一个基于粗集的决策树规则提取算法

丁春荣[1,2],李龙澍[1]
(1.安徽大学 计算机科学与技术学院,安徽 合肥 230039;
2.泉州师范学院,福建 泉州 362311)

摘 要:决策树是数据挖掘任务中分类的常用方法。在构造决策树的过程中,分离属性的选择标准直接影响到分类的效果,传统的决策树算法往往是基于信息论度量的。基于粗糙集的理论提出了一种基于属性重要度和依赖度为属性选择标准的决策树规则提取算法。使用该算法,能提取出明确的分类规则,比传统的ID3算法结构简单,并且能提高分类效率。

关键词:数据挖掘;粗糙集;决策树;属性约简
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2007)11-0110-04

A Rule Abstracting Algorithm of Decision Tree Based on Rough Set

DING Chun-rong[1,2], LI Long-shu[1]
(1. College of Computer Science and Technology, Anhui University, Hefei 230039, China;
2. Quanzhou Normal University, Quanzhou 362311, China)

Abstract: The decision tree is a usual method of classification in data mining. In the process of constructing a decision tree, the criteria of selecting attributes to split will influence the efficiency of classification directly. The decision tree algorithm traditionally is based on information theory measure. Presented a new algorithm for classification rules extraction by choosing attributes of importance of attributes and dependance based on rough set. Using this algorithm, can extract crisp rules from classification information system. Compared with the traditional ID3 algorithm, it's simpler in the structure, and can improve the efficiency of classification.

Key words: data mining; rough set; decision tree; attribute reduction

0 引 言

分类是数据挖掘领域中的一个重要课题,许多挖掘问题本质上都可以等价地转化为分类问题。分类的主要方法有决策树[1]、神经网络、统计学、粗糙集[2~5]及遗传算法等,其中决策树方法是应用最广的推理算法之一,具有分类精度高、速度快、生成的模式易于理解等优点,特别是在学习过程中不需要很多背景知识,只要训练样本集能够用"属性—值"的方式表达出来就能使用决策树学习算法进行分类,因而在数据挖掘中受到广泛关注。目前已经存在多种决策树算法,其中最经典的就是Quinlan提出的基于信息熵的算法ID3[1],C4.5[1]及后来的多个改进版本,并且已被广泛应用于实际的分类问题中。但是基于信息熵[1,6]的算法存在决策树中子树重复,及有些属性在某一路径上被多次检验等问题,以致降低了分类的效率和效果。特别当存在大量对分类无关的冗余属性时,生成的决策树过度庞大,结构性差,难以发现一些本来可以找到的有用规则。

粗糙集理论[3,4]是波兰科学家Pawlak.Z在1982年提出的对不完整数据进行分析、推理、学习、发现的新方法,借鉴了逻辑学和哲学中对不精确、模糊的各种定义,针对信息的不同分类模型,提出不精确范畴等概念,为处理模糊信息系统或不确定性问题提供了一种新型的数学工具,它不仅能够解决传统的数据分析方法如决策树法不能解决的粗糙集数据,得到传统方法如神经网络得不到的较高精度规则,而且能发现属性之间的依赖关系并对所得的结果进行简明易懂的解释。该理论已广泛应用于信息处理、数据挖掘等认知领域。

文中基于粗糙集的理论提出了一种依据属性重要度和依赖度为属性选择标准的决策树规则提取算法。使用该算法,能提取出明确的分类规则,与传统的ID3

收稿日期:2007-01-24
基金项目:国家自然科学基金项目(60273043);安徽省自然科学基金项目(050420204);安徽省高校拔尖人才基金项目(05025102)
作者简介:丁春荣(1975-),女,安徽淮北人,硕士研究生,讲师,研究方向为数据挖掘、粗糙集;李龙澍,教授,博士生导师,研究方向为智能软件和知识工程。

算法相比,结构简单,并且能提高分类效率。

1 相关概念

给定知识表达系统 $S=(U,A,V,f)$,其中非空有限集合 U 表示论域,A 是关于 U 的属性非空有限集合 $V=\bigcup_{a\in A}V_a$,V_a 是属性 a 的值域,$f:U\times A\rightarrow V$ 是一个信息函数,对于 $\forall a\in A, x\in U, f(x,a)\in V_a$,若属性集合 $A=C\cup D$,其中 C 是条件属性,D 是决策属性,则该系统也被称为决策系统或决策表。

