

粗糙集高效遗传约简算法

周江卫, 冯博琴, 刘洋

(西安交通大学计算机科学与技术系, 710049, 西安)

摘要: 提出了简化差别函数的概念及其算法, 简化差别函数不仅具有与决策表相同的决策能力, 而且剔除了由决策表导出的原始差别函数中的重复项和冗余项. 为了降低遗传约简算法适应度函数的搜索空间, 有效提高计算速度, 又提出了一种高效遗传约简算法, 其将染色体对简化差别函数的覆盖度以及染色体中包含 1 的个数作为适应度函数的参数, 从而保证了算法朝着最小约简收敛, 并从理论上证明了算法的属性约简结果为最优, 算法的时间复杂度为 $O(|f'| |C| |U|^2)$. 同时, 通过 4 个实例对算法进行了验证, 结果表明简化差别函数的项分别为原来的 0.39%、0.000 8%、0.000 08% 和 0.000 3%, 且在 500 代以内能够获得最小属性约简.

关键词: 粗糙集; 遗传约简算法; 差别函数; 属性约简

中图分类号: TP18 **文献标识码:** A **文章编号:** 0253-987X(2007)04-0444-04

Efficient Genetic Reduction Algorithm for Rough Sets

Zhou Jiangwei, Feng Boqin, Liu Yang

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: The concept and algorithm of the simplified discernment function are provided. The function not only has the same decision ability as the decision table, but also eliminates all the duplicated and redundant terms included in the original discernment function deduced by the decision table. To reduce the search space of the fitness function and improve the computing efficiency, an efficient genetic reduction algorithm is provided. By regarding the chromosome coverage of the simplified discernment function and the number of "1" in the chromosome as the parameters of the fitness function, it can be ensured that the algorithm will converge at the minimum reduction and improve the searching efficiency. It is proved theoretically that the attribute reduction calculated by the algorithm is optimal and the algorithm complexity is $O(|f'| |C| |U|^2)$. The algorithm is verified by four examples, and the results show that the number of terms of simplified discernment function is 0.39%, 0.000 8%, 0.000 08% and 0.000 3% of that of the original one respectively, and the minimal reductions can be obtained within 500 iterations.

Keywords: rough set; genetic reduction algorithm; discernment function; attribute reduction

粗糙集(Rough Set)理论^[1]于 20 世纪 80 年代由波兰科学家 Zdzislaw Pawlak 提出, 它是一种处理不精确、不完全与不相容的数学工具. 在粗糙集理论中, 属性约简是其中的重要研究内容之一. 已经证明, 最小属性约简的求解是一个 NP 难问题, 因此通过穷举属性组来获得最小属性约简几乎是不可能

的.

文献[2]将遗传算法应用于属性约简, 提出了 3 种基于遗传算法的最小属性约简算法, 其一是时间复杂度为最小的个体用二进制串表示的分类遗传算法, 计算复杂度为 $O(mn|C| |U|^2)$, 其二是由决策表导出的原始差别函数的适应度计算方法, 该方法的

操作非常耗时,仅适应于解决数据对象个数比较少且属性个数较多的问题.同时研究发现,差别函数中除了存在大量的重复项以外,还存在大量的冗余项,这些项对属性约简没有贡献,却极大地增加了计算成本.

本文提出了简化差别函数,用此剔除重复项、冗余项,又将简化差别函数作为遗传算法计算适应度函数,进而获得遗传约简算法,该约简算法可大大降低计算时间,提高计算的效率.

1 基本概念

定义 1^[1] 决策表 $S=(U, C, D, V, f)$, 其中 $U=\{x_1, x_2, \dots, x_n\}$ 是论域, $C=\{c_1, c_2, \dots, c_r\}$ 为条件属性集, $D=\{d_1, d_2, \dots, d_l\}$ 为决策属性集, $f: U \times (C \cup D) \rightarrow V$ 是信息函数, $F=C \cup D, C \cap D=\emptyset, V=\bigcup V_a, a \in F, V_a$ 表示属性 a 的值域. 每一个属性子集 $P \subseteq C \cup D$ 决定了一个二元不可区分关系

$$\text{IND}(P) =$$

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

而 $\text{IND}(P)$ 构成了 U 的一个划分, 用 $U/\text{IND}(P)$ 表示(简记为 U/P), U/P 中的任何元素 $[x]_P = \{y \mid \forall a \in P, f(x, a) = f(y, a)\}$ 称为等价类.

