

基于粗糙集的决策表属性约简方法的研究

吴守领¹, 杨颖¹, 杨磊², 刘磊³

(1. 广西大学 计算机与电子信息学院, 广西南宁 530004;

2. 广西壮族自治区计算中心, 广西南宁 530022;

3. 软通动力信息技术(集团)有限公司, 广东深圳 518129)

摘要:求核和属性约简是粗糙集理论研究的一个核心问题。文中主要针对现有的一些决策表属性约简算法存在的不足,尤其是基于信息熵的属性约简算法在较大数据集上效率不高的问题提出改进。主要通过结合粗糙集的相关理论来改进原有的属性约简算法在求核中的约束条件,进而在原有算法的基础上提出了一种改进算法。在求约简属性集时,利用新提出的约简算法,使计算复杂度降低,同时保持了高效的决策准确率。实验结果表明改进后的决策表属性约简方法能够更加快速有效地找到约简集。

关键词:约简集;属性约简;粗糙集

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2012)01-0032-04

Study of Decision Table Attribute Reduction Methods Based on Rough Set

WU Shou-ling¹, YANG Ying¹, YANG Lei², LIU Lei³

(1. Computer and Electronic Information College, Guangxi University, Nanning 530004, China;

2. Guangxi Computing Center, Nanning 530022, China;

3. Ruantong Dongli Information Technology Company, Shenzhen 518129, China)

Abstract: Searching core and attribute reduction is a main issue of the rough sets theory. To solve some existing shortcomings of the decision table attribute reduction algorithm, in particular, entropy-based algorithm has low efficiency for reduction of large data sets, so it proposed an improved algorithm based on the theory of rough sets. The new algorithm changed the constraint condition in searching core through using some rough sets theory. It has high efficiency and has low time complexity in searching core and attribute reduction. Experiment results show that the algorithm can find a good attribute subset.

Key words: reduction set; attribute reduction; rough set

0 引言

粗糙集理论是一种数学工具,它主要用来处理不确定的、模糊的知识,它最早被波兰科学家 Z. Pawlak 于 1982 年提出^[1]。通过粗糙集理论的相关知识,可以很快地找出在数据中起着关键性或决定性作用的属性,可以根据这些属性对数据进一步的简化。粗糙集理论在处理不完整的信息方面有着比较突出的优势,因此它被公认为是决策支持、数据挖掘及知识获取的高效工具。在粗糙集理论的学习中,求核以及属性约

简是它的一个重要内容。

所谓属性约简就是在保持原有知识分类能力不变的情况下,将那些不重要的或不相关的属性删除,进而简化原来的信息系统。求核以及属性约简的方法很多,无论那一种方法,它们所追求的目的都是一样的,那就是高效地找出最佳属性约简集。然而 Wong S. K. M 和 Ziarko W. 已经证明找到最佳属性约简是 NP-hard 问题^[2,3],因而目前如何寻找高效的属性约简算法是粗糙集理论研究的一大挑战。文中就是针对基于信息熵的属性约简算法在数据集较大的情况下时间复杂度及效率都不是很理想的问题,提出了改进方法,降低了时间复杂度,并能较快地生成属性约简集。

收稿日期:2011-05-31;修回日期:2011-09-10

基金项目:广西自治区科学研究与技术开发攻关计划课题(桂科攻 0816004-16)

作者简介:吴守领(1986-),男,河南商丘人,硕士研究生,研究方向为并行算法与分布式计算、数据挖掘;杨颖,教授,硕士生导师,主要研究方向为并行算法、数据挖掘。

1 粗糙集理论基础

粗糙集理论从集合的视角对知识进行定义,把知

识看作是论域的划分,构成一个信息系统,从而对知识进行分析和处理。下面介绍粗糙集的一些基本概念^[4]:

定义1 决策表。

一个决策表可以形式化的定义为: $S = (U, C \cup D, V, f)$, 其中 $U = \{X_1, X_2, X_3, \dots, X_n\}$ 是所感兴趣对象的有限集合, $C \cup D$ 是属性的有限集, 其中 C 为条件属性集, D 为决策属性集, 并且, $C \cap D = \emptyset$, V 为属性集 $C \cup D$ 的值域, $f: U \times (C \cup D) \rightarrow V$ 为一个信息函数, 表示任一对象的属性在 V 上的取值, 即 $f(x, r) \in V_r$, 它指定了 U 中每一对象 X 的属性值。 $\theta \rightarrow \varphi$ 为知识表达语言中的决策规则, 其中 θ 和 φ 分别称为 $\theta \rightarrow \varphi$ 的因和果。对于一个决策表 S , 当所有规则 $\theta \rightarrow \varphi$ 为真时, 则称决策表 S 是相容的, 否则称不相容。

