

文章编号:1001-9081(2006)03-0635-03

基于模糊粗糙集的两种属性约简算法

王丽^{1,2}, 冯山^{1,2}

(1. 四川师范大学 计算机软件实验室, 四川 成都 610066;

2. 四川师范大学 数学与软件科学学院, 四川 成都 610066)

(wl_myn@163.com)

摘要: 模糊粗糙集将模糊集合中的隶属度看作粗糙集理论中的属性值, 描述了模糊事件的可能性程度和必然隶属度。详细分析了基于模糊粗糙集的两种属性约简算法 FRSAR 和 CCD-FRSAR, 对比了它们的计算复杂性和收敛性, 并以计算实例验证了分析结论: CCD-FRSAR 总体优于 FRSAR。

关键词: 属性约简; 模糊粗糙集; 紧计算域; 计算复杂性; 算法收敛性

中图分类号: TP311.13 **文献标识码:**A

Two attribute reduction algorithms based on fuzzy-rough set

WANG Li^{1,2}, FENG Shan^{1,2}

(1. Laboratory of Computer Software, Sichuan Normal University, Chengdu Sichuan 610066, China;

2. College of Mathematics and Software Science, Sichuan Normal University, Chengdu Sichuan 610066, China)

Abstract: Fuzzy-rough set treats membership values in fuzzy sets as attribute values in rough set theory, which describes the possible degrees and the certain degrees of fuzzy events. Two attribute reduction algorithms based on fuzzy-rough set, FRSAR and CCD-FRSAR were analyzed and compared in computational complexity and convergency. The conclusion is validated by concrete experiments: as a whole, CCD-FRSAR is better than FRSAR.

Key words: attribute reduction; fuzzy-rough set; compact computational domain; computational complexity; convergency

0 引言

在粗糙集理论中, 属性约简是一个非常重要的课题。它在不降低信息系统分类能力的基础上, 用能区分所有对象的最小属性子集代替原来的属性集。计算属性约简有助于提取信息系统的规则, 实现数据挖掘的目的。对于大系统而言, 如果能删除冗余属性, 可以提高系统潜在知识的清晰度。

粗糙集数据分析不要求任何先验知识, 其结果比较客观, 也容易被人们理解。粗糙集模型适合处理离散属性值决策表, 但不适合表达属性值连续的决策表, 对于连续属性需要进行离散化。而传统离散技术因为没有考虑不同实值数据对离散值的不同隶属度, 有可能导致重要信息丢失。一种更合理的方法是使用模糊离散技术^[1], 将实数值转化为相应的隶属度值。对如何更有效地利用这些隶属度值来指导特征选择过程的问题, 文献[2]提出了一种基于模糊粗糙集的属性约简算法 (fuzzy-rough set attribute reduction, FRSAR)。实验表明^[3], 模糊粗糙特征选择比传统的基于熵、基于主成分分析以及基于随机性的维归约技术等方法的效果更好。遗憾的是, 由于算法终止准则设计上的缺陷, 导致它在许多实际数据集上并不收敛, 且算法的复杂性随属性个数呈指数阶增长, 随论域元素个数倍增。2004 年末, 文献[4]又提出了一种基于模糊粗糙集紧计算域的属性约简算法 (attribute reduction based on the compact computational domain of fuzzy-rough set, CCD-FRSAR)。

文献[2]的方法已在实际问题处理中得到了广泛运用,

分析和比较文献[4]与文献[2]方法的性能及特点, 不仅有助于进一步加深对模糊粗糙集属性约简算法的认识, 也有助于使得时效性和可靠性更优的算法得到推广使用, 为更多的人所接受。

1 FRSAR 算法

1.1 符号约定及基本概念

令 $U = \{x_1, \dots, x_n\}$ 是 n 个对象的非空有限集合, 称为论域; $P = \{P_1, \dots, P_j, \dots, P_p\}$ 是一组模糊条件属性, 每个属性通常表示为若干模糊语言项的集合 $A(P_i) = \{F_{ik}: k = 1, \dots, C_i\}$; $U/P = \{F_{ik}: i = 1, \dots, p; k = 1, \dots, C_i\}$ 是由 U 上的模糊相似关系 $R^{[5]}$ 生成的 U 的一个模糊划分; Q 是决策属性, $\forall x_i \in U$ 都被划分到类集合 $A(Q) = \{F_l: l = 1, \dots, C_Q\}$ 中, F_l 既可以是精确集, 也可以是模糊集。

