一种基于关联规则分类的改进方法

查金水　宋良图　刘现平
(中科院合肥智能机械研究所,合肥 230031)
E-mail:chajinshui@126.com

摘　要　论文首先对一种基于关联规则分类的算法做出了分析。然后对算法中的类关联规则的提取方法进行了改进,得到了一种新的基于关联规则分类的算法。并结合棉花病虫害数据运行的结果对两种算法的运行效率和实用性进行了比较。

关键词　关联规则　类关联规则　FP-树　分类

文章编号 1002-8331-(2006)10-0155-03　**文献标识码** A　**中图分类号** TP181

An Improved Method Based on Classification of Association Rules

Zha Jinshui　Song Liangtu　Liu Xianping
(Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031)

Abstract: From a concrete analysis of the classified algorithm based on association rule, we make for improvement of extraction method in class association rules and a new algorithm based on classification of association rules is acquired in this paper. Then a comparison of efficiency and practicability is made between the two algorithms according to the operation result of cotton disease data.

Keywords: association rule, class association rules, Frequent-pattern tree, classification

1　引言

对于一些大的数据库来说,建立一个精确而又有效的分类器是数据挖掘和机器学习的一个重要任务——给定一个带有类别标签的测试数据集,用它来建立一个分类器,然后预测那些未知类别的数据对象。现在的许多分类方法都是基于启发式的搜索技术,比如决策数算法[1]、Bayes 网络和一些统计学的方法。还有一些在商业化的数据挖掘领域内很少用到的方法,诸如 k-最近邻分类、基于案例的推理、遗传算法等。

分类规则的挖掘和关联规则[2]的挖掘是两种重要的数据挖掘技术。分类规则挖掘的目标就是找出数据库中的一些规则,组成一个精确的分类器。而关联规则的挖掘就是找出数据库中满足最小支持度与最小确信度约束的规则。对关联规则来说,它的目标是没有预先确定的。而对分类规则的挖掘来说,它有一个预先确定的唯一的目标,即类别标签。分类规则和关联规则的挖掘在实际中都是不可缺少的。因此,数据挖掘技术也已将关联规则挖掘用于分类问题[3]。将两种挖掘技术结合起来对使用者来说既节省时间又方便很多。这两种技术的结合可以产生一种新的分类方法:基于关联规则的分类。在关联规则分类中,规则的右侧固定为类别的属性。我们将这些规则称为类关联规则(class association rules, CARs)[4]。

数据挖掘中类关联规则的挖掘主要包括下面几个步骤(如图 1 所示):

(1)如果是连续的属性值,需要将其离散化。

(2)产生所有的类关联规则。

(3)基于已经产生的类关联规则建立一个分类器。

(4)利用分类器对未知类别数据进行分类。

如图 1 所示,我们首先假设数据集是一个正常的关系表。这个表中包含了带有 L 个不同属性值的 N 个案例,这 N 个案例已经被划分为 q 个已知的类。属性值可以是离散的也可以是连续的。对于一个连续的属性值,我们首先将其值的区间离散化成许多小区间,然后再将这些小的区间映射到一系列连续的整型值。

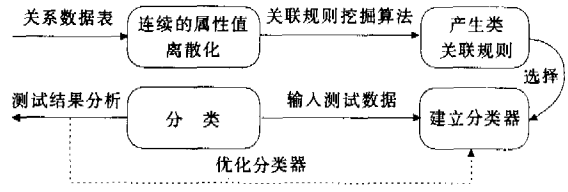


图 1　基于关联规则分类的流程图

设 D 为事务集,I 是 D 中所有项目的集合、Y 是类别标签的集合。如果 $X\subseteq d$,我们称一个数据项 $d\subseteq D$ 包含 $X\subseteq I$,I 是数据项 D 的子集。一个类关联规则(CAR)就是下面的形式 $X\rightarrow Y$,其中 $X\subseteq I$、$y\in Y$。它的支持度与确信度的定义如下:规则 R:$X\rightarrow Y$ 的支持度 sup 是指 D 中有 s%的案例包含有带有类别标签 Y 的项目 X。sup 与 D 中的含有 X 的案例数之比称为确信度

基金项目:国家 863 高技术研究发展计划资助项目(编号:2003AA118070)

