

关于解释学习逻辑构架的一些扩充

张家重 费宗铭 王岩冰

(南京大学计算机软件研究所)

摘 要:

近来,在机器学习研究领域,解释学习倍受重视,而其中心机制——解释推广却受限于一阶逻辑,为支持高阶对象和模态概念,本文介绍了两种扩充Horn逻辑表示域的方法,并通过例子说明了解释推广在扩充表示域中的实现。

一、解释学习及其传统逻辑构架

1. 解释学习和解释推广

机器学习是人工智能研究领域中最活跃的课题。所谓机器学习,内在地看,是从未知到知的过程,是知识的增加过程;外表上看,是系统本身某些性能的改善^[1,2]。

目前,机器学习研究大致分为:归纳或相似学习(SBL)和分析或解释学习(EBL)。EBL是近期提出且颇受青睐的一种学习方法。T. Mitchell将EBL总结如下:

给定: 域论DT (规则和事实集),

目标概念TC,

训练实例TE,

操作性准则OC。

找出: 满足OC的关于TC的充分条件。

EBL的一般过程是,使用DT,找出TE为什么是TC之实例的证明(即解释)。然后,根据OC对证明进行推广,从而得到关于TC的一个充分条件。由此可见,EBL的中心机制乃是推广,即解释推广(Explanation-Based Generalization, EBG),也就是T. Mitchell给出的基本构架。

2. Horn逻辑程序设计作为EBG构架

近来,人们通常把逻辑程序设计用作

EBG构架。其理论依据是,逻辑程序设计能够表示EBG的域论、训练实例、目标、推理规则和操作性准则等方面,这样便有助于用一致的方法来解释内在规律,阐明语义问题。

下面,我们在一阶逻辑程序设计中,考察一个“自杀”的例子。

EBG构成:

• 域论DT

kill A B; -hate A B, possess A C, weapon C,

hate W W; -depressed W,

possess U V; -buy U V,

weapon Z; -gun Z.

• 训练实例TE

depressed john,

buy john obj1,

gun obj1.

此例用Prolog描述。大写字母代表变量,‘,’指逻辑‘与’,并且句中变量是隐式地全称量化的。

此外,EBG还需要一个询问,设为:

kill john john.

然后,EBG便需要一个证明(或解释)来解决给出的询问。在逻辑程序设计中,这样的证明可表示为Prolog的搜索轨迹。图1给出

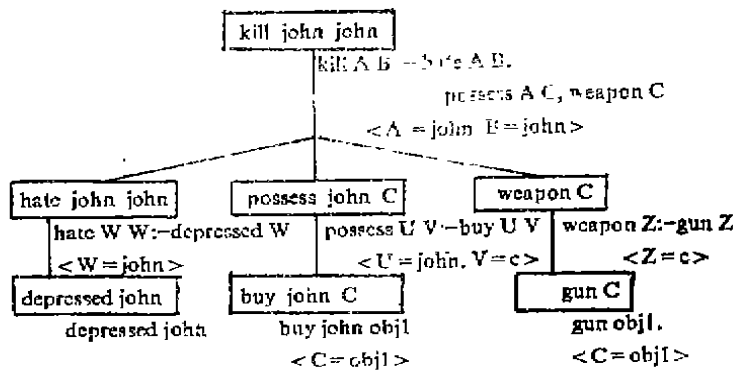


图1 一阶证明(解释)

了关于上述询问的证明。图中,方框中的内容是证明的目标,右下侧是化归特定目标的程序语句。在应用于子句过程中,变量可用常量或目标中的变量进行合一或例化。“<>”中内容是相应的合一。

EBG通过推广此解释而产生一个证明策略。图2构造了相应于具体解释的推广证明,但去除了训练实例子句。一个新的证明树的根结点是从初始询问中,通过用局部变量替换所有一阶逻辑常量而得到的推广询问,询问kill john john,成为kill X Y。第一个证明中应用的每个域论规则都相应地应用于第二个证明之中。在证明树上,通过增强合一限制而潜在地约束了最终结果。如,(kill X Y)成为(kill X X)。推广证明树的叶结点(如gun(V))对应于由训练实例规则推导所得的初始证明的子目标。这些叶结点条件的合取式构成推广询问的充分条件:kill X X, -depressed X, buy X C, gun C。我们称此导出规则为基于解释的推广,或简称

