基于粗集理论的特征子集选择算法*)

Algorithms Based on Rough Set Theory for Feature Subset Selection

赵 军[1,2] 王国胤[2] 吴中福[1] 唐 宏[1,3] 李 华[1] 廖晓锋[1]

(重庆大学计算机科学与工程学院 重庆400044)[1] (重庆邮电学院计算机科学与技术研究所 重庆400065)[2]

(重庆邮电学院移动通讯工程研究中心 重庆400065)[3]

Abstract Rough set theory is a valid mathematical tool for dealing with the problem of feature subset selection. In this paper, to break the restriction of the conception of conditional entropy and provide an effective measurement to the relative importance of redundant features, system entropy of a decision system is defined based on rough set theory; some of its algebraic characteristics are also researched. Then two similar heuristic algorithms are introduced to select features based on the notion of system entropy. Moreover, different characteristics of the two proposed algorithms are also deeply analyzed and discussed. The two new algorithms may surely maintain the discernible relation of decision systems; their space and time complexities are obviously much lower than that of analogous algorithms in literature. Simulation results on numerous UCI machine-learning databases indicate that the optimal feature subsets may always be expected through the two algorithms on almost all cases.

Keywords Rough set theory, Feature subset, System entropy

1. 引言

长期以来,特征子集选择技术一直是机器学习领域中的关键难题之一。由于学习对象的多样性,尤其是新的系统随着应用的发展而不断涌现,使人们无法用某种特定的工具或方法来完全解决这一问题,新的特征子集选择技术仍然受到人们广泛关注。20世纪80年代初,波兰数学家 Z. Pawlak 提出一种新的理论工具——"粗集",用于解决不完整和不精确信息的知识表达、学习及归纳等问题[1,2]。这一理论的特点是:除了问题所需处理的数据之外,不需要额外提供任何外界信息或先验知识[3]。这一独特优点使粗集理论在特征子集选择领域中的应用逐渐受到重视,人们已经成功地提出一些基于粗集理论的相应算法[4~14]。但是,理论分析表明,这些算法仍然具有较高的空间复杂度和/或时间复杂度。为了更有效地解决特征子集选择问题,本文提出两种相似的基于粗集理论的启发式算法,它们不但能够保证训练实例之间的分辨关系,而且具有更低的时间和空间复杂度,同时仿真结果表明,算法能够得到很好的选择结果。

本文基于粗集和信息理论定义决策系统的熵,并研究系统熵的代数特征;以系统熵作为度量属性相对重要性的启发式依据,提出两种新的特征子集选择算法,并给出算法的复杂性分析和仿真实验结果;对这两种算法进行分析和讨论;最后总结全文。

2. 系统熵及其代数特征

人们历来十分重视信息理论与粗集模型的有效结合。人们对信息熵,尤其是条件熵在特征子集选择领域中的应用已经进行了比较深入的研究,并取得了一定的研究成果:定义了基于条件熵的"属性重要性"概念,基于这一概念构造了相应的前向选择算法[11,15~16]。但是,根据这一概念难以构造任何反向删除[17]算法。这是因为按相应的定义,所有冗余属性对系统条件熵的贡献均为零,于是它们基于条件熵概念的属性重要性也全都为零,因此这一概念无法区分不同冗余属性的差别,不能根据这一概念来有效地选择待删除的"最佳"冗余属性。为了克服条件熵的这一缺点,同时仍然利用信息熵概念对系统统计特征的表达能力,我们将定义一种新的概念"系统熵",然后研究系统熵的代数性质。

定义1[3] 元组 $D=(U, V, f, A\cup\{d\})$是决策系统,其中:$U=\{x_1,x_2,\cdots,x_n\}$是有限的实例集合,称为论域;$A=\{a_1, a_2,\cdots,a_k\}$是有限的条件属性集合;$d$ 是决策属性;$V_a=[l_a,r_a]$是条件属性 $a\in A$ 的值域,$V=\cup V_a$;$f:U\to A\cup\{d\}$是实例空间对属性空间的映射函数。

实例 $x\in U$ 在属性 $a\in(A\cup\{d\})$上的取值记为 $a(x)$,即 $f(x, a)=a(x)$。

定义2[3] 在决策系统 $D=(U, V, f, A\cup\{d\})$中,若属性集合 $B\subseteq A\cup\{d\}$,则 B 在论域 U 上定义不可分辨二元关系 $IND(B)$:$IND(B)=\{(x,y)\mid((x,y)\in U\times U)\wedge(\forall\, b\in B(b(x)=b(y)))\}$

