文章编号: 1001-0920(2009)05-0738-05

一种核属性快速求解算法

葛 浩[1a], 李龙澍[2], 杨传健[1b]
(1. 滁州学院 a. 电子信息工程系, b. 计算机系, 安徽 滁州 239012; 2. 安徽大学 计算机学院, 合肥 230039)

摘 要: 针对求核算法存在所求得的核与基于正区域的核不一致以及算法的时间和空间复杂度不理想的问题,提出一种新的求核方法,并证明了由该方法所获得的核与基于正区域的核是一致的. 利用分布计数基数排序方法设计了一种高效的等价类求解算法,在此基础上给出了快速求核算法. 实验表明,所提出的算法是正确而高效的.

关键词: 粗糙集; 等价类; 正区域; 核属性
中图分类号: TP181 **文献标识码**: A

Quick algorithm for computing core attribute

GE Hao[1a], *LI Long-shu*[2], *YANG Chuan-jian*[1b]
(1a. Department of Electronic and Information Engineering, 1b. Department of Computer Science, Chuzhou University, Chuzhou 239012, China; 2. School of Computer Science, Anhui University, Hefei 230039, China. Correspondent: GE Hao, E-mail: togehao@126.com)

Abstract: The algorithms for computing the core have following shortcoming: The core acquired from these algorithms is not the core based on positive region, and the time complexity and space complexity are not good. Therefore, a new approach for computing core is provided and proved that the core is equivalent to the core based on positive region. The partition of the equivalence class is the key step for computing the core. An efficient algorithm for computing the equivalence class is designed with the approach based on radix sort by using distributing counting. On the foundation, the quick algorithm for computing the core is designed. The experimental result shows that the algorithm is correct and efficient.

Key words: Rough set; Equivalence class; Positive region; Core attribute

1 引 言

粗糙集理论[1]是由波兰数学家 Pawlak 于 1982 年提出的. 该理论能有效地分析和处理不精确、不一致、不完整等各种不完备信息,并从中发现隐含的知识,揭示潜在的规律. 求核问题是粗糙集理论研究的重要内容之一.

Hu 等[2]根据 Skowron 等[3]提出的可分辨矩阵,给出了核属性的计算方法,该算法的时间和空间复杂度均为 $O(|C||U|^2)$. 叶东毅等[4]对 Hu 的结论提出了质疑,并通过对可分辨矩阵的改进提出了一种新的算法,该算法的时间和空间复杂度仍均为 $O(|C||U|^2)$. 王国胤[5]指出,叶东毅虽然给出了修正方法,但他仍未发现产生错误的根本原因:由于不相容规则的比较,导致了 Hu 方法出错. 赵军等[6]提出了一种不需要建立可分辨矩阵的属性核计算方法,该方法的时间复杂度为 $O(|C|^2|U|\log|U|)$,空间复杂度为 $O(|U|)$,但该算法只适合相容决策表,具有一定的局限性. 为了解决因决策表存在不相容性,导致所求得的核的不正确问题,许多学者[7-10]做了大量研究:闫德勤等[7]提出了一种将决策表规范化后构造可分辨矩阵,然后求解核属性算法,算法的时间和空间复杂度为 $O(|C||U|^2)$;杨明等[8]提出了一种新的改进的可分辨矩阵及其求核方法,该算法的时间和空间复杂度分别为 $\max\{O(|C||U|\log|U|), O(|C||U||\mathrm{POS}_C(D)|)\}$ 和 $O(|C||U||\mathrm{POS}_C(D)|)$;徐章艳等[9]给出了简化可分辨矩阵的定义,并设计了一种求核算法,算法的时间和空间复杂度分别被降为 $\max\{O(|C||U/C|^2), O(|C||U|)\}$ 和 $\max\{O(|U|), O(|C||U/C|^2)\}$. 上述方法[7-10]均需要创建可分辨矩阵或简化

收稿日期: 2008-05-12; **修回日期**: 2008-07-09.
基金项目: 安徽省自然科学基金项目(050420204); 安徽高校省级自然科学研究项目(KJ2008B117).
作者简介: 葛浩(1976—),男,安徽滁州人,讲师,硕士,从事数据挖掘和粗糙集的研究; 李龙澍(1956—),男,安徽亳州人,教授,博士生导师,从事不精确信息处理和智能软件的研究.