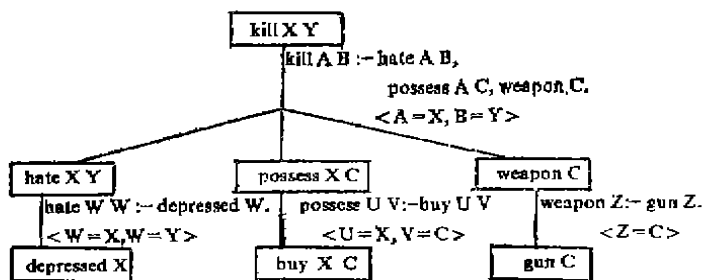


图2 一阶推广证明

推广。

3. 操作性讨论

对于前例,假定将其最后的域论规则改为:

weapon Z, -gun Z; knife Z; grenade Z;

其中,“,”指逻辑“或”。虽然这种更改能够得到同样的推广,但新的子句提示我们,在导出规则中,以weapon取代gun,或许更

具一般性:

kill X X, -depressed X, buy X C, gun C.

EBG的形式化已经提供了实现上述结果的方法。由于仅有域论规则出现在推广证明之中,因而可用如下域论和训练实例来重写前例。

• 域论

kill A B, -hate A B, possess A C, weapon C

hate W W, -depressed W.

possess U V, -buy U V.

• 训练实例

weapon Z, -gun Z; knife Z; grenade Z;

...

depressed john.

buy john objl.

gun objl.

由此可推出期望的导出规则。相应的推广证明如图3所示。

传统上,把限制推广证明树深度的手段称为操作性。通过验证一特定目标(如weapon)满足操作性准则,就可把推导出此目

标的子树从推广证明树中删去。在此EBG构架之下,训练实例和域论的差别,至少对于推广证明已无意义。本文介绍的方法的不同之处在于,把操作性表示为关于子句的谓词(不管其是否为域论),而非关于目标的谓词。

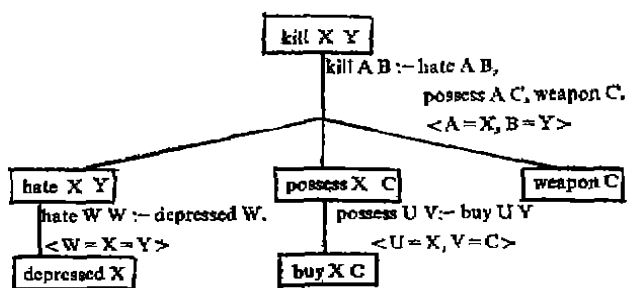


图3 改进的一阶推广证明

二、解释学习的高阶构架

高阶逻辑是在一阶逻辑中引进函数变元、谓词变元及有关量词后得到的。而所谓高阶域，是指具有像函数和谓词等高阶值的域；高阶语言或应用是关于高阶域的语言或应用；高阶EBG是指变量替换包括高阶对象的EBG；表示域是指表示应用值的语言。

推广的性能和表示语言或表示域联系紧密。如若知识在表示域中不恰当地表示，就难以用自然、精确的方法把期望的推广表示出来。特别是，高阶域之一阶表示的复杂性阻碍了推理和推广。但迄今，EBG的应用仍局限于一阶表示域。为使EBG能适用于高阶域，应该考虑应用高阶的技术。下面用例说明EBG在高阶域中的实现。其表示语言采用λProlog，它是一种支持高阶合一的高阶逻辑程序设计语言^[4]。

