

概率逻辑程序^{*}

王 洁 鞠实儿

(中山大学逻辑与认知研究所 广州 510275)

Probabilistic Logic Programming

WANG Jie JU Shi-Er

(Institute of Logic & Cognizing, Zhongshan University, Guangzhou 510275)

Abstract The purpose of this paper is to present the syntax and semantics of probabilistic logic programming to allow for the correct representation of incomplete information. General logic programming is extended by a subinterval of $[0,1]$ that describes the range for the conditional probability of the head of a clause given the range for the probability of each atom of its body. We define the semantics (answer sets semantics) of such probabilistic logic programming and illustrative their applications. We also show some properties of answer sets semantics for the probabilistic logic programs.

Keywords Uncertainty, Answer sets semantics, Probabilistic logic programming

1 引言

最近几十年来,不确定性的管理在知识描述和推理中扮演着越来越重要的角色。为了处理不确定知识,人们提出了各种不同的形式化和方法论,其中大部分是直接或间接地基于概率论的。

人们透彻研究了概率命题逻辑及其各种不同的变形,例如文[16]和[5],对概率一阶逻辑的扩展分为:概率定义于域上的一阶逻辑和概率定义于可能世界集上的一阶逻辑(特别参考 Halpern^[8]和 Bacchus 等^[2]的工作)。前者适用于描述统计知识,后者则适用于信念度的描述。用经典知识进行推理的一种重要的形式化是逻辑程序^[9,11],它是以自动定理证明的早期工作为基础,并且随着 PROLOG 的发展而逐渐繁荣起来的。现在,逻辑程序是人工智能和演绎数据库中广为人知的知识描述和推理的形式化方法。

在实际应用中,迫切需要处理不确定性知识的描述和推理系统。在逻辑程序框架中,至少存在两种处理不确定性知识的主流:多值逻辑程序和概率逻辑程序,旨在处理量化的不确定性,而析取逻辑程序则是处理析取知识和非单调“非”的。如今,概率逻辑程序被广泛认为是关于概率信息知识描述和常识推理的一种很有价值的工具。Ng 和 Subrahmanian^[13]做了开创性的工作,提出了第一个概率逻辑程序框架,并在此框架对概率信息进行描述,同时研究了这种语言的不动点语义。但是这种框架仍然存在一些限制,特别是没有关于条件概率的描述。最近几年来,在概率逻辑程序方面有了一些重要的发展^[14,15,10,11,4,12]。

在本文中,我们给出了一种新的概率逻辑程序语言,它是通过引入概率到传统的逻辑程序语言中来实现的。传统的逻辑程序通过对每个原子附加一个 $[0,1]$ 上的子区间而被扩展。它描述了在给定一个逻辑程序子句的尾部的每个原子的概率的情况下,其头部的原子发生的概率区间。在这里我们使用了

区间概率而不是点概率。粗略地讲,原因是两方面的:首先,区间概率是点概率的一般化,而且它提供了更大的可行性空间,特别是它允许描述不精确的概率知识;其次,使用区间概率是技术方面的原因,即使当我们仅仅适用点概率时,概率知识仍然可指定一个概率分布集,而不仅仅是点概率分布。

我们还在文中形式化地定义了概率程序语言,给出了它的语法和语义;使用概率逻辑程序语言编码了一个有趣的例子;讨论了概率逻辑程序语言的答案集语义的一些性质;比较了一些相关的工作。

2 语法与语义

我们在这里定义概率逻辑程序的语法和语义。考虑一阶语言 L ,它包括无限多个目标变量和有限多个常量符号、谓词符号和函数符号,并且按照通常的方式定义项。一个原子是一个形如 $A(t_1, \dots, t_k)$ 的表达式,其中 A 是任意元 $k \geq 0$ 的谓词符号, $t_1, \dots, t_k$ 是常量符号或目标变量。

