一种改进的属性约简算法

王 荣[1],陈黎伟[1],吴 涛[1,2]
(1.安徽大学 数学与计算科学学院,安徽 合肥 230039;
2.安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

摘 要:粗糙集理论是一种有效的信息处理工具,属性约简是粗糙集理论研究的一个核心内容。为了能够较为有效地获得不相容决策表较优的属性约简,在对文献[7]中属性约简算法分析的基础上,根据不相容决策表约简不改变决策表正域的原则,仅考虑相对差异比较表中与正域相关的实例对,同时结合属性重要性作为特征选取的启发式信息,提出了一种改进的启发式属性约简算法。该算法在不增加算法时间复杂度的前提下能够处理不相容决策表。最后,通过实例完整演示了该方法,表明该算法是有效的。

关键词:粗糙集;决策表;属性约简;正域
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2008)02-0146-03

An Improved Attributes Reduction Algorithm Based on Rough Set

WANG Rong[1], CHEN LI-wei[1], WU Tao[1,2]
(1.School of Mathematics and Computational Science, Anhui University, Hefei 230039, China;
2.Ministry of Education Key Lab. of IC & SP, Anhui University, Hefei 230039, China)

Abstract: Rough set is an efficient information processing tool, and attribute reduction of decision table is a kernel part of research in rough set theory. Based on the analysis of attribute reduct algorithms in reference[7], an improved heuristic reduction algorithm is proposed to effectively and efficiently achieve the better attribute reducts of inconsistent decision table. This algorithm based on the principle of remaining positive region of inconsistent decision table, and only considering objects in positive region, Then combined with the significance of attribute as heuristic information of feature selection. The algorithm is the same with consistent and inconsistent decision tables, but the time complexity does not increase. At last, demonstration section shows that this algorithm is effective.

Key words: rough set; decision table; attribute reduction; positive region

0 引 言

粗糙集理论是波兰数学家 Z. Pawlak 于 1982 年提出的一种处理模糊、不精确的分类问题的新型数学工具[1],其主要思想是在保持信息系统分类能力不变的前提下通过知识约简导出问题的决策或分类规则,属性约简是粗糙集理论中的重要内容之一。决策表的约简,就是从决策表系统的条件属性中去掉不必要的条件属性,从而分析所得约简中的条件属性对决策属性的决策规则,目前已提出了多种求属性约简的算法[2~6]。文献[7]在粗糙集的理论上构造相对差异比较表,然后设计出了 4 个算法,但这些只能处理相容数据,在数据不相容的情况下算法将失效。文中在其基础上,改进了 AR1 算法,使其能处理不相容数据,同时不增加算法的时间复杂度。文献[7]中的其他算法都是以 AR1 为基础的,因此可以使用相似的方法对其进行修改。

收稿日期:2007-05-20
基金项目:安徽省自然科学基金项目(050420208);安徽省高等学校省级自然科学研究项目(2006KJ244B);安徽大学学术创新团队和安徽大学人才队伍建设经费资助项目。
作者简介:王 荣(1982-),女,安徽滁州人,硕士研究生,研究方向为智能计算与信息处理;吴 涛,博士,副教授,主要从事机器学习、智能计算及其应用的研究。

1 属性约简的相关概念

一个决策系统 T 可以表示为四元组 $T=<U,A,V,f>$,其中 U 为对象集,A 为非空有限的属性集;$A=C\cup D$ 且 $C\cap D=\varnothing$,式中 C,D 分别为关于 U 的条件属性集和决策属性集;$V=\bigcup_{a\in A}V_a$ 表示属性的值域;$f: U\times A\rightarrow V$ 是一个信息函数,它指的是 U 中每一个对象 x 的属性值,即 $x\in U, a\in A$,有 $f(x,a)\in V_a$[8]。

定义1:令 $P \subseteq A$,定义属性集 P 的不可区分关系 $\mathrm{ind}(P)$ 为

$$\mathrm{ind}(P) = \{(x,y) \in U \times U : \forall a \in P, f(x,a) = f(y,a)\}$$

不可区分关系 $\mathrm{ind}(P)$ 是 U 上的等价关系,符号 $U/(\mathrm{ind}(P))$(简记为 U/P)表示不可区分关系 $\mathrm{ind}(P)$ 在 U 上导出的划分,即 $U/(\mathrm{ind}(P)) = \{[x]_p : x \in U\}$ 其中 $[x]_p = \{y : (x,y) \in \mathrm{ind}(P)\}$ 是 x 关于 P 的等价类。