定义 1 给定任意模糊集 $A, \mu_A(x): U \rightarrow [0, 1]$ 是隶属函数, $\forall x \in U, \forall F_{ik} \in U/P$ 元组 $(\mu_{\underline{A}}, \mu_{\bar{A}})$ 构成模糊粗糙集, $\mu_{\underline{A}}$ 和 $\mu_{\bar{A}}$ 分别称为下、上近似隶属函数, 定义如下^[5]:

$$\mu_{\underline{A}}(F_{ik}) = \inf_{x \in U} \max\{1 - \mu_{F_{ik}}(x), \mu_A(x)\}$$

$$\mu_{\bar{A}}(F_{ik}) = \sup_{x \in U} \min\{\mu_{F_{ik}}(x), \mu_A(x)\}$$

$\forall F_l \in A(Q)$, 由 U/P 中的元素 F_{ik} 可形成其近似 $(\underline{F}_l, \bar{F}_l)$ 。

F_{ik} 在 Q 的模糊正域下的隶属度定义为:

$$\mu_{POS}(F_{ik}) = \sup_{F_l \in A(Q)} \{\mu_{\underline{F}_l}(F_{ik})\}$$

x 对模糊正域的隶属度为:

$$\mu_{POS}(F_{ik}) = \sup_{F_{ik} \in A(P_i)} \min\{\mu_{F_{ik}}(x), \mu_{POS}(F_{ik})\}$$

决策属性 Q 对条件属性集 P 的依赖度定义为:

$$\gamma_P(Q) = \frac{\sum_{x \in U} \mu_{POS}(x)}{n}$$

1.2 FRSAR 算法

基本思想:根据决策属性 Q 对条件属性(集) $S \subseteq P$ 的依赖度来分层识别相关属性。

算法符号约定: R 存放每一层的选择变量; T 存放每一层的临时变量; γ'_{best} 为当前层的最大依赖度; γ'_{prev} 为前一层的最大依赖度。算法的伪代码^[2] 如下:

```

R ← {}, γ'_{best} = 0, γ'_{prev} = 0
do
    T ← R
    γ'_{prev} ← γ'_{best}
    ∀ P_i ∈ (P - R)
        if γ'_{R ∪ {P_i}}(Q) > γ'_{T}(Q)
            T ← R ∪ {P_i}
            γ'_{best} ← γ'_{T}(Q)
    R ← T
until γ'_{best} = γ'_{prev}
return R

```

显然,FRSAR 算法是一个树结构的组合搜索过程。它在每一层计算 Q 对 $S \subseteq P$ 的依赖度,计算过程由上至下进行。如果条件属性的个数为 p ,则最坏情况下,要计算 $2^p - 1$ 个可能组合的依赖度,最多计算 $\frac{p(p+1)}{2}$ 个节点。

对 FRSAR 算法,为找出 Q 对多个条件属性的依赖度,需要考虑 $\{A(P_i)\}_{i=1,2,\dots}$ 的笛卡儿乘积 $\{(F_{1k_1}, F_{2k_2}, \dots) : 1 \leq k_i \leq C_i\}$ 。因此,其计算复杂度将会随此增长,且随论域中的元素个数倍增,导致计算效率急剧下降。

2 CCD-FRSAR

由模糊 t -范数和 t -余范数算子的零元素性质,文献[4]提出了模糊粗糙集紧计算域的概念以及相应的 CCD-FRSAR 算法。

2.1 模糊逻辑算子

定义 2 模糊 t -范数(T) 和 t -余范数(S) 是递增的、结合的和可交换的映射^[5]: $T: [0, 1]^2 \rightarrow [0, 1]$, $S: [0, 1]^2 \rightarrow [0, 1]$, 且满足以下边界条件:

$$T(x, 1) = x, T(x, 0) = 0; \forall x \in U$$

$$S(x, 0) = x, S(x, 1) = 1; \forall x \in U$$

其中,模糊 t -范数算子的零元素为“1”;模糊 - 余范数算子的零元素为“0”。模糊推理中最常用的模糊 t -范数和 t -余范数分别是 min 和 max 算子。

