第 19 卷 第 11 期　　控 制 与 决 策　　2004 年 11 月
Vol. 19 No. 11　　Control and Decision　　Nov. 2004

文章编号: 1001-0920(2004)11-1208-05

基于互信息和 Beam 搜索的粗糙集属性约简算法

杨　胜[1], 施鹏飞[1], 顾　钧[2]
(1. 上海交通大学 图像处理与模式识别研究所, 上海 200030; 2. 香港科技大学 计算机系, 香港)

摘　要: 从属性集互信息的角度分析了粗糙集理论的属性约简问题. 首先在互信息的基础上定义了一个新的属性子集的冗余性和协同能力度量——属性子集的冗余协同系数; 然后将它作为属性约简度量, 提出了基于 Beam 搜索的粗糙集属性约简算法. 实验表明属性约简算法具有良好的运行效果.

关键词: 粗糙集; 属性约简; 互信息; Beam 搜索; 算法
中图分类号: TP18　　**文献标识码**: A

Rough set attribute reduction based on mutual information and Beam search

YANG Sheng[1], *SHI Peng-fei*[1], *GU Jun*[2]
(1. Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, China; 2. Department of Computer Science, Hong Kong University and Technology, Hong Kong, China. Correspondent: YANG Sheng, E-mail: yangsheng@sjtu.edu.cn)

Abstract: The attribute reduction problem for rough set is analyzed by the mutual information of attribute set. Based on mutual information, the redundancy-synergy coefficient of attribute set, a novel measure for redundancy and synergistic ability of attribute set, is defined. Then, a Beam search based attribute reduction algorithm for rough set is presented, where the redundancy-synergy coefficient is taken as the attribute reduction measure. Experiments show that the new algorithm yields satisfying attribute reduction results.

Key words: rough set; attribute reduction; mutual information; Beam search; algorithm

1 引　　言

粗糙集理论是由 Pawlak 提出的[1], 它通过属性约简和值约简得到分类规则. 最小属性约简是一个 NP-hard 问题[2]. 得到最小属性约简最直接的方法就是采用完全搜索算法, 如分支界限法[3,4]等, 但随着数据集维数的增大, 必将导致算法运行时间的增加. 目前, 通常采用求核集的方法进行属性约简[5~7]. 核集通过完全搜索策略找到最优约简, 或通过启发式策略找到一个次优约简. 这类算法用于小属性集较为合适, 对于大属性集, 尤其是没有核集的大属性集, 往往耗费大量的运算时间, 或产生一个很差的约简结果.

文献[8, 9]提出了粗糙集理论概念的信息熵表达. 本文则从属性集互信息的角度分析粗糙集理论的属性约简问题, 并定义一个属性子集的冗余性和协同能力度量——冗余协同系数. 文献[10]提出了基于 Beam 搜索[10]的粗糙集属性约简算法, 它以冗余协同系数作为属性子集选择度量. Beam 搜索算法可看作是完全搜索算法的一个简化. 它是一个启发式搜索算法, 具有多项式时间复杂度的特点.

收稿日期: 2004-02-02; **修回日期**: 2004-03-29.
基金项目: 国家自然科学基金资助项目(60075007); 国家 973 基础研究项目(G1998030401).
作者简介: 杨胜(1977—), 男, 湖南会同人, 博士生, 从事粗糙集理论、数据挖掘算法领域的研究; 施鹏飞(1939—), 男, 上海人, 教授, 博士生导师, 从事模式识别、图像处理领域的研究.

2　理论框架

2.1　基于互信息的粗糙集概念分析

定义 1　信息系统 $IS = \langle U, R, V, \xi \rangle$. 其中: U 为论域; $R = F \cup D$ 为属性集合, $F = \{f_1, f_2, \cdots, f_p\}$ 为条件属性集, $p = |F|$, $D = \{P\}$ 为决策属性集, P 为决策属性; $V = \bigcup_{f \in R} V_f$, V_f 表示属性 f 的值域; $\xi: U \times R \to V$ 为一个信息函数.