的可分辨矩阵,如果样本集对象很多,可分辨矩阵要占用很多的空间,而且计算量增大,必然影响求解效率.

本文提出一种新的核属性计算方法. 该方法不需要建立可分辨矩阵,大大减少了算法所需的空间,而且由该方法求得的核与基于正区域的核是一致的. 为了降低算法的时间复杂度,将分布计数的基数排序思想应用于等价类 U/C 的求解过程,使求解 U/C 的时间复杂度降低到 $O(|C||U|)$. 在此基础上设计求核算法,其时间复杂度为 $O(|C|^2|U|)$,空间复杂度为 $O(|U|)$.

2　基本概念

定义 1[11]　一个决策表信息系统可定义为

$$S = (U, A, V, f).$$

其中:U 为论域,是对象的集合,$U = \{x_1, x_2, \cdots, x_n\}$;$A$ 为属性集,$A = \{a_1, a_2, \cdots, a_m, d\}$,$A$ 由2部分组成,$A = C \cup D$ 且 $C \cap D = \Phi$,C 为条件属性集,D 为决策属性集,一般情况下 D 中只含有一个属性 $D = \{d\}$;V 为属性的值域,$V = \{V_{a1}, V_{a2}, \cdots, V_{am}, V_d\}$;$f$ 为信息函数 $f: U \times A \rightarrow V$, $\forall a \in A, x \in U$, 有 $f(x, a) \in V_a$.

定义 2[11]　对于决策表信息系统 $S = (U, A, V, f)$,令 $R \subseteq A$,则 $\mathrm{ind}(R) = \{(x_i, x_j) \mid f(x_i, b) = f(x_j, b), \forall b \in R\}$ 称为 S 的不可区分关系. 显然,不可区分关系是一个等价类,含 x 的等价类记为 $[x]_{\mathrm{ind}(R)}$ 或 $[x]_R$. R 在 U 上导出的划分记为 $U/\mathrm{ind}(R)$ 或 U/R.

定义 3[11]　对于决策表信息系统 $S = (U, A, V, f)$,令 $R \subseteq A$, $R\text{-}X = \{x \in U \mid [x]_{\mathrm{ind}(R)} \subseteq X\}$ 称为 X 的 R 下近似集;$R^{-}X = \{x \in U \mid [x]_{\mathrm{ind}(R)} \cap X \neq \Phi\}$ 称为 X 的 R 上近似集;$\mathrm{POS}_R(X) = R\text{-}X$ 称为 X 的 R 正区域.

定义 4[11]　决策表信息系统 $S = (U, C \cup D, V, f)$ 中,$P \subseteq C$,D 的 P 正区域记为 $\mathrm{POS}_P(D)$,定义为

$$\mathrm{POS}_P(D) = \bigcup_{X \in U/D} P\text{-}(X).$$

D 的 P 正区域是 U 中所有根据 $\mathrm{ind}(P)$ 的信息可以划分到 D 的等价关系中的对象集合.

定义 5[11]　决策表信息系统 $S = (U, C \cup D, V, f)$ 中,$a \in C$,如果 $\mathrm{POS}_C(D) = \mathrm{POS}_{C-\{a\}}(D)$,则称 a 为 C 中相对 D 不必要的;否则,称 a 为 C 中相对 D 必要的. C 的所有必要属性的集合称为 C 相对 D 的核,记为 $\mathrm{Core}(C)$.

