

用属性单值表示的决策表简化算法及属性核计算

李淑智^{1,2}, 徐光华^{1,3}, 华成¹

(1. 西安交通大学机械工程学院, 710049, 西安; 2. 西安第二炮兵工程学院, 710025, 西安;

3. 西安交通大学机械制造系统工程国家重点实验室, 710049, 西安)

摘要: 为了降低决策表的存储空间, 提高计算属性核的效率, 提出了一种用属性单值表示的决策表简化算法. 给出了条件属性的尺度、权值、属性单值和简化决策表的定义, 以及尺度运算、权值运算、每个对象的条件属性值集合转换为一个属性单值的简化运算的严密公式, 并通过决策表的可逆定理证明了简化决策表与原始决策表具有相同的信息表达能力. 利用该算法简化决策表, 条件属性的表达由多维降低为一维, 从而有效地降低了决策表的存储空间. 实验结果表明, 简化决策表属性核的计算效率明显比现有决策表属性核的计算效率高, 尤其是对于大型决策表, 这种优势更加明显.

关键词: 决策表; 属性单值表示; 简化算法; 属性核

中图分类号: TP18 **文献标志码:** A **文章编号:** 0253-987X(2010)01-0087-04

Decision Table Simplification Algorithm Based on Attribute Single-Valued Representation and Attribute Core Calculation

LI Shuzhi^{1,2}, XU Guanghua^{1,3}, HUA Cheng¹

(1. School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 2. The Second Artillery Engineering

Institute, Xi'an 710025, China; 3. State Key Laboratory for Manufacturing System Engineering,

Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: To reduce the decision table of storage space, and improve the efficiency of computing attribute core, a decision table simplification algorithm based on attribute single-valued representation is proposed. The definitions of the size and weight of condition attribute, single-valued attribute and simplified decision table are presented. Then the rigorous calculation formulas of the size, weight and the condition attribute value set of every object converting into a single-value are given. The inverse theorem is proved, which indicates that the simplified decision table are endowed the same information with the original decision table. The storage space of decision table is reduced greatly because the multi-dimension attribute values are decreased to one dimension ones. Then the attribute core algorithm based on the simplified decision table is compared with that based on existing decision table. Experimental results show that the computing efficiency of the former is obviously higher than that of the latter, especially for large-scale decision table.

Keywords: decision table; attribute single-valued representation; simplification algorithm; attribute core

在 Rough 集理论^[1-2]中, 属性核的算法和计算效率是 Rough 集理论研究的热点问题之一^[3-8]. Hu

等人^[4]给出了基于区分矩阵的属性核算法, 目前很多属性核的计算都是基于区分矩阵或在此基础上进

行改进的,它的时间复杂度一般为 $O(|A||U|^2)$,空间复杂度为 $O(|U|^2)$.文献[5-6]采用了快速排序和计算正域的属性核算法,文献[7]采用快速排序和检测决策表一致性的属性核算法,算法的时间复杂度为 $O(|A|^2|U|\lg|U|)$,空间复杂度为 $O(|A||U|)$.文献[8]提出了决策表简化算法和基于简化差别矩阵的求核算法,其时间复杂度为 $\max(O(|A||U/A|^2), O(|A||U|))$,空间复杂度为 $\max(O(|U|), O(|A||U/A|^2))$.通过对文献[3-8]的分析,发现属性核的计算复杂度与决策表的存储空间 $|A||U|$ 密切相关,因此决策表的存储空间直接影响着属性核的计算效率.

为了提高属性核的计算效率,本文提出了一种用属性单值表示的决策表简化算法,使条件属性的表达由多维降低到一维,因此决策表的存储空间被大幅度降低,为提高属性核的计算效率奠定了基础.本文的实验证明,基于简化决策表的快速属性核算法能够快速高效地计算大型决策表的属性核.

1 Rough 集的基本概念^[1-3]

定义 1 一个信息系统 $L=(U, A, V, f)$, 其中 U 是论域, 元素 $x \in U$ 则称为一个对象; A 是属性集合, 元素 $a \in A$ 称为一个属性; $V = \bigcup_{a \in A} V_a$, V_a 表示属性 a 的值域; f 是 $U \times A \rightarrow V$ 的映射, 它指定 U 中每个对象 x 的属性值, 即 $x \in U, a \in A, f(x, a) \in V_a$. 如果 A 可以分为条件属性集 C 和决策属性集 D , 即 $C \cup D = A, C \cap D = \emptyset$, 则该信息系统称为决策系统或决策表. 若决策属性集合只包含一个元素, 即 $D = \{d\}$, 则 $L=(U, C, d, V, f)$ 为单决策属性的决策表.