定义 2　$A \subseteq F$, $\mathrm{ND}(A)$ 表示 A 上的不可分辨关系. $f \in A$, 如果 $\mathrm{ND}(A) = \mathrm{ND}(A - \{f\})$, 则称 f 在 A 中为可去除属性; 否则, 称 f 在 A 中为不可去除属性(或必要属性). 所有 A 中的必要属性组成的集合称为 A 的核, 记为 $\mathrm{CORE}(A)$.

A ($A \subseteq F$) 可看作是定义在由 U 的子集组成的 σ- 代数上的一个随机变量, 设 A 在 U 上导出的划分为 $Y, U/\mathrm{ND}(A) = Y = \{Y_1, Y_2, \cdots, Y_a\}$. 其概率分布可表示如下:

$$[Y; p(\cdot)] = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_a \\ p(Y_1) & p(Y_2) & \cdots & p(Y_a) \end{bmatrix}. \tag{1}$$

其中: $p(\cdot)$ 表示概率密度函数, $p(Y_j) = |Y_j|/|U|$, $j = 1, 2, \cdots, a$, $|\cdot|$ 表示基数.

定义 3　$A \subseteq F$, 如果任意属性 $f \in A$, f 在 A 中为必要属性, 则称 A 为独立的; 否则, 称 A 为相依的.

定义 4　$A \subseteq B \subseteq F$, 如果 $\mathrm{ND}(A) = \mathrm{ND}(B)$, 称 A 为 B 的一个等价属性子集; 如果 $\mathrm{ND}(A) = \mathrm{ND}(B)$, 且 A 为独立的, 则称 A 为 B 的一个约简, 记作 $\mathrm{RED}(B)$. 有 $\mathrm{CORE}(B) = \cap \mathrm{RED}(B)$.

定义 5　条件属性集 F 与决策属性 P 之间的互信息[11] 记为

$$I(F; P) = I(P) - E(P|F). \tag{2}$$

式中: $I(F; P)$ 为 F 和 P 之间的互信息, $I(P)$ 为类属性 P 的期望信息(熵), $E(P|F)$ 为类属性 P 的条件熵.

设 P 和 F 在 U 上导出的划分分别为 $U/\mathrm{ND}(P) = \{X_1, X_2, \cdots, X_u\}$, $U/\mathrm{ND}(F) = \{Y_1, Y_2, \cdots, Y_v\}$. $I(P)$ 和 $E(P|F)$ 可以分别由下式计算:

$$I(P) = -\sum_{i=1}^{u} p(X_i) \log_2 p(X_i) = -\sum_{i=1}^{u} \frac{|X_i|}{|U|} \log_2 \frac{|X_i|}{|U|}, \tag{3}$$

$$E(P|F) = -\sum_{i=1}^{u} \sum_{j=1}^{v} \frac{|X_i \cap Y_j|}{|U|} \log_2 \frac{|X_i \cap Y_j|}{|Y_j|}. \tag{4}$$

单调性: $A \subseteq B \subseteq F$, 则 $I(A; P) \leq I(B; P)$[11].

引理 1　如果 $A \subseteq B \subseteq F$, 则 $\mathrm{ND}(A) = \mathrm{ND}(B) \Leftrightarrow E(P|A) = E(P|B)$.

证明　1) 充分性 ($\mathrm{ND}(A) = \mathrm{ND}(B) \Rightarrow E(P|A) = E(P|B)$), $A \subseteq B$ 且 $\mathrm{ND}(A) = \mathrm{ND}(B)$, 说明 A 和 B 导出完全相同的划分, 设 $U/\mathrm{ND}(P) = \{X_1, X_2, \cdots, X_u\}$, $U/\mathrm{ND}(A) = U/\mathrm{ND}(B) = \{Y_1, Y_2, \cdots, Y_v\}$, 根据式 (4), 易证 $E(P|A) = E(P|B)$.