1.1 不可区分关系

对决策系统 S,$R\in A$ 是条件属性集合的一个子集,称二元关系 $\mathrm{Ind}(R)$ 为 S 的不可区分关系:$\mathrm{Ind}(R)=\{(x,y)\in U\times U\mid \forall a\in R, f(x,a)=f(y,a)\}$,表示样本对象 x 和 y 关于属性子集 B 是不可区分的。$U/\mathrm{Ind}(R)$ 即等价关系 $\mathrm{Ind}(R)$ 的所有等价类,有时也记作 U/R。

1.2 上、下近似

给定决策系统 S,R 是 U 上的一个等价关系,X 是 U 的一个子集,则上近似 $R^-(X)=\bigcup\{Y\in U/R\mid Y\cap X\neq\varnothing\}$,$R^-(X)$ 是根据知识 R,U 中一定能和可能归入 X 的元素的集合。下近似 $R_-(X)=\bigcup\{Y\in U/R\mid Y\subseteq X\}$,$R_-(X)$ 是根据知识 R,U 中一定能归入 X 的元素的集合。$\mathrm{Pos}_R(X)=R_-(X)$ 是 X 的 R 正域。

1.3 知识依赖

如果知识 D 的部分初等范畴能用知识 C 的某些初等范畴来定义,称知识 D 部分依赖于知识 C。有 $k=\gamma_C(D)=\dfrac{\mathrm{Card}(\mathrm{Pos}_C(D))}{\mathrm{Card}(U)}$,其中 Card 表示集合基数,则称 D 是 $k\in[0,1]$ 依赖于 C。k 反映了知识 C 对 D 的分类能力,若 $k=0$,则 D 完全不依赖于 C;若 $k=1$,则 D 完全依赖于 C;若 $0<k<1$,则 D 部分依赖于 C。

1.4 属性重要性

在确定决策目标时,不同属性的重要性是不同的。粗集根据属性的分类能力来确定该属性的重要性。属性子集 $C'\subseteq C$ 关于 D 的重要性定义为 $\sigma_{CD}(C')=\gamma_C(D)-\gamma_{C-C'}(D)$,当 $C'=\{a\}$ 时,属性 a 的重要性为 $\sigma_{CD}(a)=\gamma_C(D)-\gamma_{C-\{a\}}(D)$,$\sigma_{CD}(a)$ 的值越大,说明属性 a 对分类的影响越大,分类能力越强。

1.5 属性约简

对决策系统 S,属性 $r\in R\subseteq A$ 是 R 中必要的,当且仅当 $\mathrm{Ind}(R)\neq\mathrm{Ind}(R-\{r\})$,否则,属性 r 在 R 中是冗余或可省略的。属性 R 的约简是一个集合 $R'\in R$,当且仅当满足:

①R' 是独立的;

②$\mathrm{Ind}(R')=\mathrm{Ind}(R)$。

1.6 规则可信度和支持度

对于完全相容的决策表进行规则挖掘,得到确定性的规则。而不相容的决策表挖掘到的规则是不确定的,规则不确定性可以通过可信度和支持度进行度量。

对于决策表 $S=(U,A,V,f)$,$A=C\cup\{d\}$:决策规则 $A\rightarrow B$ 的可信度 $\mathrm{CD}(A\rightarrow B)=\dfrac{|X\cap Y|}{|X|}$,其中 $X=\{x\mid x\in U\cap A_x\}$,$Y=\{y\mid y\in U\cap B_y\}$,$A_x$ 表示实例 x 的条件属性值满足公式 A,B_x 表示实例 x 的决策属性值满足公式 B。支持度 $\mathrm{CD}(A\rightarrow B)=\dfrac{|X\cap Y|}{|U|}$。

2 提取规则算法

粗糙集中的规则提取,一般是先对条件属性集合进行约简,约简后的属性个数直接影响着决策规则的繁简和性能,而找到具有最少条件属性的约简是 NP 难问题。利用一些启发式约简算法可以得到约简属性集,合并决策表中相同的行后得到约简表,可以认为该约简表就是规则,但此时规则可能还含有很多冗余信息,如何得到最简或相对最简的规则是粗糙集规则提取算法要处理的问题。一般做法是:先求得约简表的核值属性,然后再在核中添加属性,直到某条规则可以在约简表中是唯一的表达为止。但求核后逐个增加条件属性是个组合问题,最坏的情况是计算 $2^{|A|}-1$ 次。