作者简介:查金水(1978-),男,硕士研究生,主要研究方向:数据挖掘,复杂系统。宋良图(1963-),男,副研究员,主要研究方向:智能化农业信息系统。刘现平(1979-),男,硕士研究生,主要研究方向:图像检索系统,图形与图像处理。

conf。我们的目标就是根据使用者给定的最小支持度 *minsup* 和最小置信度 *minconf* 阀值来产生所有的 CARs 集，然后根据产生的 CARs 建立一个分类器。

2 CBA 算法的描述

CBA 算法[5]包含两个步骤：类关联规则的产生和分类器的建立。

2.1 基本概念

产生规则的首要条件就是要找出所有大于最低支持度阀值的规则。一个规则项 ruleitem 的形式如下：<*condset*,*y*>。这里的 *condset* 是一项集，$y \in Y$ 是一个类别标签。*Condset* 的支持度(condsupCount)是指 D 中包含 *condset* 的数。规则项 *ruleitem* 的支持度(rulesupCount)是指 D 中类别标签是 y 的 *condset* 的数。每一个 *ruleitem* 可以表示一条规则：*condset*→*y*。它的支持度是(*rulesupCount*/|*D*|)*100%，这里|*D*|是数据集的大小。它的确信度是(*rulesupCount*/*condsupCount*)*100%。

对于有同样 condset 的项集来说，确信度最高的将被选为可能的规则 PR(possible rules)来代表这个项集。如果有超过一个的项集具有相同的最高的确信度，我们将随机的选择一个项集。如果一条规则的确信度大于最低确信度阀值，我们说这条规则是精确的。而类关联规则集就是包含那些既频繁又精确的所有的 PR。

2.2 产生类关联规则

CBA 产生类关联规则的算法 CBA-RG 如下：

1 F_1={large 1-*ruleitems*};
2 CAR_1=*genRules*(F_1);
3 $prCAR_1$=*pruneRules*(CAR_1);
4 for(k=2;$F_{k-1}\neq\emptyset$;k++)do
5 C_k=*candidateGen*(F_{k-1});
6 for each data case $d \in D$ do
7 C_d=*ruleSubset*(C_k,d);
8 for each candidate $c \in C_d$ do
9 *c.condsupCount*++;
10 if *d.class*=*c.class* then *c.rulesupCount*++
11 end
12 end
13 F_k={$c \in C_k$|*c.rulesupCount*≥*minsup*};
14 CAR_k=*genRules*(F_k);
15 $prCAR_k$=*pruneRules*(CAR_k);
16 end
17 *CARs*=$\cup_k CAR_k$;
18 *prCARs*=$\cup_k prCAR_k$;

由上述的算法可以看出，在算法的每个循环中都要进行四个主要的操作。如在第 k 循环中，首先通过第 k-1 循环中的频繁项集 F_{k-1} 来产生频繁候选 k 项集 C_k，这一步主要是通过使用了 candidateGen 函数来实现。接着扫描数据库来更新 C_k 中各个候选集的支持度计数，然后这些新的频繁项组成新的 F_k。算法使用 genRules 函数来产生规则 CAR_k。最后使用对这些 CAR_k 规则进行剪枝。

2.3 建立分类器

为了在已经获得的规则上面建立一个最好的分类器，需要选择那些错误最少的规则。设 R 是所有已经产生的规则，D 是训练数据。算法的基本目的就是从 R 中选择一些优先度比较高的规则来替代 D。在这里优先度的定义为：给定两条规则，r_i 和 r_j，$r_i > r_j$(即 r_i 的优先度比 r_j 高)需满足下列条件：

(1)如果 r_i 的确信度比 r_j 的高，或

(2)两者的确信度相同，但是 r_i 的支持度比 r_j 的大，或

(3)两者的确信度和支持度相同，但 r_i 产生的比 r_j 早(也就是在规则的左手边 r_i 有更少的属性)；

我们建立的分类器的形式如下：

<r_1,r_2,…,r_n,*default_class*>，这里 $r_i \in R$，如果 $b>a$，则 $r_a > r_b$。default_class 是默认的类。在对一条未知类别的案例进行分类时，第一条满足这个案例的规则即可以分类这个案例。如果没有规则满足，则将这个案例归为默认的类。