一个 p -原子是一个形如 F/μ 的表达式,其中 F 是原子, $\mu = [a, b] \subseteq [0,1]$, $0 \leq a \leq b \leq 1$ 是概率常量,称为 F 发生的概率。特别当 $\mu = [1,1]$ 时, $F/[1,1]$ 意味着 F 是真的,我们常常把 $F/[1,1]$ 简写为 F ;当 $\mu = [0,0]$ 时, $F/[0,0]$ 意味着 F 是假的;当 $\mu \subset (0,1)$ 时, F/μ 意味着 F 可能发生,即 F 是概率真的。直觉上, F/μ 意味着 F 发生的概率属于区间 μ 。这样,如果 toss 是语言 L 中的一个二元谓词, $\text{toss}(\text{head})/[0.49, 0.51]$ 的意思是:投掷一枚硬币时;头像的一面朝上的概率属于区间 $[0.49, 0.51]$ 。一个概率逻辑子句 r 是这样的一个表达式:

$$F_0/\mu_0 \leftarrow F_1/\mu_1, \dots, F_m/\mu_m, \text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n \quad (1)$$

其中 $n \geq m \geq 0$, 每个 F_i 是一个原子, μ_i 是单位区间 $[0,1]$ 上的子区间。即 $0 \leq \mu_i \leq 1$, F_i/μ_i 是一个 p -原子 ($0 \leq i \leq n$)。not(非)是联结词,称为缺省非。not F/μ 的意思是:没有理由相信 F 发生的概率属于区间 $\mu \subseteq [0,1]$ 。规则(1)的直觉意思是:如果

^{*} 本文得到了教育部重点研究基地重大项目:开放世界中认知的逻辑结构的基金资助。王 洁 博士研究生,研究方向为:数理逻辑、不确定推理和逻辑程序。鞠实儿 博士,教授,博导,主要从事数理逻辑、归纳逻辑及认知科学的研究。

我们分别以 $\mu_i (1 \leq i \leq m)$ 的概率相信 F_i 发生, 并且没有理由相信 $F_{m+1}, \dots, F_n$ 发生的概率分别为 $\mu_{m+1}, \dots, \mu_n$, 那么 F_0 发生的概率是 μ_0 。在规则(1)中, F_0/μ_0 称为它的头部, $F_1/\mu_1, \dots, F_m/\mu_m$ 称为它的正的尾部, $\text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n$ 称为它的负的尾部。我们常常分别用 $\text{Head}(r)$, $\text{Body}^+(r)$ 和 $\text{Body}^-(r)$ 来表示一个规则的头部、出现在正的尾部的 p-原子集和出现在负的尾部的 p-原子集。当规则的尾部为空时, 我们称这样的规则为一个事实。规则(1)可以简写为

$$\text{Head}(r) \leftarrow \text{Body}^+(r), \text{Body}^-(r).$$

概率逻辑程序是概率逻辑子句的有限集合。

特别假定我们想描述: 如果 $F_i (1 \leq i \leq m)$ 是真的, 并且没有理由相信 $F_i (m+1 \leq i \leq n)$ 发生的概率是 μ_i , 那么 F_0 发生的概率是 μ_0 。这个关系用一个概率子句 r 表示:

$$F_0/\mu_0 \leftarrow F_1/[1, 1], \dots, F_m/[1, 1], \\ \text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n \quad (2)$$

另外, 若数据库中下列事实成立:

$$F_i/\mu_i \leftarrow 1 \leq i \leq m \quad (3)$$

则由规则(2)、(3), 我们知道下列规则成立:

$$F_0/\mu_0 \leftarrow F_1/\mu_1, \dots, F_m/\mu_m, \text{not } F_{m+1}/\mu_{m+1}, \dots, \\ \text{not } F_n/\mu_n \quad (4)$$

它的意思是: 当我们分别以 $\mu_i (1 \leq i \leq m)$ 的概率相信 F_i 发生, 并且没有理由相信 $F_{m+1}, \dots, F_n$ 发生的概率分别为 $\mu_{m+1}, \dots, \mu_n$, 那么 F_0 发生的概率是 μ^* 。我们把由形如(2)、(3)的规则产生规则(4)的变换称为概率逻辑程序的一般化过程, 由此变换产生的新的概率逻辑程序称为一般化的概率逻辑程序。注意: 由于我们仅仅知道 F_i 发生的概率为 $\mu_i (1 \leq i \leq m)$, 因此 F_0 发生的概率不仅依赖于 μ_i , 而且依赖 F_i 之间的相互关系。