定义2:设 R 为论域 U 的一个子集,$R \subseteq C$,X 关于 R 的下近似为

$$\underline{R}\,X = \{x \in U : [x]_R \subseteq X\}$$

定义3:令 P 和 Q 为 U 中的等价关系,Q 的 P 正域记为 $\mathrm{POS}_P(Q)$,即

$$\mathrm{POS}_P(Q) = \bigcup_{X \in U/Q} \underline{P}\,X$$

Q 的 P 正域是 U 中所有根据分类 U/P 的信息可以准确地划分到关系 Q 的等价类中去的对象集合。

定义4:令 P 和 Q 为等价关系,$a \in P$,如果

$$\mathrm{POS}_{\mathrm{ind}(P)}(\mathrm{ind}(Q)) = \mathrm{POS}_{\mathrm{ind}\{P-\{a\}\}}(\mathrm{ind}(Q)),$$

则称 a 为 P 中 Q 不必要的;否则 a 为 Q 中必要的,通常用 $\mathrm{POS}_p(Q)$ 代替 $\mathrm{POS}_{\mathrm{ind}(P)}(\mathrm{ind}(Q))$,如果 P 中的每个 a 都为 Q 必要的,则称 P 为 Q 独立的。

定义5:设 $S \subseteq P$,S 为 P 的约简当且仅当 S 是 P 的 Q 独立子族且 $\mathrm{POS}_S(Q) = \mathrm{POS}_P(Q)$,记 P 的所有 Q 约简为 $\mathrm{red}(C)$。

定义6:对于任一个条件分类 $E_i \in U/\mathrm{ind}(C)$,如果属于它的所有记录值都有相同的决策值,则称 E_i 为一致的;否则称 E_i 为不一致的。对于任意决策系统下,如果所有的条件分类都是一致的,则称 T 为相容决策表,否则称 T 为不相容决策表。

定义7[7]:设 B 是一个 $[\frac{(n^2-n)}{2}]$ 行 $(N+1)$ 列的表。表的每一列对应一个属性,每一行对应决策表中的一对不同记录的比较,最后一列对应着决策属性。设 $b((s,t),i)$ 是 B 中的一个元素,对应着属性 i 和决策表中的一对记录 (x_s,x_t),$s < t$,对于 $i \in \{1,2,\cdots,N\}$:

$$b((s,t),N+1) = \begin{cases} 1; d(x_s) \neq d(x_t) \\ 0; d(x_s) = d(x_t) \end{cases}$$

如果 $b((s,t),N+1) = 1$,则

$$b((s,t),i) = \begin{cases} 1; c(x_s) \neq c(x_t) \\ 0; c(x_s) = c(x_t) \end{cases}$$

否则 $b((s,t),i) = -1, i \in \{1,2,\cdots,N\}$。

将 B 中所有包含元素的行及 B 的 $N+1$ 列去掉后,剩余部分就被称为相对差异比较表 B_1。

2　不相容数据的属性约简算法

2.1　AR1 算法的分析

相对差异比较表中,各行 1 的数目实际上表示的是可以区分该实例对的属性的数目。对于相容决策表,如果行和为 1,说明能区分该实例对的属性只有一个,该属性就是必要的,这就说明行和越小,则元素 1 所对应的属性就越重要;同样,各列 1 的数目越多,说明该属性能区分的实例对越多,属性越重要,但对于不相容的决策表,就会出现行和为零的情况,算法 AR1 将失效。表 1 和表 2 举出反例:

表1　决策表 T_1

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

表2　决策表 T_1 的相对差异比较表 B_1

	a	b	c
(1,2)	0	0	0
(1,4)	1	0	0
(2,3)	1	0	0
(2,5)	0	1	0
(3,4)	0	0	0
(4,5)	1	1	0

容易看出 T_1 是不相容决策表,根据相对差异比较表的定义得到 B_1。

容易看出 B_1 中第2行和第6行行和为0,AR1算法没有提及此类情形。如果选行和为1的行中元素1对应的属性加入约简集容易得到 $\mathrm{red}(C) = \{a,b\}$,而 $\mathrm{POS}_{\{b\}}(D) = \mathrm{POS}_C(D)$,所以 $\mathrm{red}(C) = \{b\}$ 为约简,AR1 算法不适用于不一致决策表。

2.2　改进的属性约简算法

首先,考虑到不相容决策表的约简是考查去掉某些属性后分类能力是否发生变化来判定该属性是否是可省的,故在相对差异比较表中仅考察正域内和正域与非正域之间的实例对,而不考虑非正域内的实例对。其次,算法对列和相同的属性采取随机选取的方法。文中综合考虑行和与列和,将属性按其重要性由大到小排列,依次选取重要性较大的属性。

算法基本思想:去掉相对差异比较表 B_1 中非正域内的实例对,得到一个新表 B_2。令初始约简集为空集。将 B_2 中行和为 1 的行中元素 1 所对应的属性加入约简集,其余属性按列和由大到小排序,如果列和相同,则将列对应的属性所在的列值为 1 的行的 1 的数目相加,将和小的属性排在前。选择重要性最大的属性加入约简集,如果终止条件不满足,则按顺序加入下一个属性。

