

属性约简算法 CARRDG 的改进及其实现技术研究

陈可赢^{1,2}, 曾文华¹, 施明辉³

CHEN Ke-ying^{1,2} ZENG Wen-hua¹ SHI Ming-hui³

1. 厦门大学 软件学院 福建 厦门 361005

2. 厦门华夏学院 福建 厦门 361005

3. 厦门大学 智能科学与技术系 福建 厦门 361005

1. School of Software, Xiamen University, Xiamen, Fujian 361005, China

2. Xiamen Huaxia College, Xiamen, Fujian 361005, China

3. Department of Cognition Science, Xiamen University, Xiamen, Fujian 361005, China

E-mail: smh@xmu.edu.cn

CHEN Ke-ying ZENG Wen-hua SHI Ming-hui. Research on technologies for improvement and implementation of attribute reduct algorithm CARRDG. Computer Engineering and Applications 2008 44(36) :160-163.

Abstract : The attribute reduct algorithm CARRDG, which can efficiently compute total attribute reducts for large scale information system has been proposed recently. Although it has rigorous theoretical foundation further improvement may be possible especially in the implementation level. In addition to the existed three heuristic deep-first searching principles (Member Executive Principle MEP, Friend Persuade Principle FPP, Stranger Enter Principle SEP) based on reduct discernibility graph, a new heuristic searching principle——Blocking Layer Block Principle (BLBP) has been proposed in this paper to improve the searching efficiency. Since the reasonable data structures have been developed, BLBP will not increase the implementing complexity of the algorithm. In contrast, the experimental results by using UCI data show that BLBP exceeds MEP and FPP in trimming efficiency for some large information systems.

Key words : attribute reduct, reduct discernibility graph, data mining, knowledge discovery

摘要 : 属性约简算法 CARRDG 是近来提出的能计算大型信息系统中所有属性约简的高效算法。针对属性约简算法 CARRDG 在实现技术层面的可改进之处, 在原有的三种约简分辨图深度优先搜索原则(成员独占原则、友人劝阻原则、陌生人吸纳原则)的基础上, 增加新的深度优先搜索原则——阻挡层阻挡原则。由于采用了恰当的数据结构和实现技术, 使得增加阻挡层阻挡原则不会增加原算法的程序实现的复杂性, 也几乎不会增加程序的运行时间。相反, UCI 数据实验结果表明, 阻挡层阻挡原则对于某些大型信息系统的约简分辨图的剪枝效率超过了成员独占原则与友人劝阻原则。

关键词 : 属性约简, 约简分辨图, 数据挖掘, 知识发现

DOI : 10.3778/j.issn.1002-8331.2008.36.045 **文章编号 :** 1002-8331(2008)36-0160-04 **文献标识码 :** A **中图分类号 :** TP182

1 引言

从海量数据中获取信息的关键技术之一在于消除原始数据中的冗余信息。属性约简为解决该问题提供了有效途径。其目的是在保持已有数据信息分类能力不变的前提下, 消除冗余属性, 使高维数据降为低维数据, 从而降低信息处理的难度与复杂性。因此, 属性约简在数据挖掘、机器学习、知识发现、决策支持等领域具有重要意义。

找到一个信息系统的所有最小属性约简是 NP-hard 问题^[1]。目前, 国内外学者利用启发式信息, 已提出了许多的相关算法。这些算法可分为^[2]: 基于分辨矩阵(Discernibility Matrix)的方法^[3-6]、基于二进制分辨矩阵的方法^[7]、基于正区域的方法^[8]、基于

信息熵的方法^[9-10]、基于遗传算法的方法^[11]等。这些算法有的不完备, 有的只能找到一个或少数几个最小属性约简, 有的不能应用于实际的大型信息系统。最近, 文献[2]提出了对于现实中的大型信息系统, 能高效地计算所有属性约简的算法 CARRDG, 并从理论上严密论证了算法 CARRDG 的正确性、完全性与高效性。