定义 2 在 $L=(U, C, D, V, f)$ 中, 若 $P \subseteq A$, 则 P 对 U 的划分为 $U/P = \{P_1, P_2, \dots, P_n\}$, $0 < n \leq |U|$, $P_i (1 \leq i \leq n)$ 是等价类.

定义 3 在 $L=(U, C, D, V, f)$ 中, 若 $P \subseteq A, X \subseteq U$, $U/P = \{P_1, P_2, \dots, P_n\}$, 称 $P_-(X) = \bigcup \{P_i \mid P_i \in U/P, P_i \subseteq X\}$ 为 X 关于 P 的下近似集.

定义 4 在 $L=(U, C, D, V, f)$ 中, 若 $P, Q \subseteq A$, 则 P 关于 Q 的正区域 Z 为 $Z_P(Q) = \bigcup_{x \in U/Q} P_-(x)$.

定义 5 在 $L=(U, C, D, V, f)$ 中, 若 $P \subseteq C$, $Z_P(D) = Z_C(D)$, 且不存在 $R \subseteq P, Z_P(D) = Z_R(D)$, 则 P 为 C 相对于 D 的一个约简, 即所有约简集合的交集称为 C 的核, 记为 $H_D(C)$.

2 简化决策表

2.1 基本定义

定义 6 在 $L=(U, C, d, V, f)$ 中, 若 $P \subseteq C$, $P = \{p_1, p_2, \dots, p_m\}$, 则 $p_j (1 \leq j \leq m)$ 的尺度为 s_j , P 的尺度集合为 $S = \{s_1, s_2, \dots, s_m\}$, 而

$$s_j = \max(V_{p_j}) + 1, 1 \leq j \leq m \quad (1)$$

定义 7 在 $L=(U, C, d, V, f)$ 中, 若 $P \subseteq C$, $P = \{p_1, p_2, \dots, p_m\}$, 则 P 的权值集合为 $W = \{w_1, w_2, \dots, w_m\}$. $p_j (1 \leq j \leq m)$ 的权值为

$$w_j = \begin{cases} 1, & j = 1 \\ \prod_{k=1}^{j-1} s_k, & 2 \leq j \leq m \end{cases} \quad (2)$$

定义 8 在 $L=(U, C, d, V, f)$ 中, 若 $P \subseteq C$, 则 P 的为 P_w . 对于 $\forall x \in U$, P_w 对应的属性单值为

$$f(x, P_w) = \sum_{j=1}^m (f(x, p_j) w_j), 1 \leq j \leq m \quad (3)$$

定义 9 在 $L=(U, C, d, V, f)$ 中, 若 $P \subseteq C$, P_w 为 P 的加权属性, V_{P_w} 表示 P_w 的值域, $V_w = V_{P_w} \cup V_d$, 则 $L_w = (U, P_w, d, V_w, f)$ 为 $L_w = (U, P, d, V, f)$ 的简化决策表.

定理 在 $L=(U, C, d, V, f)$ 中, 若 $P \subseteq C, L_w = (U, P_w, d, V_w, f)$ 为 $L_w = (U, P, d, V, f)$ 的简化决策表, 则对于 $\forall x \in U, f(x, P) = \{f(x, p_1), f(x, p_2), \dots, f(x, p_m)\}$ 和 $f(x, P_w)$ 是可以逆向转换的, 即

$$f(x, p_m) = \text{int} \left(\frac{f(x, P_w)}{w_m} \right) \quad (4)$$

$$f(x, p_j) = \text{int} \left(\frac{f(x, P_w) - \sum_{k=j+1}^m (f(x, p_k) w_k)}{w_j} \right) \quad (5)$$

式中: w_k 为 p_k 的权值.

证明 由式(1)和式(2)可推出 $f(x, p_j) + 1 \leq s_j$, $f(x, p_j) \leq s_j$, 由式(3)可推出 $f(x, p_1) w_1 < s_1 = w_2$,

$$\sum_{k=1}^j (f(x, p_k) w_k) < w_{j+1}, \text{ 则}$$

$$\text{int} \left(\left(\sum_{k=1}^j (f(x, p_k) w_k) \right) / w_{j+1} \right) = 0 \quad (6)$$

$$f(x, P_w) = f(x, p_m) w_m + \sum_{k=1}^{m-1} (f(x, p_k) w_k)$$

$$\frac{f(x, P_w)}{w_m} = f(x, p_m) + \left(\sum_{k=1}^{m-1} (f(x, p_k) w_k) \right) / w_m$$

式中 $f(x, p_m)$ 是整数, 对式(6)两边取整数得到

$$\text{int}(f(x,P_W)/w_m) = f(x,p_m) +$$
$$\text{int}((\sum_{k=1}^{m-1}(f(x,p_k)w_k))/w_m)$$
$$f(x,p_m) = \text{int}(f(x,P_W)/w_m)$$

式(5)的证明与式(4)相似. 可逆定理表明了简化决策表和原始决策表具有相同的信息量,因此 Rough 集的相关定义同样适合于简化决策表.