一个原子, p-原子, 概率逻辑子句和概率逻辑程序是基本的, 当且仅当它不包括任何的目标变量。一个概率逻辑子句 $A/\mu \leftarrow F_1/\mu_1, \dots, F_n/\mu_n, \text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n$ 的基例是一个基本概率逻辑子句 $A\theta/\mu \leftarrow F_1\theta/\mu_1, \dots, F_n\theta/\mu_n, \text{not } F_{m+1}\theta/\mu_{m+1}, \dots, \text{not } F_n\theta/\mu_n$, 其中 θ 是正好出现在概率逻辑程序子句中的所有目标变量的一个基替代。替代和基替代的概念按通常的方式定义。概率逻辑程序的语义是建立在 Gelfond 和 Lifschitz 所提出的稳定模型的基础上的回答集语义^[6,7]。一个概率逻辑程序的回答集与一个理智的推理者所拥有的信念是一致的。为了定义概率逻辑程序的语义, 我们首先给出下面的定义。

定义 1 任给 p-原子 $A/\mu_1, A/\mu_2$, 如果 $\mu_1 \subseteq \mu_2$, 那么 $A/\mu_1 \subseteq A/\mu_2$ 。特别地, 如果 $\mu = [\alpha, \alpha]$, 那么 $A/[\alpha, \alpha] \subseteq A/[\alpha, \alpha]$ 。

定义 2 一个概率合取策略是这样的一个映射, 它把一对概率区间映射到一个概率区间, 并且满足下列公理:

$$(1) \text{ 下限: } [a_1, \beta_1] \otimes [a_2, \beta_2] \subseteq [\min(a_1, a_2), \min(\beta_1, \beta_2)]$$

其中 $[x, y] \subseteq [x', y'] \Leftrightarrow x \leq x' \wedge y \leq y'$ 。

$$(2) \text{ 无知: } [a_1, \beta_1] \otimes [a_2, \beta_2] \subseteq [\max(0, [a_1 + a_2 - 1]), \min(\beta_1, \beta_2)]$$

$$(3) \text{ 恒等: 当 } (e_1 \wedge e_2) \text{ 一致且 } [a_2, \beta_2] = [1, 1] \text{ 时, } [a_1, \beta_1] \otimes [a_2, \beta_2] = [a_1, \beta_1]$$

$$(4) \text{ 零化子: } [a_1, \beta_1] \otimes [0, 0] = [0, 0]$$

$$(5) \text{ 交换律: } [a_1, \beta_1] \otimes [a_2, \beta_2] = [a_2, \beta_2] \otimes [a_1, \beta_1]$$

$$(6) \text{ 结合律: } ([a_1, \beta_1] \otimes [a_2, \beta_2]) \otimes [a_3, \beta_3] = [a_1, \beta_1] \otimes ([a_2, \beta_2] \otimes [a_3, \beta_3])$$

* 由定义 2 的无知公理计算得到。

$$([a_2, \beta_2] \otimes [a_3, \beta_3])$$

$$(7) \text{ 单调性: } [a_1, \beta_1] \otimes [a_2, \beta_2] \subseteq [a_1, \beta_1] \otimes [a_3, \beta_3] \text{ if } [a_2, \beta_2] \subseteq [a_3, \beta_3]$$

直觉上, 按上面的定义, $[a_1, \beta_1]$ 和 $[a_2, \beta_2]$ 分别是事件 e_1, e_2 发生的概率区间, 并且由 $[a_1, \beta_1] \otimes [a_2, \beta_2]$ 可计算得到两个事件同时发生时的概率区间。下限公理描述: 合取的概率小于单个事件的概率。当我们对于两个事件 e_1, e_2 之间的关系一无所知的时候, Boole^[3] 给出它们的合取的概率属于区间 $[\max(0, [L_1 + L_2 - 1]), \min(p_1, p_2)]$, 这是无知公理所陈述的内容。恒等公理和零化子公理指出了当其中一个事件是明确发生或不发生的时候概率合取策略所应遵守的原则。交换律和结合律是明显的。单调公理陈述了当两个事件中其中一个事件的概率边界是不断下降的时候, 其合取事件的概率边界也是不断下降的。

定义 3 如果 p-原子 $A/\mu_1, A/\mu_2$ 同时发生, 并且 $\mu_1 \not\subseteq \mu_2, \mu_2 \not\subseteq \mu_1$, 那么称 A/μ_1 和 A/μ_2 是不一致的。