2.2 基于紧计算域的模糊粗糙集

根据模糊逻辑算子的零元素性质,重新定义的上、下近似隶属函数。

定义 3 给定任意模糊集 A , $\mu_A(x): U \rightarrow [0, 1]$ 是隶属函数, $\forall x \in U, F_{ik} \in U/P$ 。元组 $(\mu_{\underline{A}}, \mu_{\bar{A}})$ 记为紧计算域上的模糊粗糙集,其中, $\mu_{\underline{A}}$ 和 $\mu_{\bar{A}}$ 分别定义为^[4]:

$$\mu_{\underline{A}}(F_{ik}) = \inf_{x \in D_{\underline{A}}(F_{ik})} \max\{1 - \mu_{F_{ik}}(x), \mu_A(x)\}, D_{\underline{A}}(F_{ik}) \neq \emptyset$$

$$\begin{cases} 1, & D_{\underline{A}}(F_{ik}) = \emptyset \end{cases}$$

$$\mu_{\bar{A}}(F_{ik}) =$$

$$\begin{cases} \sup_{x \in D_{\bar{A}}(F_{ik})} \min\{\mu_{F_{ik}}(x), \mu_A(x)\}, D_{\bar{A}}(F_{ik}) \neq \emptyset \\ 0, & D_{\bar{A}}(F_{ik}) = \emptyset \end{cases}$$

其中,

$$D_{\underline{A}}(F_{ik}) = \{x \in U: \mu_{F_{ik}}(x) \neq 0 \wedge \mu_A(x) \neq 1\}$$

$$D_{\bar{A}}(F_{ik}) = \{x \in U: \mu_{F_{ik}}(x) \neq 0 \wedge \mu_A(x) \neq 0\}$$

并称 $D_{\underline{A}}(F_{ik}) \subseteq U$ 和 $D_{\bar{A}}(F_{ik}) \subseteq U$ 分别为下、上近似隶属函数的紧计算域。

如果记 $f_{ik} = \{x \in U: \mu_{F_{ik}}(x) > 0\}$, $A_c = \{x \in U: \mu_A(x) = 1\}$, $A_S = \{x \in U: \mu_A(x) > 0\}$, \bar{A}_c 表示 A_c 的补集,即 $\bar{A}_c = \{x \in U: \mu_A(x) \neq 1\}$, 则有:

$$D_{\underline{A}}(F_{ik}) = f_{ik} \cap \bar{A}_c$$

$$D_{\bar{A}}(F_{ik}) = f_{ik} \cap A_S$$

由于 $(\mu_{\underline{A}}, \mu_{\bar{A}})$ 的计算只需考虑紧计算域 $D_{\underline{A}}, D_{\bar{A}}$ 内的元素,而不是针对所有的 $x \in U$,故称此时的 $(\mu_{\underline{A}}, \mu_{\bar{A}})$ 为基于紧计算域的模糊粗糙集。紧计算域是论域 U 的子集,它的引入省去了不必要的计算,提高了计算效率。

2.3 CCD-FRSAR 算法

此算法与 FRSAR 的基本思想是一致的,也是根据决策属性 Q 对条件属性(集) $S \subseteq P$ 的依赖度来分层识别相关属性。

在第一层, $S = \{P_i\} (i = 1, \dots, p)$, $\gamma_S(Q)$ 计算如下:

1) $\forall k, \forall l$, 下近似计算:

$$\mu_{\underline{F}_{lk}}(F_{ik}) =$$

$$\begin{cases} \inf_{x \in D_{\underline{F}_{lk}}(F_{ik})} \max\{1 - \mu_{F_{ik}}(x), \mu_{F_l}(x)\}, D_{\underline{F}_{lk}}(F_{ik}) \neq \emptyset \\ 1, & D_{\underline{F}_{lk}}(F_{ik}) = \emptyset \end{cases}$$