2.1 算法及测试属性选择标准

ID3 算法建立的决策树是单变量决策树,在每一个非叶结点只能选择唯一属性,势必会造成子树重复,上述问题促使人们考虑多变量决策树[7],或者构造一种超属性[6]。文中提出一种基于决策树的规则提取算法(AAID,an Algorithm by Attribute Importance and Dependance),利用该算法从约简表中提取规则,即使不能保证得到最简规则,但也得到相对最简规则。算法既可以处理相容决策表,也可以处理不相容决策表。在决策树构造过程中,对每个结点分裂属性的选择是至关重要的,总希望能得到最佳划分,使尽可能多的样本被正确分类。传统的决策树构建方法是利用信息论中信息增益[2,5]寻找数据库中具有最大信息增益的属性来建立分支结点,现在通过粗糙集方法可以依据属性对分类的影响即属性重要性来选择属性建立分支,但是有时在属性重要性相等情况下仍难以选择分裂属性。如表 1 所示[8]。

根据属性重要性公式计算条件属性 a,b,c 的重要性,结果 $\sigma_{CD}(a)=\sigma_{CD}(b)=\sigma_{CD}(c)$,此时根据属性重要

性无法选择用哪个属性进一步分裂,但从条件属性 a,b,c 和决策属性 d 分别对 U 的划分:

表 1 决策表

U	a	b	c	D
1	0	0	1	1
2	0	1	1	0
3	1	0	1	1
4	1	1	0	0
5	1	1	1	1

$U/\{d\} = \{(2,4),(1,3,5)\}, Y1 = (2,4), Y2 = (1,3,5)$

$U/\{a\} = \{(1,2),(3,4,5)\}, U/\{b\} = \{(1,3),(2,4,5)\}, U/\{c\} = \{(4),(1,2,3,5)\}$

可以看出,由 b 取某一值能正确分类 Y2 中的 2 个对象,由 c 取某一值能正确分类 Y1 中的 1 个对象,而 a 取任何值都不能正确分类任何样例。可见选择属性对分类效果更好,如果选择 a 决策树的复杂度将大于前者。由此,选取一个参数 K 来反映决策分类对条件属性集的依赖度:$K = \max\{|X_i \subseteq Y_j| / |Y_j|\}$,上例中,$a$,$b$,$c$ 的 K 值分别为 0,2/3,1/2。

2.2 AAID 算法

输入:一个信息系统 S;

输出:决策规则表 DT;

(1) 对 S 进行属性约简,删除冗余属性。

(2) 计算约简属性集中各属性的重要性,选择值最大的属性作为分裂属性,if 各个属性重要性相等,then 选择 K 值最大的属性作为分裂属性,if 各属性 K 值相等,then 选择序号靠前的属性建立分支结点。

(3)重复以上过程,直到各分类的决策属性一致或属性集合为空,产生一个叶子结点。被选属性选择后,在属性集中删除掉,以保证各分支不重复选择属性。

(4)读树,生成规则,并计算规则的 CD 和 SD。

(5)验证规则最简性。对每一条规则从第一个属性结点逐个去掉,如果去掉后该规则在约简表中对应唯一的实例,则继续去除下一个规则属性结点,直到没有可供去除的规则属性结点为止;如果规则对应多个样例则停止验证该规则转向下一条规则。

步骤 1 是约简掉冗余属性过程,步骤 2 和 3 是建树过程,步骤 4 是生成规则过程,由于决策树本身特点,进行验证可保证得到最简规则集。

3 算例分析

假设下面数据集是一个信息系统(如表 2 所示)。

对表 2 进行属性约简,知 A4,A6 相对于决策属性 D 是可以省略的,将其去除后不会改变系统的分类能力,去除冗余属性后的决策表如表 3 所示。

表 2 一个信息系统

U	Condition attributes(C)						Decision attribute(D)
	A1	A2	A3	A4	A5	A6	
1	0	1	0	1	1	1	0
2	1	0	0	0	1	0	1
3	0	0	1	0	0	0	0
4	0	0	1	1	0	0	0
5	1	0	1	1	1	1	0
6	1	0	0	1	1	1	1
7	1	1	0	1	1	1	1
8	1	1	1	1	0	1	0
9	0	1	1	0	1	1	0
10	1	1	1	1	1	1	1
11	1	1	1	1	1	1	0