定义 2^[3] 设 $X \subseteq U$ 为论域的一个子集, $P \subseteq C$, 则 X 的关于 P 的下近似

$$PX = \{x \in U: [x]_P \subseteq X\}$$

式中: $[x]_P$ 表示 U 中所有与 x 在关系 $\text{IND}(P)$ 下由等价元素构成的集合.

定义 3^[3] 设 $P \subseteq C$, 则对划分 $\{\Psi_1, \Psi_2, \dots, \Psi_k\}$ 的 P -近似精度 $\gamma_P = \sum_{i=1}^k |P\Psi_i| / |U|$.

定义 4^[3] 设 $P \subseteq C$, 若 $\gamma_P = \gamma_C$, 且不存在 $R \subseteq P$, 使得 $\gamma_R = \gamma_C$, 则称 P 为 C 的一个(相对于决策属性 D 的)属性约简, 所有 C 的属性约简的交称为 C 的核, 记为 $\text{Core}(C)$.

2 简化差别函数

差别函数

$$\begin{aligned} f = & (a) \wedge (b \vee c) \wedge (a \vee b \vee c) \wedge \\ & (a \vee d) \wedge (b \vee c \vee d) \wedge (b \vee c) \wedge \\ & (c \vee d) \wedge (a \vee b \vee c \vee d) \end{aligned}$$

利用合取范式以及析取范式的相关数学知识, 对差别函数进行化简后可以得到

$$f' = (a) \wedge (b \vee c) \wedge (c \vee d)$$

则称 f' 为 f 的简化差别函数. 简化差别函数不仅消

除了所有重复项(如 $b \vee c$), 而且也消除了其中的所有冗余项(如 $a \vee b \vee c, a \vee d, b \vee c \vee d, a \vee b \vee c \vee d$), 从而减少了差别函数的项数, 即项数从原来的 8 减少到了 3, 这有利于降低遗传算法的计算复杂度.

就数据量非常大的信息决策表而言, 要把差别函数所有的项都存储起来, 需要相当大的空间, 考虑到这个原因, 本文采用边求函数项边化简的方法, 以降低存储需求.

简化差别函数的算法描述如下.

输入: $S=(U, C, D, V, f), U=\{x_1, x_2, \dots, x_n\}, C=\{c_1, c_2, \dots, c_r\}, D=\{d_1, d_2, \dots, d_l\}$.

输出: 简化差别函数 f' .

(1) 采用基数排序算法^[4]得到属性核 $\text{Core}(C)$ 、正区域 $U_{\text{pos}} = \text{pos}_C(D)$ 以及负区域 $U_{\text{neg}} = U/U_{\text{pos}}$.

(2) $f' = \emptyset$.

(3) 对 U 中的任意 2 个元素 $x, y \in U$, 若 $x, y \in U_{\text{pos}}$ 且决策属性不相等, 或者 $x \in U_{\text{pos}}, y \in U_{\text{neg}}$, 则计算 x, y 的差异度 term ; 若 $x, y \in U_{\text{pos}}$, $\text{term} = \emptyset$, 或者 term 中含有 1 个或者多个核属性, 则舍弃 term ; 若存在 $\text{term}' \in f', \text{term} \wedge \text{term}' = \text{term}'$, 则舍弃 term , 否则将所有满足 $\text{term} \wedge \text{term}' = \text{term}, \text{term}' \in f'$ 的 term' 从 f' 中删除, 且 $f' = f' \cup \text{term}$.

3 高效遗传约简算法

3.1 高效遗传约简算法

3.1.1 个体表示 个体用长度 $|C|$ (条件属性个数) 的进制编码位串表示. 染色体的每一位对应一个条件属性, 染色体上的某个基因位若为 1 表示可选择其对应的条件属性, 若为 0 表示不可选择其对应的条件属性. 本文中的初始种群的个体基因值用均匀分布的随机数发生器产生, 并将每个个体的所有核属性对应的基因位置 1.