2.2 简化算法及复杂度分析

原始决策表 $L=(U,C,d,V,f), P\subseteq C$, 将 $L=(U,P,d,V,f)$ 简化为 $L_W=(U,P_W,d,V_W,f)$ 的算法如图 1 所示. 当 $|P_W|=1, |d|=1$ 时, 步骤 1 的时间复杂度为 $O(|P||U|)$, 空间复杂度为 $O(|P|)$; 步骤 2 的时间复杂度为 $O(1)$, 空间复杂度为 $O(|P|)$; 步骤 3 的时间复杂度为 $O(|P||U|)$, 空间复杂度为 $O(|U|)$. 在大多数情况下, $|P|<|U|$, 简化决策表算法的时间复杂度为 $O(|P||U|)$, 空间复杂度为 $O(|U|)$.

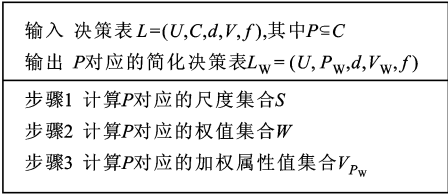


图 1 决策表简化算法

表 1 是一个一致决策表, $c_1\sim c_7$ 表示条件属性, d 表示决策属性. 设 $P=\{p_1,p_2,p_3\}$, 其中 $p_1=c_1, p_2=c_2, p_3=c_3$, 简化决策表 $L_W=(U,P_W,d,V_W,f)$ 的计算过程如下: 由决策表简化算法的步骤 1 计算 $S=\{3,3,3\}$, 由步骤 2 计算 $W=\{1,3,9\}$, 由步骤 3 计算 $V_{P_W}=\{0,15,5,13,21,13,9,13\}$, 简化后的决策表如表 2 所示.

表 1 原始决策表

U	c_1	c_2	c_3	c_4	c_5	c_6	c_7	d
x_1	0	0	0	0	1	1	0	0
x_2	0	2	1	2	2	0	0	0
x_3	2	1	0	2	0	1	1	1
x_4	1	1	1	2	2	0	1	1
x_5	0	1	2	2	2	0	1	0
x_6	1	1	1	1	1	1	1	0
x_7	0	0	1	2	1	0	0	1
x_8	1	1	1	2	1	1	1	1

表 2 简化决策表

U	P_W	d
x_1	0	0
x_2	15	0
x_3	5	1
x_4	13	1
x_5	21	0
x_6	13	0
x_7	9	1
x_8	13	1

3 快速属性核算法及复杂度分析

由于简化决策表和原始决策表具有相同的信息表达能力, 因此属性核的计算可以建立在简化决策表上. 本文给出了基于简化决策表的快速属性核算法(见图 2), 该算法首先对决策表进行简化, 然后通过检测一致决策表中是否出现冲突样本的方法计算属性核.

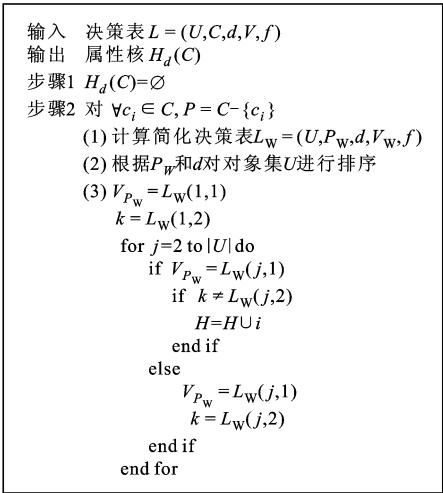


图 2 属性核算法

对属性核算法的复杂度进行分析: 步骤 1 的时间和空间复杂度为 $O(1)$, 步骤 2 计算 L_W 的时间复杂度为 $O(|C||U|)$, 空间复杂度为 $O(|U|)$; 排序的时间复杂度为 $O(|U|\lg|U|)$, 空间复杂度为 $O(|U|)$. 判断核属性的时间复杂度为 $O(|U|)$, 空间复杂度为 $O(1)$, 步骤 2 需要计算 $|C|$ 次, 其时间复杂度为 $(O(|C|^2|U|), O(|C||U|\lg|U|))$, 空间复杂度为 $O(|U|)$. 因此, 该属性核算法的时间复杂度为 $(O(|C|^2|U|), O(|C||U|\lg|U|))$, 空间复杂度为 $O(|U|)$.

