

基于熵的不相容规则修正算法

朱颢东 钟 勇

(中国科学院成都计算机应用研究所 成都 610041) (中国科学院研究生院 北京 100039)

摘 要 不相容规则可能是信息系统中的异常信息,在这个信息时代,这些异常信息与那些“正常”信息同样重要。在总结了许多不相容规则处理算法的不足之后,提出了一种基于熵的不相容规则的修正算法。根据设定的可信度值,使用这个算法可以判断出一个信息系统中不相容规则是否是异常规则。最后结合一个实例阐明了该算法的基本思路。

关键词 不相容规则,异常信息,熵,异常规则

中图法分类号 TP301 文献标识码 A

Incompatible Rules Correcting Algorithm Based on Entropy

ZHU Hao-dong ZHONG Yong

(Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China)

(The Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

Abstract Exceptional rules may be exceptional information of information systems, these exceptional information is as important as those of normal information in modern times. This paper summed up much shortcoming of many algorithms of processing incompatible rules and presented a correcting algorithm of incompatible rules based on entropy. According to confidence which the paper assigned, it can judge whether rules of information system are exceptional rules. An example was given to demonstrate the main idea of this algorithm.

Keywords Incompatible rules, Exceptional information, Entropy, Exceptional rules

1 引言

近年来,随着粗糙集理论在各领域的广泛应用,基于等价关系的经典粗糙集理论也得到了迅速发展,原来的等价关系也放宽到了相容关系(自反性、对称性)、相似关系(自反性、传递性),甚至于一般二元关系(一般要求满足自反性)等中,这些都极大地丰富了粗糙集理论的内涵,为粗糙集理论的应用拓宽了道路。虽然粗糙集理论在许多方面取得了令人鼓舞的成果,但也存在一些问题,如应用粗糙集理论来推理决策规则时,要求提供的决策数据是完备的;对于数据信息不完备的情况,可将其视为完备的,或做某种形式的处理后再进行推理决策。这样虽然简单,但常常出现不相容规则问题。对于不相容规则问题,管等人在文献[1]提出了基于最大相容类的确定性决策规则获取,但该文献的不足之处在于只提取相容规则而忽略了不相容规则;朱等人虽然在文献[2]对不相容规则作了一些处理,但是却删除了那些分类错误较大的不相容规则,这就忽略了这些异常规则所具有的异常信息,而且该文献不适合规则库较小的系统以及不完备信息系统,同时还存在如何提高修正后分类精度问题;武等人在文献[3]以及吕等人在文献[4]提出的规则提取方法也只对完备系统以及相容规则有效;黄等人在文献[5]提出的规则提取方法虽然适合不完备系统,也能将不相容规则提取出来,但并没有对这些不相容规则作区

分以备系统出现异常时使用。

不相容规则的产生可能是信息系统中存在异常信息,在这个信息时代,这些异常信息同那些“正常”信息同样重要,比如在信息分类或聚类中,这些异常信息可能就是孤立点,这对孤立点的检测有很大帮助;再比如在银行系统中,这些异常信息可能代表着“欺诈”信息,挖掘出这些信息对维护银行利益起着重要作用。因此,认为有必要对不相容规则进行一些专门的处理,以便发挥它们的作用。本文正是基于这种需要,根据信息论的一些基本观点,定义了规则的信息量,并在此基础上对不相容规则进行了讨论,提出了一种基于熵的不相容规则的修正算法。该算法主要是针对属性简化后的信息系统中的不相容规则进行处理,它首先将简化后的信息系统分成两部分:前一部分是相容规则,这是“正常部分”,不作处理;后一部分是不相容规则,这是“异常部分”,这部分是算法处理的重点。

2 基础理论

信息系统可以表示为 $S = \langle U, R, V, f \rangle$, U 为对象集合, $R = C \cup D$ 是属性集合,其中 C 为条件属性集, D 为决策属性集, $V = \bigcup_{r \in R} v_r$ 是属性值的集合, V_r 表示属性 r 的值域, $f: U \times R \rightarrow V$ 是一个映射函数,它指定 U 中每一个对象 X 的属性值。含有未知属性值 * 的信息系统称为不完备信息系统^[6]。在本

到稿日期:2008-10-15 返修日期:2009-03-09 本文受四川省科技计划项目(2008GZ0003),四川省科技厅科技攻关项目(07GG006-014)资助。

朱颢东(1980—),男,博士,CCF会员,主要研究方向为软件过程技术与方法、文本挖掘, E-mail: zhuhaodong80@163.com; 钟 勇(1966—),男,研究员,博士生导师,主要研究方向为软件过程技术与方法。