2) 必要性 ($\mathrm{ND}(A) \neq \mathrm{ND}(B) \Rightarrow E(P|A) \neq E(P|B)$ 或 $E(P|A) = E(P|B) \Rightarrow \mathrm{ND}(A) = \mathrm{ND}(B)$). 取 $\mathrm{ND}(A) \neq \mathrm{ND}(B)$, 设 $U/\mathrm{ND}(P) = \{X_1, X_2, \cdots, X_u\}$, $U/\mathrm{ND}(A) = \{YA_1, YA_2, \cdots, YA_{va}\}$, $U/\mathrm{ND}(B) = \{YB_1, YB_2, \cdots, YB_{vb}\}$. 因为 $A \subseteq B$, 根据属性子集在 U 上划分的定义, B 在 U 上的划分可看作是基于属性子集 $B-A$ 对 A 在 U 上划分的进一步细分, 因此至少存在一个等价类被属性子集 $B-A$ 划分. 设等价类 YA_j ($YA_j \in U/\mathrm{ND}(A)$) 被划分为 $YB_{j1}, YB_{j2}, \cdots, YB_{jb}$ ($YB_{j1}, YB_{j2}, \cdots, YB_{jb} \in U/\mathrm{ND}(B)$), 得 $YB_{j1} \cup YB_{j2} \cup \cdots \cup YB_{jb} = YA_j$, $|YB_{j1}| + |YB_{j2}| + \cdots + |YB_{jb}| = |YA_j|$, $(YB_{j1} \cap X_i) \cup (YB_{j2} \cap X_i) \cup \cdots \cup (YB_{jb} \cap X_i) = (YA_j \cap X_i)$, $|YB_{j1} \cap X_i| + |YB_{j2} \cap X_i| + \cdots + |YB_{jb} \cap X_i| = |YA_j \cap X_i|$. 而

$$E(P|A) - E(P|B) = \sum_{j=1}^{v} \sum_{i=1}^{u} \left(\sum_{k=1}^{jb} \frac{|X_i \cap YB_{jk}|}{|U|} \log_2 \frac{|X_i \cap YB_{jk}|}{|YB_{jk}|} - \frac{|X_i \cap YA_j|}{|U|} \log_2 \frac{|X_i \cap YA_j|}{|YA_j|} \right). \tag{5}$$

利用递推方法易证等式 (5) > 0, $E(P|A) \neq E(P|B)$, 所以必要性成立. 综合可证引理 1 成立. □

定理 1　$A \subseteq B \subseteq F$, $\mathrm{ND}(A) = \mathrm{ND}(B) \Leftrightarrow I(A; P) = I(B; P)$.

根据互信息的定义和引理 1 简单可证.

推论 1　$A \subseteq B \subseteq F$, $I(A; P) = I(B; P)$, 则 A 为 B 的等价属性子集.

定理 2　如果 $I(A; P) = I(B; P)$, 且 A 独立, 则 A 为 B 的一个约简的充分必要条件是 $A \subseteq B \subseteq F$.

根据定义 4 和定理 1 简单可证.

定理 3　$A \subseteq F$, 如果 $I(A; P) > I(A - \{f | f \in A\}; P) \Leftrightarrow f$ 为 A 的必要属性.

根据单调性, 定义 2 和定理 1 简单可证.

定理 4　$A \subseteq B \subseteq F$, 如果 $I(A; P) = I(B; P)$, 则 B 的必要属性也是 A 的必要属性.

证明 设f ($f \in B$) 是B 的一个必要属性, $I(B - \{f\}; P) < I(B; P)$, 如果$f \notin A$, 则$A \subseteq B - \{f\}$, 根据单调性得$I(A; P) < I(B; P)$, 与条件矛盾, 所以$f$ 必定属于A; 又因为$I(B - \{f\}; P) < I(B; P) = I(A; P)$, $A - \{f\} \subseteq B - \{f\}$, 从而$I(A - \{f\}; P) \leq I(B - \{f\}; P) < I(A; P)$, 所以$f$ 为A 的必要属性 □

定理5 $A \subseteq B \subseteq F$, 如果$I(A; P) = I(B; P)$, 则$A$ 中至少有一个B 的约简

根据单调性和定义4简单可证

定理6 $A \subseteq B \subseteq F$, 如果$I(A; P) = I(B; P)$, 则$A$ 的约简必为B 的约简

根据单调性和定义4简单可证

2.2 属性子集冗余协同系数

根据信息编码理论, Brenner 等人定义了冗余协同指数作为两个神经元f_1 和f_2 传递刺激P 的信息组合协同能力和冗余度度量[12]. 这里从信息量熵的角度描述属性子集的冗余程度和组合协同能力, 称为冗余协同系数. 它是冗余协同指数的扩展, 是一个相对信息度量的概念. 其取值范围为$(0, \infty)$[13]. 冗余协同系数越小, 属性的组合能力越弱, 说明属性间包含类信息的冗余越大, 可被删除的属性越多, 而保持互信息不减少.