显然,$IND(B)$是一种等价关系,由此导出的对 U 的划分记为 $U/IND(B)$,其中包含实例 x 的等价类记为$[x]_{IND(B)}$。$[x]_{IND(\{d\})}$称为包含实例 x 的决策类,为方便起见,亦记为 $Dec(x)$。

定义3[3] 在决策系统 $D=(U, V, f, A\cup\{d\})$中,属性集合 $B\subseteq A\cup\{d\}$在决策类 $Dec(x)\subseteq U$ 上导出划分 $Dec(x)/IND(B)=\{X_1, X_2, \cdots, X_n\}$,则 B 在由 $Dec(x)$构成的 σ 代数上的概率分布为:

*)国家自然科学基金、攀登特别支持费、重庆市科委攻关基金资助。赵 军 博士生,研究领域为数据挖掘、网络安全等。王国胤 博士,教授,研究领域为智能数据分析和处理、网络安全等。吴中福 教授,博导,研究领域为计算机网络与通信、现代远程教育技术等。唐 宏 博士生,研究领域为移动 IP、数据挖掘等。李 华 副教授,研究领域为计算机网络与通信、现代远程教育技术等。廖晓锋 博士后,教授,博导,研究领域为计算机网络与通信、网络安全等。

$$[Dec(x)/IND(B):P]=\begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ p(X_1) & p(X_2) & \cdots & p(X_n) \end{bmatrix}$$

其中：$p(X_i)=\frac{|X_i|}{|Dec(x)|}, i=1,2,\cdots,n$

定义4 在决策系统 $D=(U,V,f,A\cup\{d\})$中，决策类 $Dec(x)\subseteq U$ 的熵定义为：

$$H(Dec(x):B)=-\sum_{X_i\in Dec(x)/IND(B)} p(X_i)\log(p(X_i))$$

定义5 决策系统 $D=(U,V,f,A\cup\{d\})$的熵定义为：

$$H(U:B)=\sum_{Dec(x)\in U/IND(\{d\})} p(Dec(x))H(Dec(x):B)$$

定理1 决策系统 $D=(U,V,f,A\cup\{d\})$，决策类 $Dec(x)\subseteq U$。若属性集合 $B'\subset B\subseteq A$，则：

$H(Dec(x):B')\leqslant H(Dec(x):B)$。

证明：设 $Dec(x)/IND(B)=\{X_1,X_2,\cdots,X_m\}$，$Dec(x)/IND(B')=\{Y_1,Y_2,\cdots,Y_n\}$。由于 $B'\subset B$，因此有：

$\forall X_i\exists Y_j(X_i\subseteq Y_j)$，即 $Dec(x)/IND(B)$是 $Dec(x)/IND(B')$的细分。于是：

$$Y_j=\bigcup_{X_i\subseteq Y_j} X_i, p(Y_j)=\sum_{X_i\subseteq Y_j} p(X_i)(j=1,2,\cdots,m)$$，因此：

$$H(Dec(x):B')=-\sum_{Y_j\in Dec(x)/IND(B')} p(Y_j)\log(p(Y_j))$$

$$=-\sum_{Y_j\in Dec(x)/IND(B')}\left(\sum_{X_i\subseteq Y_j} p(X_i)\right)\log\left(\sum_{X_i\subseteq Y_j} p(X_i)\right)$$

由于$\forall X_i\subseteq Y_j$，有：$\log\left(\sum_{X_i\subseteq Y_j} p(X_i)\right)\geqslant\log(p(X_i))$。因此：

$$H(Dec(x):B')\leqslant-\sum_{Y_j\in Dec(x)/IND(B')}\sum_{X_i\subseteq Y_j}(p(X_i)\log(p(X_i)))$$

$$=-\sum_{X_i\in Dec(x)/IND(B)} p(X_i)\log(p(X_i))=H(Dec(x):B)$$

因此 $H(Dec(x):B')\leqslant H(Dec(x):B)$成立。

定理1说明，若从决策系统中删除任何属性，将导致决策类的熵非严格单调地递减。

推论1 决策系统 $D=(U,V,f,A\cup\{d\})$，若属性集合 $B'\subset B\subseteq A$，则：$H(U:B')\leqslant H(U:B)$。