定义 6[11]　一个决策表信息系统 $S = (U, C \cup D, V, f)$,若存在 $x_i, x_j \in U$,当 $i \neq j$ 时,有 $f(x_i, C) = f(x_j, C)$ 且 $f(x_i, D) \neq f(x_j, D)$,则称该系统为不相容决策表信息系统,x_i 和 x_j 称为不相容对象(冲突对象);否则,称为相容决策表信息系统.

3　求核性质

对于相容的决策表,Hu 的方法是正确的;但对于不相容决策表,Hu 的方法却不能保证获得正确的核. 针对决策表中存在不相容性,提出一种新的求核方法. 该方法不需要创建可分辨矩阵.

定义 7　决策表信息系统 $S = (U, C \cup D, V, f)$ 中,$a \in C$,核属性集 $\mathrm{GCore}(C)$ 表示为

$$\mathrm{GCore}(C) = \begin{cases} \{a \mid a \in C\}, & |\mathrm{ConSet}(C - \{a\})| > |\mathrm{ConSet}(C)|; \\ \Phi, & \text{otherwise.} \end{cases}$$

其中 $\mathrm{ConSet}(C) = \{x_i \mid x_i \in U, \exists x_j, f(x_i, C) = f(x_j, C) \wedge f(x_i, D) \neq f(x_j, D)\}$. 也就是说,$\mathrm{ConSet}(C)$ 是冲突对象的集合,$|\mathrm{ConSet}(C)|$ 表示 $\mathrm{ConSet}(C)$ 中冲突对象的数目.

定义 7 表明,若删除某个属性 a_i 后,如果冲突对象的个数增加了,则说明 a_i 为核属性;否则,a_i 不是核属性.

定理 1　对于决策表信息系统 $S = (U, C \cup D, V, f)$,有 $\mathrm{Core}(C) = \mathrm{GCore}(C)$.

证明　设 $U/C = \{X_1, X_2, \cdots, X_n\}$, $U/D = \{Y_1, Y_2, \cdots, Y_m\}$.

1) 首先证明 $\mathrm{GCore}(C) \subseteq \mathrm{Core}(C)$. 对于 $\forall a \in \mathrm{GCore}(C)$,由定义 7 可知,删除属性 a 后,ConSet 中冲突对象个数增加了. 设 $x \notin \mathrm{ConSet}(C)$, $x \in \mathrm{ConSet}(C - \{a\})$,则存在 $y \notin [x]_C$ 且 $y \in [x]_{C-\{a\}}$,有 $f(x, D) \neq f(y, D)$. 设 $x \in Y_q$, $y \in Y_p$, $y \notin Y_q$,因为 $y \in [x]_{C-\{a\}}$ 且 $y \notin Y_q$,由下近似定义知 $x \notin (C - \{a\})\text{-}Y_q$,故 $x \notin \mathrm{POS}_{C-\{a\}}(D)$. 又 x 在原系统 S 中无冲突对象且 $x \in Y_q$,因为 $x \in C\text{-}Y_q$,故 $x \in \mathrm{POS}_C(D)$,于是 $\mathrm{POS}_C(D) \neq \mathrm{POS}_{C-\{a\}}(D)$. 所以 $a \in \mathrm{Core}(C)$. 因此,$\mathrm{GCore}(C) \subseteq \mathrm{Core}(C)$.

2) 证明 $\mathrm{Core}(C) \subseteq \mathrm{GCore}(C)$. 对于 $\forall a \in \mathrm{Core}(C)$,有 $\mathrm{POS}_C(D) \neq \mathrm{POS}_{C-\{a\}}(D)$,则存在 x 使得 $[x]_C \subseteq Y_k$ 且 $[x]_{C-\{a\}} \not\subseteq Y_k$. 于是 $\exists y, y \in [x]_{C-\{a\}}$ 且 $y \notin [x]_C$. 有 $f(x, C - \{a\}) = f(y, C - \{a\})$ 且 $f(x, D) \neq f(y, D)$. 因此,在删除属性 a 后会产生冲突. 下面证明 x 在原系统 S 中为非冲突对象. 采用反证法. 假设存在 z, $z, x \in [x]_C \subseteq Y_k$ 且 $f(x, D) \neq f(z, D)$,则 x, z 属于 U/D 的不同等价类,这与 $z, x \in [x]_C \subseteq Y_k$ 相矛盾,所以 x 在原系统 S 中没有与之相冲突的对象. 因此,删除 a 后新增加了冲突对象,即 $|\mathrm{ConSet}(C - \{a\})| > |\mathrm{ConSet}(C)|$, $a \in$

GCore(C). 故 Core(C) ⊆ GCore(C).

