决策系统的快速属性约简算法

李金海，吕跃进

(广西大学数学与信息科学学院 南宁 530004)

【摘要】针对决策系统提出了一种高效的属性约简算法；讨论了合理刻画属性重要性的新指标，并设计了一种快速计算划分的方法；在此基础上，得到了一种快速计算属性约简的算法。与现有算法相比，该算法具有较大的灵活性，能从搜索空间中逐次删除不重要属性，避免了对其重要性的重复计算；并且时间复杂度低。通过实例和实验表明了该算法的可行性与有效性。

关 键 词 属性约简； 高效算法； 正区域； 粗糙集
中图分类号 TP18 文献标识码 A

Quick Attribute Reduction Algorithm on Decision System

LI Jin-hai，LÜ Yue-jin

(School of Mathematics and Information Science, Guangxi University Nanning 530004)

Abstract This paper puts forward an efficient algorithm for reduction of attribute in decision systems. A relatively reasonable formula measuring attribute significance is discussed and a quick method to compute partition is proposed. Then a quick algorithm for reduction of attribute is obtained. Compared with those existed algorithms, its flexibility has been increased because calculating the important value of unimportant attributes repeatedly can be avoided by removing unimportant attributes gradually from the search space. The theoretical analysis shows that this algorithm is much less time complexity than those existed algorithms. A real example and experimental results demonstrate its feasibility and effectiveness, respectively.

Key words attribute reduction; efficient algorithm; positive region; rough set

文献[1]提出的粗糙集理论是一种新的处理模糊和不确定知识的数学工具。经过二十多年的研究和发展，粗糙集理论已被成功地应用于人工智能、数据挖掘、模式识别与智能信息处理等领域[2-5]。属性约简是粗糙集理论中的核心内容之一，是在保持信息系统分类能力不变的条件下，删除其中的冗余属性。粗糙集的有效算法方面的研究主要集中在属性约简方面[6-12]。

目前，很多学者已从不同的角度提出了一些属性约简的方法[6-11]，但这些约简方法的时间复杂度至少为 $O(|A|^2|U|^2)$。当数据量很大时，这些方法的有效性面临巨大的挑战。文献[12]提出了一种快速计算属性约简的方法，将信息系统属性约简的时间复杂度降低到 $O(|C|^2|U|\log_2|U|)$，但其时间复杂度不太理想。由于现有属性约简算法的时间复杂度较高的主要原因是没有找到快速计算划分，本文对此进行研究，提出了一种快速计算划分的方法，从而设计出高效的属性约简算法。

1 粗糙集的基本概念

粗糙集的基本概念[1-3]如下：

定义 1 信息系统 $S=(U,A,V,f)$，其中，U 为对象集；A 为属性集；$V=\bigcup_{a\in A}V_a$，V_a 为属性 a 的值域；$f:U\times A\to V$ 为一个信息函数，它指定 U 中每一个对象的属性值，即对任意 $a\in A, x\in U$，有 $f(x,a)\in V_a$。

如果属性集 A 可以分为条件属性集 C 和决策属性集 D，即 $C\cup D=A$，$C\cap D=\varnothing$，则该信息系统称为决策系统，D 一般只有一个属性。

定义 2 在信息系统 S 中，对于 A 中的每个属性子集 P 都可决定一个二元不可区分关系为 $\mathrm{IND}(P)=\{(x,y)\in U\times U\mid\forall a\in A, f(x,a)=f(y,a)\}$，

收稿时间：2007－07－15
基金项目：广西教育厅科研项目(桂教科研[2006]26号)
作者简介：李金海(1984－)，男，硕士生，主要从事数据挖掘、粗糙集理论等方面的研究；吕跃进(1958－)，男，教授，主要从事控制与决策、数据挖掘、粗糙理论等方面的研究.

其中，IND(P) 为U上的一个等价关系。对象 $x\in U$ 在属性集 P 上的等价类 $[x]_{\mathrm{IND}(P)}$ 定义为：

$$[x]_{\mathrm{IND}(P)}=\{y\mid y\in U, y\mathrm{IND}(P)x\}$$

将 $[x]_{\mathrm{IND}(P)}$ 简记为 $[x]_P$。所有等价类 $[x]_P$ 组成的集合 $\{[x]_P\mid x\in U\}$ 构成了 U 的一个划分，用 U/P 表示。

性质 1　在信息系统 S 中，若 $P,Q\subseteq A$，$x\in U$，则 $x\in \mathrm{POS}_P(Q)$，当且仅当 $[x]_P\subseteq[x]_Q$。