根据定义5和定理1，推论1很容易得到证明。

推论1说明，若从属性集合中删除属性，由于各决策类的熵非严格单调地递减，系统的熵将随之非严格单调地递减。

定理2 决策系统 $D=(U,V,f,A\cup\{d\})$，系统熵的理论上限和下限分别是 $H(U:A)$和0。

证明：由推论1知：从 A 中删除任何属性都将会使系统熵非严格单调地递减，因此 $H(U:A)$最大，是系统熵的上限；同时，$H(U:\Phi)$最小，是系统熵的下限。根据定义4和5易知：$H(U:\Phi)=0$。因此结论成立。

3. 基于系统熵的特征子集选择算法

3.1 特征子集和最优特征子集

我们首先基于决策系统的一致性定义系统的特征子集，然后提出两种相似的算法。这两种算法都按照典型的反向删除方式来选择特征子集，并且都以系统熵作为度量和区分属性的相对重要性的启发式依据。两种算法的根本区别在于他们确定“最佳”冗余属性的策略不同。

定义6 决策系统 $D=(U,V,f,A\cup\{d\})$是一致的，当且仅当：$\forall x,y\in U(d(x)\neq d(y))\rightarrow\exists a\in A(a(x)\neq a(y))$。

定义7 一致的决策系统 $D=(U,V,f,A\cup\{d\})$，$A'\subseteq A$ 是 A 的特征子集，当且仅当：$D'=(U,V,f,A'\cup\{d\})$仍是一致的，但$\forall A''\subset A'\rightarrow D''=(U,V,f,A''\cup\{d\})$不是一致的。

根据相关的定义容易证明，这里定义的特征子集本质上与粗集理论中条件属性集合相对于决策属性的约简概念是一致的。

由所有特征子集构成的集合记为 $Red(A)$，即：$Red(A)=\{A'|A'$是 A 的特征子集$\}$。

定义8 一致的决策系统 $D=(U,V,f,A\cup\{d\})$，$A'\in Red(A)$，A'是 A 的最优特征子集，当且仅当：

$\forall A''\in Red(A)\rightarrow|A''|\geqslant|A'|$。

显然，仅就特征子集的基数而言，一个决策系统可以同时存在多个最优特征子集。

定理3[18] 最优特征子集选择问题是 NP 完全问题。

在属性空间中搜索最优子集的时间代价为 $O(2^kJ)$，其中 k 是候选属性集合的基数，J 是评估每一属性子集所需要的时间。对具有丰富实例和较大（甚至中等规模）属性集合的训练数据来说，启发式的方法更为切实可行。因此，研究高效的启发式算法具有重要的现实意义。

3.2 特征子集选择算法 BSE(Backward Stepwise Elimination)

首先分析从属性集合 A 中删除某一属性 $a(a\in A)$时，决策系统 $D=(U,V,f,A\cup\{d\})$的变化规律。对一致决策系统 D，显然 $U/IND(A)$是 $U/IND(\{d\})$的细分。从 A 中删除属性 a，将导致 $U/IND(A)$中的某些等价类合并，如果 $U/IND(A-\{a\})$仍是 $U/IND(\{d\})$的细分，即 a 的删除并没有破坏决策系统的一致性，那么根据定义7，属性 a 显然在系统中是冗余的，选择特征子集的反向删除方式就需要删除这样的冗余属性。算法 BSE 就是反复循环，每次从属性集合中删除一个冗余属性，直至在余下的集合中已经不存在这样的冗余属性为止，即算法 BSE 按照典型的反向删除方式来选择特征子集。

在删除冗余属性的任一轮循环中，属性集合中可能同时存在多个冗余属性，删除不同的属性最终将可能得到不同的结果，但穷尽试探无疑具有指数阶的搜索空间，因此启发式算法就必须根据一定的依据和策略，有效地度量和区分各冗余属性的相对重要性，从中选择“最佳”的冗余属性来作为被删除对象。由于信息熵能够全面地反映系统的统计特征，因此这类概念是一种非常有效的度量属性重要性的方式。但是，已有的“条件熵”概念无法分辨冗余属性的相对重要性，而本文定义的系统熵概念则完全可以做到这一点。此外系统熵不但计算简单方便，并且具有确定的变化规律，因此在本文提出的算法中，我们将考察冗余属性删除前后决策系统熵的变化量，并把熵的变化量作为最终确定被删除属性的启发式依据。实践中可以有两种选择：(1)每次优先删除最小变化量对应的冗余属性；(2)每次优先删除最大变化量对应的冗余属性。这两种不同的策略正是算法 BSE1和 BSE2的唯一区别。正是这一区别表明这两种算法基于不同的思想和出发点，并使它们具有不同的特点，可以得到不同的结果。