然而, 算法 CARRDG 仍有可改进之处。本文首先阐述了改进型算法 CARRDG 的思想, 然后介绍了约简分辨图的数据结构的设计与实现技术, 最后给出了六种典型的 UCI 机器学习数据(ftp://ftp.ics.uci.edu/pub/machine-learning-databases/)的实验结果与分析。

基金项目: 国家 985 工程基金资助项目, 厦门大学科技创新项目(No.XDKJ CX20063011)。

作者简介: 陈可赢(1976-), 女, 实验师, 研究方向: 软件工程, 信息管理系统, 软计算; 曾文华, 男, 教授, 博士生导师, 研究方向: 嵌入式系统, 神经网络, 并行演化计算; 施明辉, 男, 博士, 讲师, 研究方向: 神经网络, 模糊理论, 粗集理论。

收稿日期: 2008-08-25 修回日期: 2008-10-27

2 属性约简算法 CARRDG 及其改进

2.1 基本概念

(1) 信息系统

在 Rough 集理论中, 信息系统 (Information System, IS) 定义为 $IS=(U, A, V, f)$, 其中 $U=\{u_1, u_2, \dots, u_n\}$ 是对象的集合; A 是属性的集合; $V=\{V_{a_1}, V_{a_2}, \dots, V_{a_m}\}$, V_{a_i} 表示属性 a_i 的值域; f 是信息函数: $f: U \times A \rightarrow V$ 。

(2) 决策系统

若信息系统的属性集 A 可分为条件属性集 C 和决策属性集 D , 即 $C \cup D = A$, $C \cap D = \emptyset$, 则该信息系统称为决策系统 (Decision System, DS)。一般决策属性集 D 中仅含一个决策属性 d , 若含多个决策属性, 则可转化为一个决策属性 $d^{[4]}$ 。此时, 记 $DS=(U, C \cup d, V, f)$, 其中 $U=\{u_1, u_2, \dots, u_n\}$, $C=\{a_1, a_2, \dots, a_m\}$ 。

(3) 分辨属性、分辨属性集、分辨属性簇^[2]

设决策系统 $DS=(U, C \cup d, V, f)$, 其中 $U=\{u_1, u_2, \dots, u_n\}$, $C=\{a_1, a_2, \dots, a_m\}$, 条件属性 $a_k \in C$ 称 a_k 为分辨属性 (Discernibility Attribute, DA)。若存在 $u_i \in U, u_j \in U, u_i \neq u_j$, 使得 $f(u_i, a_k) \neq f(u_j, a_k)$, $f(u_i, d) \neq f(u_j, d)$ 。特别地, 称 a_k 为关于对象 u_i, u_j 的分辨属性。关于对象 u_i, u_j 的分辨属性集 $DAS(u_i, u_j)$, 定义为:

$$DAS(u_i, u_j) = \begin{cases} \phi & f(u_i, d) = f(u_j, d) \\ \{a_k \mid a_k \in C, f(u_i, a_k) \neq f(u_j, a_k), f(u_i, d) \neq f(u_j, d)\} & \text{其他} \end{cases} \quad (1)$$

关于 DS 的分辨属性簇 (Discernibility Attribute Set Family, DASF) 记为 $F(DS)$,

$$F(DS) = \{DAS(u_i, u_j) \mid 1 \leq i \leq n-1, i+1 \leq j \leq n\}$$

设 $DS=(U, C \cup d, V, f)$, $F_i \in F(DS)$, 若存在 $F_j \in F(DS)$, 使得 $F_j \subset F_i$, 则称 F_i 是可被吸收的 (分辨属性集), 特别地称 F_i 是可被 F_j 吸收的 (分辨属性集)。

(4) 约简分辨属性簇^[2]