分类器的建立有三个步骤：

(1)对所有 R 中的规则根据关系按降序排列。这确保我们的分类器可以选到优先度最高的规则。

(2)对于每条规则 $r \in R$，我们到 D 中去寻找可以被 r 替代(即它们满足规则 r 的左手边属性值)的案例，如果 r 至少可以正确分类，即可以替代，一个案例，它将是我们分类器的一条潜在的规则。对于那些可以被分类的案例将其从 D 中移出来。对于 D 中那些不能被规则 r 替代的案例，我们用 default_class 来标识。然后来计算由分类器和默认的类别号分类的错误的案例数。这里的 default_class 是指 D 中剩余案例中大部分案例所属的那个类。

(3)将分类器中那些不能增加分类器准确率的规则抛弃，剩下的未被抛弃的规则和 default_class 一起组成我们的分类器。具体的算法参见[5]。

3 改进的基本措施

由上述对 CBA 算法的描述我们可以看出，类关联规则的产生算法与 Apriori 算法类似。与 Apriori 不同的是在算法过程中要对两项进行支持度的计算，即 *condset* 和 *ruleitem*。这个主要是为了后面可以计算 *ruleitem* 的确信度。以前针对 CBA 算法的一些改进主要是集中在 CBA-RG 阶段，为了使数据库可以一次性的载入到内存中，对数据库进行划分，每部分采用单独的支持度计数。它对于数据库的划分和规则的产生采用不同的算法，选择效率最高的一个[6]。

在类关联规则的挖掘过程中，由于采用了 Apriori 算法[7]，需要不断地产生候选集，虽然利用 Apriori 性质，可以对候选集进行缩减以达到提高挖掘效率的目的，仍然存在两个问题：(1)产生的候选集过多；(2)需要对数据库进行反复扫描，通过一定的模式匹配的方式对大量候选集进行检验。为了避免产生的候选集过多，以及提高挖掘的效率，我们提出了一个新的基于关联规则分类的算法——NCBA。

为了发现分类规则，NCBA 首先挖掘训练数据，通过支持度和置信度阀值来发现所有的频繁项集。这也是一个典型的频繁模式关联规则挖掘任务。为了使挖掘的效率更高，NCBA算法使用 FP-树[8]算法。FP-树的频繁模式生成方法比 Apriori 类的方法更快，特别是对于大数据集、低的支持度阀值、长模式来说效率更高。通过使用 FP-树挖掘类关联规则来改进 CBA 算法的主要思想通过如下的例子来说明：

给定一个如表 1 所示的训练数据集 T。假定设最小支持度阀值是 2，确信度阀值是 50%。我们首先对数据集 T 进行扫描

一次，然后找出那些支持度大于2的那些项，项集 $F=\{a,d,f,k\}$ 称为频繁项集。其它的支持度小于最小支持度的项不能在关联规则中起到作用，所以将被剪枝。

表1 训练数据集

id	ID的列表	类别标签
1	a,c,f,i,l	A
2	a,d,f,j,m	B
3	b,e,g,k	A
4	a,d,h,k	C
5	a,d,f,k,n	C

然后对 F 中的项，按照支持度计数的降序排列，排列的结果是 F-list=a-d-f-k。然后再扫描一次训练数据集来构建一棵FP-树（如图2所示）。先创建树的根结点，用Null表示。接着我们按照F-list中出现的项和项的顺序对训练数据集中的每个元组进行选取。例如：在第一个元组中，只有(a,f)出现在F-list中，将其选取出来，作为最左边的一个分枝插入到树中。类别标签放在路径的最后一个节点上。

在训练集中的元组将会在树中分享一个共同的前缀。例如，第二个元组的属性值(a,d,f)。这样在F-list中将会和第一个元组分享同一个前缀 a。因此在FP-树中也同时分享最左边分枝的 a 的子路。所有相同属性值的节点作为一个队列从头节点开始连起来。

根据F-list，我们可以将类关联规则集划分为无重复的四个子集：(1)含有 k 的集；(2)含有 f 但是不含有 k 的集；(3)含有 d 但不含有 k 和 f 的集；(4)只含有 a 的集。