由于决策系统的熵是非严格单调地递减的，因此系统熵变化量的相对大小完全可以通过仅考察冗余属性删除后系统的熵来确定。这可以进一步减小算法的计算量。

系统熵在算法中仅作为选择“最佳”冗余属性的启发式依据，它并不影响对属性冗余性的判定，因此算法能够保证决策

系统的分辨关系。

两种算法描述如下：

Algorithm BSE1:
Input: Consistent Decision System $D=(U, V, f, A\cup\{d\})$
Output: Feature Subset of A
1. $temp:=A$; $result:=\Phi$;
2. While ($|temp|>0$)Do
3. If($|temp|=1\wedge|result|=0$)//*At least one attribute of A should be remained.*
4. $result:=temp$; *Exit Loop While*
5. End If
6. $bEnd:=True$;
7. For each $a\in temp$;
8. $L:=U/IND(temp-\{a\})$;
9. *If*($\exists\ l\in L(\exists\ x,y\in l(d(x)\neq d(y))))$ //*D becomes inconsistent, feature a is indispensable.*
10. $result:=result\cup\{a\}$; $temp:=temp-\{a\}$;
11. Else //*feature a is redundant.*
12. $entropy[a]:=H(U:temp-\{a\})$;$bEnd:=False$;
13. End If
14. End For
15. If ($bEnd=True$) //*no feature still left in set temp*
16. Exit Loop While
17. Else
18. $attr:=a$
Where $a\in temp\wedge|V_a|=$MIN. cardinality of V_a among features with MAX. $entropy[a]$ in this iteration;
19. $temp:=temp-\{attr\}$; //*attr eliminated*
20. End If
21. End While
22. Return *result*

Algorithm BSE2:
This algorithm is just the same as algorithm BSE1 except the line 18:
18. attr := a
Where $a\in temp\wedge|V_a|=$MAX. cardinality of V_a among features with MIN. $entropy[a]$ in this iteration;

3.3 算法复杂性分析

算法 BSE1和算法 BSE2具有相同的时间复杂度和空间复杂度。

(1)*空间复杂度* 在文献报道的基于粗集理论的特征子集选择算法中，众多的算法都利用了系统的分辨矩阵[19]。这类算法都需要 $O(|U|^2\cdot|A|)$的辅助空间来存储分辨矩阵，而本文建议的算法仅需要 $O(|U|)$的辅助空间来存储属性子集对论域 U 的划分结果，因此算法 BSE1和 BSE2的空间复杂度都显著地低于基于分辨矩阵的任何算法。

(2)*时间复杂度* 算法 BSE1和 BSE2的时间复杂度取决于不可分辨关系的确定。确定不可分辨关系的最大次数为：$\sum_{i=0}^{|A-1|}(|A|-i)$，对应的属性集合的基数分别为：$|A|$-$i$-1，$i=0,1,\cdots,|A$-1|。按文[7]，通过对属性值排序的方式，可以在 $O(|B|\cdot|U|\log(|U|))$时间内确定 $U/IND(B)$。因此，确定所有不可分辨关系的时间代价，即算法的时间复杂度为：

$$T=\sum_{i=0}^{|A-1|}(|A|-i)\times((|A|-i-1)\cdot|U|\cdot\log(|U|))$$
$$=O(|A|^3\cdot|U|\log|U|)$$

文献报道的同类算法的时间复杂度分别为：四种 SBS 算法的时间复杂度均为：$O(|A|^3\cdot|U|^2)$[9]；基于分辨矩阵的任何启发式算法的时间性能取决于对分辨矩阵的扫描次数，其时间复杂度为：$O(|A|^2\cdot|U|^2)$；对基于信息论的 CEBARKCC、CEBARKNC、MIBARK 算法，其时间复杂度分别为：$O(|A|\cdot|U|^2)+O(|U|^3)$、$O(|A|^2\cdot|U|)+O(|A|\cdot|U|^3)$、$O(|A|\cdot|U|^2)+O(|U|^3)$[11]；Jelonek 及其改进算法的时间复杂度分别为 $O(|A|^3\cdot|U|^2)$和 $O(|A|^2\cdot|U|^2)$[14]。对任意实际的决策系统而言，通常有 $|A|\ll|U|$，否则实例的数目不足以表达其统计分布规律，这样的系统难以取得好的学习结果。因此算法 BSE1和 BSE2的时间复杂度通常都远低于参考算法。