设 $DS=(U, C \cup d, V, f)$, 从 $F(DS)$ 中删去所有可被吸收的分辨属性集后, 得到的集合称为约简分辨属性簇 (Reduct Discernibility Attribute Set Family, RDASF), 记为 $RF(DS)$ 。

(5) 分辨子图^[2]

设 $DS=(U, C \cup d, V, f)$, $F(DS)=\{F_1, F_2, \dots, F_l\}$, $F_i \in F(DS)$, $F_j \in F(DS)$, $i \neq j$, $F_i=\{a_{i1}, a_{i2}, \dots, a_{is}\}$, $F_j=\{a_{j1}, a_{j2}, \dots, a_{jt}\}$, 则由 F_i, F_j 产生的分辨子图 (Sub Discernibility Graph, SDG) 是一个有向完全二部有序图, 记为 $SDG(F_i, F_j)$, 如图 1 所示。

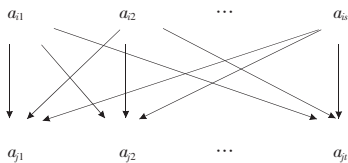


图 1 分辨子图

(6) 约简分辨子图^[2]

设 $DS=(U, C \cup d, V, f)$, $RF(DS)=\{F_1, F_2, \dots, F_h\}$, $F_i \in RF(DS)$, $F_j \in RF(DS)$, $i \neq j$, $F_i=\{a_{i1}, a_{i2}, \dots, a_{is}\}$, $F_j=\{a_{j1}, a_{j2}, \dots, a_{jt}\}$, 则由 F_i, F_j 产生的约简分辨子图 (Reduct Sub Discernibility Graph, RSDG) 定义为 $RSDG(F_i, F_j)=(V(RSDG), E(RSDG), \mu(RSDG))$, 简记 $RSDG(F_i, F_j)=(V, E, \Psi)$, 其中 V, E, Ψ 的定义与分辨子图类似。

(7) 约简分辨图^[2]

设 $DS=(U, C \cup d, V, f)$, $RF(DS)=\{F_1, F_2, \dots, F_h\}$, 关于 DS 的约简分辨图 (Reduct Discernibility Graph, RDG) $RDG(DS)$ 定义

为 $RDG(DS)=\bigcup RSDG(F_i, F_j)$, 其中 $F_i \in RF(DS)$, $F_j \in RF(DS)$,

$1 \leq i \leq h-1, j=i+1$; 记 $RDG(DS)=(V(RDG), E(RDG), \mu(RDG))$, 简记 $DG(F_i, F_j)=(V, E, \Psi)$ 。

2.2 属性约简算法 CARRDG 的改进

文献[2]给出了基于约简分辨图的属性约简算法 CARRDG, 其基本思想是: 在对于给定的决策系统建立约简分辨图之后, 采用合适的启发式深度优先搜索原则, 获取约简完整路径, 然后将约简完整路径转化为属性约简。

文献[2]已证明, 在约简分辨图中可采用三种启发式深度优先搜索原则计算属性约简。这三种启发式原则是成员独占原则、友人劝阻原则和陌生人吸纳原则。

(1) 成员独占原则 (Member Executive Principle, MEP)

在 RDG 的深度优先搜索过程中, 设 P_{i-1} 是一条未完成路径, 若 $Sons(P_{i-1}) \cap MemberSet(P_{i-1}) \neq \emptyset$, 则 P_{i-1} 必经顶点 v_{ik} 继续延伸, 其中顶点 v_{ik} 满足 $v_{ik} = FirstMember(P_{i-1}, k)$ 。

(2) 友人劝阻原则 (Friend Persuade Principle)

在 RDG 的深度优先搜索过程中, 设 P_{i-1} 是一条未完成路径, 若 $Sons(P_{i-1}) \cap MemberSet(P_{i-1}) = \emptyset$, 且顶点 v_{ik} 满足 $v_{ik} \in Sons(P_{i-1}) \cap FriendSet(P_{i-1})$, 则 P_{i-1} 不再经顶点 v_{ik} 继续延伸。