论文中,为了使算法适用于不完备系统,对于信息表 $S=\langle U, C\cup D, V, f\rangle, x\in U, y\in U, C_i\in C, V_i=\{v_{i1}, v_{i2}, \dots, v_{im_i}\}$ 对象 x, y 在属性 C_i 上相似的程度 $R_i(x, y)$ 假设 * 按照如下规则进行补充:

$$R_i(x, y) = \begin{cases} \text{MAX}(\{v_{i1}, v_{i2}, \dots, v_{im_i}\}) & c_i(x) = c_i(y) = * \\ 0 & c_i(x) \neq *, c_i(y) \neq *, \text{且 } c_i(x) \neq c_i(y) \\ 1 & c_i(x) \neq *, c_i(y) \neq *, \text{且 } c_i(x) = c_i(y) \\ v_{ij} & c_i(x) = *, c_i(y) = v_{ij} \text{ 或 } c_i(x) = v_{ij}, c_i(y) = * \end{cases} \quad (1)$$

以此来确定未知值 * 的取值概率,这不但充分利用了系统已有信息,而且也更加符合人们对不确定性的认知规律和直观感觉。

在决策逻辑语言中,命题联词的集合 $\{\neg, \vee, \wedge, \rightarrow\}$ 可以分别表示逻辑联词“非”,“或”,“与”和“当...,则...”。公式 (a, v) , 简记为 a_v , 称为初等原子公式,其中以 $a\in R, v\in V_a$ 。 θ, φ 为决策逻辑语言的公式,且 $\neg\theta, \theta\vee\varphi, \theta\wedge\varphi, \theta\rightarrow\varphi$ 也是决策逻辑语言的公式。记 $[(a, v)] = [a_v] = \{x\in U \mid f(a, x) = v\}$ 为论域中满足公式 a_v 的对象集。

说明:文中所有的对数计算均取“2”为底。

3 规则的信息熵以及可信度

定义1 称公式 $\theta\rightarrow\varphi$ 为知识表达系统 S 的决策规则 r, θ 是由原子公式 $\{a_{v_i}^i \mid a^i\in C, v_i\in V_{a^i}, i=1, 2, \dots, n, \text{其中 } n\leq \text{Card}(C)\}$ 和逻辑联词 $\{\neg, \vee, \wedge\}$ 构成的公式。 φ 是由原子公式 $\{d_{v_j}^j \mid d^j\in D, v_j\in V_{d^j}, j=1, 2, \dots, m, \text{其中 } m\leq \text{Card}(D)\}$ 和逻辑联词 $\{\neg, \vee, \wedge\}$ 构成的公式。

定义2 称 $H(\theta)$ 为公式 θ 的信息熵:

当 $\theta=(a, v)$ 时,

$$H(\theta) = -\log p(a_{ij}) = -\log(p([a_{ij}])) \\ = -\log \frac{\text{card}([a_v])}{\text{card}(U)}, a\in C, v\in V_a \quad (2)$$

当 $\theta=\bigwedge_{i=1}^k a_{v_i}^i$ 时,

$$H(\theta) = \frac{\sum_{i=1}^k p(a_{v_i}^i) H(a_{v_i}^i)}{\sum_{i=1}^k p(a_{v_i}^i)}, a^i\in R, v_i\in V_{a^i}, k\leq \text{card}(R) \quad (3)$$

当 $\theta=\bigwedge_{i=1}^k a_{v_i}^i$ 时,

$$H(\theta) = \frac{\prod_{i=1}^k p(a_{v_i}^i) H(a_{v_i}^i)}{\prod_{i=1}^k p(a_{v_i}^i)}, a^i\in R, v_i\in V_{a^i}, k\leq \text{card}(R) \quad (4)$$

定义3 称 $H(r, \theta\rightarrow\varphi)$ 为决策规则 r 的信息熵:

当 $\theta=a_v, \varphi=d_v$ 时,

$$H(r, \theta\rightarrow\varphi) = \log \frac{p(d_v/a_v)}{p(d_v)} \quad (5)$$

当 $\theta=\bigwedge_{i=1}^k a_{v_i}^i, \varphi=d_v$ 时,

$$H(r, \theta\rightarrow\varphi) = \log \frac{\sum_{i=1}^k p(a_{v_i}^i/d_v) H(r_i, a_{v_i}^i\rightarrow d_v)}{\sum_{i=1}^k p(a_{v_i}^i/d_v)}, \\ k\leq \text{card}(C) \quad (6)$$

当 $\theta=\bigwedge_{i=1}^k a_{v_i}^i, \varphi=d_v$ 时,

$$H(r, \theta\rightarrow\varphi) = \log \frac{p(d_v/\prod_{i=1}^k a_{v_i}^i)}{p(d_v)}, k\leq \text{card}(C) \quad (7)$$