由 1) 和 2) 得证 Core(C) = GCore(C). □

4 快速求核算法

4.1 快速等价类求解算法

由定理 1 可知,求核过程中等价类划分是一个关键步骤. 求等价类的一般方法是对样本集 U 中未分类的对象进行两两比较,比较它们对于条件属性集 C,每个属性取值是否相同. 如果相同,则属于同一个等价类;否则,不属于同一等价类. 上述等价类划分算法的时间复杂度为 $O(|C||U|^2)$.

性质 1 一个决策表信息系统 $S = (U, A, V, f)$,两个样本 $x_i, x_j \in U$ 相对于属性集 C 同属于一个等价类当且仅当 $\forall a \in A$,有 $f(x_i, a) = f(x_j, a)$.

由定义 2,可以证明性质 1 成立.

根据性质 1,可先对决策系统 S 按属性集 C 排序;然后分析排序后的决策系统 S,划分等价类. 赵军等[6] 使用了快速排序,使等价类划分算法的时间复杂度降低为 $O(|C||U|\log|U|)$;徐章艳等[9] 利用链式基数排序算法,将时间复杂度降低到 $O(|C||U|)$. 本文提出一种分布计数的基数排序方法,按属性集 C 对决策表 S 排序. 该算法的时间复杂度也为 $O(|C||U|)$,空间复杂度为 $O(|U|)$,且相对于徐章艳的方法更易于理解和实现.

对决策表采用分布计数的基数排序的思想是:设 $S = (U, C \cup D)$,其中 $C = \{a_i \mid i = 1, \cdots, m\}$, $D = \{d\}$,决策表一行为一个数据对象,则 $S = \{S_i \mid i = 1, \cdots, n\}$ 为数据对象的集合,S_i 为一个 $m+2$ 的元组,$S_i = (x_i, a_1, a_2, \cdots, a_m, d)$. 其中:$S_i x_i$ 为对象的编号,$S_i a_j$ 表示对象 i 的 a_j 属性值,$S_i d$ 表示对象 i 的决策属性值.

按照属性集 C 对 S 排序,即依次以每个属性 a_i 对 S 排序. 首先,将需要离散化的属性 a_i 离散化,将其分布在整型区间 $[1 \cdots e]$,其中 $0 < e \leqslant |U|$;然后,构造一个计数表 countPos$[0 \cdots e]$,countPos 中元素个数为 ind(a_i) 等价类的个数,每个元素用于存放 ind(a_i) 中每个等价类当前最后一个元素在有序决策表中的位置,根据 countPos 表可直接将每个对象 S_i 放到有序决策表最终的位置. 在这个过程中,需要使用 2 个辅助空间,一个是 countPos,一个是存放有序决策表的 sortedS.

算法 1 等价类划分算法

输入:决策系统 $S = (U, C \cup D, V, f)$, $U = \{x_i \mid i = 1, \cdots, n\}$, $C = \{a_i \mid i = 1, \cdots, m\}$;

输出:U/C.

Step1: for i = 1 to $|C|$ Do // 分布计数的基数排序

Step1.1: { 初始化 countPos 表: countPos$[0 \cdots e] = 0$.

Step1.2: 统计 ind(a_i) 中每个等价类中对象的个数,由此计算每个等价类最后一个对象在有序决策表的位置,存放到 countPos 中.