(3) 陌生人吸纳原则 (Stranger Enter Principle, SEP)

在 RDG 的深度优先搜索过程中, 设 P_{i-1} 是一条未完成路径, 若 $Sons(P_{i-1}) \cap MemberSet(P_{i-1}) = \emptyset$, 且顶点 v_{ik} 满足 $v_{ik} \in Sons(P_{i-1}) \wedge v_{ik} \notin FriendSet(P_{i-1})$, 则称此顶点 v_{ik} 为 P_{i-1} 的陌生人 (顶点)。陌生人吸纳原则 (SEP) 规定: 此时 P_{i-1} 可以经陌生人 v_{ik} 继续延伸。

从实现技术层面上看, 文献[2]的算法仍有可改进之处。

为此, 首先定义阻挡层的概念。在约简分辨图中, 若某一层中的所有节点都是某搜索路径的友人节点, 则该层称为此搜索路径的阻挡层。

根据友人劝阻原则, 显然, 阻挡层是搜索路径无法逾越的层。基于此性质, 可增加一条启发式搜索原则: 阻挡层阻挡原则。

(4) 阻挡层阻挡原则 (Blocking Layer Block Principle, BLBP)

对于某搜索路径, 在约简分辨图的深度优先搜索过程中, 若在其搜索方向的前方存在阻挡层, 则此搜索路径不必继续延伸。

改进的具体办法是: 只须改动原算法的过程 Explore 中关于陌生人节点的处理 (这是因为, 成员节点与友人节点都不会对搜索路径的友人集产生影响), 当搜索路径经过一个非叶子的陌生人节点之后, 若不存在新增友人, 则继续延伸; 若存在新增友人, 则判断搜索方向的前方是否存在阻挡层, 若存在, 则此搜索路径不必继续延伸, 否则此搜索路径继续延伸。

改进后的过程 Explore 如下所示:

Procedure: Explore // 改进的 Explore

Input:

RDG	// 约简分辨图
P	// 初始路径
$curlayer$	// 路径 P 延伸的当前层层号
$curvertex$	// 路径 P 延伸的当前顶点
$SearchResult$	// 初始结果

Begin

$S \Leftarrow MemberSet(P) \cap VertsInLayer(curlayer)$;

If $S \neq \emptyset$ then

```

{
    P := P + FirstMember(P, curlayer);
    If Sons(P) = φ then
        AddToResult(P); //将路径 P 添加到 SearchResult 中
    Else
    {
        curlayer := curlayer + 1;
        curvertex := HeadInLayer(curlayer);
        Explore (RDG, P, curlayer, curvertex, SearchResult); //继续搜索
    }
}
Else if curvertex ∈ FriendSet(P) then
{
    If R = Brother(curvertex) ≠ φ then
    {
        curvertex := DR = Brother(curvertex);
        Explore (RDG, P, curlayer, curvertex, SearchResult); //继续搜索
    }
}
Else
{
    P := P + curvertex;
    If Sons(P) = φ then
        AddToResult(P); //将路径 P 添加到 SearchResult 中
    Else
    {
        UpdateFriendSet(P); //更新路径 P 的友人集
        If HasBlockLayer(P) return; //若存在路径 P 的阻挡层, 返回
        curlayer := curlayer + 1;
        curvertex := HeadInLayer(curlayer);
        Explore (RDG, P, curlayer, curvertex, SearchResult); //继续搜索
    }
}
End

```

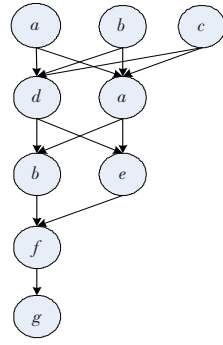


图2 约简分辨图逻辑结构示例(1)