4 实验结果及分析

选用 UCI 机器学习数据库^[9]中的 5 个一致决策表,如表 3 所示,其中 Mushroom 数据集的第 1 个属性因缺失大量数据,所以去掉了该属性。

表 3 决策表的数据描述

决策表 ^[9]	实例数	条件属性数	分类数
zoo	101	16	7
Ionosphere	351	34	2
Tic-tac-toe	958	9	2
Chess	3 196	36	2
Mushroom	8 124	21	2

文献[7]的属性核算法简称算法 1,文献[8]提出了去掉重复样本的决策表简化方法,将论域 U 简化为 U' ,该简化决策表的属性核算法简称算法 2。将算法 1、算法 2 与本文的属性核算法在 T2300PC 机(512 MB Windows XP matlab7.0)上进行了对比实验,结果如表 4 所示,表中的执行时间 t 为 10 次实验的平均值。

表 4 不同属性核算法的比较

决策表 ^[9]	核属性条数	算法 1	算法 2		本文算法
		t/s	$ U' $	t/s	t/s
zoo	2	0.015	59	0.010	0.005
Ionosphere	1	0.141	335	0.150	0.015
Tic-tac-toe	0	0.078	958	0.094	0.022
Chess	27	1.047	3 196	1.078	0.187
mushroom	0	2.078	7 548	1.782	0.218

由表 4 可以看出,本文属性核算法在效率上明显高于算法 1 和算法 2,随着决策表存储空间 的增加,其优越性更加突出,这是由于该算法具有较低的时间复杂度和空间复杂度。算法 2 中如果简化决策表的存储空间明显减小(如决策表 zoo),则算法 2 在效率上高于算法 1,否则算法 2 在效率上要低于算法 1。本文算法的高效性表明该算法非常适合计算大型决策表的属性核。

本文算法的不足之处为:如果决策表的条件属性很多,同时属性值又过于离散, $f(x, P_w)$ 的值有可能超出计算机的显示范围,造成 $f(x, P_w)$ 的值失真,从而导致计算属性核失败。对于该类决策表,条件属性的尺度以降序排列,条件属性按其序列重新排列,构造新的决策表,再利用本文算法计算属性核,可有效避免由 $f(x, P_w)$ 的失真而引起的属性核计算失败。

5 结 论

为了降低决策表的存储空间,提高属性核算法的效率,本文提出了属性单值表示的决策表简化方法,使条件属性的表达由多维降低到一维,利用简化的决策表计算属性核,其时间复杂度为 $(O(|C|^2|U|), O(|C||U|\lg|U|))$,空间复杂度为 $O(|U|)$ 。理论分析和实验对比表明,该属性核算法在效率上高于现有的属性核算法,对于大型决策表,本文算法的优势更加明显。

参考文献:

[1] PAWLAK Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11(5):341-356.

[2] PAWLAK Z. Rough set theory and its application to data analysis [J]. Cybernetics and Systems, 1998, 9(5): 661-668.

[3] 王国胤. Rough 集理论与知识获取 [M]. 西安:西安交通大学出版社,2001.

[4] HU Xiaohua, NICK C. Learning in relational database: a rough set approach [J]. International Journal of Computational Intelligence, 1995, 11(2):323-338.

[5] 刘少辉,盛秋戡,史忠植. 一种新的快速计算正区域的方法 [J]. 计算机研究与发展,2003,40(5):637-642. LIU Shaohui, SHENG Qiujian, SHI Zhongzhi. A new method for fast computing positive region [J]. Journal of Computer Research and Development, 2003,40(5): 637-642.

[6] 刘少辉,盛秋戡,吴斌,等. Rough 集高效算法的研究 [J]. 计算机学报,2003,26(5):524-529. LIU Shaohui, SHENG Qiujian, WU Bin. Research on efficient algorithms for rough set methods [J]. Chinese Journal of Computer, 2003,26(5):524-529.

[7] 赵军,王国胤,吴中福,等. 一种高效的属性核计算方法 [J]. 小型微型计算机系统,2003,24(11):1950-1953. ZHAO Jun, WANG Guoyin, WU Zhongfu, et al. An efficient approach to computer the feature core [J]. Mini-Micro Systems,2003,24(11):1950-1953.

[8] 徐章艳,杨炳儒. 一个基于差别矩阵的快速求核算法 [J]. 计算机工程与应用,2006,42(6):4-6. XU Zhangyan, YANG Bingru. Quick computing core algorithm based on discernibility matrix [J]. Computer Engineering and Application, 2006, 42(6):4-6.

[9] ASUNCION A, NEWMAN D J. UCI machine learning repository [DB/OL]. [2009-02-26]. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.