定义6 A ($A = \{f_i \mid f_i \in A, i = 1, \cdots, a\}$)$F$, $RSC(A)$ 称为属性子集A 的冗余协同系数, 其计算如下:

$$RSC(A) = \frac{I(A; P)}{\sum_{i=1}^{a} I(f_i; P)}. \tag{6}$$

定理7 如果A 为B 的等价属性子集, 则$RSC(A) \geq RSC(B)$.

根据定义2和冗余协同系数的定义简单可证

定理8 对于属性子集$A \subseteq F$, $A = \{f_1, f_2, \cdots, f_a\}$, 如果$I(f_1; P) < I(f_2; P) < \cdots < I(f_a; P)$, 且$I(A - \{f_i \mid i = 1, 2, \cdots, a\}; P) = I(A; P)$, 则$RSC(A - \{f_1\}) < RSC(A - \{f_2\}) < \cdots < RSC(A - \{f_a\}) < RSC(A)$.

证明 因为A 中属性按互信息从小到大排列, 因此$\sum_{i=1}^{a} I(f_i; P) - I(f_i; P)$, $i = 1, 2, \cdots, a$ 从大到小排列; 而$I(A - \{f_i\}; P) = I(A; P)$, 根据RSC 的定义, $RSC(A - \{f_i\})$ 从小到大排列 □

3 属性约简算法

粗糙集属性约简就是要搜索到初始属性集F 的尽可能小的等价属性子集. 冗余协同系数是属性集的一个属性协同表达类属性的冗余性和协同能力的度量, 冗余协同系数越小, 冗余度越大, 越可能找到一个更小的F 的等价属性子集. 因此, 将冗余协同系数作为属性子集选择度量. 结合Beam 搜索算法, 提出了以冗余协同系数为度量的粗糙集属性约简算法, 称为Beam 保持互信息最小化冗余协同系数方法(BEMM_MRSC), 描述如下:

算法: BEMM_MRSC

输入: IS- 信息系统

Beam-Beam 队列

M -Beam 宽度

Queue- 一个队列 // 长度为M^2

输出: F 的约简

Step1: 将F 中的属性按照互信息从小到大排列;

Step2: 找到M 个冗余协同系数较小的F 的孩子等价属性子集, 存入Beam;

Step3: 清空Queue, 对于Beam 中的每个属性子集, 找到其冗余协同系数较小的M 个孩子等价属性子集, 存入Queue;

Step4: 如果Queue 为空, 转 Step6; 否则, 转 Step5;

Step5: 清空Beam, 从Queue 中选择M 个冗余协同系数较小的属性子集存入Beam, 转 Step3;

Step6: 输出Beam 中的所有属性子集

在BEMM_MRSC 中首先将初始属性集F 中的属性按照互信息从小到大排列. 根据定理8, 运用这个排列只需要找到每个父属性子集的前M 个孩子等价属性子集, 而不需考虑这个父属性子集所有的孩子属性子集. 因此, 节约了大量的运算时间. 在Step3中, 需要为每个Beam 中的属性子集找到M 个孩子等价属性子集, 有时可能会出现找到的孩子等价属性子集数小于M, 这种情况可忽略; 同时, 为了防止属性子集被重复评价, 本文采用与ABB 算法相同的孩子属性子集产生框架处理这个问题[4]: 一个Beam 中的属性子集可以用一个1和0的序列表示(如F 为一个全1序列), 1表示属性子集包含相应的属性, 反之0表示不包含; 在序列中从左往右找到最后一个0作一个标记(F 的标记在最左边); 通过依次删除标记右边的一个属性依次产生孩子属性子集. 注意, 当$M = 1$时, BEMM_MRSC 是一个典型的best-first 启发式算法