算法:改进的 AR1 算法

输入:一个决策表 $T = < U, A, V, f >, A = C \cup D$ 且 $C \cap D = \varnothing$;

输出:最后约简 red(C)。

步骤 1:计算正域 $POS_C(D)$;red(C) = $\varnothing$;$j = 1$;

步骤 2:生成相对差异比较表 B_1;

步骤 3:如果 B_1 中任意行和不为 0,则 $B_2 = B_1$;否则 B_2 为 B_1 去掉非正域内的实例对后剩余的部分;

步骤 4:如果存在 i 使 $h_i = 1$(h_i 为行和),则 red(C) = a(a 为 1 对应的属性),$A = A - \{a\}$;否则 red(C) = $a_j, a_j \in A, j = j + 1$;

步骤 5:在 B_2 中,将属性按 l_{a_j}(列和)由大到小排序。$A = \{a_1, a_2, \cdots, a_k\}$,$k$ 为属性数目。如果 $l_{a_j} \neq l_{a_i}$,$i \neq j$,则将属性所在列值为 1 的行的 1 数目相加和小的排在前面;

步骤 6:计算 $POS_{red(C)}(D)$;如果 $POS_{red(C)}(D) = POS_C(D)$ 结束;否则 red(C) = red(C) + a_j;$j = j + 1$;转步骤 6;

步骤 7:最后得到 red(C) 就是 C 相对于 D 的约简。

3 实例分析

例 1:

在上例中有 B_2:

	a	b	c
(2,5)	0	1	0
(4,5)	1	1	0

B_2 中各行行和均不为 0 且第一行行和为 1,将元素 1 对应的属性 b 加入到约简集,有 $POS_b(D) = POS_C(D)$,故 red(C) = $\{b\}$。

例 2:

表 2 一致决策表 T_2:

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

得到 B_2:

	a	b	c	d
(1,3)	1	0	0	1
(1,4)	0	1	0	0
(1,5)	0	1	0	1
(1,6)	1	1	1	1
(1,7)	1	0	0	0
(2,3)	0	1	0	1
(2,4)	0	1	0	1
(2,5)	1	1	0	1
(2,6)	1	1	1	1
(3,5)	1	1	0	1
(3,6)	1	1	0	1
(3,7)	1	1	1	1
(4,5)	0	0	0	1
(4,6)	1	1	0	1
(4,7)	1	1	1	1

1) 计算 $POS_C(D)$,$POS_C(D) = \bigcup_{X \in U/D} C X = \{1,2,3,4,5,6,7\}$;

2)B_2 中容易看出第 3 行,第 6 行,第 14 行和为 1,元素 1 所对应的属性分别是 b, a, d,故 red(C) = $\{a, b, d\}$;

3) 计算 $POS_{\{a,b,c\}}(D) = \{1,2,3,4,5,6,7\} = POS_C(D)$;

4) 所以 red(C) = $\{a, b, d\}$。

与文献[7]中的结果一致。

例 3:

不一致决策表 T_3:

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

得到 B_2:

	a	b	c
(1,2)	0	0	1
(1,3)	0	0	1
(1,6)	1	0	1
(1,7)	1	1	1
(2,4)	1	0	1
(2,5)	1	1	0
(2,8)	1	0	1
(3,4)	1	0	1
(3,5)	1	1	1
(3,8)	1	0	0
(4,6)	0	0	1
(4,7)	0	1	1

(1) 计算 $POS_C(D)$,$POS_C(D) = \bigcup_{X \in U/D} C X = \{1,2,3,4\}$;

(2)B_2 中容易看出第 2 行,第 3 行,第 11 行,第 13 行行和为 1,元素 1 所对应的属性分别是 c, a,故 red(C) = $\{a, c\}$;

(3) 计算 $POS_{\{a,c\}}(D) = \{1,2,3,4\} = POS_C(D)$;

(4) 所以 red(C) = $\{a, c\}$。

(下转第 166 页)

4.2 持久层的实现

Hibernate 是一个轻量级数据持久化框架，是一个面向 Java 环境的 ORM(Object/Relation Mapping)工具[5]。ORM 是指把对象模型表示的对象映射到基于 SQL 的关系模型数据结构中去。ORM 的最大作用不是分布式和安全检查，而是让程序员能够完全用面向对象的思维来分析和设计系统，让系统能够支持多种不同的数据库平台。

Hibernate 没有采用处理字节码和代码生成的办法，而是采用了运行时反射(runtime reflection)来决定一个类的持久化属性。Hibernate 对每一种数据库都有对应的操作优化，从而提高它在各种情况下的效率。