定义2 知识约简。

令 R 为一族等价关系, $r \in R$, 如果 $\text{ind}(R) = \text{ind}(R - \{r\})$, 则称 r 为 R 中不必要的; 否则称 r 为 R 中必要的。

设 $Q \subseteq P$ 。如果 Q 是独立的, 且 $\text{ind}(Q) = \text{ind}(P)$, 则称 Q 为 P 的一个约简。 P 中所有必要关系组成的集合称为 P 的核, 记做 $\text{core}(P)$ 。

核与约简有如下关系: $\text{core}(P) = \cap \text{red}(P)$, 其中 $\text{red}(P)$ 表示 P 的所有约简。

定义3 知识的依赖。

令 $K = (U, R)$ 为一知识库, 且 $P, Q \in R$ 当 $k = \gamma_P(Q) = |\text{pos}_P(Q)| / |U|$ 称知识 Q 是 k 度依赖于知识 P 的, 记作 $P \Rightarrow_k Q$ 。当 $k = 1$ 时, 我们称 Q 完全依赖于 P ; 当 $0 < k < 1$ 时, 称 Q 粗糙依赖于 P ; 当 $k = 0$ 时, 称 Q 完全独立于 P 。

定义4 重要性和核。

如果 (U, A) 是一个决策表, $A = C \cup D, C \cap D = \emptyset$, 其中 C 是条件属性集, D 是决策属性集。令 $X \subseteq C, Y \subseteq D, U/Y \neq \{U\}$, 给定 $x \in X$, 如果 $S_x(Y) \supset S_{X-\{x\}}(Y)$, 则称 x 在 X 中是重要的(对于 Y 而言);

如果 $S_x(Y) = S_{X-\{x\}}(Y)$, 则称 x 在 X 中是不重要的(对于 Y 而言), 则属性子集 $C' \subseteq C$ 关于 D 的重要性定义为:

$$\sigma_{cd}(C') = \gamma_C(D) - \gamma_{C-C'}(D) \quad (1)$$

定义5 相对约简。

令 (U, A) 是一个决策表, $A = C \cup D, C \cap D = \emptyset$, 其中 C 是条件属性集, D 是决策属性集。令 $X \subseteq C, Y \subseteq D, U/Y \neq \{U\}$ 。总是可以找到一个极小子集 $X_0 \subseteq X$ 使得 $S_{X_0}(Y) = S_X(Y)$ 。即:

(1) $S_{X_0}(Y) = S_X(Y)$, 即 $X_0 \leftrightarrow X(Y)$

(2) 如果 $X' \subset X_0$, 则 $S_{X'}(Y) \supset S_X(Y)$

则称 X_0 是 X 的一个约简(对于 Y 而言)。

2 基于粗糙集的决策表属性约简方法研究

2.1 常见的决策表属性约简方法

决策表是一类特殊重要的知识表达系统, 多数决策问题可以用决策表形式来表达, 它在决策分析、智能控制、规则获取和近似推理等领域有着广泛的应用^[4]。决策表的属性约简及其算法是实现决策表信息智能处理的关键内容, 所谓的知识约简就是指在保持原始决策表条件属性和决策属性之间的依赖关系不发生变化的前提下删除冗余的属性和属性值^[5,6]。

一般属性约简算法主要是利用粗糙集的基本概念, 如正域、重要性和核等来进行计算, 文中首先就一些常见的对于决策表属性约简算法做出了概述。

(1) 利用区分矩阵的属性约简方法^[7]。

基于差别矩阵的决策表的知识表达方式简洁明了, 易于求出决策表的相对 D 核和所有的相对 D 约简, 但该方法只适用于相容的决策表^[4,7], 因此在实际应用中受到限制。

(2) 基于属性重要性的属性约简方法^[8]。

对于决策表的属性约简有别于一般的属性约简, 在文献[2,3]已经证明求决策表的属性约简集是一个 NP-hard 问题, 因此在实际的应用中一般只求出其相对约简。在决策表的条件属性中, 有些是重要属性, 有些是非重要属性, 基于属性重要性的约简方法主要是以决策表的区分矩阵为着手点, 根据属性重要性来顺序地选出最重要的属性加入到核属性中, 进而得到决策表的一个相对属性约简集。然而计算属性的重要性度及排序都增加了算法的时间复杂度, 在数据集较大的情况下该算法运行的时间也随之增加。