表 3 约简后的决策表

U		Condition attributes(C)				Decision attribute(D)
new	old	A1	A2	A3	A5	
1	1	0	1	0	1	0
2	2,6	1	0	0	1	1
3	3,4	0	0	1	0	0
4	5	1	0	1	1	0
5	7	1	1	0	1	1
6	8	1	1	1	0	0
7	9	0	1	1	1	0
8	10	1	1	1	1	1
9	11	1	1	1	1	0

这是一个不相容决策表,针对表 2,论域 $U = \{1,2,3,4,5,6,7,8,9\}$,条件属性 $C = \{A1,A2,A3,A5\}$,决策属性是 D,依据 AAID 算法建树,过程如下:

$U/D = \{(1,3,4,6,7,9),(2,5,8)\}$;

$U/C = \{(1),(2),(3),(4),(5),(6),(7),(8,9)\}$

$pos_C(D) = \{1,2,3,4,5,6,7\}$;

则 $\gamma_C(D) = \dfrac{Card(Pos_C(D))}{Card(U)} = 7/9$

对于属性 $A1$,$U/Ind(C - \{A1\}) = \{(1,5),(2),(3),(4),(6),(7,8,9)\}$

$pos_{C-\{A1\}}(D) = \{2,3,4,6\}$,则 $\gamma_{C-\{A1\}}(D) = 4/9$;

属性 $A1$ 的重要性 $\sigma_{CD}(A1) = \gamma_C(D) - \gamma_{C-\{A1\}}(D) = 1/3$;

对于属性 $A2$,$U/Ind(C - \{A2\}) = \{(1),(2,5),(3),(6),(7),(4,8,9)\}$

$pos_{C-\{A2\}}(D) = \{1,2,3,5,6,7\}$,则 $\gamma_{C-\{A2\}}(D) = 2/3$;

属性 $A2$ 的重要性 $\sigma_{CD}(A2) = \gamma_C(D) - \gamma_{C-\{A2\}}(D) = 1/9$;

对于属性 $A3$,$U/Ind(C - \{A3\}) = \{(1,7),(2,4),(3),(5,8,9),(6)\}$

$pos_{C-\{A3\}}(D) = \{1,3,6,7\}$,则 $\gamma_{C-\{A3\}}(D) = 4/9$;

属性 $A3$ 的重要性 $\sigma_{CD}(A3) = \gamma_C(D) - \gamma_{C-\{A3\}}(D) = 1/3$;

对于属性 $A5$,$U/\mathrm{Ind}(C-\{A5\})=\{(1),(2),(3),(4),(5),(6,8,9),(7)\}$

$\mathrm{pos}_{C-\{A5\}}(D)=\{1,2,3,4,5,7\}$,则 $\gamma_{C-\{A5\}}(D)=2/3$;

属性 $A5$ 的重要性 $\sigma_{CD}(A5)=\gamma_C(D)-\gamma_{C-\{A5\}}(D)=1/9$。

由此知,条件属性中 $A1$ 和 $A3$ 重要性最大,且相同,此时再计算 K 值,属性 $A1$ 与 $A3$ 的 K 值也相等,此时选择序号靠前的 $A1$ 作为根结点,当 $A1=0$ 时,{1,3,7};当 $A1=1$ 时,{2,4,5,6,8,9}。对于 $A1=0$ 这个分支,分类集合决策为一类,停止选择属性过程。规则可信度 CD = 1,支持度 SD = 3/11 * 100% = 27.3%。

对另一个分类集合继续选择属性。分类集合如表 4 所示。

表 4　$A1=$'1' 分类集合

U		Condition attributes(C)			Decision
new	old	$A2$	$A3$	$A5$	attribute(D)
2	2,6	0	0	1	1
4	5	0	1	1	0
5	7	1	0	1	1
6	8	1	1	0	0
8	10	1	1	1	1
9	11	1	1	1	0