当 $\theta=a_v, \varphi=\bigwedge_{i=1}^k d_{v_i}^i$ 时,

$$H(r, \theta\rightarrow\varphi) = \log \frac{p(\prod_{i=1}^k d_{v_i}^i/a_v)}{p(\prod_{i=1}^k d_{v_i}^i)}, k\leq \text{card}(D) \quad (8)$$

当 $\theta=a_v, \varphi=\bigvee_{i=1}^k d_{v_i}^i$ 时,

$$H(r, \theta\rightarrow\varphi) = \log \frac{\sum_{i=1}^k p(d_{v_i}^i/a_v) H(r_i, a_v\rightarrow d_{v_i}^i)}{\sum_{i=1}^k p(d_{v_i}^i/a_v)}, \\ k\leq \text{card}(D) \quad (9)$$

文中所定义的规则的信息量体现的是规则中蕴涵的关于规则前件和后件间相关程度的知识,体现的是一种与因果性相关的信息,逻辑公式的信息熵则是关于某事件本身所具有的不确定性,是获知此事时得到的信息。

很容易证明 $H(r, \theta\rightarrow\varphi) = H(r, \varphi\rightarrow\theta)$, 由于篇幅有限在此不做证明^[7]。

定义4 称 $B(r, \theta\rightarrow\varphi)$ 为决策规则 $r, \theta\rightarrow\varphi$ 的可信度:

$$B(r, \theta\rightarrow\varphi) = \frac{H(r)}{H(\varphi)} \quad (10)$$

下面的算法就是基于上面的定义进行的。

4 不相容规则的修正算法

该算法的核心思想:对于属性简化后的信息系统,这里的信息系统可以是完备系统也可以是不完备系统,如果是不完备系统,则对于 * 的值按照式(1)取值。把信息系统分成两部分:前一部分是相容规则的集合,对这部分不作处理;后一部分为不相容规则的集合,如果这部分为空则算法结束,否则作如下处理:设定一个信息基于熵的可信度阈值 g , 然后以此计算每个决策规则 r 的可信度 $B(r, \theta\rightarrow\varphi)$, 如果 $B(r, \theta\rightarrow\varphi) \geq g$, 则认为该决策规则正常,把它放入前一部分中;如果 $B(r, \theta\rightarrow\varphi) < g$, 则认为该决策规则为异常规则,则给该决策规则做个标记,然后把它放入前一部分中,以后凡是符合这个决策规则的信息都被看作是异常信息。

该算法的伪码表示:

Input: 属性简化后的信息系统 $S=\langle U, R, V, f\rangle$ 以及规则可信度阈值 g , 将 U 分成两部分: M (相容规则集合), IM (不相容规则的集合)

While ($IM \neq \text{null}$)

{ 从 IM 任取一规则 r ;

$IM = IM - \{r\}$; 计算 $B(r, \theta\rightarrow\varphi)$;

If $B(r, \theta\rightarrow\varphi) < g$ then $r = r_{\text{exception}}$

$M = M \cup \{r_{\text{exception}}\}$;

Output: 带异常标记的信息系统 $S=\langle M, R, V, f\rangle$

5 算法例证

表1是一张某检测系统的属性简化后决策表, $U=\{1, 2, \dots, 8\}$ $C=\{\text{Pressure}, \text{Temperature}, \text{Noise}\}$, $D=\{\text{Speed}\}$, 已经把相容规则与不相容规则分开了, 其中 5, 6 不相容, 7, 8 不相容。

表 1 某检测系统的属性简化后决策表

U	Pressure	Temperature	Noise	Speed
1	Low	Low	high	decrease
2	Low	medium	high	increase
3	normal	low	high	decrease
4	normal	high	normal	increase
5	normal	medium	normal	normal
6	normal	medium	normal	increase
7	Low	high	normal	increase
8	Low	high	normal	normal

根据经验给定的可信度阈值 $g=0.5$ 。

计算:对于规则 5

$$H(normal \wedge medium \wedge normal \rightarrow normal) = 0.8$$

$$H(normal) = 0.9, B(normal \wedge medium \wedge normal \rightarrow normal) = 0.8/0.9 \approx 0.9 > 0.5$$

对于规则 6

$$H(normal \wedge medium \wedge normal \rightarrow increase) = 0.6$$

$$H(increase) = 1.5, B(normal \wedge medium \wedge normal \rightarrow increase) = 0.6/1.5 \approx 0.4 < 0.5$$

该规则为异常规则记为:

$$normal \wedge medium \wedge normal \rightarrow increase^{exception}$$

对于规则 7

$$H(low \wedge high \wedge normal \rightarrow increase) = 0.7$$

$$H(increase) = 1.5, B(low \wedge high \wedge normal \rightarrow increase) = 0.7/1.5 < 0.5$$

