

不相容决策表的属性约简与规则提取算法

林江毅¹, 马亨冰²

(1.福州大学数学与计算机学院 福建 福州 350002 2.福建省经济信息中心 福建 福州 350003)

[摘要]: 决策表属性约简与规则提取是粗糙集理论中的重要问题。本文引入一种改进的决策表广义信息表的方法来求解属性核与相对约简,并在此基础上提出一种不相容决策表的规则提取算法。

[关键词]: Rough 集;约简;核属性;不相容决策表

1.引言

Rough 集理论自 Pawlak 教授提出以来,已经在机器学习、数据挖掘等领域中得到了较为广泛的应用。决策表信息系统是 Rough 集理论的主要研究对象。根据决策表中数据的相容性,决策表分为相容(或一致)和不相容(或不一致)的决策表。由于含有不相容的数据,不相容决策表的规则提取相对相容决策表而言要复杂一些。文献[1]介绍了一个提取不相容决策表决策规则的分解处理方法,该方法将一个不相容决策表分解为两个不相交的子表,然后分别进行属性约简及规则提取。本文进一步对这种分解方法进行考察,结合广义信息表的概念提出一种对完全不相容的决策表的规则提取。

2.基本概念

限于篇幅,粗糙集的一些基本概念请参见文献[1]-[3]。这里为以后叙述方便,只介绍一些重要的定义。

定义 1 一个决策系统可以用一个四元组来表示: $T=(U, A, V, f)$ 。其中, U 是论域。不妨设该论域有 n 个对象,则 U 可表示为 $U=\{x_1, x_2, \dots, x_n\}$ 。 A 是属性集合,将 A 进一步划分为不相交的属性集 C 和 D 的并集 ($A=C \cup D$ 且 $C \cap D = \emptyset$),其中 C 为条件属性集, $C=\{c_1, c_2, \dots, c_m\}$ 。 D 为决策属性集,一般考虑只有一个决策属性的情况,而多决策属性问题可以化为单决策属性问题处理。 $V=U \cup V_a, a \in A, V_a$ 为属性 a 的值域集。 f 是信息函数,

$f:U \times A \rightarrow V$,对任意 $x \in U, a \in A$,有 $f(x, a) = V_a$ 。有时也将决策表记为: $T=(U, C \cup D)$ 。

定义 2 构造决策系统 $T=(U, C \cup D)$ 的广义信息表 $S^*=(U^*, A^*, V^*, f^*)$ 定义为:

$U^*=\{(x_i, x_j) \in U \times U | f(x_i, D) \neq f(x_j, D) \wedge \min\{d(x_i), d(x_j)\}=1\}$, $A^*=\{a | a \in C\}$, $V^*=\{1, 0\}$,任意的 $a \in A^*, f^*:U^* \times A^* \rightarrow V^*$ 。

若 $f(x_i, a) \neq f(x_j, a)$,则 $f^*((x_i, x_j), a)=1$;否则 $f^*((x_i, x_j), a)=0$ 。根据广义信息表的构造可知,对任意的 $a \in A^*, f^*((x_i, x_j), a)=1$ 表示在原决策表中属性 a 能区分对象 x_i 与 x_j 。

定理 1 若广义信息表某行中只有 1 个属性的取值为 1,表明此 1 所对应的属性是唯一能区分该行所对应的实例对属性,因此这个属性为核属性。

3.分解矩阵

文献[1]提出的一种分解不相容决策表的方法。其基本思路为:给定一个不相容的决策表 $T=(U, C \cup D)$,则 T 可唯一分解为两个决策子表 $T_1=(U_1, C \cup D), T_2=(U_2, C \cup D)$,其中 $U_1=PosC(D), U_2=U-U_1$,即 T_1 是完全相容的决策表, T_2 是完全不相容的决策表。

倘若按照文献[1]的方法求得 T_2 ,那在求属性约简时,那些矛盾的记录会被重复比较;并且在求 T_2 中每条记录的决策规则时,每次计算粗糙算子都要遍历该表一次,从而造成时间上的浪费。所以本文做以下改进:融合不相容的数据,用两个字段(RHS, LHS)来记录数据的数目。其中:RHS 表示条件属性相同的矛盾记录总数,LHS 表示对应于每个决策属性值的记录个数。并且规定: T_2 中的记录间的决策属性是不相同的,并且它跟 T_1 中记录的决策属性也是不同的。分解表 1 后, T_1 与 T_2 分别如表 2,3 所示。

表 1 不相容的决策表

u	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
3	2	1	1	1	2
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	0	0	1	1
8	0	1	1	0	1

表 2 完全一致的决策表

u	a	b	c	d	e
3	2	1	1	1	2
4	1	1	0	2	2
6	2	2	0	1	1
7	2	0	0	1	1

表 3 改进的完全不一致的决策表

u	a	b	c	d	e	RHS	LHS
15	1	0	2	20	00	2	10
28	0	1	1	10	20	2	10

4.属性约简算法

本文在分解决策表的基础上,提出一种用改进的广义信息表来求取属性约简的算法。对原来广义信息表做以下修改:

(1) 由于原决策表已经分解,因此将广义信息表的 U^* 定义为:

$$U^*=\{(x_i, x_j) \in U \times U | f(x_i, D) \neq f(x_j, D) \wedge x_i \in T_1, x_j \in T_1 \cup T_2\}$$

(2) 修改广义信息表的存储内容为核属性不能区分的记录的比较信息。换言之,在求取广义信息表时,若 $f^*((x_i, x_j), a)=1$,且 a 为核属性,则 x_i 与 x_j 能被核属性区分,因此不将比较信息加入到广义信息表中。同时,用 Val-one 来表示当前广义信息表各列 1 的数目,Sum 用来表示各行 1 的数目,并多增加一行 Sum-one 用来保存当前广义信息表中每个属性(列)中值为 1 对应的行中 1 的数目的总和。

4.1 算法主要思想

先按上述修改求取 $T_1 T_2$ 对应的广义信息表及核集,然后约简集合 $Reduct=核集$ (核集可能为空),选取最重要的属性加入到 $Reduct$ 中,修改广义信息表,直到广义信息表为空。 $Reduct$ 即为属性约简。

4.2 算法描述

算法 1 改进的属性约简算法

输入 信息系统决策表 $T=(U, C \cup D), U=\{x_1, x_2, \dots, x_n\}, C=\{c_1, c_2, \dots, c_m\}$ 。

输出 决策表相对约简 $Reduct$ 。

Step1 按 3 的方法将不相容的决策表 T 分解,得到完全相容的子表 T_1 ,完全

不相容的子表 T_2 ;核集 $CoreD(C)=\emptyset; Reduct=\emptyset$;

Step2 求广义信息表:

for $i=1$ to $Card(T_1)$ { // $Card(T_1)$ 表示 T_1 中的记录个数
for $j=i+1$ to $Card(T_1)$ {

求取记录 x_i, x_j 的比较信息,若有核值出现,则将该核属性加入到 $CoreD(C)$;

若无核,且不能被 $CoreD(C)$ 区分,将比较结果加入广义信息表中;

for $j=1$ to $Card(T_2)$ { // $Card(T_2)$ 表示 T_2 中的记录个数

求取记录 x_i, x_j 的比较信息,若有核值出现,则将该核属性加入到 $CoreD(C)$;

若无核,且不能被 $CoreD(C)$ 区分,将比较结果加入广义信息表中;}}

Step3 遍历广义信息表,对任意的 $a \in CoreD(C)$:

if(广义信息表的某一行在该属性上的取值为 1) 将该行剔除出广义信息表;
 else 将值为 1 的属性对应的 Val-one 值+1,同时 Sum-one 值+ Sum;
 Step4 if(CoreD(C)=Φ) 转 Step5.
 else{ Reduct= CoreD(C);
 if(广义信息表中无记录) 输出 Reduct,算法终止。
 else 转 Step5。}
 Step5 while(广义信息表有记录)
 {取 Val-one 中值最大对应的属性加入到 Reduct 集合中(若最大值不止一个,则 取 Sum-one 值最小的属性加入到 Reduct。又若 Sum-one 中最小值也不止一个,则 随机选取)。将广义信息表中在该属性上取值为 1 的行去掉,并将该行中值为 1 的对应的属性的 Val-one 值-1, Sum-one 的值- Sum。} 输出 Reduct,算法终止。

4.3 实例分析

U*	a	b	c	Sum
(3,4)	1	0	1	2
(3,6)	0	1	1	2
(3,7)	0	1	1	2
(3,15)	1	1	1	3
(4,15)	0	1	1	2
Val-one	0	0	0	
Sum-one	0	0	0	

表4 表2,3对应的广义信息表

U*	a	b	c	Sum
(3,6)	0	1	1	2
(3,7)	0	1	1	2
(4,15)	0	1	1	2
Val-one	0	3	3	
Sum-one	0	6	6	

表5 表4约简后的广义信息表

以表 1 来说明算法 1 的思想:

- (1) 经过 Step1 将表 1 分解为表 2,表 3;
- (2) 经过 Step2 得到表 2,表 3 对应的广义信息表,如表 4 所示;
- (3) 经过 Step3 得到表 5;
- (4) 按照 Step4 求得属性约简为{a,b}。

5.规则提取

根据文献[1]的规则提取算法易求出 T1 的一致规则集(如图 1 所示)。对于 T2 的决策规则的提取,我们有如下算法:

算法 2 完全不一致决策子表的规则提取

Step1 求取 T2 的决策规则的提取所需要的属性集合:

将 U* 的取值范围改为: $U^* = \{(x_i, x_j) \in U \times U \mid x_i \text{ 与 } x_j \text{ 属于完全不一致的决策子表}\}$