性质 2　在信息系统 S 中，若 $P\subseteq C$，则对于任意 $x_i\in U$，有 $[x_i]_P=\bigcap_{a\in P}[x_i]_{\{a\}}$。

定义 3　在信息系统 S 中，若 $P,Q\subseteq A$，则 Q 的 P 正域 $\mathrm{POS}_P(Q)$ 定义为：

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

式中　$\underline{P}X$ 为 X 的 P 下近似。

2　属性重要性的度量

在决策系统 $T=(U,C\cup D,V,f)$ 中，任意 $P\subseteq C$，记 $D(P)=\sum_{x\in \mathrm{POS}_C(D)}\left|[x]_P-[x]_D\right|$。其中，$[x]_P-[x]_D$ 为所有满足 $y\in[x]_P$、且 $y\notin[x]_D$ 的 y 的集合；$\left|[x]_P-[x]_D\right|$ 为 $[x]_P-[x]_D$ 的基数。

定理 1　在决策系统 T 中，对任意 $P\subseteq C$，$\mathrm{POS}_P(D)=\mathrm{POS}_C(D)$，当且仅当 $D(P)=0$。

证明　(1) 必要性。因为 $\mathrm{POS}_P(D)=\mathrm{POS}_C(D)$，故对任意 $x\in\mathrm{POS}_C(D)$，有 $x\in\mathrm{POS}_P(D)$。由性质1可得 $[x]_P\subseteq[x]_D$，因此 $[x]_P-[x]_D=\varnothing$，$\left|[x]_P-[x]_D\right|=0$，所以 $D(P)=0$。(2) 充分性。由于 $D(P)=0$，故任意 $x\in\mathrm{POS}_C(D)$，有 $\left|[x]_P-[x]_D\right|=0$，即 $[x]_P-[x]_D=\varnothing$，$[x]_P\subseteq[x]_D$；根据性质1可得 $x\in\mathrm{POS}_P(D)$。另一方面，对任意 $x\in\mathrm{POS}_P(D)$，有 $[x]_P\subseteq[x]_D$；而 $P\subseteq C$，因此 $[x]_C\subseteq[x]_P$，所以 $[x]_C\subseteq[x]_D$，也即 $x\in\mathrm{POS}_C(D)$，由此可得 $\mathrm{POS}_P(D)=\mathrm{POS}_C(D)$。　证毕

定理 2　在决策系统 T 中，若 $P\subseteq Q\subset C$，$a\in C-Q$，且 $D(P\cup\{a\})=D(P)$，则 $D(Q\cup\{a\})=D(Q)$。

证明　因为 $P\subset P\cup\{a\}$，故对任意 $x\in\mathrm{POS}_C(D)$，则有：

$$[x]_{P\cup\{a\}}-[x]_D\subseteq[x]_P-[x]_D \tag{1}$$

因此有：

$$|[x]_{P\cup\{a\}}-[x]_D|\leq|[x]_P-[x]_D| \tag{2}$$

已知 $D(P\cup\{a\})=D(P)$，则有：

$$\sum_{x\in\mathrm{POS}_C(D)}\left|[x]_P-[x]_D\right|=\sum_{x\in\mathrm{POS}_C(D)}\left|[x]_{P\cup\{a\}}-[x]_D\right| \tag{3}$$

由式(2)～(3)可得，对任意 $x\in\mathrm{POS}_C(D)$ 有：

$$\left|[x]_{P\cup\{a\}}-[x]_D\right|=\left|[x]_P-[x]_D\right| \tag{4}$$

由式(1)、(4)可得，对任意 $x\in\mathrm{POS}_C(D)$ 有：

$$[x]_{P\cup\{a\}}-[x]_D=[x]_P-[x]_D \tag{5}$$

由式(5)可得：

$$[x]_{P\cup\{a\}}\cap([x]_D)^{\mathrm{T}}=[x]_P\cap([x]_D)^{\mathrm{T}} \tag{6}$$

由式(6)可得，对任意 $x\in\mathrm{POS}_C(D)$ 有：

$$\begin{aligned}|[x]_{Q\cup\{a\}}-[x]_D|&=|[x]_{Q\cup\{a\}}\cap([x]_D)^{\mathrm{T}}|=\\&|[x]_{P\cup\{a\}\cup(Q-P)}\cap([x]_D)^{\mathrm{T}}|=\\&|[x]_{P\cup\{a\}}\cap([x]_D)^{\mathrm{T}}\cap[x]_{Q-P}|=\\&|[x]_P\cap([x]_D)^{\mathrm{T}}\cap[x]_{Q-P}|=\\&|[x]_Q\cap([x]_D)^{\mathrm{T}}|=|[x]_Q-[x]_D|\end{aligned}$$