该规则为异常规则记为:

$$low \wedge high \wedge normal \rightarrow increase^{exception}$$

对于规则 8

$$H(low \wedge high \wedge normal \rightarrow normal) = 0.5$$

$$H(normal) = 0.9, B(low \wedge high \wedge normal \rightarrow normal) = 0.5/0.9 > 0.5$$

此时 $M = \{1, 2, 3, 4, 5, 6, 7, 8\}$, 其中 6, 7 为异常规则, 符合它们的信息为异常信息。

结束语 本文在总结了许多不相容规则处理算法的不足之后, 提出了一种基于熵的不相容规则的修正算法。该算法的优点在于: 不论是完备系统中的不相容规则还是不完备系统中的不相容规则, 只要给定了规则的可信度阈值, 它就能找出哪些不相容规则是异常信息, 这在那些专门检测异常信息的场合大有用途, 而且, 只要对该算法稍作修改还可以计算得到信息中各个规则的可信度, 为以后做决策时使用。

参考文献

- [1] 管延勇, 薛佩军, 王洪凯. 不完备信息系统的可信决策规则提取与 E-相对约简[J]. 系统工程理论与实践, 2005, 12: 76-82
- [2] 朱荣刚, 许浒, 姜长生. 粗集理论中不相容规则的一种修正算法[J]. 电光与控制, 2006, 4: 29-32
- [3] 武志峰, 吉林林. 一种基于决策矩阵的属性约简及规则提取算法[J]. 计算机应用, 2005, 3: 639-642
- [4] 吕韶, 谢先明. 一种基于差别矩阵的决策表规则提取算法[J]. 现代机械, 2006, 3: 72-74
- [5] 黄兵, 周献中, 史迎春. 基于一般二元关系的知识粗糙熵与粗集粗糙熵[J]. 系统工程理论与实践, 2004(1): 93-96
- [6] 曾黄麟. 智能计算[M]. 重庆: 重庆大学出版社, 2004
- [7] 周荫清. 信息理论基础(第三版)[M]. 北京: 北京航空航天大学出版社, 2006

(上接第 150 页)

4 NGridFTP 的性能测试

并行传输是 GridFTP 提高传输效率最重要的途径, 我们使用并行传输对 NGridFTP 进行性能测试。测试使用电影《飘》作为传输数据。测试界面如图 9 所示, 性能结果如图 10 所示。

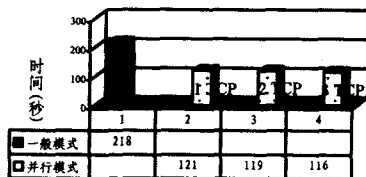


图 10 并行传输性能测试

可见, 并行模式的传输耗时确实明显少于一般模式。但是也可看到, 使用双倍的并行流并没有使性能提高一倍, 三条数据流与两条甚至一条并没有很大区别, 这主要是因为使用多个流实际上是增加了线程处理每条数据。因此, 除非使用真正的多个 CPU, 否则在并行模式下增加并行性的同时也增加了线程的开销。

结束语 本文基于 GT4 和 Java CoG 提供的 GridFTP

API 实现了一个可视化的网格数据传输工具-NGridFTP。此工具包括了 GridFTP 的数据传输、第三方控制的数据传输、并行传输等主要功能。详细介绍了开发工具使用的 API, 给出了实现的核心代码以及实现情况, 并对 NGridFTP 并行传输性能进行了测试。使用本工具, 只要用鼠标点击就可以进行相应的网格数据传输。下一步将研究 NGridFTP 如何实现分块传输功能以及如何进一步提高传输效率的问题。

参考文献

- [1] Foster I, Kesselman C. 网格计算(第二版)[M]. 金海, 袁平鹏, 石柯, 译. 北京: 电子工业出版社, 2004
- [2] 徐志伟, 冯百明, 李伟. 网格计算技术[M]. 北京: 电子工业出版社, 2004
- [3] The globus' alliance. GT 4. 0 GridFTP[EB/ OL]. <http://www.globus.org/toolkit/docs/4.0/data/gridftp/>, 2007-7-2
- [4] The globus' alliance. CoG jglobus[EB/ OL]. http://dev.globus.org/wiki/CoG_jglobus, 2007-7-2
- [5] 刘冀, 王茜. 基于网络服务的网格环境下数据传输[J]. 计算机科学, 2006, 33(1): 56-59
- [6] 徐清宇, 刘福岩. 网格数据传输协议 GridFTP[J]. 中北大学学报, 2005, 26(5): 341-244