G SSD 固态硬盘; Windows 10 专业版操作系统. 实验选取了 UCI 标准数据库中的 5 个常用数据集 Pima Indians Diabetes、Lymphography、Wine、Car Evaluation、Iris[7]分别涵盖医疗卫生、食品检测、汽车评估、生物研究等领域, 每个数据集的相关数据信息如表 4 所示. 本实验算法程序使用 Java 语言实现.

表 4 实验所用数据集相关信息统计

数据集	事务数	属性数	分类类别数
Pima Indians Diabetes	768	7	2
Lymphography	148	18	4
Wine	178	13	3
Car Evaluation	1728	6	4
Iris	150	4	3

3.2 实验结果分析

本实验使用了 10 折交叉验证方法来避免过度拟合, 从每个数据集中随机选取 80% 的样本作为训练数据集, 其余 20% 作为测试数据集测试算法的分类性能. 针对数据集中存在的数据缺失, 根据缺失的属性值是离散还是连续, 分别采用众数原理将其设定为该属性在数据集中出现次数最多的取值, 或是设定为数据集中该属性其他非缺失值的平均数. 本文选取了现有的 CBA 算法以及传统分类算法中的 C4.5 决策树算法进行对照试验, 实验中最小支持度 *minSupp* 和最小置信度 *minConf* 分别设定为 2% 和 60%, 结果如表 5 所示.

本文采用正确分类的样本数占测试样本总数的比例即分类准确率来衡量分类器模型的优劣. 从表 5 中可以看出, 关联分类算法的准确率整体高于传统的 C4.5 决策树算法. 在部分数据集上 CBA 算法的分类准确率等于或略小于 C4.5 算法, 而改进后的关联分类算法 ACCP 则在全部数据集上都明显优于 C4.5 和 CBA, 平均分类准确率分别提高了 5.29 和 3.37 个百分点. 与此同时, 基于分类修剪并加入了预先剪枝的 ACCP 算法在实验所采用的所有数据集上的运行时间均较 CBA 算法有所降低, 在数据集属性较多、事务数较大更为明显. 实验结果证明, ACCP 算法取得来了良好的应用效果.

表 5 实验结果对比

数据集	分类准确率 (%)			运行时间 (s)		
	C4.5	CBA	ACCP	C4.5	CBA	ACCP
Pima Indians Diabetes	75.65	74.48	79.43	6.0	7.9	7.2
Lymphography	79.05	85.14	90.54	12.9	13.6	9.5
Wine	92.70	94.38	96.07	12.0	13.9	13.1
Car Evaluation	84.09	86.69	89.00	23.0	29.4	21.6
Iris	94.67	94.67	96.67	1.7	1.5	1.5
Average	85.23	87.07	90.34	11.1	13.3	10.6

4 结语

本文提出了一种基于分类修剪的新关联分类算法 ACCP, 通过将事务数据集根据分类标识分块挖掘, 极大地节省了内存空间, 提高了挖掘效率, 同时在分类器构造过程中加大规则修剪力度, 剪除了规则集中分类性能较差的冗余规则, 进一步优化了分类模型. 实验证明, 本文提出的方法相比传统的 C4.5 决策树和 CBA 分类模型具有更优的分类性能.

基于关联规则产生分类器的过程并不有助于人们对分类模型的理解, 反而会影响分类器的性能. 因此, 如何有效减少构建分类器时所使用到的规则数量, 提高单个规则的适用性, 是接下来要研究解决的问题.

参考文献

1 Liu B, Hsu W, Ma YM. Integrating classification and association rule mining. Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining. New York, NY, USA. 1998. 80–86.

2 Wang K, Zhou SQ, He Y. Growing decision trees on support-less association rules. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, USA. 2000. 265–269.

3 Li WM, Han JW, Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules. Proceedings of 2001 IEEE International Conference on Data Mining. San Jose, CA, USA. 2001. 369–376.

4 许立莎. 基于关联规则挖掘的分类算法研究[硕士学位论文]. 西安: 西安科技大学, 2012.

5 王振武. 数据挖掘算法原理与实现. 2 版. 北京: 清华大学出版社, 2017.

6 柴艺. 关联分类算法研究及其在海量慢病医疗数据挖掘中的应用[硕士学位论文]. 北京: 北京邮电大学, 2016.

7 UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/.