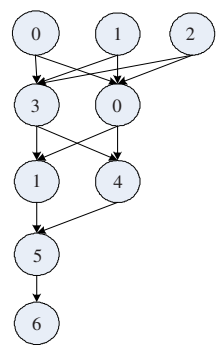


图3 约简分辨图逻辑结构示例(2)

3 约简分辨图的数据结构设计及实现

3.1 约简分辨图的逻辑结构设计

约简分辨图数据结构的设计是实现属性约简算法 CAR-RDG 的关键, 需要综合考虑如何有利于实现各启发式搜索原则。约简分辨图的形式化描述已在第 2.1 节介绍, 此处从直观上解释约简分辨图的逻辑结构。

约简分辨图是一个有向图。其中, 每一个节点代表一个分辨属性; 上一层的节点可以到达其下一层的每一个节点, 每一层节点所代表的属性构成一个分辨属性集。例如, 假设决策系统中, 条件属性集为 {a, b, c, d, e, f, g}, 则决策系统的约简分辨图类似于图 2。在程序中, 属性用其序号表示。若 a, b, c, d, e, f, g 的序号分别为 0, 1, 2, 3, 4, 5, 6, 则图 2 可表示为图 3。

3.2 约简分辨图的数据结构设计

通常用于表示图的邻接矩阵和邻接链表数据结构并不适合于算法 CARRDG。为此, 针对算法 CARRDG 的搜索特点, 设计如图 4 所示的数据结构。图 4 示意了图 3 所示的约简分辨图

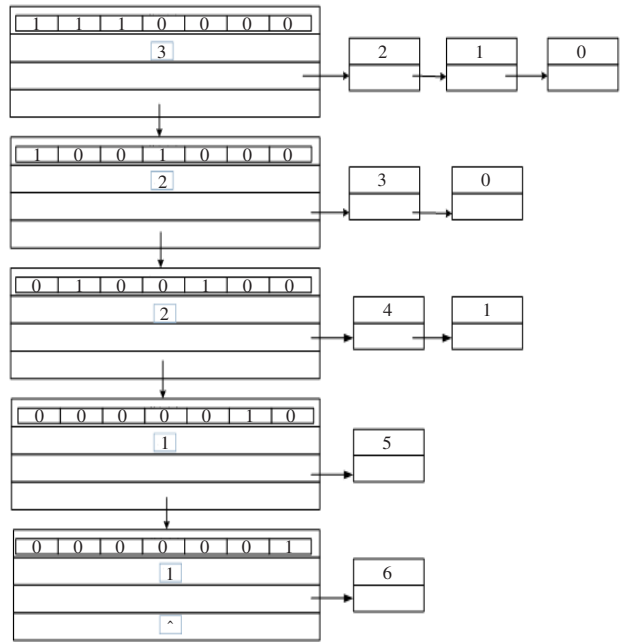


图4 约简分辨图数据结构示例

的逻辑结构的实现方法。图 4 中最左边一列称为主干(或主干链表), 主干中的节点称为主干节点(其成员的含义在 3.3 节中介绍), 按其容量由大到小排序; 每一层中, 在主干节点右边的节点称为分支节点(或分辨属性节点), 每一层中的分支节点构成一个链表, 称为分支(或分支链表), 分支链表的节点按属性序号由大到小排序。

3.3 约简分辨图的数据结构实现

(1) 主干节点

主干节点的结构体定义如图 5 所示。其中, CATTCOUNT 是宏名, 表示条件属性的个数, 数组 arrAttsInLayer[CATTCOUNT] 表达了该层的分辨属性的集合, 若 arrAttsInLayer[i]=1, 则表示

```

typedef struct AttsInLayerCodeNode /* 主干节点 */
{
    int arrAttsInLayer[CATTCOUNT]; /* 该层的分辨属性集数组 */
    int intCapacity; /* 该层中含属性个数——容量 */
    ATTCODENODE * ptrFirstAttInLayerCode; /* 分支 */
    struct AttsInLayerCodeNode * ptrSon; /* 下一层 */
} ATTSINLAYERCODENODE;