定义 4 一个 p-原子集 S 在一个规则下是封闭的, 如果 S 满足规则的头部或者不满足它的尾部。

一个概率逻辑程序 Π 的回答集语义指定给 Π 一个回答集的集合——基本的 p-原子集合, 它一致于在 Π 的规则集的基础上由一个理智的推理者所建立的信念。概率逻辑程序的语义把具有变量的规则看作它的基例集的简写。这样, 我们有充足的理由来定义不含变量的概率逻辑程序的回答集语义。一个概率逻辑程序的精确的语义(回答集语义)由下列步骤来定义:

步骤 1 首先假定 Π 是任意一个的概率逻辑程序。令 PA 表示 Π 的语言中所有基本 p-原子的集合, 对任意集合 $S \subseteq PA$, 如果一个规则具有下列形式:

$$F_0/\mu_0 \leftarrow F_1/[1, 1], \dots, F_m/[1, 1], \\ \text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n$$

并且 $F_i/\mu_i \in S$ 或 $F_i/\mu_i \subseteq F_i/\nu \in S$, 那么我们对此概率逻辑程序做一般化变换, 得到一个一般化的概率逻辑程序。

步骤 2 假定 Π 是一个不包括‘非’(not)的一般化概率逻辑程序, PA 是语言 Π 中所有基本 p-原子组成的集合, 那么 Π 的回答集是满足下列条件的 PA 的最小子集:

(i) 对 Π 中的任何规则 $F_0/\mu_0 \leftarrow F_1/\mu_1, \dots, F_m/\mu_m$, 如果 $F_i/\mu_i \in S$ 或 $F_i/\mu_i \subseteq F_i/\nu \in S$, 那么 $F_0/\mu_0 \in S$;

(ii) 如果 S 包括不一致的 p-原子, 那么 $S = PA$ 。

我们用 $\alpha(\Pi)$ 表示不包括‘非’(not)的概率逻辑程序的回答集。

步骤 3 假定 Π 是任意一个不包括变量的一般化概率逻辑程序, 由 PA 表示 Π 中所有基本 p-原子组成的集合, 对任给的 $S \subseteq PA$, 令 Π^S 表示对 Π 通过下列删除获得的概率逻辑程序:

(iii) 每个其尾部出现 $\text{not } F/\mu$ 并且 $F/\mu \in S$ 或 $F/\mu \subseteq F/\nu \in S$ 成立的规则。

(iv) 剩余的规则中所有形如 $\text{not } F/\mu$ 的 p-原子。

显然, Π^S 不包括‘非’, 实际上其回答集已经由步骤 2 所定义。如果这个回答集与 S 一致, 那么我们称 S 是 Π 的回答集。也就是说, 一般化的概率逻辑程序的回答集满足下列等式:

$$S = \alpha(\Pi^S) \quad (5)$$

回答集语义对一个查询有 4 种可能的回答:‘是’、‘不是’、‘可能’和‘不知道’。对一个一致的逻辑程序,如果它的回答集中包括 $L/[1,1]$ 或 $L/[0,0]$,说明 L 是真的或 L 是假的,那么对查询 L 的回答相应‘是’或‘不是’;如果其回答集中包括 L/μ ,则说明 L 是可能发生的。我们以 μ 表示发生的可能性的值,否则对一个查询 L 的回答是‘不知道’。

3 举例

某国有一种疾病 M 严重损害人们的健康,它与吸烟及家族病史有很大的关系。在临床上,一般通过 3 种基本症状 A, B, C 对此疾病做出初步的诊断。只有在认为患者有较大可能患有此病时,医生才对其做一项费用昂贵且给病人带来极大痛苦的病理检验,并最终确认是否患有此病。医生使用下列规则来决定是否对患者做此项检验:

(1)如果 3 种症状 A, B, C 出现的概率均大于 80%,那么患病的可能性大于 80%。

(2)如果 3 种症状 A, B, C 出现的概率均大于 60%,并且有吸烟史,那么患病的可能性大于 80%。

(3)如果 3 种症状 A, B, C 出现的概率均大于 70%,并且有家族病史,那么患病的可能性大于 80%。

(4)如果 3 种症状 A, B, C 出现的概率均大于 50%,并且有吸烟史及家族病史,那么患病的可能性大于 80%。

(5)如果在初步诊断中认为患病的可能性大于 80%,那么患者需要做进一步病理检验。

(6)如果没有理由认为患者需要做进一步检验,那么认为其不需要做病理检验。

这些规则可用逻辑程序如下编码:

(1) $M(x)/[0.80,1] \leftarrow A(x)/[0.8,1], B(x)/[0.8,1], C(x)/[0.8,1]$

(2) $M(x)/[0.80,1] \leftarrow A(x)/[0.6,1], B(x)/[0.6,1], C(x)/[0.6,1], \text{Smoke}(x)$

(3) $M(x)/[0.80,1] \leftarrow A(x)/[0.7,1], B(x)/[0.7,1], C(x)/[0.7,1], \text{Mfamily}(x)$

(4) $M(x)/[0.80,1] \leftarrow A(x)/[0.5,1], B(x)/[0.5,1], C(x)/[0.5,1], \text{Smoke}(x), \text{Mfamily}(x)$

(5) $\text{Test}(x) \leftarrow M(x)/[0.80,1]$

(6) $\text{Test}(x)/[0,0] \leftarrow \text{not Test}(x)$

假定在医疗数据库中 3 名患者 Martin, John 和他的姐姐 Jane,医生在诊断中发现 John 和 Jane 出现 3 种症状的可能性都在 60%以上,但 John 吸烟。Martin 的 3 种症状比较明显,都在 80%以上。

由第 2 节中所定义的逻辑程序的语义,可知包括以上规则及事实的逻辑程序有一个回答集,这个回答集由

$\{M(\text{John})/[0.80,1], \text{Test}(\text{John}), M(\text{Martin})/[0.80,1], \text{Test}(\text{Martin}), \text{Test}(\text{Jane})/[0,0]\}$

和上面提到的关于 John, Jane 和 Martin 的事实组成。

由于回答集中包括了 $\text{Test}(\text{John}), \text{Test}(\text{Martin})$ 和 $\text{Test}(\text{Jane})/[0,0]$,因此对 John 和 Martin 是否需要病理检验的回答是“是的”,而对 Jane 是否需要检验的回答是“不需要”。

4 回答集语义的性质

定理 1 对任意一个不包括‘非’(not)的逻辑程序,由步骤 3 定义的回答集语义等价于步骤 2 所给出的语义。

证明 这个结论是下列事实的一个直接后承:任意一个

不包括‘非’(not)的逻辑程序 Π ,有 $\Pi^S = \Pi$,所以其固定点条件变为 $S = \alpha(\Pi)$ 。

定义 5 一个基本的 p -原子 A/μ 是逻辑程序 Π 的回答集语义的后承当且仅当 A/μ 属于 Π 的回答集,表示为 $\Pi \models A/\mu$ 。

定理 2 令 Π 是任意一个逻辑程序, Π 必满足下列条件之一:

- Π 没有回答集合。
- Π 的回答集存在且其回答集是一致的。
- 唯一的回答集是 Π 中所有的 p -原子的集合。

如果一个程序的后承集是一致的,则这个程序是一致的,否则是不一致的。前者相应于定理中的前两种情况,后者相应于定理的第三种情况。

对任意的逻辑程序 Π ,都可以由回答集语义中的步骤 1 转化为一般化的逻辑程序。因此,下面我们只讨论一般化的逻辑程序的性质。我们说一个 p -原子集的子集 X 被一般化的逻辑程序支撑,如果对 Π 中每个 p -原子 $A/\mu \in X$,都存在 Π 中的规则:

$F/\mu \leftarrow F_1/\mu_1, \dots, F_m/\mu_m, \text{not } F_{m+1}/\mu_{m+1}, \dots, \text{not } F_n/\mu_n$

使得

(1) $\{F_1/\mu_1, \dots, F_m/\mu_m\} \subseteq X$,

并且

(2) $\{F_{m+1}/\mu_{m+1}, \dots, F_n/\mu_n\} \cap X = \emptyset$

定理 3 令 Π 是一个一般化的逻辑程序,下面的性质成立:

- Π 的任意的一个一致的回答集被 Π 所支撑。
- Π 的一致回答集中的每个基本的 p -原子都是 Π 的头部。

• 如果 x 和 y 分别是 Π 的回答集,并且 $x \subseteq y$,那么 $x = y$ 。

证明 前两种结论是明显的。后者的证明如下:因为 x, y 是 Π 的回答集,且 $x \subseteq y$,明显地有 $\Pi^x \subseteq \Pi^y$ 。由此可得 $\alpha(\Pi^x) \subseteq \alpha(\Pi^y)$,即 $y \subseteq x$,所以 $x = y$ 。