(3) 基于信息熵的属性约简方法。

文献[9]中提及: 基于信息熵的属性约简算法的基本思想是一种基于启发式信息的属性约简算法, 此属性约简算法引入了一种新的启发式信息—信息熵, 从而从新的角度获得高效的属性约简算法。但是此方法在数据集较大的情况下, 时间复杂度会随之增加。

对上述基于信息熵的属性约简方法中仔细分析后, 在进行属性约简时的步骤中, 要计算每一个剩余属性的信息熵增益, 这在很大程度上增加了算法的时间复杂度, 降低了该算法的执行效率, 又由于传统的基于信息熵的决策表属性约简方法采用的启发函数是基于信息增益的^[10-12], 而且终止条件也需要计算 $\text{pos}_C(D) = \text{pos}_B(D)$ 是否相等, 在条件属性较多的情况下, 该方法的效率就会变得很低, 根据以上分析, 对算法加以改进, 把 $\gamma_R(D) = \gamma_C(D)$ 作为终止条件, 使得该算法能够更好地适应较大的数据集的属性约简

2.2 改进的基于属性依赖度的约简算法

改进后的算法伪代码描述如下:

求核:

(1) 初始化 $\text{core} = C$, 并设 C 中所包含的属性个数为 n ;

(2) 令 $B = C - \{C_i\}$, 计算 $\text{pos}_C(D)$ 是否等于 $\text{pos}_B(D)$, 如果相等, 则 $\text{core} = \text{core} \cap B$, 否则返回 (2) 从头开始 $i = 1, 2, 3, \dots, n$;

(3) 返回 core 。

求约简集:

(1) 数据初始化, $R = \text{core}$;

(2) $T = R, x_i \in (C - R)$, 如果 $\gamma_{R \cup \{x_i\}}(D) > \gamma_R(D)$, 则令 $T = R \cup \{x_i\}$, $R = T$ 并转到 (3), 否则继续循环 (2); 若没有剩余属性可选, 也转到 (3); $i = 1, 2, 3, \dots, n (n \leq |C - R|)$;

(3) 判断 $\gamma_R(D) = \gamma_C(D)$, 如果是则结束, 转到 (4), 否则转到 (2);

(4) 返回 R 。

该算法中采用 $\gamma_R(D) = \gamma_C(D)$ 为终止条件, 这在数据集较大的情况下会降低运行的时间, 提高了算法的执行效率。

2.3 应用实例

文中选取了广西医科大学附属医院的门诊数据作为测试数据, 经过分析后, 形成的决策表如表 1 所示:

表 1 选取应用测试数据的决策表

病人	条件属性			决策属性
	头痛	肌肉痛	体温	流感
M1	是	是	正常	否
M2	是	是	高	是
M3	是	是	很高	是
M4	否	是	正常	否
M5	否	否	高	否
M6	否	是	很高	是
M7	否	否	高	是
M8	否	是	很高	否

对以上决策表进行分析后, 设 $S = (U, A, V, f)$ 为一个信息系统, 根据此表可以得出 $U = \{e_1, e_2, \dots, e_8\}$, $C = \{\text{头痛}, \text{肌肉痛}, \text{体温}\}$, $D = \{\text{流感}\}$, 令 $C_1 = \text{头痛}$, $C_2 = \text{肌肉痛}$, $C_3 = \text{体温}$, 则根据求核及改进后的求约简集的相关算法, 求核及求约简集的过程如下所示:

第一步求核:

$$U/\{C_1\} = \{\{M_1, M_2, M_3\}, \{M_4, M_5, M_6, M_7, M_8\}\}$$

$$U/\{C_2\} = \{\{M_1, M_2, M_3, M_4, M_6, M_8\}, \{M_5, M_7\}\}$$

$$U/\{C_3\} = \{\{M_1, M_4\}, \{M_2, M_5, M_7\}, \{M_3, M_6, M_8\}\}$$

$$U/\{C_1, C_2\} = \{\{M_1, M_2, M_3\}, \{M_4, M_6, M_8\}, \{M_5, M_7\}\}$$

$$U/\{C_1, C_3\} = \{\{M_1\}, \{M_2\}, \{M_3\}, \{M_4\}, \{M_5, M_7\}, \{M_6, M_8\}\}$$

$$M_7\}, \{M_6, M_8\}\}$$

$$U/\{C_2, C_3\} = \{\{M_1, M_4\}, \{M_2\}, \{M_5, M_7\}, \{M_3, M_6, M_8\}\}$$

$$U/C = \{\{M_1\}, \{M_2\}, \{M_3\}, \{M_4\}, \{M_5, M_7\}, \{M_6, M_8\}\}$$

$$U/D = \{\{M_3, M_2, M_6, M_7\}, \{M_1, M_4, M_5, M_8\}\}$$

由以上可知 C 的 core 为 $\{C_1, C_3\}$, 因为

$$\text{POS}_{(C-\{C_1\})}(D) = \{M_1, M_2, M_4\} \neq \text{POS}_C(D)$$

$$\text{POS}_{(C-\{C_3\})}(D) = \{M_1, M_2, M_3, M_4\} = \text{POS}_C(D)$$

$$\text{POS}_{(C-\{C_1\})}(D) = \emptyset \neq \text{POS}_C(D)$$

$$\text{POS}_{(C-\{C_1, C_3\})}(D) = \{M_1, M_4\} \neq \text{POS}_C(D)$$

$$\text{POS}_{(C-\{C_1, C_3\})}(D) = \emptyset \neq \text{POS}_C(D)$$

$$\text{所以 } \text{core} = C - \{C_2\} = \{C_1, C_3\}$$

第二步求约简:

第一步可知 $R = \{C_1, C_3\}$, 所以 $x_1 = C_2$, 又因为 $\gamma_{R \cup \{x_1\}}(D) = \gamma_R(D)$, 继续选择 x_i , 由于没有剩余属性可选, 判断 $\gamma_R(D) = \gamma_C(D)$ 是否成立, 由于 $\gamma_R(D) = 0.5 = \gamma_C(D)$, 所以结束, 约简集 $R = \{C_1, C_3\}$ 。

但是在某些情况下, 经过约简后得到的约简集不一定是唯一的, 如何选出一个比较合理的候选约简集作为最简约简集合, 作为本算法的补充, 可以采用粗糙集理论的属性重要度量来进一步计算每一个候选约简集的属性重要度量, 哪一个候选集的重要度量的和最大就选取哪一个作为最简的约简集。

例如集合 R_1, R_2 都为 C 的约简集情况下可以计算

$$\sigma_{CD}(R_1) = \gamma_C(D) - \gamma_{C-R_1}(D) \quad (2)$$

$$\sigma_{CD}(R_2) = \gamma_C(D) - \gamma_{C-R_2}(D) \quad (3)$$

的值, 并比较它们的大小, 若 $\sigma_{CD}(R_1) \geq \sigma_{CD}(R_2)$, 则选择 R_1 作为约简集, 否则选择 R_2 作为约简集。

2.4 实验数据集测试对比

文中选取了最常用的测试数据集 UCI 来作为测试数据, 它收集了曾在各种机器学习方法中使用过的数据库。文中选取了最常用的几个数据库来进行试验测试, 实验结果如表 2 所示:

表 2 算法对比表

数据库名称	原属性个数	约简后的属性个数	
		原算法	改进的算法
Kinship Domain	2	2	2
Mushroom database Domain	22	12	10
German Credit Data	24	13	10
Chess End-Game	36	28	24
Standardized Audiology Database	38	14	7

从表 2 中可以看出, 当需要约简的属性个数较少时, 改进后的算法与原算法得到的约简结果基本相同, 但是随着需要约简属性个数的增加即数据集的增大, 改进后的算法具有较好的优越性, 能够得出属性个数较少的约简集, 这也正是属性约简算法所追求的。

此外,文中还对基于信息熵的属性约简算法与改进的算法在时间效率上做了比较,各个数据集的规模如图1所示,比较结果如图2所示:

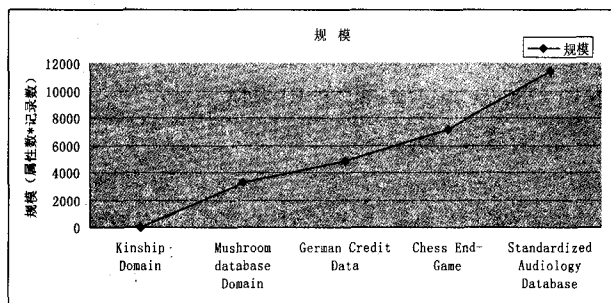


图1 数据集规模

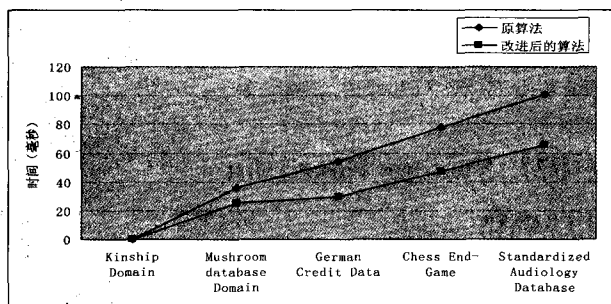


图2 时间对比图

从图1及图2中可以看出,当数据集较小时,改进后的算法与原来的算法效率基本相等,但当数据集较大时,本算法的效率高于原来的算法,在实际中可以运用。

3 结束语

针对大多数属性约简方法在数据集较大的情况下计算效率降低以及运行时间较长等问题,提出了基于属性依赖度的约简算法,尤其是在对大的数据集的属

性进行约简时较为有效,并用具体的应用验证了该算法的有效性。

参考文献:

- [1] Pawlak Z. Rough sets and decision tables[J]. Lecture Notes in Computer Science, 1985, 208: 187-196.
- [2] Pawlak Z. Rough set theory and its applications to data analysis[J]. Cybernetics and Systems: An International Journal 1998, 29: 661-688.
- [3] Wong S K M, Ziarko W. On optimal tables[J]. Bulletin of Polish Academy of Sciences, 1985, 33(11-12): 693-696.
- [4] 苗夺谦, 李道国. 粗糙理论、算法与应用[M]. 北京: 清华大学出版社, 2008.
- [5] 周爱武, 周闪闪, 邹武, 等. 一种基于变精度粗糙集理论的属性约简算法[J]. 计算机技术与发展, 2009, 19(7): 35-37.
- [6] 王荣, 陈黎伟, 吴涛. 一种改进的属性约简算法[J]. 计算机技术与发展, 2008, 18(2): 147-148.
- [7] 孙凌宇, 彭宜戈, 冷明. 基于动态区分矩阵的属性约简算法[J]. 计算机工程, 2008, 34(24): 216-217.
- [8] 杨成福, 舒兰. 基于属性重要性的决策表属性约简方法[J]. 计算机技术与发展, 2006, 16(11): 63-66.
- [9] 吴尚志, 苟平章. 粗糙集和信息熵的属性约简算法及其应用[J]. 计算机工程, 2011, 37(7): 57-61.
- [10] 丁守祯, 桑琳, 朱全英, 等. 基于信息熵的粗糙集属性约简及其应用[J]. 计算机工程与应用, 2007, 43(35): 245-247.
- [11] 徐章艳, 侯伟, 宋威. 一个有效的基于信息熵的启发式属性约简算法[J]. 小型微型计算机系统, 2009, 30(9): 1806-1809.
- [12] 舒文豪, 徐章艳, 杨炳儒, 等. 一种新的信息熵属性约简算法[J]. 计算机工程与应用, 2009, 45(32): 109-113.
- [6] 薛小平. 基于 Pub/Sub 系统的 RFID 网络及其路由研究[D]. 北京: 北京交通大学, 2008.
- [7] 夏明望. 钱塘 RFID 中间件平台关键技术研究[D]. 杭州: 浙江大学, 2007.
- [8] 郭跃辉, 艾君锐. 基于 ALE 规范的 RFID 中间件的研究与设计[J]. 现代计算机(专业版), 2010(10): 79-82.
- [9] EPCglobal. The Application Level Events(ALE) Specification Version 1.0[R]. Boston: EPCglobal, 2005.
- [10] Nath B, Reynolds F, Want R. RFID technology and applications[J]. IEEE Pervasive Computing, 2006, 5(1): 22-24.
- [11] 肖楠, 郑文岭. 一种基于 RFID 的物流管理系统的设计[J]. 计算机技术与发展, 2008, 18(7): 237-243.
- [12] EPCglobal. The Application Level Events(ALE) Specification Version 1.1[R]. Boston: EPCglobal, 2009.
- [1] 张晓鹏. RFID 中间件事件处理模型的研究与实现[D]. 南昌: 南昌大学, 2010.
- [2] 邓海生, 李军怀. RFID 中间件研究与设计[J]. 计算机技术与发展, 2008, 18(11): 188-191.
- [3] 邓海生, 李军怀. 基于 RFID 的数据采集中间件[J]. 计算机技术与发展, 2007, 17(9): 55-57.
- [4] 徐德兴. RFID 中间件动态信息实时数据处理研究与实现[D]. 南昌: 南昌大学, 2010.
- [5] 郝兴贞. 基于 ALE 的 RFID 中间件的设计与实现[D]. 青岛: 中国海洋大学, 2007.

(上接第31页)

的实现,目前也仅仅停留在概念阶段,在实际部署应用时,网络中的海量数据的传输处理,阅读器节点的管理和协调等都需要进一步研究。