```

图5 主干节点的结构体定义

序号为 i 的属性属于该层的分辨属性的集合,否则,若 $\text{arrAttsInLayer}[i]=0$,则表示序号为 i 的属性不属于该层的分辨属性的集合; intCapacity 指示了该层中包含的属性的个数,称为容量;指针 $\text{ptrFirstAttInLayerCode}$ 指向该层的分支节点;指针 ptrSon 指向下一层的主干节点。

(2)分支节点

分支节点的结构体定义如图 6 所示。其中 intAttNum 表示该分支节点所代表的属性的序号;指针 ptrBrother 指向同一层的下一个分支节点。

```
typedef struct AttCodeNode /* 分支节点 */
{
    int intAttNum; /* 属性序号,从 0 开始计 */
    struct AttCodeNode * ptrBrother;
}ATTCODENODE;
```

图 6 分支节点的结构体定义

(3)约简分辨属性集

分辨属性集由每一层的分支链表实现,如图 4 所示。其中,分支链表中的分支节点按属性序号(即 intAttNum)由大到小排序。

(4)约简分辨属性集簇

约简分辨属性集簇由主干链表实现,如图 4 所示。其中,主干链表中的主干节点按容量(即 intCapacity)由大到小排序。

4 实验结果与分析

由于连续型带缺失值的系统,可以用不同的方法转化为离散型无缺失值的系统^[12],故采用 UCI 机器学习数据库中,六种典型的离散型无缺失值的数据(参见表 1)进行实验。这些数据集的说明信息和数据可从 UCI 机器学习数据库中找到。

4.1 实验数据基本信息及实验基本结果

实验数据基本信息及实验基本结果见表 1。

4.2 三原则剪枝数占总的剪枝数的比例情况

三原则剪枝数占总的剪枝数的比例情况见表 2。表 2 中数据 2 的剪枝比例为 0,是因为对于数据 2,仅存在一条路径。从表 2 中可以看出,多数情况下,成员独占原则对剪枝的贡献最大,但有时阻挡层阻挡原则也起了很大作用。表中加了下划线的数据都表明阻挡层阻挡原则的剪枝数占总的剪枝数的比例大于 30%,其中对于数据 3 采用 MEP/A-FPP 方案,其比例达到 100%,意味着,此时三项原则中,仅有阻挡层阻挡原则起到剪枝的作用。由此可见,对于一些大型信息系统,阻挡层阻挡原则可以显著提高属性约简算法 CARRDG 的搜索效率。

5 结束语

属性约简算法 CARRDG 是近来提出的能计算大型信息系统中所有属性约简的高效算法^[2]。本文针对属性约简算法 CARRDG 在实现技术层面上的可改进之处,在原有的三项约简分辨图深度优先搜索原则(成员独占原则、友人劝阻原则、陌生人吸纳原则)的基础上,增加了新的深度优先搜索原则——阻挡层阻挡原则。由于采用了恰当的数据结构和实现技术,使得增加阻挡层阻挡原则不会增加原算法的程序实现的复杂性,也几乎不会增加程序的运行时间。相反,UCI 数据实验结果表明,阻挡层阻挡原则对于某些大型信息系统的约简分辨图的剪枝效率超过了成员独占原则与友人劝阻原则。

论文阐述的属性约简算法 CARRDG 的实现程序具有通用性与实用性。一方面,对于大多数现实中的信息系统,只要符合基本格式要求,该程序都能快速地计算出所有的属性约简。另一方面,属性约简算法 CARRDG 本质上解决的是数学领域中析取范式与或取范式的相互转换问题^[2],因此,该程序也可以解决这类数学问题,从而具有广阔的应用领域与前景。