例 考虑下面符号积分的高阶规则。其首先处理幂（ $a \neq -1$ ），之后处理常因子，最后分解和，

$$\begin{aligned} & \forall a. \{ \lambda x. x^a = \lambda x. x^{(a+1)} / (a+1) \\ & \forall a \forall f \forall f'. \{ f = f' \Rightarrow \lambda x. a * f(x) \\ & \quad = \lambda x. a * f'(x) \\ & \forall f \forall f' \forall g \forall g'. \{ f = f' \wedge g = g' \\ & \quad \Rightarrow \lambda x. f(x) + g(x) = \lambda x. f'(x) + g'(x) \end{aligned}$$

其中，以λ抽象进行函数的显式定义，而未采用传统的名字约束dx。

在支持名字约束的高阶域中，限制x不

自由出现于a中，不需要复杂的辅助条件。为表示积分规则，引入一个谓词“integr”来联结定义和不定积分。

$$\begin{aligned} & \text{pi } A \backslash (\text{integr } X \backslash (\text{expn } X \ A) \ X \backslash \\ & \quad ((\text{expn } X \ (A+1)) \text{div } (A+1))). \\ & \text{pi } A \backslash (\text{pi } F \backslash (\text{pi } G \backslash (\\ & \quad \text{integr } X \backslash (A * (F X)) X \backslash \\ & \quad (A * (G X)), -\text{integr } F \ G))). \\ & \text{pi } F \backslash (\text{pi } Fi \backslash (\text{pi } G \backslash (\text{pi } Gi \backslash (\\ & \quad \text{integr } X \backslash ((F X) + (G X)) X \backslash \\ & \quad ((Fi \ X) + (Gi \ X)) \\ & \quad , -\text{integr } F \ Fi, \text{integr } G \ Gi))). \end{aligned}$$

其中，“\”起λ抽象的作用，pi指全称量化。如， $\forall y. B$ 表示为 $\text{pi } Y \backslash B$ ，即Y在B中约束出现。

对训练实例：integr cos sin.

询问：integr X \ (3 * (expn X 2) + (cos X) F

推出： $F = X \backslash (3 * (\text{expn } X \ (2+1) \text{div } (2+1)) + \sin X)$

推广为：

$$\begin{aligned} & \text{integr } G \ Gi \Rightarrow \\ & \text{integr } x \backslash (A * \text{expn } X \ B + G \ X) X \backslash (A * \\ & \quad (\text{expn } X \ (B+1) \text{div } (B+1)) + Gi \ X) \end{aligned}$$

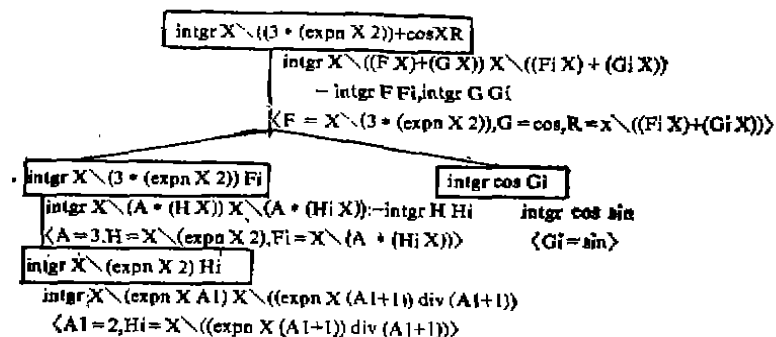


图4 高阶证明（解释）

其中，“ $\Rightarrow$ ”代替“ $:-$ ”，如， $A \Rightarrow B$ 即 $B :- A$ 。

由此，所得证明和推广证明分别如图4、图5所示。可见，高阶EBG的推广空间远远大于一阶EBG，这是因为高阶常量受变量替换的限制，在图2中，目标(kill X Y)是完全一般化的，而对于高阶，涉及命题的简单变量GG是完全一般化的。