2) $\forall k = 1, \dots, C_i$, 计算 F_{ik} 的模糊正域:

$$\mu_{POS}(F_{ik}) =$$

$$\begin{cases} \sup_{F_l \in A(Q)} \{\mu_{F_l}(F_{ik})\}, \forall F_l \in A(Q), D_{\underline{F}_{lk}}(F_{ik}) \neq \emptyset \\ 1, & \exists F_l \in A(Q), D_{\underline{F}_{lk}}(F_{ik}) = \emptyset \end{cases}$$

3) 计算 $x \in U$ 对模糊正域的隶属度:

$$\mu_{POS}(x) = \sup_{F_{ik} \in A(P_i)} \min\{\mu_{F_{ik}}(x), \mu_{POS}(F_{ik})\}$$

4) 依赖度计算:

$$\gamma_S(Q) = \frac{\sum_{x \in U} \mu_{POS}(x)}{n}$$

算法符号约定: S 存放每一层的选择变量; C 存放每一层的拒绝变量; T 存放每一层的临时变量; $temp$ 存放每一层中选择和拒绝后的剩余变量; γ_{best} 为当前层的最大依赖度; γ_{prev} 为前一层的最大依赖度。算法伪代码如下^[4]:

$$\forall k, \forall i, \text{计算 } f_{ik} = \{x \in U: \mu_{F_{ik}}(x) > 0\}$$

$$\forall l, \text{计算 } f_l = \{x \in U: \mu_{F_l}(x) > 0\}$$

$$S \leftarrow \{\}, C \leftarrow \{\}, temp \leftarrow P, T \leftarrow \{\}$$

$$\gamma_{prev} = 0, \gamma_{best} = 0, v \leftarrow 0$$

do

$$T \leftarrow S$$

$$\gamma_{prev} \leftarrow \gamma_{best}$$

$$level-v: \forall P_i \in temp$$

$$\text{计算 } D_{\underline{F}_{lk}}(F_{ik}); k = 1, \dots, C_i$$

$$\text{计算 } \gamma_{S \cup \{P_i\}}$$

$$\text{If } (\gamma_{S \cup \{P_i\}} < \gamma_{prev}) \text{ Or } (\gamma_{S \cup \{P_i\}} < \gamma_{best})$$

$$\text{Then } C \leftarrow C \cup \{P_i\}$$

$$\text{If } (\gamma_{S \cup \{P_i\}} > \gamma_{best})$$

```

Then
 $T \leftarrow S \cup \{P_i\}$ 
 $\gamma_{best} \leftarrow \gamma_T$ 
End of level-v
 $v \leftarrow v + 1$ 
 $S \leftarrow T$ 
 $temp \leftarrow P - S - C$ 
While ( $temp \neq \emptyset$ )
Return S

```

新算法与FRSAR不同之处在于:一是引入了迭代思想,只需计算出搜索树第一层的紧计算域,其余层的计算域则可通过上一层计算域的迭代交产生,减少了计算工作量。二是它并入了两个新的改进步骤:1)与第 v 层选择属性的依赖度相比,如果它与任何其他条件属性的组合在第 $v+1$ 层的依赖度降低,则将新加入的属性从搜索树中去除;2)如果层 v 的两个节点的依赖度相同,且同时为最大,就出现了所谓的“多枝”问题,后继搜索将在两个不同的分支上进行,在第 $v+1$ 层分别计算这两个节点与其他属性组合的依赖度,保留产生更大的那个。

3 算法分析及计算实验

为了比较两种算法的性能,我们以文献[3]选用的决策表为例。如表1所示,论域 $U = \{0, \dots, 8\}$; $P = \{P_1, P_2, P_3\}$ 是模糊条件属性集,每个属性都表示为若干模糊项的集合,如 $A(P_1) = \{F_{11}, F_{12}, F_{13}\}$; $U/P = \{F_{ik}: i = 1, 2, 3; k = 1, \dots, c_i\}$ 是由 U 上的模糊相似关系 $R^{[5]}$ 生成的 U 的一个模糊划分; $Q = \{Plan\}$ 是决策属性,每个 $x_i \in U$ 都被划分到类集合 $A(Q) = \{F_1, F_2, F_3\}$ 中,这里的 F_i 也是模糊集。