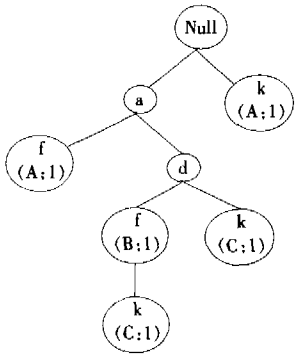


图2 FP-树

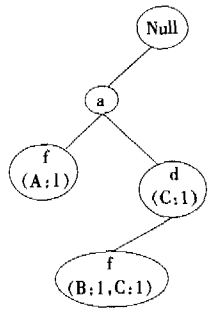


图3 合并节点k之后的FP-树

为了找到含有 k 的子集的规则，我们观察FP-树，遍历含有 k 指针的节点组成一个 k-db 数据库，我们可以发现 k-db 含有三个元组：$(a,d,f,k):C,(a,d,k):C,k:A$，这就是所有含有 k 的元组。在训练集中找出所有含有的频繁模式的问题就简化为在 k-db 数据库中挖掘频繁模式。

由上述可知，在 k-db 中，a 和 d 都是频繁的属性值，因为它们都大于或等于支持度的门槛值。又因为在 k-db 中，k 在每一个元组中都出现，因此必定是频繁的。所以我们也不需要计算 k 的支持度。我们可以通过循环的构造FP-树和db数据库来挖掘db数据库中的类关联规则。

在 k-db 数据库中，a 和 d 正好都是同时出现，因此 ad 是一个频繁模式。a 和 d 是 ad 的两个子集，与 ad 有相同的支持度。基于类别标签的信息，我们可以产生三条规则：$ak\rightarrow C,dk\rightarrow C,adk\rightarrow C$。它们三个的支持度皆为2、确信度皆为100%。

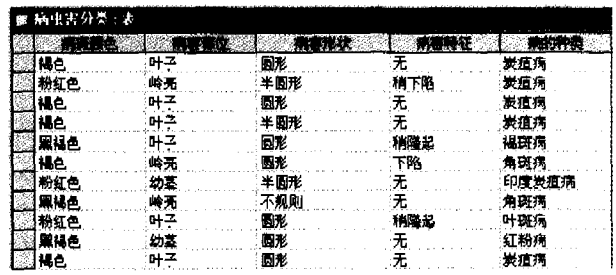
在搜寻到所有的含有 k 的规则之后，所有的 k 的节点都分别和它们各自的父节点合并。也就是说在 k 结点中的类别标签将在其父节点的标注。缩减之后的树如图3所示。余下的规则的提取同上述的类似。

由此我们可以看出，在对树的挖掘过程中，将原有的发现较长频繁模式的问题转化为反复寻找较短的模式而后再连接其前缀的过程。因此和CBA中采用的Apriori算法相比，不必重复扫描数据库。可以降低搜索成本，极大的提高效率。

由于FP-树在扫描数据库的过程中，需要将数据库一次性装入内存中来构建树。因此对机器的内存有一定的要求，如果数据库比较大的话，我们可以首先对数据库进行分割，然后再对每一个分割后的数据库用FP-树算法提取类关联规则。

4 实验结果及分析

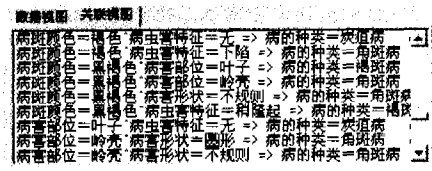
我们采用了某一地区的棉花病虫害数据为例测试算法的效果。本实验以Delphi 6.0为开发环境，数据存储在access数据库中，即为分类的样本空间。部分数据如图4所示。


棉虫害分类：表

病斑颜色	病害部位	病害形状	病害特征	病的种类
褐色	叶子	圆形	无	炭疽病
粉红色	岭秃	半圆形	稍下陷	炭疽病
褐色	叶子	圆形	无	炭疽病
褐色	叶子	半圆形	无	炭疽病
黑褐色	叶子	圆形	稍隆起	褐斑病
褐色	岭秃	圆形	下陷	角斑病
粉红色	幼茎	半圆形	无	印度炭疽病
黑褐色	岭秃	不规则	无	角斑病
粉红色	叶子	圆形	稍隆起	叶斑病
黑褐色	幼茎	圆形	无	红粉病
褐色	叶子	圆形	无	炭疽病