3.1.2 适应度函数 适应度函数依赖于 2 个参数, 染色体 $r(L_r)$ 中 1 的个数, 以及被 $r(L_r)$ 覆盖的简化差别函数 $f'(C_r)$ 的项数, 这 2 个变量值均归一化, 得到适应度函数

$$F(r) = \frac{|C| - L_r}{|C|} + \frac{C_r}{|f'|} \quad (1)$$

式中: $|C|$ 为条件属性个数; L_r 为 r 中基因位为 1 的个数; $|f'|$ 表示简化差别函数的项数; C_r 表示被 r 覆盖的简化差别函数的项数. 所谓 r 覆盖简化差别函数的项, 就是指被覆盖项所包含的条件属性中至少存在一个条件属性, 其对应的 r 基因位置的值为 1.

3.1.3 选择方法 每个 r 的适应度经计算后,需进行选择处理.本文采用适应度比例选择方法,从当前群体中选择优良个体,并将其复制到下一代群体之中,具体过程描述如下.

(1)每个 r 的相对适应度即为复制到下一代群体的概率,即

$$F_N(r) = \frac{F(r)}{\sum F(r)} \quad (2)$$

(2)利用模拟赌盘方式(0~1之间的随机数)决定所选择的个体.

3.1.4 交叉操作 采用单点交叉算子,其交叉概率 $p_c=0.7$,则交叉操作的执行过程为:按 3.1.3 节选择方法选择 2 个个体,然后进行随机配对;对每一对相互配对的个体,随机设置某一基因座之后的位置为交叉点,依设定的交叉概率 p_c 相互交换 2 个个体在其交叉点之后的部分染色体,从而产生出 2 个新个体.

3.1.5 变异操作 采用基本位变异算子,变异概率 $p_m=0.01$,则变异操作执行过程为:对个体的每一个基因座,依变异概率 p_m 指定其变异点;对每一个指定的变异点,核属性对应的基因位不发生变异,其他的应对其基因值做取反运算,从而产生出 1 个新个体.

3.1.6 最优保存策略 将初始最优个体设为 r 基因位为全 1 的个体,适应度设为 1.对于新一代个体,如果其最劣个体(适应度最小)的适应度小于最优个体的适应度,则用最优个体替换最劣个体,利用该方法能确保收敛.同时,新一代个体,如果其适应度最优个体(适应度最大)的简化差别函数 f' 的覆盖度 $\frac{C_r}{|f'|}=1$,且其适应度大于当前最优个体,那么将新一代中适应度最优的个体替换当前最优个体.该操作能确保最优个体与信息决策表决策能力保持一致,即最终得到的最优个体确实为信息决策表的一个约简.

3.2 算法描述

输入: $S=(U, C, D, V, f)$, $U=\{x_1, x_2, \dots, x_n\}$, $C=\{c_1, c_2, \dots, c_r\}$, $D=\{d_1, d_2, \dots, d_l\}$.

输出: 属性约简 R

(1)采用基数排序算法,得到决策属性 D 相对于条件属性 C 的依赖度 $\gamma_C(D)$ 以及属性核 $Core(C)$,如果 $\gamma_{Core}(D)=\gamma_C(D)$,则结束运算;否则下转(2).

(2)根据第 2 节中的算法计算简化差别函数.

(3)随机产生 m 个长度为 $|C|$ (条件属性的个数)的二进制串初始群体,每个个体的所有核属性对应的基因位置 1,其他位随机置 0 和 1,并计算初始群体中每个个体的适应度.

(4)依轮盘赌方法选择个体,依交叉概率 p_c 和变异概率 p_m 产生新一代群体.变异时保持核属性所对应的基因位不发生变异.

(5)依据式(1)计算新一代群体中每个个体的适应度.

(6)依据最优保存策略将最优个体复制到下一代群体中,并且依最优保存策略保存最优个体.

(7)如果连续 k 代的最优个体适应度不发生变化,或者世代数达到预先设定的最大值,算法终止;否则上转(4).

3.3 可行性分析和复杂度分析

命题 算法得到的属性约简 R 为问题的最优解.

证明 约简算法基于的是遗传算法,而遗传算法是一种最优化方法,通过进化和遗传机理,从初始群体出发,在适应度函数的约束下,不断进化、产生新的群体,最后收敛到一个特定的个体,即最优解.本文采用适应度函数,它可以控制 r 朝着最小约简的方向进化,即 C_r 值越大,该染色体覆盖简化差别函数的项就越多, L_r 可控制染色体朝着最小属性约简方向进化,从而算法得到的属性约简 R 为问题的最优解.