所以，$D(Q\cup\{a\})=D(Q)$。　证毕

定义 4　在决策系统 T 中，$a\in C$ 在 C 中相对于 D 的重要性为：

$$\mathrm{SIG}(a,C,D)=D(C-\{a\})-D(C)$$

定理 3　$a\in C$ 在 C 相对于 D 是必要的，当且仅当 $\mathrm{SIG}(a,C,D)>0$。

推论 1　$\mathrm{core}_C(D)=\{a\in C\mid \mathrm{SIG}(a,C,D)>0\}$

定义 5　在决策系统 T 中，$R\subset C$，则对于任意属性 $a\in C-R$ 在 R 中相对于 D 的重要性为：

$$\mathrm{SIG}(a,R\cup\{a\},D)=D(R)-D(R\cup\{a\})$$

当 $\mathrm{SIG}(a,R\cup\{a\},D)=0$ 时，属性 $a\in C-R$ 在 R 中相对于 D 是不重要的；否则属性是重要的。

定义5表明，$\mathrm{SIG}(a,R\cup\{a\},D)$ 的值越大，说明属性 $a\in C-R$ 在 R 中相对于 D 就越重要。本文把 $\mathrm{SIG}(a,R\cup\{a\},D)$ 作为寻找最小属性约简的启发式信息，以减少搜索空间。

由定理2可得以下推论：

推论 2　在决策系统 T 中，$P,Q\subset C$，$P\subseteq Q$，$a\in C-Q$。若属性 a 在 P 中相对于 D 是不重要的，则属性 a 在 Q 中相对于 D 也是不重要的。

定理 4　在决策系统 T 中，$P\subseteq C$。如果 $D(P)=0$，且对任意 $a\in P$ 有 $\mathrm{SIG}(a,P,D)>0$，则 P 为 C 相对于 D 的一个约简。

3　快速属性约简算法

本文设计以下算法求解属性约简。

算法 1　属性约简算法。输入：决策系统 $T=(U,C\cup D,V,f)$；输出：C 相对于 D 的一个相对约简。算法过程如下：(1) 计算 $\mathrm{POS}_C(D)$。(2) 令 $\mathrm{core}_C(D)\leftarrow\varnothing$。(3) 对于 C 中的每个属性 a 计算 $\mathrm{SIG}(a,C,D)$。如果 $\mathrm{SIG}(a,C,D)>0$，则 $\mathrm{core}_C(D)\leftarrow\mathrm{core}_C(D)\cup\{a\}$。(4) 如果 $D(\mathrm{core}_C(D))=0$，则转步

骤9；否则转步骤5。(5) 令 $E \leftarrow \text{core}_C(D)$。(6) 对于 $C-E$ 中每个属性 a 计算 $\text{SIG}(a, E\cup\{a\}, D)$。如果 $\text{SIG}(a, E\cup\{a\}, D)=0$，则 $C \leftarrow C-\{a\}$。(7) 在 $C-E$ 中，选择满足 $\text{SIG}(a, E\cup\{a\}, D)=\max\limits_{b\in C-E}\{\text{SIG}(b, E\cup\{b\}, D)\}$ 的属性 a，执行 $E \leftarrow E\cup\{a\}$。(8) 如果 $D(E)=0$，则转步骤9；否则返回步骤6。(9) 结束。

在算法1的步骤6中，从搜索空间 $C-E$ 中逐次删除不重要属性，避免了对其重要性的重复计算。由推论2可知，这种方式既不影响求解属性约简，又可以提高搜索效率。

根据性质1，下面给出一种快速计算划分 U/P 的方法。

算法 2　计算划分 U/P。

子算法：计算 $\text{IND}(\{a_i\})$ 的各等价类 $[x_1]_{\{a_i\}}, [x_2]_{\{a_i\}}, \cdots, [x_n]_{\{a_i\}}$。输入：对象集 U 和属性 a_i 的属性值组成的集合 V_{a_i}；输出：$\text{IND}(\{a_i\})$ 的各等价类 $[x_1]_{\{a_i\}}, [x_2]_{\{a_i\}}, \cdots, [x_n]_{\{a_i\}}$。算法过程如下：(1) 求 V_{a_i} 中的最大属性值 M_i 与最小属性值 m_i。(2) For $(j=0, j\le M_i-m_i, j++)$，$S_{ij}=\varnothing$。(3) For $(j=1, j\le n, j++)$，$t=f(x_j, a_i)-m_i$，$S_{it}=S_{it}\cup\{x_j\}$。(4) For $(j=1, j\le n, j++)$，$T_{ij}=\varnothing$。(5) For $(j=1, j\le n, j++)$，$T_{ij}=S_{i, f(x_j, a_i)-m_i}$。