表1 样本数据集

对象	P_1			P_2			P_3			$Plan$		
	F_{11}	F_{12}	F_{13}	F_{21}	F_{22}	F_{23}	F_{31}	F_{32}	F_1	F_2	F_3	
0	0.3	0.7	0.0	0.2	0.7	0.1	0.3	0.7	0.1	0.9	0.0	
1	1.0	0.0	0.0	1.0	0.0	0.0	0.7	0.3	0.8	0.2	0.0	
2	0.0	0.3	0.7	0.0	0.7	0.3	0.6	0.4	0.0	0.2	0.8	
3	0.8	0.2	0.0	0.0	0.7	0.3	0.2	0.8	0.6	0.3	0.1	
4	0.5	0.5	0.0	1.0	0.0	0.0	0.0	1.0	0.6	0.4	0.0	
5	0.0	0.2	0.8	0.0	1.0	0.0	0.0	1.0	0.0	0.7	0.3	
6	1.0	0.0	0.0	0.7	0.3	0.0	0.2	0.8	0.7	0.3	0.0	
7	0.1	0.8	0.1	0.0	0.9	0.1	0.7	0.3	0.0	0.0	1.0	
8	0.3	0.7	0.0	0.9	0.1	0.0	1.0	0.0	0.0	0.0	1.0	

3.1 约简运算

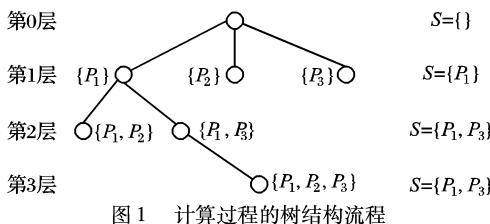


图1 计算过程的树结构流程

根据算法流程,逐次计算决策属性 Q 对条件属性集 P 中元素所有组合的依赖度。就此决策表而言,两种算法的最终约简结果都相同。

$$\text{第1层: } \gamma_{|P_1|}(Q) = 3.8/9, \gamma_{|P_2|}(Q) = 2.1/9,$$

$$\gamma_{|P_3|}(Q) = 2.7/9, \text{故 } S = \{P_1\};$$

$$\text{第2层: } \gamma_{|P_1, P_2|}(Q) = 4.0/9, \gamma_{|P_1, P_3|}(Q) = 5.1/9, \text{故}$$

$S = \{P_1, P_3\}$;

第3层 $\gamma_{|P_1, P_3, P_2|}(Q) = 5.1/9 = \gamma_{|P_1, P_3|}(Q)$, 故 $S = \{P_1, P_3\}$;

因此最后的约简结果为 $S = \{P_1, P_3\}$ 。

计算过程的树结构流程如图1所示。

3.2 算法分析

3.2.1 计算复杂度分析

在第 v 层,FRSAR的计算代价为:

$$\sum_{i \in temp} C_S \times C_i + 2 \times C_S \times C_i \times C_Q \times n + n,$$

$$C_S = \prod_{j \in S} C_j$$

其中, S 表示到第 v 层的选择属性集, P 为原条件属性集, $temp = P - S$ 。在第1层, $S = \emptyset$, $C_S = 1$, $temp = \{1, \dots, p\}$ 。

CCD-FRSAR:首先计算每个条件属性在第1层的紧计算域,第1层的计算量为:

$$\sum_{i \in temp} 2 \times C_i + C_Q \times C_i \times C_Q + domain_i + n$$

$$domain_i = \sum_{l=1}^{c_Q} \sum_{k \in [1, \dots, c_l]} |D_{F_l}(F_{ik})|$$

在随后的任意层 v ,计算量为:

$$\sum_{i \in temp} C_i \times C_S + 2 \times domain_i + n$$

$$C_S = \prod_{j \in S} C_j$$

其中, S 表示到第 v 层的选择属性集, P 为原条件属性集, $temp = P - S - C$, C 为到第 v 层的拒绝属性集。在第1层, $S = \emptyset$, $C_S = 1$, $temp = \{1, \dots, p\}$, $domain_i = \sum_{l=1}^{c_Q} \sum_{k \in [1, \dots, c_l \times C_S]} |D_{F_l}(F_{ik})|$ 。