算法的时间复杂度由 3 部分组成,第 1 部分为计算核的时间复杂度 $O(|C||U|)$,第 2 部分为计算简化差别函数的时间复杂度 $O(|f'|||C||U|^2)$,第 3 部分为遗传算法的时间复杂度 $O(mn|f'| |C|)$.遗传算法中的 m 为群体中的个体数, n 为遗传算法终止的世代数.所以,总的时间复杂度

$$O(|C||U| + |f'| |C||U|^2 + mn|f'| |C|) = \max(O(|f'| |C||U|^2), O(mn|f'| |C|))$$

通常情况下, $O(mn|f'| |C|)$ 远小于 $O(|f'| |C||U|^2)$,从而总的时间复杂度为 $O(|f'| |C| \cdot |U|^2)$.

4 举例

采用 California 大学提供的 UCI 知识库^[5],其中包括 zoo 数据(16 个条件属性和 1 个决策属性,共有 101 项记录)、Chess End-Game 数据(36 个条件属性和 1 个决策属性,共有 3 196 项纪录)、mushroom 数据(22 个条件属性和 1 个决策属性,共有 8 124 项纪录),同时采用了 crx 数据(<http://www>.

fmt. vein. hu/softcomp/ucidata/dataset/[6],有9个离散属性和1个决策属性,共有690项记录),并用它们分别组成测试实例.算法A表示文献[2]中的方法1,算法B表示本文算法,2种遗传算法的属性约简比较如表1所示.

从表1的实验数据可以得出,简化差别函数的项数分别仅为原始项数的0.39%、0.0008%、0.00008%和0.0003%,这使得算法B在计算适应

度时的搜索空间比算法A大为降低,因而尽管算法B增加了计算简化差别函数的成本,但在效率上比算法A仍有明显的优势,随着对象数的增加,算法B的计算效率就越高.由于本文算法采用了将核属性的 r 基因位置1,采用了最优保存策略,所以使得最优约简代数比较小.在上面的实例中,采用本文算法均在500代以内得到了期望的最小属性约简,这表明本文算法适应于大数据集的最小属性约简.

表1 2种遗传算法属性约简比较

数据名称	对象数	条件属性个数	算法A			算法B			简化差别函数项数的执行时间/s
			差别函数项数	最优约简产生的代数	进化500代的执行时间/s	差别函数项数	最优约简产生的代数	进化500代的执行时间/s	
zoo	101	16	3 874	31	0.215	15	17	0.014	0
crx	690	9	117 462	223	0.665	1	1	0.022	0
Chess End-Game	3 196	36	2 509 685	429	68	2	1	0.040	0.794
mushroom	8 124	22	9 308 797	297	314	30	24	0.167	2.364

注:运行环境为P42.8机器,256 MB RAM,Windows XP操作系统;最大进化代数为500,种群数为50,交叉概率 $p_c=0.7$,变异概率 $p_m=0.01$.

5 结束语

在文献[2]遗传算法属性约简的基础上,通过简化差别函数来设计适应度函数,使得适应度的计算时间大为减少.通过将核属性的 r 基因位置1,并采用最优保存策略,使得算法能够快速收敛且保证达到全局最优.实验结果表明,本文算法的属性约简高效、可行,且对大数据集有很好的适应性.

参考文献:

- [1] Pawlak Z. Rough sets [J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356.
- [2] Wroblewski J. Finding minimal reducts using genetic algorithm, ICS research report 16/95 [R]. Warsaw, Poland: Warsaw University of Technology, 1995: 186-

189.

- [3] Pawlak Z. Rough set approach to multi-attribute decision analysis [J]. European Journal of Operational Research, 1994, 72: 443-459.
- [4] 徐章艳,刘作鹏. 一个复杂度为 $\max(O(|C|^2|U/C|, O(|C||U|)))$ 的快速属性约简算法 [J]. 计算机学报, 2006, 29(3): 391-399.
Xu Zhangyan, Liu Zuopeng. A Quick attribute reduction algorithm with complexity of $\max(O(|C|^2|U/C|, O(|C||U|)))$ [J]. Chinese Journal of Computers, 2006, 29(3): 391-399.
- [5] Forsyth R, Shapiro A, Alfred A. Knopf [EB/OL]. [2006-07-10]. <http://www.ics.uci.edu/~mlearn/MLSummary.html>.
- [6] Sano C. CRX [EB/OL]. [2006-07-10]. <http://www.fmt.vein.hu/softcomp/ucidata/dataset/>.

(编辑 苗凌)