3.4 算法仿真

要验证特征子集选择算法的有效性，可以将算法应用于现实生活中的数据集合，通过分析其输出结果来得到相应的结论。由于这类仿真数据真实的特征子集是不可知的，因此为了比较算法的性能，需要以某些参考算法为基准，通过对比这些算法的输出来衡量算法的有效性。

我们对 UCI 机器学习数据库中众多的数据集合进行了仿真实验，并以文[4]报道的算法为基准算法。该参考算法能够求出决策系统的全部特征子集(NP 复杂度)，我们可以从中挑选基数最小的特征子集作为其输出，并代表系统的最优特征子集。

在应用特征子集选择算法之前，首先对数据库进行了预处理：如果数据库中存在 ID 属性，则删除之；对具有连续属性的数据库，采用 Nguyen 贪心算法[20]进行离散化。

算法的仿真结果如表1所示。

表1 算法仿真结果

序号	数据库名称	$\|A\|$	$\|U\|$	|Result|		
				基准算法	算法 BSE1	算法 BSE2
1	Balance Scale Weight&Distance	4	625	4	4	4
2	Balloon (1)	4	20	2	2	2
3	Balloon (2)	4	16	4	4	4
4	Bupa Liver Disorders	6	345	3	3	4
5	Fitting Contact Lenses	4	24	4	4	4
6	The Monk's Problem(1)	6	124	3	3	3
7	The Monk's Problem (2)	6	169	6	6	6
8	The Monk's Problem(3)	6	122	4	4	4
9	Tie_Tac_Toe Endgame	9	958	8	8	8
10	Diagnostic Breast Cancer	30	569	10	10	10
11	Galss Identification Database	9	214	8	8	8
12	Iris Plants Database	4	150	4	4	4
13	John Hopkins Ionosphere	34	351	10	10	10
14	Image Segmentation	19	210	8	8	8
15	New Thyroid Disease Database	5	215	4	4	4
16	Wine Recognition Database	13	178	5	5	5

仿真结果表明：算法 BSE1无一例外地都能够得到与文[4]相同的最优特征子集，算法 BSE2除 Bupa 数据库之外，也得到了相同的结果。

4. 讨论

由推论1易知，删除任何属性之后，决策系统的熵将会非严格单调地递减，因此算法 BSE1每次优先删除最大熵对应的无关属性，就是期望该属性删除之后对系统的影响最小，从而减缓算法达到终态的速度，使更多的属性有机会被删除；与此相反，算法 BSE2每次选择与最小熵(熵的变化量最大，即信息增益最大)对应的无关属性，该属性的删除对系统的影响最强

烈，这种方式能够使算法迅速达到终态，因而可以缩短算法的运行时间。显然，我们可以预期，单就结果特征子集的基数（|result|）而言，算法 BSE1比 BSE2更有机会获得更好的结果。表1中对 Bupa 数据库的实验结果就反映了这一规律。

为表明算法 BSE1和 BSE2的这一特点，我们给出它们作用于 Bupa 数据库的详细结果，如表2、表3所示。

表2 Bupa 特征

$\|V_{A_1}\|$	$\|V_{A_2}\|$	$\|V_{A_3}\|$	$\|V_{A_4}\|$	$\|V_{A_5}\|$	$\|V_{A_6}\|$
26	78	67	47	94	16

（V_{A_i}指属性 A_i 的值域）

表3 Bupa 属性选择结果

基准算法	算法 BSE1	算法 BSE2
$\{A_2, A_3, A_5\}$	$\{A_2, A_3, A_5\}$	$\{A_1, A_2, A_4, A_6\}$

另一方面，算法 BSE1优先删除的属性对实例的分辨力弱，正因为如此，我们才能够保证它的删除对系统的影响最小。因此，在算法 BSE1中，那些具有较小值域空间的属性有着被优先删除的倾向，其直接后果就是结果子集中余下的属性都对实例具有较强的分辨力。对这一特征子集进行归纳学习，所得到的分类规则将可能仅具有较低的聚类能力，并且对新的待识别实例中的数据噪音也比较敏感，从而使学习结果对新实例进行预测的准确率降低。算法 BSE2则与此相反，这也正是算法 BSE2的优点。

我们用5次交叉验证的方式对 Bupa 数据库进行进一步的仿真研究：随机将数据库分为5个大致相等的子集，依次以其中的一个子集作为测试实例，余下的四个子集构成训练实例。每次首先对训练实例采用不同的算法来选择特征子集，然后对选择的结果都以0.5为阈值，运用 Skowron 算法进行归纳学习，最后对学习结果进行测试。表4给出有关的平均结果。对各次训练实例，算法 BSE1选择特征子集的结果全部同于基准算法，始终为{A_2, A_3, A_5}。