各层的计算量加起来就得到了算法的总计算代价。

就前面计算实例中的样本数据集而言,其计算代价分别为:FRSAR,2309;CCD-FRSAR,878。可见后者的计算代价明显小于前者。

3.2.2 算法收敛性分析

CCD-FRSAR将在有限时间内收敛到某个 $S \subseteq P$,并能有效处理“多枝”问题,也不会陷于无限搜索循环。其理由是:第 v 层的搜索完成后,第 $v+1$ 层有3种情况可能出现:1)没有一个拒绝属性,并且选择一个最优属性;2)拒绝多个属性,同时选择一个最优属性;3)拒绝所有属性,没有选择任何属性。所以在第 v 层,若以 $|temp|_v$ 表示 $temp$ 集合的势,则 $|temp|_{v+1} < |temp|_v$ 。

对FRSAR算法,则不能保证总可以在有限时间内收敛:如果第 $v+1$ 层的最大依赖度高于第 v 层,FRSAR将添加一个新变量到选择属性集;若至少有一个属性组合不满足此条件,算法在树的新搜索层将不添加任何属性,这将导致无限搜索循环。

此外,在FRSAR中,如果在某一层 v 达到条件 $\gamma'_{best} = \gamma'_{prev}$,搜索过程将终止,此时,算法只选出了到第 v 层的约简属性集,但其他组合可能会产生更高的函数依赖。可见,FRSAR最终选出的属性子集并不总是可靠的。而CCD-FRSAR算法的终止条件为 $temp = \emptyset$,即它将计算决策属性对条件属性的所有可能组合的依赖度,从而保证了约简结果的可靠性。

4 结语

模糊粗糙集属性约简算法FRSAR是处理连续属性值决策的(下转第672页)

4 算法分析与对比实验

决策树的生成和事前修剪是两个交织在一起的过程,当树生成结束时,事前修剪也就结束了。基于节点支持度、纯度的事前修剪算法 PDTBS、PDTBP 简单易行,它们仅仅需要在决策树的节点上增加一(数)个数据域,用于记录节点对应样例集中样例的总(各个类别的)数目。前者在判断节点是否要扩展时仅需一次比较,其时间复杂度为 $O(1)$;后者需在比较前计算一下熵的值,其时间复杂度为 $O(C)$, C 为类别个数。由于引入了队列,使得算法 PDTBS、PDTBP 消除了传统决策树生成算法(如 ID3)中的递归调用,提高了效率;同时,这两个算法并没有对节点相关属性的选择标准做硬性的规定,在实际使用时可以根据需要定义,比较灵活。

采用 UCI^[8] 中关于机器学习的 tic-tac-toe 数据库做对比实验。此库中的记录分为两个类别共 958 条,每次实验随机抽取 70% 的记录作为训练样例,分别用 ID3、卡方测试(阈值分别设为 1,2,4)做事前修剪的 ID3、PDTBS 算法(支持度阈值分别设为 958 的 0.5%, 1%, 1.5%; 节点相关属性的选择以信息增益最大为标准)、PDTBP 算法(纯度阈值分别设为 0.2, 0.4, 0.6; 节点相关属性的选择以信息增益最大为标准)生成 10 棵决策树,对剩下 30% 的记录分类。独立地实验 100 次,统计得出这 10 棵决策树的平均节点数与平均分类精度于表 2 中,在此表中还列出了判断一个节点是否需扩展的时间复杂度。

表 2 分类实验结果

算法	阈值	平均分类精度 (%)	平均节点数	时间复杂度
ID3		84.681	275	$O(1)$
ID3 (Chi-square test)	1	84.764	268	
	2	84.685	253	$O(K)$
	4	82.991	158	
PDTBS	0.5%	84.244	197	
	1%	82.291	135	$O(1)$
	1.5%	80.092	105	
PDTBP	0.2	84.681	275	
	0.4	84.579	269	$O(C)$
	0.6	82.881	208	

其中, K 为属性的平均属性值个数; C 为类别个数。

从表中可以发现, 卡方测试的方法对树的修剪较为保守, 但精度控制得好; PDTBS 算法对树的修剪幅度最大, 但相对

(上接第 637 页)