结论 本文陈述了逻辑程序的语法和语义,以对不完备信息进行正确的描述。逻辑程序是通过对传统的逻辑程序附加一个 $[0,1]$ 上的子区间而被扩展,它描述了在给定一个逻辑程序的尾部的每个原子的概率的情况下其头部的原子发生的概率的区间。我们定义了逻辑程序的回答集语义,并且举例说明了它的应用。同时给出了逻辑程序的回答集的一些性质。

自从 Ng 和 Subrahmanian^[13] 的开创性工作以来,特别是最近几年,逻辑程序有了很大的发展。虽然不同研究者所关注的方面不尽相同,但目的是相似的,即使用逻辑程序方法描述不确定信息,特别是概率信息。Ngo 和 Haddawy^[15] 研究了一个逻辑程序框架,它允许对条件概率进行描述,并且定义了不动点理论,同时讨论了所研究的逻辑程序与贝叶斯网之间的联系,朝着建立与自动推理两种方法合一的方向发展。相比较而言,Thomas Lukasiewicz^[16] 陈述了可能世界语义下的逻辑程序框架,其中对概率演绎使用了线形程序的方法。而本文所提出的逻辑程序框架则使用回答集语义作为知识描述与推理的有效工具。

参考文献

- 1 Apt K R. Logic programming. Handbook of theoretical computer science, volume B, chapter 10, MIT Press, 1990. 493~574

(下转第 11 页)

- mation Processing System. Cambridge, MA. MIT Press, 1996, 8:535~541
- 15 Buntine W L. A theory of learning classification rules: [Ph. D. thesis]. University of Technology, School of Computing Science, Sydney, Australia. 1990
 - 16 Wolpert D. Stacked generalization. *Neural Network*, 1992, 5(2):241~260
 - 17 Gama J. Combining classification algorithm: [Ph. D. Thesis]. 1999
 - 18 Musick R, Catlett J, Russell S. Decision theoretic subsampling for induction on large database. In: *Utgoff, Proc. of the Tenth Intl. Conf. on Machine Learning*. San Francisco, CA. Morgan Kaufmann, 1992. 212~219
 - 19 Shafer J, Agrawal R, Mehta M. SPRINT: A scalable parallel classifier for data mining. In: *Proc. of the Twenty-second VLDB Conf.* San Francisco, CA. Morgan Kaufmann, 1996. 544~555
 - 20 Ruggieri S. Efficient C4.5. *IEEE Transactions on Knowledge and Data Engineering*, 2001, 14(2):438~444
 - 21 Chan P K, Stolfo S J. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In: *Proc. of the First Intl. Conf. on Knowledge Discovery and Data Mining*. Menlo Park, CA. AAAI Press, 1995. 39~44
 - 22 Catlett J. On changing continuous attributes into ordered discrete attributes. In: *Proc. of the Fifth European Working Session on Learning*, Heidelberg: Springer-Verlag, 1991. 164~178
 - 23 Fayyad U M, Irani K B. Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proc. of the Thirteenth Intl. Joint Conf. on Artificial Intelligence*. San Francisco, CA: Morgan Kaufman, 1993. 1022~1027
 - 24 Fulton T, Kasif S, Salzberg S. Efficient algorithms for finding multi-way splits for decision trees. *Machine Learning*. In: *Proc. of the Twelfth Intl. Conf.* San Francisco, CA: Morgan Kaufmann, 1995. 244~251
 - 25 Elomaa T, Rousu J. General and Efficient Multisplitting of Numerical Attributes. *Machine Learning*, 1999. 1~49
 - 26 Kononenko I. Estimating attributes: analysis and extension of relief. In: *Proc. of the 1994 European Conf. on Machine Learning*. Amsterdam. Springer Verlag, 1994. 171~182
 - 27 Kononenko I, Šimec E, Robnik-Šikonja M. Overcoming the myopic of inductive learning algorithms with RELIEFF. *Applied Intelligence Journal*, 1997, 7(1):39~66
 - 28 John G, Kohavi R, Pfleger K. Irrelevant features and the subset selection problem. In: *Proc. of the eleventh Intl. Conf. on Machine Learning*. San Francisco, CA. Morgan Kaufmann. 1994. 121~129
 - 29 Moore A W, Lee M S. Efficient algorithm for minimizing cross validation error. In: *Proc. of the eleventh Intl. Conf. on Machine Learning*. San Francisco, CA. Morgan Kaufmann, 1994. 190~198
 - 30 Lowe D G. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 1995, 7(1):72~85
 - 31 Fu L. Rule Learning by Searching on Adapted Nets. In: *Proc. of the 9th National Conf. on Artificial Intelligence*, Anaheim, CA: AAAI Press, 1991. 590~595
 - 32 Towell G G, Shavlik J W. Knowledge-Based Artificial Neural Networks. *Artificial Intelligence*, 1994, 70(1-2):119~165
 - 33 Setiono R. Extracting Rules from Neural Networks by Pruning and Hidden-Unit Splitting. *Neural Computation*, 1997, 9(1):205~225
 - 34 Setiono R, Leow W K. FERNN: An algorithm for Fast Extraction of Rules from Neural Networks. *Applied Intelligence*, 2000, 12(1-2):15~25
 - 35 Saito K, Nakano R. Rule Extraction from Facts and Neural Networks. In: *Proc. of the Intl. Neural Network Conf.*, North-Holland: Kluwer, 1990. 379~382
 - 36 Craven M W. Extracting comprehensible models from trained neural network: [Ph. D thesis]. University of wisconsin, madison, 1996
 - 37 Thrun S. Extracting Rules from Artificial Neural Networks with Distributed Representations. In: *Tesauro G, Touretzky D, Leen T, eds. Advances in Neural Information Processing Systems (Volume 7)*, Cambridge, MA: MIT Press, 1995. 505~512
 - 38 Benitez J M, Castro J L, Requena I. Are Artificial Neural Networks Black Boxes? *IEEE Transactions on Neural Networks*, 1997, 8(5):1156~1164.
 - 39 Boz O. Converting a trained neural network to a decision tree. *DECTEX-Decision tree extractor*. The 2002 Intl. conf. on machine learning and applications, 2002. 24~27