BEMM MRSC 属性约简算法的运行时间与两个因素有关: 1) 属性子集互信息的计算; 2) 被评价属性子集的个数. 一个属性子集评价的时间取决于属性子集对U 的划分, 可采用散列法进行划分, 一个属性子集的评价时间复杂度为$O(m)$. 假设属性约简子集的大小为 r, 则被评价的属性子集个数不大于 $0.5 \times M \times (p - r) \times (p - 1 + r) + p + 1$, 所以, BEMM MRSC 的时间复杂度为$O(mMp^2)$. 实际上, 因为通过属性排序和孩子属性子集产生框架减少了多余的属性子集评价, 因此BEMM MRSC 的搜索空间远小于 $0.5 \times M \times (p - r) \times (p - 1 + r) + p + 1$. 当$M = 1$ 时, BEMM MRSC 的时间复杂度为$O(mp)$.

因为 BEMM MRSC 的时间复杂度为$O(mMp^2)$, M 越大, 运行时间越多, 因此必须选择一个合适的Beam 宽度平衡运算时间和属性约简质量 M 的选取原则如下: 从小到大选取M; m 和 p 的初始值越小, M 的取值可以越大

4　实　　验

实验选用 10 个UCI 分类问题数据集对 BEMM MRSC 进行测试. 它们是: Corral, Monk1, Monk3, Parity5 + 5, Parity5 + 2, Vote, Lenses, Zoo, Mushroom, Sonar, M 分别取$2p$, p 和1. ABB 算法用于作为一个比较, 而 F 的互信息作为ABB 的界.

表 1 和表 2 为知识约简的结果, 分别记录了属性约简结果和识别率. 其中识别率的验证采用 10 折交叉验证方法. BEMM MRSC($M = 2p$) 几乎可以得到与ABB 算法相同的结果, 能够找到多个属性约简. 即使在$M = 1$ 的情况下, BEMM MRSC 仍能得到一个好的约简结果. 对于Mushroom, $M = p$ 找到了 15 个属性约简. 注意, $M = p$ 比$M = 1$ 找到更小