策表的有效算法,并已在实际应用中得到了广泛运用。而基于模糊粗糙集累计算域的属性约简算法 CCD-FRSAR 是最新提出的,本文系统地研究并比较了这两种算法的特点及性能。我们的理论分析和实验计算结果均表明: CCD-FRSAR 在时间复杂度和计算结果的可靠性上均优于 FRSAR。在算法的收敛性上, 算法 FRSAR 无法保证总可以在有限时间内收敛, 而算法 CCD-FRSAR 利用剪枝的思想方法,使得各搜索层剩余条件属性集合的势总是向减小的趋势发展,从而保证了收敛。

需要指出的是,尽管 CCD-FRSAR 有这些优点,但它并没有考虑到大型数据集分析中,由于人为错误或噪声可能导致的某些数据被误分的分类问题,缺乏抗噪声干扰的能力,这将在一定程度上制约其处理复杂应用问题的有效性。

参考文献:

于树的大幅度修剪(例如 $support = 1.5\%$, 相对于 ID3 生成的树, 节点从 275 变为 105, 树被修剪了 61.8%), 精度并没有如此幅度的降低(上例, 树的精度从 84.681% 变为 80.092%, 仅仅降低了 4.589%), 并且此方法的时间复杂度最小; 使用 PDTBP 算法时要注意对阈值的选择, 例如表中当 $purity = 0.2$ 时树根本没有被修剪, 这种方法对树的修剪最保守, 但精度控制得较好。

5 结语

假设训练样例有 n 个属性, 每个属性平均有 m 个属性值, 那么生成的一棵决策树, 理论上其节点数可达 m^n 个, 这是一个很巨大的数字。事前修剪的目的就在于控制决策树的规模, 避免生成较大的树, 这样做一方面节省了空间, 另一方面在利用它分类时也节省了时间。虽然过于复杂的树降低分类精度的可能性存在^[2] 并在实验中再次得到了验证(表 2 中卡方测试的第 1 行), 但绝大多数事前修剪的方法都会降低分类精度, 关键是要在树的复杂性和分类精度之间取得一种平衡, 即要求在(大幅度)修剪决策树的同时精度损失要尽可能的小, 本文提出的事前修剪算法 PDTBS 和 PDTBP 都很好地达到了这一目的。

参考文献:

- [1] QUINLAN JR. Induction of Decision Tree[J]. Machine Learning, 1986, 1(1): 81 - 106.
- [2] FAYYARD UM, IRAN KB. What Should Be Minimized in a Decision Tree? [A]. Proceedings of Eighth National Conference on Artificial Intelligence[C]. Boston. MA. USA, 1990. 749 - 754.
- [3] QUINLAN JR. Simplifying decision trees[J]. International Journal of Man-Machine Studies, 1987, 27(3): 221 - 234.
- [4] BRESLOW LA, AHA DW. Simplifying decision trees: a survey[J]. Knowledge Engineering Review, 1997, 12(1): 1 - 40.
- [5] WEI HL. Comparison among Methods of Decision Tree Pruning[J]. Journal of Southwest Jiaotong University, 2005, 40(1): 44 - 48.
- [6] HOGG RV, CRAIG AT. Introduction to mathematical statistics [M]. London: Collier-Macmillan, 1970.
- [7] MITCHELL TM. Machine Learning [M]. China Machine Press, 2003.
- [8] University of California. Irvine repository of machine learning database[DB/OL]. <ftp://ics.uci.edu> in the /pub/machine-learning-databases directory, 2005 - 05 - 12.

- [1] SHEN Q, CHOUCOULAS A. A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems[J]. Engineering Applications of Artificial Intelligence, 2000, 13 (3): 263 - 278.
- [2] JENSEN R, SHEN Q. Fuzzy-rough attribute reduction with application to web categorization[J]. Fuzzy Sets and Systems, 2004, 141 (3): 469 - 485.
- [3] JENSEN R, SHEN Q. Fuzzy-rough data reduction with ant colony optimization[J]. Fuzzy Sets and Systems, 2005, 149 (1): 5 - 20.
- [4] BHATT RB, COPAL M. On fuzzy-rough sets approach to feature selection[J]. Pattern Recognition Letters, 2005, 26 (7): 965 - 975.
- [5] RADZIKOWSKA AM, KERRE EE. A comparative study on fuzzy-rough sets [J]. Fuzzy Sets and Systems, 2002, 126 (2): 137 - 155.