(上接第3页)

- 2 Bacchus F, Grove A, Halpern J Y, Koller D. From statistical knowledge bases to degrees of belief. *Artificial intelligence*, 1996, 87:75~143
- 3 Boole G. The law of thought. Macmillan, London. 1854
- 4 Dekhtgar M I, Dekhtgar A, Subrahmanian V S. Hybrid probabilistic programs: Algorithms and complexity. In: *Proc. UAI-99*, 1999. 160~169
- 5 Fagin R, Halpern J, Meggido N. A logic for reasoning about probabilities. *Information and Computation*, 1990, 87:78~128
- 6 Gelfond M, Lifschitz V. The stable model semantics for logic programming. *Logic programming*. In: *Proc. of the Fifth Int. Conf. and Symp.* 1988. 1070~1080
- 7 Gelfond M, Lifschitz V. Classical negation in logic program and disjunctive databases. *New generation computing*, 1991. 365~387
- 8 Halpern J Y. An analysis of first-order logics of probability. *Artificial Intelligence*, 1990, 46(3): 311~350
- 9 Lloyd J. *Foundations of logic programming*. Springer, 1984
- 10 Lukasiewicz T. Probabilistic logic programming. In: *Proc of the 13th Biennial European Conf. On Artificial Intelligence*, 1998. 388~392
- 11 Lukasiewicz T. Many-Valued Disjunctive Logic Programs with Probabilistic Semantics. In: *Proc. of the 5th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning*, Volume 1730 of LNAI, 1999. 277~289
- 12 Lukasiewicz T. Probabilistic logic programming with conditional constraints. *ACM Trans. Computat. Logic*, 2001, 2(3):289~337
- 13 Ng R, Subrahmanian V S. Probabilistic logic programming. *Information and computation*, 1992, 101(2): 150~201
- 14 Ng R, Subrahmanian V S. Stable semantics for probabilistic deductive databases. *Information and computation*, 1995, 110(1):42~83
- 15 Ngo L, Haddawy P. Probabilistic Logic Programming and Bayesian Networks. *ASIAN 1995*, 1995. 286~300
- 16 Nilsson N J. Probabilistic logic. *Artificial Intelligence*, 1986, 28(1): 71~87