表 1　属性约简结果

Dataset	BEMM MRSC($M = 2p$)	BEMM MRSC($M = p$)	BEMM MRSC($M = 1$)	ABB
Corral	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$
Monk1	$\{f_1, f_2, f_5\}$	$\{f_1, f_2, f_5\}$	$\{f_1, f_2, f_5\}$	$\{f_1, f_2, f_5\}$
Monk3	$\{f_2, f_4, f_5\}$	$\{f_2, f_4, f_5\}$	$\{f_2, f_4, f_5\}$	$\{f_2, f_4, f_5\}$
Parity5 + 5	$\{f_2$-$f_4, f_6, f_8\}$	$\{f_2$-$f_4, f_6, f_8\}$	$\{f_2$-$f_4, f_6, f_8\}$	$\{f_2$-$f_4, f_6, f_8\}$
Parity5 + 2	$\{f_3$-$f_7\}^{(1)}$ $\{f_1, f_3$-$f_5, f_7\}^{(2)}$ $\{f_2$-$f_6\}^{(3)}$ $\{f_1$-$f_5\}^{(4)}$	$\{f_3$-$f_7\}^{(1)}$ $\{f_1, f_3$-$f_5, f_7\}^{(2)}$ $\{f_2$-$f_6\}^{(3)}$	$\{f_1$-$f_5\}$	$\{f_1$-$f_5\}^{(1)}$ $\{f_1, f_3$-$f_5, f_7\}^{(2)}$ $\{f_2$-$f_6\}^{(3)}$ $\{f_3$-$f_7\}^{(4)}$
Vote	$\{f_1$-$f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$	$\{f_1$-$f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$	$\{f_1$-$f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$	$\{f_1$-$f_4, f_9, f_{11}, f_{13}, f_{15}, f_{16}\}$
Lenses	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$	$\{f_1$-$f_4\}$
Zoo	$\{f_3, f_4, f_6, f_8, f_{13}\}^{(1)}$ $\{f_3, f_4, f_6, f_9, f_{13}\}^{(2)}$ $\{f_3, f_6, f_8, f_{10}, f_{13}\}^{(3)}$ $\{f_4, f_6, f_8, f_{12}, f_{13}\}^{(4)}$ $\{f_3, f_6, f_8, f_{13}, f_{16}\}^{(5)}$ $\{f_4, f_6, f_9, f_{12}, f_{13}\}^{(6)}$	$\{f_3, f_4, f_6, f_8, f_{13}\}^{(1)}$ $\{f_3, f_4, f_6, f_9, f_{13}\}^{(2)}$ $\{f_3, f_6, f_8, f_{10}, f_{13}\}^{(3)}$ $\{f_4, f_6, f_8, f_{12}, f_{13}\}^{(4)}$	$\{f_3, f_4, f_6, f_8, f_{13}\}$	$\{f_3, f_4, f_6, f_8, f_{13}\}^{(1)}$ $\{f_3, f_4, f_6, f_9, f_{13}\}^{(2)}$ $\{f_3, f_6, f_8, f_{10}, f_{13}\}^{(3)}$ $\{f_4, f_6, f_8, f_{12}, f_{13}\}^{(4)}$ $\{f_3, f_6, f_8, f_{13}, f_{16}\}^{(5)}$ $\{f_4, f_6, f_9, f_{12}, f_{13}\}^{(6)}$ $\{f_3, f_6, f_9, f_{13}, f_{16}\}^{(7)}$
Mushroom	$\{f_5, f_{20}, f_{21}, f_{22}\}^{(1)}$ $\{f_4, f_5, f_{12}, f_{22}\}^{(2)}$	$\{f_5, f_8, f_{12}, f_{19}, f_{20}\}^{(1)}$ $\{f_5, f_8, f_{13}, f_{19}, f_{20}\}^{(2)}$ $\{f_5, f_{12}, f_{19}, f_{20}, f_{21}\}^{(3)}$ $\{f_5, f_{13}, f_{19}, f_{20}, f_{21}\}^{(4)}$ $\{f_5, f_{12}, f_{19}, f_{20}, f_{22}\}^{(5)}$ $\{f_5, f_8, f_{12}, f_{15}, f_{20}\}^{(6)}$ $\{f_5, f_{13}, f_{19}, f_{20}, f_{22}\}^{(7)}$ $\{f_5, f_8, f_{13}, f_{15}, f_{20}\}^{(8)}$ $\{f_5, f_{14}, f_{19}, f_{20}, f_{22}\}^{(9)}$ $\{f_5, f_{15}, f_{19}, f_{20}, f_{22}\}^{(10)}$ $\{f_5, f_8, f_{12}, f_{20}, f_{21}\}^{(11)}$ $\{f_4, f_5, f_8, f_{12}, f_{20}\}^{(12)}$ $\{f_5, f_8, f_{13}, f_{20}, f_{21}\}^{(13)}$ $\{f_4, f_5, f_8, f_{13}, f_{20}\}^{(14)}$ $\{f_5, f_{12}, f_{15}, f_{20}, f_{22}\}^{(15)}$	$\{f_5, f_8, f_{12}, f_{19}, f_{20}\}$	—
Sonar	$\{f_{10}, f_{11}, f_{36}, f_{48}\}$	$\{f_{10}, f_{11}, f_{36}, f_{48}\}$	$\{f_{10}, f_{11}, f_{12}, f_{36}, f_{49}\}$	—