此时,计算 $A2,A3,A5$ 的属性重要性,方法同上,结果 $A3$ 重要性大于另外两个属性,此时选择 $A3$ 作为分裂结点进行分类,当 $A3=0$ 时,{2,5};当 $A3=1$ 时,{4,6,8,9}。当 $A3$ 取值 0 时,决策属性类别为 1,停止选择属性过程,规则可信度 CD = 1,支持度 SD = 2/11 * 100% = 18.2%;当 $A3$ 取值为 1 时,继续选择属性。重复以上过程,直到最后得到完整决策树,如图 1 所示。

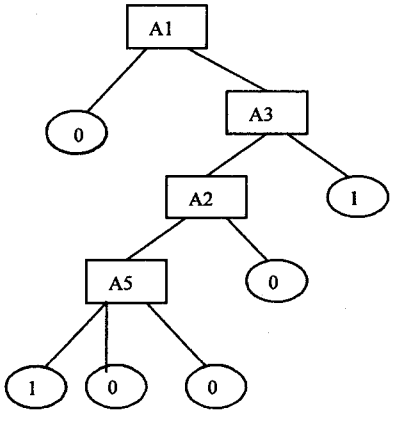

图 1　决策树

根据生成的决策树提取规则,得到规则集(如表 5 所示)。

表 5 经过验证后,规则 4 可将 $A1$ 和 $A3$ 属性除去,所以化简后的决策表如表 6 所示。

表 5　得到的规则集

序号	If	Then	CD	SD
1	$A1=0$	$D=0$	1	36.36%
2	$A1=1\&A3=0$	$D=1$	1	27.27%
3	$A1=1\&A3=1\&A2=0$	$D=0$	1	9.09
4	$A1=1\&A3=1\&A2=1\&A5=0$	$D=0$	1	9.09
5	$A1=1\&A3=1\&A2=1\&A5=1$	$D=1$	0.5	9.1
6	$A1=1\&A3=1\&A2=1\&A5=1$	$D=0$	0.5	9.1

表 6　简化后规则集

序号	If	Then	CD	SD
1	$A1=0$	$D=0$	1	36.36%
2	$A1=1\&A3=0$	$D=1$	1	27.27%
3	$A1=1\&A3=1\&A2=0$	$D=0$	1	9.09
4	$A2=1\&A5=0$	$D=0$	1	9.09
5	$A1=1\&A3=1\&A2=1\&A5=1$	$D=1$	0.5	9.09
6	$A1=1\&A3=1\&A2=1\&A5=1$	$D=0$	0.5	9.09

通过以上处理,得到了相对最简规则,并且完全保留了原信息表中的信息。上述例子采用传统 ID3 算法进行处理后可得到完全相同的决策树,因此 AAID 算法的执行效率优于 ID3 算法。

4　结束语

文中结合粗糙集与决策树的优点,提出了一种新的基于粗糙集和决策树的决策规则提取算法,结合粗糙集和决策树的优点,可以处理相容表,也可以处理不相容表。虽然不一定得到最简的规则,但却能方便地得到相对最简规则,避开了 NP 问题。并且执行效率优于 ID3 算法,通过与其它实验数据的比较,用 AIAD 算法得到的规则所用的属性有时比较少,但规则个数不一定会比 ID3 少。实际应用表明该算法是有效的。

参考文献:

[1] Quinlan J R. Induction of decision trees[J]. Machine Learning,1986,1(1):81 - 106.

[2] Pawlak Z. Rough sets[J]. International Journal of Computer Information Science,1982,11:341 - 356.

[3] 曾黄麟. 粗糙集理论及其应用[M]. 修订版. 重庆:重庆大学出版社,1998:119 - 129.

[4] 张文修. 粗糙集理论与方法[M]. 北京:北京科学工业出版社,2001:3 - 24.

[5] Pawlak Z. Rough sets[J]. Communications of ACM,1995,38(11):89 - 95.

[6] 赵卫东,盛昭瀚. 粗糙集在决策树生成中的应用[J]. 东南大学学报,2000,30(4):134 - 136.

[7] 苗夺谦,王　珏. 基于粗糙集的多变量决策树构造方法[J]. 软件学报,1997,8(6):425 - 431.

[8] Zhu H. The Research of the Simplest Decision Tree Production Algorithm Based on Rough Set[J]. Computer Engineering and Application,2003(13):129 - 131.