4.3 MVC 多层集成

整个系统架构分为五层：客户层、表现层、业务层、数据持久层和数据库层。其中客户层主要是指客户浏览器；表现层是由 JSP 页面结合 Struts 提供的强大的标签库 TagLib 实现；业务层由 Struts 控制组件 ActionServlet，Action，ActionForm，JavaBeans 等实现；而数据持久层则通过 Hibernate 框架实现对象—关系的映射；数据库层则主要指存放数据的关系数据库系统[7]。以登录为例，图 5 是集成 Struts 和 Hibernate 的 Web 应用架构。

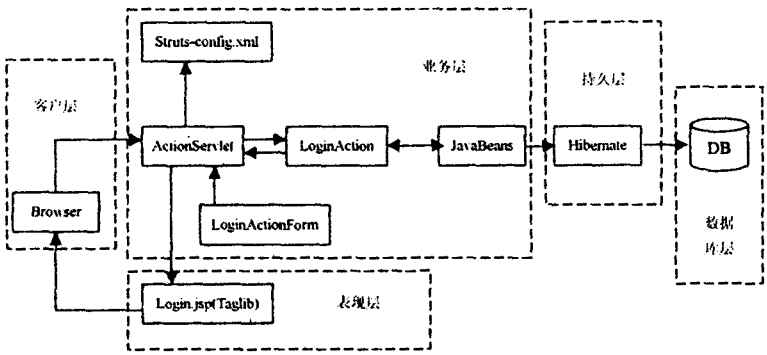

图 5 集成 Struts 和 Hibernate 的 Web 应用架构

5 系统优缺点和结论

文中是在完成设备管理信息系统实例之后总结出来的，系统运行良好。把多 Agent 技术和 MVC 模式引入设备管理信息系统，可以实现各子系统之间信息的(智能)交换。应用计算机管理信息系统，能够提高企业的服务效益和质量、减员增效、加强企业管理、辅助高层领导决策，体现出很大优越性。系统有的模块还暂时不完善，如系统安全性等，还待进一步升级。把多 Agent 的思想与技术和 MVC 模式引入管理信息系统的开发，使系统更有实用价值，为设备管理信息系统的开发提供了一种新的模式。

参考文献：

[1] 王文杰，叶世伟. 人工智能原理与应用[M]. 北京：人民邮电出版社，2004.

[2] 元继学，吴祈宗，王未今. 基于 Agent 的供电企业管理信息系统[J]. 计算机应用技术，2004(6)：196－197.

[3] 朱广武. 基于 Agent 的人力资源管理信息系统[J]. 宁夏工程技术，2004(3)：71－73.

[4] 朱福喜，朱三元，伍春香. 人工智能基础教程[M]. 北京：清华大学出版社，2006.

[5] 任文娟，王 华，鞠宏伟，等. 基于 Struts 和 Hibernate 框架的 Web 应用的设计与实现[J]. 微计算机信息，2006，22：3－9.

[6] 雷 钧，徐洪胜，付勇智. MVC 设计模式在 J2EE 平台上的应用[J]. 微计算机信息，2006(3)：1－3.

[7] 孙卫琴. 精通 Struts 基于 MVC 的 Web 设计与开发[M]. 北京：电子工业出版社，2004.

(上接第 148 页)

4 结 论

对文献[7]中提出的 AR1 算法作出改进，提出一种可处理不相容决策表的属性约简算法。该算法的时间复杂度与文献[7]中提出的算法复杂性相当，均为 $O(|A||U|^2)$，意义比较明确，概念比较清晰，并以属性的重要性作为特征选取的启发信息，计算较为简单，效果较明显，并能够在大多数情况下找到令人满意的结果。

参考文献：

[1] Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning about Data[M]. Boston, London, Dordrecht: Kluwer Academic Publishers, 1991.

[2] 王 钰，王 任，苗夺谦. 基于 Rough set 理论的"数据浓缩"[J]. 计算机学报，1998，21(5)：393－400.

[3] 叶东毅，黄翠微，赵 斌. 粗糙集属性约简的一个贪心算法[J]. 系统工程与电子技术，2000，22(9)：63－65.

[4] 苗夺谦，胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展，1999，36(6)：681－684.

[5] 王国胤. 决策表核属性的计算方法[J]. 计算机学报，2003，26(5)：611－615.

[6] 覃志华，唐承超，王加阳. 不相容决策表的核属性计算[J]. 计算机工程与应用，2005，41(3)：44－46.

[7] 潘 丹，政启伦. 属性约简自寻优算法[J]. 计算机研究与发展，2001，38(8)：904－910.

[8] 张文修. 粗糙集理论与方法[M]. 北京：科学出版社，2001.