图4 棉花病虫害数据

由图4可以看出，我们将前面四个属性：病斑颜色、病害部位、病害形状、病害特征作为 *condset* 集，最后的一个属性：病的种类作为类别标签 Y。然后我们对数据进行转化，将这些文本数据转化为布尔型数据，以方便规则的挖掘。接着选定 *minsup*=15%，*minconf*=80%来进行类关联规则的挖掘，然后对挖掘得到的类关联规则建立分类器。我们使用样本中的90%的数据为训练数据，其余的为测试数据。图5即为得到的分类器中的类关联规则。


数据视图 关联视图

病斑颜色=褐色 病虫害特征=无 => 病的种类=炭疽病
病斑颜色=褐色 病虫害特征=下陷 => 病的种类=角斑病
病斑颜色=黑褐色 病害部位=叶子 => 病的种类=褐斑病
病斑颜色=黑褐色 病害部位=岭秃 => 病的种类=角斑病
病斑颜色=黑褐色 病害形状=不规则 => 病的种类=角斑病
病斑颜色=黑褐色 病虫害特征=稍隆起 => 病的种类=褐斑
病害部位=叶子 病虫害特征=无 => 病的种类=炭疽病
病害部位=岭秃 病害形状=圆形 => 病的种类=角斑病
病害部位=岭秃 病害形状=不规则 => 病的种类=角斑病

图5 训练样本为90%时得到的分类器中的类关联规则

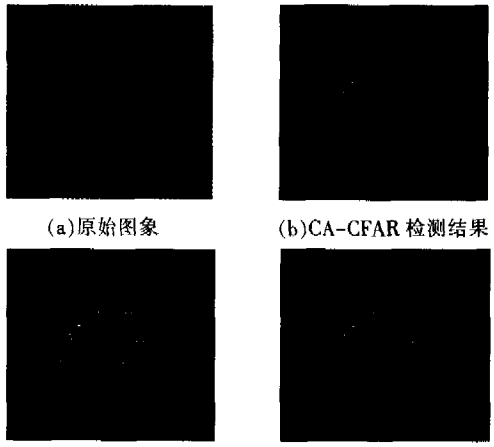
我们分别使用了三组数据进行了测试，样本数分别为40、200和1 000个。两种算法测试比较的结果如表2所示。

表2 两种算法运行时间比较 s

运行时间	样本数		
	40	200	1 000
CBA	0.3	2.5	21.3
NCBA	0.2	1.3	9.5

由实验结果可以看出，随着样本数目的增加，NCBA算法的运行时间明显比CBA算法少，效率增加。随着数据库的不断增大，CBA算法运行效率低的瓶颈就暴露出来了。因此改进后

（下转203页）



(a)原始图象 (b)CA-CFAR 检测结果

(c)本文 CFAR 方法检测结果 (d)本文方法最终结果

图 5 实验结果图像

检测结果中,CA-CFAR 方法有 2 个虚警、2 个漏警,本文方法有 1 个虚警,无漏警,且轮廓的完整性保持更好。可以看出,本文方法相比 CA-CFAR 方法,具有更好的检测效果,说明该方法是有效的。对检测结果进一步判别,虚警目标被滤除,得到目标的 ROI。

表 1 给出了三种方法在其他几幅图像的检测结果,比较得出:CA-CFAR 和 GO-CFAR 分别只对部分图像有较好的检测效果,本文方法对所有图像都有较好检测效果。在背景平稳区域,本文方法性能接近 CA-CFAR;在杂波边缘,CA—CFAR 虚警增多,而本文方法较好地抑制了虚警,检测性能接近 GO-CFAR;在多目标区域,CA-CFAR 由于目标间的相互干扰,检测出目标轮廓不完整甚至漏警,本文方法由于自动排除了干扰目标影响,比 CA-CFAR 和 GO-CFAR 具有更好的检测效果。在一幅图像中,杂波边缘和多目标情况经常可能同时存在,CSSO-CFAR 或 CSGO-CFAR 方法只是单纯地选大或者选小,不能对图象灰度分布变化自适应,当图像复杂时难以有好的效果。本方法的最大优势在于结合了各种方法的优点,智能判定区域类型,在各种复杂环境下都具有较好的检测性能。