表 1 实验数据基本信息及实验基本结果

	数据 1	数据 2	数据 3	数据 4	数据 5	数据 6
数据名称	Zoo	Car	Tic-tac-toe	Soybean-small	Letter-recognition	Pocker-hand-training
对象数	101	1 728	958	47	20 000	25 010
属性数(包括决策属性)	17	7	10	36	17	11
分辨图的层数	14	6	36	99	65	9
总的路径数	1.17×10^7	1	6.87×10^{10}	1.80×10^6	7.27×10^{29}	144
核属性数	2	6	0	0	3	5
最小属性约简数	7	1	9	4	11	8
最小属性约简含属性数	5	6	8	2	11	7
属性约简总数	33	1	9	765	61	8
核属性	6,13	1,2,3,4,5,6	无	无	4,15,8	2,4,6,8,10

表 2 三原则剪枝数占总的剪枝数的比例表

		数据 1	数据 2	数据 3	数据 4	数据 5	数据 6
MEP/A-FPP	成员独占原则	45.304 6%	0	0	85.738 9%	35.993 6%	70.714 3%
	友人劝阻原则	24.162 0%	0	0	10.679 1%	0	29.285 7%
	阻挡层阻挡原则	<u>30.533 3%</u>	0	<u>100%</u>	3.582 0%	<u>64.006 4%</u>	0
MEP/R-FPP	成员独占原则	89.839 6%	0	23.516 8%	93.928 4%	67.702 5%	64.705 9%
	友人劝阻原则	8.539 6%	0	1.433 6%	5.912 3%	19.612 0%	35.294 1%
	阻挡层阻挡原则	1.620 8%	0	<u>75.049 6%</u>	0.159 3%	12.685 5%	0
MEP/L-FPP	成员独占原则	68.246 8%	0	29.666 0%	91.109 4%	58.433 3%	90.441 2%
	友人劝阻原则	31.685 8%	0	7.675 3%	8.379 1%	1.405 6%	9.558 8%
	阻挡层阻挡原则	0.067 4%	0	<u>62.658 7%</u>	0.511 5%	<u>40.161 0%</u>	0

的项严格按照支持度降序的方式进行排列, 并且保证不影响头表的随机存取性, 算法的效率将得到进一步的提升。这也是今后可能的研究方向。

参考文献:

- [1] Agrawal R, Srikant R. Fast algorithms for mining association rules[C]// Proc Int'l Conf Very Large Data Bases, Sept 1994: 487-499.
- [2] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]// Proc ACM-SIGMOD Int'l Conf Management of Data, May 2000: 1-12.
- [3] Zaki M J. Scalable algorithms for association mining[J]. IEEE Trans Knowledge and Data Mining, 2000, 12(3): 372-390.
- [4] Zou Q, Chu W, Lu B. SmartMiner: A depth first algorithm guided by tail information for mining maximal frequent itemsets[C]// Proc IEEE Int'l Conf Data Mining, Dec 2002.
- [5] Gopalan R, Sucahyo Y. High performance frequent patterns extraction using compressed FP-Tree[C]// Proceedings of the SIAM International Workshop on High Performance and Distributed Mining, Orlando, USA, 2004.
- [6] Grahne G, Zhu Jianfei. Fast algorithms for frequent itemset mining using FP-Trees[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(10): 1347-1362.
- [7] Orlando S, Lucchese C, Palmerini P, et al. kDCI: A multi-strategy algorithm for mining frequent sets[C]// Proc IEEE ICDM Workshop Frequent Itemset Mining Implementations, CEUR Workshop Proc, Nov 2003: 80.
- [8] Pietracaprina A, Zandolin D. Mining frequent itemsets using Patricia tries[C]// Proc IEEE ICDM Workshop Frequent Itemset Mining Implementations, CEUR Workshop Proc, Nov 2003: 80.
- [9] Burdick D, Calimlim M, Gehrke J. MAFIA: A maximal frequent itemset algorithm for transactional databases [C]// Proc Int'l Conf Data Eng, Apr 2001: 443-452.
- [10] 严蔚敏, 吴伟民. 数据结构(C语言版)[M]. 北京: 清华大学出版社, 2002.
- [11] 韩家炜, 坎伯. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2005.
- [12] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large database[C]// Proc ACM SIGMOD Int'l Conf Management of Data, May 1993: 207-216.