初始化 CoreD (C)=Reduct, Reduct 为算法 1 求得的属性约简。求 T₂ 对应的广义信息表, 调用算法 1 的 Step4 ,Step5 对 T₂ 进行属性约简。

Step2 根据 Step1 的结果按文献[1]的规则提取算法求得对应的带粗糙算子的决策规则。根据算法 2,Step1 的结果为{a,b}。亦即{a,b}可将完全不一致决策子表中的记录完全区分开来。容易求得表 3 的带置信度的不一致规则集(如图 1 所示)。该结果与文献[1]的结果一致(参见文献[1]中 162 页)。

一致规则集:	不一致规则集:
$r_1: a_2 b_1 \rightarrow d_1 e_2$	$r_1: a_1 b_0 \rightarrow_{0.5} d_2 e_0$
$r_2: a_2 b_0 \rightarrow d_1 e_1$	$r_2: a_1 b_0 \rightarrow_{0.5} d_0 e_1$
$r_3: b_2 \rightarrow d_1 e_1$	$r_3: a_0 \rightarrow_{0.5} d_1 e_2$
$r_4: a_1 b_1 \rightarrow d_2 e_2$	$r_4: a_0 \rightarrow_{0.5} d_0 e_1$

图 1 规则集

6.结论

本文根据粗糙集理论,提出一种高效的属性约简算法,并提出对不一致决策表的规则提取算法,该算法对有效的获取规则十分有意义。

参考文献:

- 1.史忠植.知识发现[M].北京:清华大学出版社,2001.
- 2.Pawlak Z. Rough set approach to multi-attribute decision analysis[J].European Journal of Operational Research,1994,11:443-459.
- 3.刘清.粗糙集及粗糙推理[M].北京:科学出版社,2001.
- 4.X.Wang,T.Zhang,Y.Huang,J.Xiao "A New Algorithm for Relative Attribute Reduction in Decision Table" in Proceedings of the 6th World Congress on Intelligent Control and Automation,June 21-23, 2006,Dalian,China,pp.4051-4054.

(上接第 67 页)

户的方法有两种:1.使用[本地用户和组],在帐户[属性]对话框中选中[帐户已停用]复选框 2.在[运行]中输入 Net user username /active:no 按[确定]按钮

4.5 配置安全性高的登录过程

安全登录过程,是一种防止未授权用户物理访问您的计算机而进行登录的方式,要求关闭 Windows 中某些便利性的功能。在默认安装的 Windows XP 中,开机的时候就会出现欢迎屏幕,它提供了一个很友好的界面并且使您可以只需要点击就可以即可登录,如果您的用户有密码则需要输入密码。欢迎屏幕会给只要能打开您电源的人显示所有用户的用户名,攻击者只要知道密码就很容易通过身份验证。关闭欢迎屏幕的方法:在[控制面板]中打开[用户帐户],单击[更改用户登录或注销的方式],清除[使用欢迎屏幕]复选框,再单击[应用选项]。在 XP 中,我们可以像 Windows 2000 一样,用户必须按 Ctrl+Alt+Delete 键来显示[登录到]Windows 对话框,设置的方法:打开[用户和密码],单击[高级]标签,选中[要求用户按下 Ctrl+Alt+Delete]复选框。同时我们也可以选中[用户]标签下[要使用本机,用户必须输入用户名和密码],已防止自动登录。

4.6 设置用户锁定策略

帐户锁定策略允许您在用户输入了太多次数的错误密码之后锁定那个帐户。设置这个策略是一个对付密码破解企图的有效保卫措施。当一个用户因为太多的错误密码输入被锁定时,管理员可以在[本地用户和组]中对该用户解锁。如果您的计算机出

现这种情况,那么您应该警觉起来,可能您的系统正在受到攻击。您可以使用[本地安全策略]控制台来设置用户锁定策略。打开[本地安全设置]→[帐户策略]→[帐户锁定策略]。它包括:帐户锁定时间、帐户锁定阈值、复位帐户锁定计数器。帐户锁定时间指定用户被锁定的时间,经过指定的时间之后,该用户会被自动解锁;帐户锁定阈值是指在指定的时间内如果用户输入的错误密码次数达到了指定的数字,系统就会被阻止该用户登录;复位帐户锁定计数器是指在指定的时间内如果用户输入密码错误达到了指定的次数,就会被锁定。

5.结束语

迄今为止,有越来越多的趋于不同目的的黑客们正虎视眈眈地窥视着我们的计算机,用户又是我们进入计算机的最基本的构件,所以对于自己用户的安全也越来越受到人们所重视。本文提到的非常简单而实用的操作可以帮助大家更好地提高计算机系统的安全性。

参考文献:

- 1.[美]Ed Boot,Carl Siechert 著精通《Windows 2000 和 Windows XP 安全技术》清华大学出版社 2006-6-1
- 2.梅筱琴 蒲韵 廖凯生 编《计算机病毒防治与网络安全手册》海洋出版社 2001-6-1
- 3.王锐 影响网络安全的因素及需要考虑的问题 [J] 计算机教育,2005