主算法：计算划分 U/P。输入：决策系统 T，其中，$U=\{x_1, x_2, \cdots, x_n\}$，$P\subseteq A$，$P=\{a_1, a_2, \cdots, a_k\}$；输出：划分 U/P。算法过程如下：(1) 调用子算法求 $T_{1i}(i=1,2,\cdots,n)$。(2) 令 $W_{Pi}=T_{1i}(i=1,2,\cdots,n)$。(3) For $(j=2,\ j\le k, j++)$。调用子算法求 $T_{ji}(i=1,2,\cdots,n)$，For $(i=1, i\le n, i++)$，$W_{Pi}=W_{Pi}\cap T_{ji}$，End。(4) For $(i=1, i\le n, i++)$，如果 $W_{Pi}=\varnothing$，删除 W_{Pi}。

算法2的时间复杂度为 $O(|P||U|)$。根据算法2并结合性质1，得到以下命题：

命题 1　在决策系统 T 中，计算 $\text{POS}_C(D)$ 的时间复杂度为 $O((|C|+|D|)|U|)$。

在算法1中，步骤1求 $\text{POS}_C(D)$，由命题1可得，执行步骤1的时间复杂度为 $O((|C|+|D|)|U|)$；步骤3需要计算 $|C|$ 次 $\text{SIG}(a, C, D)$。由算法2可知，计算每个 $\text{SIG}(a, C, D)$ 的时间复杂度为 $O((|C|+|D|)|U|)$，故执行步骤3的时间复杂度为 $O(|C|(|C|+|D|)|U|)$；步骤6、7在最坏的情况下，需要计算 $|C|(|C|+1)/2$ 次 $\text{SIG}(a, E\cup\{a\}, D)$。从 $E=\text{core}_C(D)$ 开始，在操作的过程中，用中间变量替换并保存已经得到的划分 U/E，故计算 $\text{SIG}(a, E\cup\{a\}, D)$ 的时间复杂度为 $O(|U|)$。因此执行步骤6、7的时间复杂度为 $O(|C|^2|U|)$，所以算法1的总时间复杂度为 $O(|C|(|C|+|D|)|U|)$。决策属性 D 一般只有一个，所以算法1的时间复杂度为 $O(|C|^2|U|)$。

4　实例与实验

4.1　实例分析

本文利用算法1，求出决策系统的相对约简，如表1所示。

表1　决策系统的相对约简

U	a_1	a_2	a_3	a_4	A_5	a_6	a_7	a_8	a_9	D
x_1	0	0	0	0	1	0	1	1	0	1
x_2	1	1	1	0	1	0	1	0	1	1
x_3	1	1	2	0	1	0	1	2	1	2
x_4	1	0	2	0	1	1	1	1	1	1
x_5	1	0	2	0	1	1	0	0	1	3
x_6	1	0	2	0	1	2	0	3	0	3
x_7	1	0	2	0	1	2	0	2	1	2
x_8	1	0	2	0	1	2	0	2	1	4
x_9	1	2	1	1	0	3	2	3	1	4
x_{10}	1	2	1	2	0	3	2	3	1	5

相对约简过程如下：

(1) 对 C 中的每个属性 a 计算 $\text{SIG}(a, C, D)$，可以得到：

$$\text{SIG}(a_i, C, D)=0 \quad i=1,2,\cdots,9$$

$$\text{SIG}(a_4, C, D)=2>0 \quad \text{core}_C(D)=\{a_4\}$$

(2) $D(\text{core}_C(D))=33\neq 0$，转步骤3。

(3) $B=\text{core}_C(D)=\{a_4\}$。

(4) 对于 $C-B$ 中每个属性 a，计算 $\text{SIG}(a, B\cup\{a\}, D)$，即：$\text{SIG}(a_1, B\cup\{a_1\}, D)=8$，$\text{SIG}(a_2, B\cup\{a_2\}, D)=15$，$\text{SIG}(a_3, B\cup\{a_3\}, D)=16$，$\text{SIG}(a_5, B\cup\{a_5\}, D)=0$，$\text{SIG}(a_6, B\cup\{a_6\}, D)=26$，$\text{SIG}(a_7, B\cup\{a_7\}, D)=23$，$\text{SIG}(a_8, B\cup\{a_8\}, D)=30$，$\text{SIG}(a_9, B\cup\{a_9\}, D)=14$。