(上接 33 页)

到, Yang X P 在极小化公理集的过程中所忽略的一点: 在模糊粗糙近似算子公理化体系中, 上、下近似算子满足对偶性情况下, 其最小化公理集也可以是下面两种情况: 对于任意 $\alpha \in [0, 1]$, $A \in F_L(U)$,

$$(1) K(A) = \sim H(\sim A), \hat{\alpha} \wedge H(A) = H(\hat{\alpha} \wedge A), H(A) \vee H(B) = H(A \vee B);$$

$$(2) H(A) = \sim K(\sim A), \hat{\alpha} \vee K(A) = K(\hat{\alpha} \vee A), K(A) \wedge K(B) = K(A \wedge B).$$

参考文献:

- [1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Science, 1982, 11: 341-356.
- [2] Wu W Z, Zhang W X. Constructive and axiomatic approaches of fuzzy rough approximation operators[J]. Information Science, 2004,

159: 233-254.

- [3] Morsi N N, Yakout M M. Axiomatics for fuzzy rough sets[J]. Fuzzy Sets and Systems, 1998, 100: 327-342.
- [4] Yang X P, Li T J. The minimization of axiom sets characterizing generalized approximation operators[J]. Information Science, 2006, 176: 887-899.
- [5] Yang X P. Minimization of axiom sets on fuzzy approximation operator[J]. Information Science, 2007, 177: 3840-3854.
- [6] Pavelka J. On fuzzy logic I: Many-valued rules of inference, II: Enriched residuated lattices and semantics of propositional calculi, III: Semantical completeness of some many-valued propositional calculi[J]. Zeitschr F Math Logik and Grundlagend Math, 1979, 25: 45-52, 119-134, 447-464.
- [7] Wu Zhengjiang, Qin Keyun. L-fuzzy rough set based on complete residuated lattice[J]. The Journal of Southwest Jiaotong University: English Edition, 2008, 16(1): 95-98.

(上接 163 页)

参考文献:

- [1] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of Polish Academy of Sciences, 1985, 33(11/12): 693-696.
- [2] 施明辉. 面向中医辨证计算的粗糙集知识获取及其应用研究[D]. 厦门大学, 2008.
- [3] Shi M H, Zhou C L. Approach to knowledge discovery for capturing experts' reasoning based on rough set theory[J]. Journal of Computational Information Systems, 2005(1): 887-894.
- [4] Walczak B, Massart D L. Rough sets theory[J]. Chemometrics and Intelligent Laboratory Systems, 1999, 47: 1-16.
- [5] 刘银山, 吴孟达, 王丹. 粗糙集中求取所有最小属性约简快速算法[J]. 计算机工程与科学, 2007, 29(1): 97-101.
- [6] 支天云, 苗夺谦. 二进制可辨矩阵的变换及高效属性约简算法的构造[J]. 计算机科学, 2002, 29(2): 140-142.
- [7] 黄龙军, 章志明, 周才英, 等. 一种基于布尔矩阵的属性约简方法[J]. 计算机工程与应用, 2006, 42(9): 160-161.
- [8] 刘少辉, 盛秋骥, 吴斌, 等. 粗糙集高效算法的研究[J]. 计算机学报, 2003, 26(5): 524-529.
- [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.
- [10] 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.
- [11] Woblewski J. Finding minimal reducts using genetic algorithms[C]// Wang P P. Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS 95), Wrightsville Beach, North Carolina, USA, 1995: 186-189.
- [12] Lin T Y, Cercone N. Rough sets and data mining—analysis of imperfect data[M]. Boston, USA: Kluwer Academic Publishers, 1997.