Step1.3: for $j = |U|$ to 1 Do

Step1.3.1: { 根据 $S_j a_i$ 的值,在 countPos 表中找到 S_j 在有序决策表中的位置 pos;

Step1.3.2: 将 S_j 存入有序决策表 sortedS 的第 pos 位置;

Step1.3.3: 修改等价类 $[S_j a_i]_{\text{ind}(a_i)}$ 当前最后一个元素在有序表中的位置(即位置前移一位); }

} //end_for_i

Step2: $s = 1$, $E_1 = \{x_1\}$; // s 用于标识等价类的个数

Step3: for i = 2 to $|U|$ Do

{if(x_i 与 x_{i-1} 对于 C 的每个属性值均相等)

then $E_s = E_s \cup \{x_i\}$;

else $s = s + 1$, $E_s = \{x_i\}$;

}

Step4: 输出 E(即 U/C) 和 s.

算法 1 中,Step1 内部的执行时间为 $O(e) + O(|U|) + O(|U|) = O(|U|)$. 由于循环次数为 $O(|C|)$, Step1 的总的时间复杂度为 $O(|C||U|)$,Step3 的时间复杂度为 $O(|U|)$. 在 Step1 中,辅助空间 countPos 容量最大为 $|U|+1$,Step3 中辅助空间 sortedS 容量可以控制为 $2|U|$. 因此,算法 2 的空间复杂度为 $O(|U|)$. 从而可得出算法 1 总的时间复杂度为 $O(|C||U|) + O(|U|) = O(|C||U|)$,空间复杂度为 $O(|U|)$.

4.2 高效核属性求解算法

根据定义 7 和定理 1,给出如下求核算法:

算法 2 高效求核算法

输入:决策表信息系统 $S = (U, A, V, f)$. 其中:U 为论域;A 为属性集,$A = C \cup D$ 且 $C \cap D = \Phi$, C 为条件属性集;D 为决策属性集.

输出:决策表的核 Core(C).

Step1: Core$(C) = \Phi$, ConSet$(C) = \Phi$.

Step2: 依据算法 1,按属性 C 排序决策表 S,得 E 和 s.

Step3: 初始化 countPos,并计算每个等价类中最后一个元素的位置,存于 countPos 中.

Step4: for j = 1 to s Do

if((E_j 中存在 x_t 和 x_k 使得 $x_t \cdot d \neq$

x_k ·d)

then ConSet(C) = ConSet(C) ∪ E_j.

Step5: for i = 1 to | C| Do

Step5.1: {依据算法 1,按属性 C - { a_i} 排序决策表 S,得 E 和 s;

Step5.2: 初始化 countPos,并计算每个等价类中最后一个元素的位置,存于 countPos 中;

Step5.3: ConSet(C - { a_i}) = Φ;

Step5.4: for j = 1 to s Do

if(E_j 中存在 x_t 和 x_k 使得 x_t ·d ≠ x_k ·d)

then ConSet(C - { a_i}) = ConSet(C - { a_i}) ∪ E_j;

Step5.5: if(| ConSet(C - { a_i}) | > | ConSet(C) |)

then Core(C) = Core(C) ∪{ a_i};

} //end_for_i.

Step6: 输出 Core(C).

Step7: 结束.

算法 2 中,Step2 的时间复杂度为 $O(|C||U|)$,Step3 和 Step4 的时间复杂度为 $O(|U|)$. Step5.1 的时间复杂度为 $O(|C||U|)$,Step5.2 和 Step5.4 的时间复杂度为 $O(|U|)$,Step5 循环体的总的时间复杂度为 $O(|C||U|)+2O(|U|)=O(|C||U|)$,而 Step5 循环次数为 | C| 次,则 Step5 的时间复杂度为 $O(|C|^2|U|)$. 因此,算法 2 的时间复杂度为 $O(|C||U|)+2O(|U|)+O(|C|^2|U|)=O(|C|^2|U|)$,空间复杂度为 $O(|U|)$.