表 1 几种方法的检测结果比较

图像序号	1	2	3
实有目标数目	17	14	23
CA-CFAR 结果	17	13	21
GO-CFAR 结果	15	14	20
本文方法结果	17	14	23

6 结论

本文提出了一种在 SAR 图像中检测目标的方法。该方法采用基于 weibull 分布模型的 CFAR 检测技术,对背景区域分块,根据每个子块的统计参数和空间分布,确定子块类型,根据各子块类型不同,选择不同的参考单元确定阈值。同时根据目标灰度、方差特征剔除明显不可能为目标的像素,利用多数滤波器和目标形状特征,进一步排除虚警。相比于 CA-CFAR 方法,本文方法保留了算法简单、同质区检测性能好的优点,同时,对存在杂波边缘或多目标干扰的情况,也能有很好的检测效果。实验证明本文方法检测性能好,自适应性强,适应于大多数 SAR 图像的目标检测。(收稿日期:2005 年 9 月)

参考文献

1.Quoc H Pham,Timothy M Brosnan,Mark J T Smith.Multistage Algorithm for Detection of Targets in SAR Image Data[J].SPIE,1997;3070

2.王世锦,孟健青.单元筛选后作最小选择的 CFAR 自适应检测器[J].雷达与对抗,2004;(4)

3.贾承丽,计科峰,匡纲要等.利用 Gamma CFAR 进行 SAR 图像目标检测[J].系统工程与电子技术,2005;(1)

4.何友,关键,孟祥伟.雷达自动检测和 CFAR 处理方法综述[J].系统工程与电子技术,2001;23(1)

5.Michael E Smith,Pramod K Varshney.Intelligent CFAR Processor Based on Data Variability[J].IEEE Transactions on Aerospace and Elect ronic Systems,2000

6.Char Ming Wong,Chee Hang Chang,Weixian Liu et al.CA-CFAR in Weibull Background[C].In:2nd International Conference on Microwave and Millimeter Wave Technology Proceedings,2000

7.M Skolnik,Radar Handbook[M].2nd edn.,McGraw Hill,1990

(上接 157 页)

的 NCBA 算法的实际应用性也较 CBA 有了很大的提高。

5 结论

本文通过对基于关联规则的分类方法 CBA 的分析,指出它的效率不足之处:它需要反复扫描数据库,而且会产生大量的候选集,通过一定的模式匹配方法对大量候选集进行检验。我们新的 NCBA 算法通过使用 FP-树来提取类关联规则,由于 FP-树算法比较简单,只需扫描一遍数据库,可以将较长的频繁模式转化为先寻找较短的模式而后再连接其前缀的过程。有效的降低了搜索成本。通过对算法的改进使运行效率有了很大的提高,并举例说明了获取类关联规则的具体过程。最后通过一个实际的例子说明两种算法的效果,证明了改进后算法的实用性和效率都有了很大的提高。(收稿日期:2005 年 11 月)

参考文献

1.Quinlan J R.C4.5:Programs for Machine Learning.California:Morgan Kaufmann,1993

2.Agrawal R,Imielinski T,Swami A.Mining Association Rules between Sets of Items in Large Databases[C].In:Proc of the ACM SIGMOD International conference on Management of Data,Washington D C,1993:207~216

3.Han J W,Kamber M.数据挖掘:概念与技术[M].北京:机械工业出版社,2001

4.Lent B,Swami A,Widom J.Clustering association rules[C].In:Proc of the 13th International Conference on Data Engineering,Birmingham,1997:220~231

5.Liu B,Hsu W,Ma Y.Integrating Classification and Association Rule Mining[C].In:Proc of the 4th International Conference on Knowledge Discovery and Data Mining,New York,1998

6.Liu B,Ma Y,Wong K.Improving an Association Rule Based Classifier[C].In:Proc of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases,Lyon,2000

7.Agrawal R,Srikant R.Fast Algorithms for Mining Association Rules[C].In:Proc of the 20th International Conference on Very Large Databases,Santiago,Chile,1994;9:487~499

8.Han J W,Pei J,Yin Y.Mining Frequent Patterns without Candidate Generation[C].In:Proc of 19th ACM SIGMOD International Conference on Management of Data,Dallas,2000:207~216