(5) 由于 $\text{SIG}(a_5, B\cup\{a_5\}, D)=0$，所以有 $C \leftarrow C-\{a_5\}$。

(6) 因为 $a_8\in C-B$，满足 $\text{SIG}(a_8, B\cup\{a_8\}, D)=\max\limits_{b\in C-B}\{\text{SIG}(b, B\cup\{b\}, D)\}$，执行 $B \leftarrow B\cup\{a_8\}$

(7) $D(B)=3>0$，返回步骤4。

(8) $\text{SIG}(a_1, B\cup\{a_1\}, D)=0$ $\text{SIG}(a_2, B\cup\{a_2\}, D)=3$，$\text{SIG}(a_3, B\cup\{a_3\}, D)=2$ $\text{SIG}(a_6, B\cup\{a_6\}, D)=3$，$\text{SIG}(a_7, B\cup\{a_7\}, D)=3$ $\text{SIG}(a_9, B\cup\{a_9\}, D)=0$。

(9) $B \leftarrow B\cup\{a_6\}$。

(10) $D(B)=0$，结束，输出 $B=\{a_4,a_6,a_8\}$。

以上求得的 $B=\{a_4,a_6,a_8\}$ 恰好是 C 相对于 D 的一个最小约简。

4.2 实验

本文选择表1中的决策系统(记为决策系统1)与University of California中的两个决策系统BUPA Liver Disorders(记为决策系统2)及Chess End-Game(记为决策系统3)，对文献[11]提出的属性约简算法(记为算法A)与本文中的算法1的有效性进行比较。本文使用VC6.0+SQL Server 2000在Windows Advanced Server 2000的服务器(Pentium(R) Ⅳ，CPU:1.60 GHz，Memory: 512 MB)上进行算法比较实验，实验结果如表2所示。

表2 算法比较

决策系统	对象个数	条件属性数	算法A		算法1	
			是否约简	运行时间/s	是否约简	运行时间/s
1	10	9	是	0.006	是	0.002
2	345	6	是	8.620	是	0.680
3	3 196	36	是	3 864.600	是	186.700

从表2可以看出，当数据库比较大时，本文提出的算法1在效率上比文献[11]的约简算法有显著的提高，这是因为当数据库很大时存在大量的不重要属性，而算法1能从搜索空间中不断地删除不重要属性，避免了对其重要性的重复计算，从而使搜索效率得到显著的提高。

5 结 束 语

在基于正区域的属性约简算法中，无论是用分析的方法还是用差别矩阵的方法，都要计算划分 U/P，因此在基于正区域的属性约简中，设计求划分 U/P 的低复杂度的算法具有实际意义。本文给出了一个高效求 U/P 的算法，其时间复杂度为 $O(|P||U|)$；在此基础上设计出一种高效的、基于决策系统的属性约简算法，其时间复杂度为 $O(|C|^2|U|)$，低于文献[6-12]提出的约简算法的时间复杂度。此外，与现有约简算法相比，本文提出的约简算法有较大的灵活性，它能从搜索空间中逐次删除不重要属性，避免了对其重要性的重复计算，提高了搜索效率。

参 考 文 献

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

[2] 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.

[3] 张文修, 姚一豫, 梁 怡. 粗糙集与概念格[M]. 西安: 西安交通大学出版社, 2006.

[4] 彭 宏. 基于粗糙集理论的入侵检测方法研究[J]. 电子科技大学学报, 2006, 35(1): 108-110.

[5] 张 鹏, 张 靖, 刘玉增, 等. 粗集在交通事故黑点成因分析中的应用[J]. 电子科技大学学报, 2007, 36(2): 267-270.

[6] SKOWRON A， RAUSZER C. The discernibility matrics and functions in information system[C]//Intelligent Decision Support Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer Academic Publishers, 1992: 331-362.

[7] WANG J. Reduction algorithms based on discernibility matrix: The ordered attributes method[J]. Journal of Computer & Science, 2001, 16(6): 489-504.

[8] HU X H, CERCONE N. Learning in relational databases: A rough set approach[J]. International Journal of Computational Intelligence. 1995, 11(2): 323-338.

[9] 戎晓霞, 刘家壮, 马红英. 基于Rough集的决策表属性最小约简的整数规划算法[J]. 计算机工程与应用, 2004, 11(2): 24-25.

[10] ZHENG Z, WANG G Y. RRIA: A rough set and rule tree based incremental knowledge acquisition algorithm[J]. Fundamental Information, 2004, 59(2-3): 299-313.

[11] 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.

[12] 刘少辉, 盛秋戬, 吴 斌, 等. Rough集高效算法研究[J]. 计算机学报, 2003, 26(5): 524-529.

编 辑 黄 莘