4.3　与其他求核算法的比较

Hu,叶东毅和闫德勤求核算法的时间和空间复杂度均为 $O(|C||U|^2)$,显然本文求核算法的时间和空间复杂度均明显优于他们.

赵军等将快速排序应用到等价类求解的过程中,使求核算法的时间和空间复杂度降低到 $O(|C|^2|U|\log|U|)$ 和 $O(|U|)$,但时间复杂度还是高于本文求核算法.

杨明算法的时间和空间复杂度分别降为 $\max\{O(|C||U|\log|U|), O(|C||U||\mathrm{POS}_C(D)|)\}$ 和 $O(|C||U||\mathrm{POS}_C(D)|)$;徐章艳求核算法的时间和空间复杂度分别降到 $\max\{O(|C|\,|U/C|^2), O(|C||U|)\}$ 和 $\max\{O(|U|), O(|C||U/C|^2)\}$,对于一个大型数据集而言,| U|,| U/C| 和 | $\mathrm{POS}_C(D)$| 是远大于 | C| 的,例如 UCI 数据库中的 Car evaluation 数据集共有 1728 个样本对象,条件属性集 C 中属性的个数为 6,而 | U/C| = 972, | $\mathrm{POS}_C(D)$| = 1196,可见 | U|, | U/C| 和 | $\mathrm{POS}_C(D)$| 远大于 | C|. 因此,本文算法的时间和空间复杂度低于杨明和徐章艳的算法.

5　实例和实验分析

5.1　实例分析

为了说明本文算法,下面采用文献[4]中的 2 个实例进行说明.

例 1　表 1 为一个决策表,有 5 个样本对象,条件属性 $C=\{a,b,c\}$,D 为决策属性.

表 1　决策表信息系统 S_1

U	a	b	c	D
x_1	1	0	1	1
x_2	1	0	1	0
x_3	0	0	1	1
x_4	0	0	1	0
x_5	1	1	1	1

根据算法 2 分析表 1 可知,x_1 与 x_2,x_3 与 x_4 分别为冲突对象. 因此,首先求得 ConSet(C) = { x_1, x_2, x_3, x_4}; 然后对每个条件属性得 ConSet(C - { a}) = { x_1, x_2, x_3, x_4}, ConSet(C - { b}) = { x_1, x_2, x_3, x_4, x_5}, ConSet(C - { c}) = { x_1, x_2, x_3, x_4}. 可见,只有 | ConSet(C - { b}) | > | ConSet(C) |,从而 Core(C) = { b}.

例 2　表 2 中有 4 个样本对象,条件属性 $C=\{a,b,c\}$,D 为决策属性.

表 2　决策表信息系统 S_2

U	a	b	c	D
x_1	1	0	1	0
x_2	0	0	1	1
x_3	0	0	1	0
x_4	1	1	1	1

根据算法 2,首先求得 ConSet(C) = { x_2, x_3}; 然后对每个条件属性得 ConSet(C - { a}) = { x_1, x_2, x_3}, ConSet(C - { b}) = { x_1, x_2, x_3, x_4}, ConSet(C - { c}) = { x_2, x_3}. 可见,有 | ConSet(C - { a}) | > | ConSet(C) | 和 | ConSet(C - { b}) | > | ConSet(C) |,从而 Core(C) = { a, b}.

5.2　实验比较

采用文献[6]中的气象实例和 UCI 数据库中 6 个决策表为测试数据,在 Petium 4 2.8 GHz,RAM 512 M 微机上,分别采用叶东毅的算法、赵军的 EABKF 算法、徐章艳的算法以及本文给出的求核算法进行比较实验,结果如表 3 所示. 用 ALG1 表示叶东毅的算法,ALG2 表示赵军的 EABKF 算法,ALG3 表示杨明算法,ALG4 表示徐章艳算法,