表 2 识别率

Dataset	BEMM MRSC($M = 2p$)	BEMM MRSC($M = p$)	BEMM MRSC($M = 1$)	ABB
Corral	99	99	99	99
Monk1	97	97	97	97
Monk3	100	100	100	100
Parity5 + 5	93	93	93	93
Parity5 + 2	$96^{(1)}, 96^{(2)}, 96^{(3)}, 96^{(4)}$	$96^{(1)}, 96^{(2)}, 96^{(3)}$	96	$96^{(1)}, 96^{(2)}, 96^{(3)}, 96^{(4)}$
Vote	93	93	93	93
Lenses	84	84	84	84
Zoo	$91^{(1)}, 92^{(2)}, 87^{(3)}, 89^{(4)}, 91^{(5)}, 93^{(6)}$	$91^{(1)}, 92^{(2)}, 87^{(3)}, 89^{(4)}$	91	$91^{(1)}, 92^{(2)}, 87^{(3)}, 89^{(4)}, 91^{(5)}, 93^{(6)}, 91^{(7)}$
Mushroom	$99^{(1)}, 96^{(2)}$	$98^{(1)}, 99^{(2)}, 95^{(3)}, 93^{(4)}, 98^{(5)}, 97^{(6)}, 99^{(7)}, 96^{(8)}, 93^{(9)}, 95^{(10)}, 93^{(11)}, 99^{(12)}, 98^{(13)}, 94^{(14)}, 94^{(15)}$	98	-
Sonar	72	72	69	-

表 3 属性约简算法搜索空间大小和运行时间 t ms

Dataset	All	BEMM MRSC($M = 2p$)			BEMM MRSC($M = p$)			BEMM MRSC($M = 1$)			ABB		
		Ev	Ratio	*t*	*Ev*	Ratio	*t*	*Ev*	Ratio	*t*	*Ev*	Ratio	*t*
Corral	2^6	12	0.188	2	12	0.188	2	7	0.109	2	12	0.188	3
Monk1	2^6	24	0.375	13	24	0.375	13	7	0.109	4	20	0.313	19
Monk3	2^6	32	0.500	14	32	0.500	14	7	0.109	4	20	0.313	19
Parity5 + 5	2^{10}	50	0.049	385	50	0.049	385	11	0.011	50	112	0.109	406
Parity5 + 2	2^{10}	88	0.086	403	74	0.072	397	11	0.011	49	228	0.223	650
Vote	2^{16}	458	0.007	985	330	0.005	765	17	0.000	42	908	0.014	2 697
Lenses	2^4	5	0.313	1	5	0.313	1	5	0.313	1	5	0.313	1
Zoo	2^{16}	1 790	0.027	487	1 088	0.017	375	17	0.000	5	25 344	0.387	132 456
Mushroom	2^{22}	6 893	0.000	659 219	3 762	0.000	369 640	23	0.000	2 389	-	-	-
Sonar	2^{60}	178 883	0.000	1 494 828	92 538	0.000	317 828	61	0.000	104	-	-	-

注: All 为整个搜索空间的大小; *Ev* 为搜索空间大小; Ratio 为搜索空间压缩率, 等于 *Ev* 除以 All

更多的约简, 而$M = 2p$ 比$M = p$ 有更好的约简结果, 说明M 越大, 约简结果越好. 从表 3 可以看出, 属性约简的结果能够很好地进行值约简, 得到较高的识别率. 注意, 由于$p > 20$, ABB 算法运行时间大于 12 h, 认为对于 Mushroom 和 Sonar 不适用.

表 3 为以上属性约简算法中评价的属性子集数(搜索空间大小) 和运行时间. 从表 3 可以看出, BEMM MRSC 算法能够在较短时间内完成属性约简. 注意,M 越小, 搜索空间和运行时间越小.

5 结 论

通过从互信息角度分析, 粗糙集理论的属性约简问题的实质就是要找到尽量小的初始属性集 F 的等价属性子集. 在定义和分析了属性子集冗余协同系数的基础上, 提出了基于 Beam 搜索的启发式粗糙集属性约简算法—BEMM MRSC 算法, 与一般的 best-first 算法相比, 它可以找到多个属性约简, 这样可以得到更多的规则, 因而可提高识别率. 实验采用了 10 个标准的 UCI 数据集, 从 4 个方面(约简、识别率、搜索空间大小和运行时间) 对 BEMM MRSC 算法进行测试, 属性集大小从 4～60, 样本集大小从 24～8 124. 实验表明, BEMM MRSC 粗糙集属性约简算法能够得到高质量的解(近似于最优解), 特别是对于大属性集.