表4 算法性能比较

算法名称	|result|	正确识别率	错误识别率	拒绝识别率
基准算法	3.0	51.8%	44.8%	3.4%
算法 BSE1	3.0	51.8%	44.8%	3.4%
算法 BSE2	3.6	56.2%	43.2%	0.6%

结论 特征子集选择是机器学习领域的关键难题之一，学习系统的多样性使单一的工具或方法难以完全解决这一问题；最优特征子集选择问题是 NP 完全的。这些原因使进一步研究启发式的特征子集选择算法具有重要的现实意义。为了有效地度量学习系统中冗余属性的相对重要性，克服“条件熵”概念的局限，本文基于不可分辨关系定义了决策系统的熵，研究了系统熵的代数特征。在此基础上，提出两种启发式算法来解决特征子集选择问题。这两种算法都以系统熵为度量属性重要性的启发式依据，都按照典型的反向删除方式来选择特征子集，其根本区别在于它们根据不同的策略来确定“最佳”冗余属性。理论分析及仿真结果表明，尽管这两种算法各具特色，但在时间和空间性能上都明显优于同类算法，同时都能够有效地删除属性集合中的冗余属性，取得很好的特征子集选择结果。最后对这两种算法的特点进行了讨论。讨论中涉及的类似问题在其他的启发式特征子集选择算法中也同样存在，给出的结论同样对它们具有一定的参考价值。

参考文献

1 Pawlak Z. Rough Set. Int. J. of Computer and Information Science, 1982, 11: 341～456

2 Pawlak Z, et al. Rough Set. Communications of the ACM, 1995, 38(11): 89～95

3 王国胤. Rough 集理论与知识获取. 西安：西安交通大学出版社，2001. 23～36

4 常犁云，等. 一种 Rough Set 理论的属性约简及规则提取方法. 软件学报，1999，10(11)：1206～1211

5 吴福保，等. 基于粗集理论知识表达系统的一种归纳学习方法. 控制与决策，1999，14(3)：206～211

6 王珏，等. 基于 Rough Set 理论的数据浓缩. 计算机学报，1998，21(5)：393～400

7 Nguyen S H, et al. Some efficient algorithms for Rough Set methods. In: Proc. of the Conf. of Information Processing and Management of Uncertainty in Knowledge Based Systems, Granada, Spain, 1996. 1451～1456

8 Choubey S K, et al. A comparison of feature selection algorithms in the context of rough classifiers. In: Proc. of 5th IEEE Int. Conf. on Fuzzy Systems, 1996, 2: 1122～1128, New Orleans, LA

9 Choubey S K, et al. On Feature Selection and Effective Classifiers. J. of ASIS, 1998, 49(5): 423～434

10 苗夺谦，等. 知识约简的一种启发式算法. 计算机研究与发展，1999，36(6)：681～684

11 于洪，等. 基于 Rough Set 理论的知识约简算法. 计算机科学，2001，28(5. 专刊)：31～34

12 Hu X H, et al. Learning in relational databases: a Rough Set approach. Computational Intelligence, 1995, 11(2): 323～337

13 Jelonek J, et al. Rough set reduction of attributes and their domains for neural networks. Computational Intelligence, 1995, 11(2): 338～347

14 叶东毅，等. Jelonek 属性约简算法的一个改进. 电子学报，2000，28(12)：81～82

15 苗夺谦，等. 粗糙集理论中概念与运算的信息表示. 软件学报，1999，10(2)：113～116

16 Wang G Y. Algebra view and information view of rough set theory. In: Proc. of SPIE, 2001, 4384: 200～207

17 Caruana R, et al. Greedy attribute selection. In: Proc. of the 11th Int. Conf. on Machine Learning, New Brunswick, 1994. 28～36

18 Blum A L, et al. Training a 3-node neural network is NP-complete. Neural Networks, 1992, 5: 117～127

19 Skowron A, et al. The discernibility matrices and functions in information systems, Intelligent Decision Support-Handbook of Applications and Advances of the Rough Set Theorem. Dordrecht: Kluwer, 1992. 331～362

20 Nguyen H S, et al. Quantization of real values attributes, rough set and boolean reasoning approaches. In: Proc of 2nd Joint Annual Conf. on Information Science, 1995. 34～37