表 3 5 种求核算法时间复杂度比较

决策表	条件属性个数	样本对象个数	U/C 中等价类的个数	$POS_C(D)$ 中元素的个数	算法执行时间/ ms				
					AL G1	AL G2	AL G3	AL G4	AL G5
文献[6]中实例	4	14	14	14	0.022	0.018	0.018	0.019	0.016
Patient data	8	90	66	66	0.895	0.831	0.833	0.834	0.471
Flare data	12	323	174	289	11.951	8.437	9.912	9.657	5.301
Balance data	4	625	248	625	32.671	8.516	26.790	21.046	0.607
Monkey data	17	556	432	556	40.316	24.513	31.819	19.546	8.962
Car evaluation data	6	1728	972	1196	292.195	9.650	19.381	16.054	2.682
Led17 data	26	2000	1998	2000	12595.141	297.028	943.281	873.651	57.173

AL G5 表示本文提出的求核算法(算法 2).

从表 3 可以看出, AL G5 的执行时间低于 AL G1～AL G4,并随着数据集中样本对象数目的增加,AL G5 的效率更加明显,说明本文的求核算法更加适合大数据集的处理.

6 结 论

由于决策表中存在不相容性, Hu 算法求得的核与正区域的核不一致. 本文依据删除某个条件属性后冲突对象的数目是否增加为标准,提出一种新的求核方法,并证明了由该方法所获得的核与正区域的核是一致的. 等价类划分是求核算法的重要步骤,为了提高算法的效率,提出采用分布计数的基数排序方法求解等价类 U/C,使其时间复杂度和空间复杂度分别降至 $O(|C||U|)$ 和 $O(|U|)$. 在此基础上设计求核算法,算法的时间复杂度和空间复杂度分别为 $O(|C|^2|U|)$ 和 $O(|U|)$. 实验表明,本文算法是正确而高效的.

参考文献(References)

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

[2] Hu X H, Cercone N. Learning in relational databases: A rough set approach[J]. Computational Intelligence, 1995, 11(2): 323-337.

[3] Skowron A, Rauszer C. The discernibility matrices and functions in information systems [C]. Intelligent Decision Support-handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publisher, 1991: 331-362.

[4] 叶东毅, 陈昭炯. 一个新的差别矩阵及其求核方法[J]. 电子学报, 2002, 30(7): 1086-1088.
(Ye D Y, Chen Z J. A new discernibility matrix and the computation of a core [J]. Acta Electronica Sinica, 2002, 30(7): 1086-1088.)

[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.
(Wang G Y. Calculation methods for core attributes of decision table [J]. Chinese J of Computers, 2003, 26(5): 611-615.)

[6] 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 J, Wang G Y, Wu Z F, et al. An efficient approach to computer feature core [J]. Mini-Micro Systems, 2003, 24(11): 1590-1593.)

[7] 闫德勤, 刘菲斐. 属性约简中的差别矩阵与近似精度[J]. 小型微型计算机系统, 2005, 26(11): 1975-1977.
(Yan D Q, Liu F F. Discernibility matrix and approximate quality in attribute reduction [J]. Mini-Micro Systems, 2005, 26(11): 1975-1977.)

[8] 杨明, 孙志挥. 改进的差别矩阵及其求核方法[J]. 复旦大学学报, 2004, 43(5): 865-868.
(Yang M, Sun Z H. Improvement of discernibility matrix and the computation of a core [J]. J of Fudan University, 2004, 43(5): 865-868.)

[9] 徐章艳, 杨炳儒, 宋威, 等. 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 Z Y, Yang B R, Song W, et al. Quick computing core algorithm based on discernibility matrix [J]. Computer Engineering and Applications, 2006, 42(6): 4-6.)

[10] 杨明, 杨萍. 基于差别矩阵的属性核快速更新算法[J]. 控制与决策, 2007, 21(8): 857-862.
(Yang M, Yang P. Fast updating algorithm of computation of a core based on discernibility matrix [J]. Control and Decision, 2007, 21(8): 857-862.)

[11] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001.
(Zhang W X, Wu W Z, Liang J Y, et al. Theory and method of rough set [M]. Beijing: Science Press, 2001.)