(下转第 1217 页)

般次梯度算法(表中 * 号处), 这是因为迭代步长的影响. 比较式(9)和式(15), 不难发现, 模糊次梯度在迭代步长 t_k 中引入了变量 a, 这可能造成步长的减小, 从而影响了个别情况下算法的短期收敛速度. 但这种情况并不多见(15 组数据中只出现 3 次), 综合起来, 本文提出的模糊次梯度算法在计算时间几乎不变的情况下, 收敛效果要明显优于一般次梯度算法.

5　结　　论

本文提出的模糊次梯度算法利用一个隶属度函数给出了历史次梯度的权重系数, 从而简单有效地利用了历史次梯度的信息. 仿真实验表明, 与传统的次梯度算法相比, 该算法在基本不增加计算时间的情况下, 可以更有效地抑制振荡现象, 其收敛效果明显优于一般的次梯度算法.

参考文献(References):

[1] Naum Z Shor. *Nondifferentiable Optimization and Polynomial Problems* [M]. Boston: Boston Kluwer, 1998.

[2] Kiwiel K C. An aggregate subgradient method for nonsmooth convex minimization [J]. *Mathematical Programming*, 1983, 27(3): 320-341.

[3] Camerini P M, Fratta L, Maffioli F. On improving relaxation methods by modified gradient techniques [J]. *Mathematical Programming Study*, 1975, 3: 26-34.

[4] Kim S, Ahn H. Convergence of a generalized subgradient method for nondifferentiable convex optimization [J]. *Mathematical Programming*, 1991, 50(1): 75-80.

[5] Francesca F. A modified subgradient algorithm for Lagrangean relaxation [J]. *Computers and Operations Research*, 2000, 28(1): 33-52.

[6] Zhao X, Luh P B. New bundle methods for solving Lagrangian relaxation dual problems [J]. *J of Optimzation Theory and Applications*, 2002, 113(2): 373-397.

[7] Kiwiel K C. The efficiency of subgradient projection methods for convex optimization, part I: General level methods [J]. *SIAM J Control and Optimization*, 1996, 34(2): 660-676.

(上接第 1212 页)

参考文献(References):

[1] Pawlak Z. *Rough Sets: Theoretical Aspects of Reasoning about Data* [M]. Amsterdam: Kluwer Academic Publishers, 1991.

[2] Bazan J. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision system [A]. *Rough Sets in Knowledge Discovery* [C]. Herdelberg: Phyisca-Verlag, 1998. 321-365.

[3] Narendra P, Fukunaga K. A branch and bound algorithm for feature subset selection [J]. *IEEE Trans on Computer*, 1977, 26 (9): 917-922.

[4] Liu H, Motoda H. *Feature Selection for Knowledge Discovery and Data Mining* [M]. Boston: Kluwer Academic Press, 1998.

[5] 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.

[6] Lin M. *Software System for Intelligent Data Processing and Discovering Based on the Fuzzy-Rough Sets Theory* [M]. San Diego: San Diego University, 1995.

[7] 唐建国, 谭明术. 粗糙集理论中的求核与约简[J]. 控制与决策, 2003, 18(4): 449-452.
(Tang J G, Tan M S. On finding core and reduction in rough set theory [J]. *Control and Decision*, 2003, 18 (4): 449-452.)

[8] 苗多谦, 王钰. 粗糙集理论中概念与运算的信息表示[J]. 软件学报, 1999, 10(2): 113-116.
(Miao D Q, Wang J. An Information representation of the concepts and operations in rough set theory [J]. *J of Software*, 1999, 10(2): 113-116.)

[9] 王国胤, 于洪, 杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7): 759-766.
(Wang G Y, Yu H, Yang D C. Decision table reduction based on conditional information entropy [J]. *Chinese J of Computers*, 2002, 25(7): 759-766.)

[10] Aha D, Bankert R. A comparative evaluation of sequential feature selection algorithms [A]. *5th Int Workshop on Artificial Intelligence and Statistics* [C]. New York: Springer, 1991.

[11] Cover T M. *Elements of Information Theory* [M]. New York: Wiley, 1991.

[12] Brenner N, Strong S P, Koberle R, et al. Synergy in a neural code [J]. *Neural Computation*, 2000, 13(7): 1531-1552.

[13] Yaglom A M, Yaglom IM. *Probability and Information* [M]. Hardbound: D Reidel Publishing Company, 1983.