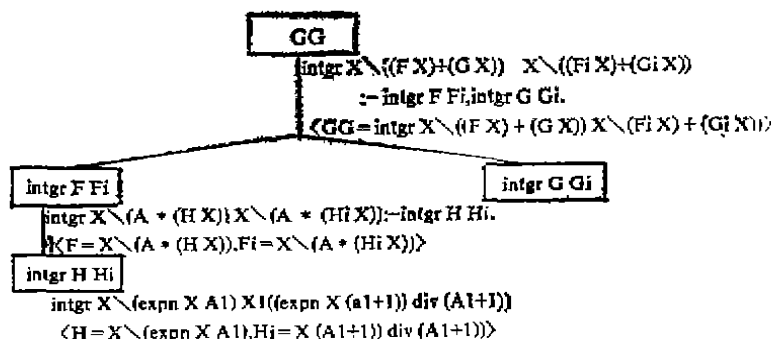


图5 高阶推广证明

不同于一阶证明，积分证明利用了高阶合一，它允许使用函数和一阶常量来例化变量。如， $(F \ a)$ 和 $(g \ a \ a)$ 可通过例化 F 为 $X \setminus (g \ X \ X)$ ， $X \setminus (g \ a \ X)$ ， $X \setminus (g \ X \ a)$ 或 $X \setminus (g \ a \ a)$ 而合一。由于这些例化不存在一个是另一个之例的情况，这说明了高阶合一的不确定性，这可通过对自由变元和约束变元施加限制来解决。

三、模态逻辑与EBG

除一阶逻辑，现在计算机科学界谈论得最多的要数模态逻辑了。它不仅是程序语义描述的有力工具，而且在知识表示方面表现出特别的优越性。根据对模态词的不同解释，分为不同的模态逻辑。较为古典的有真理论模态逻辑和认识论模态逻辑。

真理论模态逻辑的模态算子是“必然”和“可能”^[5]。

必然 A ，记为 $\Box A$ ，即：无论在什么场合（现实或想象的、非现实场合）均有事实 A ；

可能 A ，记为 $\Diamond A$ ，即：对某个场合（也许只有一个，也许仅是想象到的某个场合）有事实 A 。

在EBG中，由于只有域论规则才能用到推广证明中，故EBG仅依赖于域论和实例间的差别。为了区别，把域论子句前加模态必然算子“ $\Box$ ”，以指其后子句必然为真，或者说子句在任何可能状态或任何时间均真。而未加“ $\Box$ ”者仅在当前状态或当前时间可能真。

下面说明“ $\Box$ ”在前面“自杀”例中的应用。表示语言采用 $\lambda^{\Box}$ Prolog，是一种在 λ Prolog中引入模态必然算子的逻辑程序设计语言，其元解释器由美国CMU用 λ Prolog实现^[3]。在 $\lambda^{\Box}$ Prolog中，“ $\Box$ ”用box代替。例中

规则和事实集现表示为：

```
box(pi a\ (pi B\ (pi C\ (kill A B
:-hate A B, possess A C, weapon
C))))).
box (pi W\
(hate W W :-depressed W)).
box (pi U\ (pi V\
(possess U V :-buy U V))).
box (pi Z\
(weapon Z :-gun Z)).
depressed john.
buy john objl.
gun objl.
```

由于引用box，就不再需 λ Prolog的隐式全称量化。这是因为EBG算法认为子句 $\text{box}(\pi X \setminus (G \ X))$ 和 $\pi X \setminus (\text{box}(G \ X))$ 是不同的。

从询问(kill john john)，EBG产生推广：

```
kill X X, -depressed X, buy X C,
weapon C.
```

该规则包括所有的 X 和 C ，且相对于域论而言，必然真，因而最好将其表示为：

```
box (pi X\ (pi C\ (kill X X, -dep-
ressed X, buy X C, gun C))).
```

EBG在域论和实例间的区分相应于模态逻辑在必然真和可能真间的区分。加“ $\Box$ ”子句（域论）表示已经推广了的知识，它必然真。而未加“ $\Box$ ”者（实例）表示可能真或特定的知识。当训练实例子句被推广证明所排除时，它们就可被删除，这可以理解为语义上的状态改变。

假定在“自杀”例中，把最后一个程序语

句变为 (box (gun obj1)), 对于同一询问结果 box (pi X \ (depressed X, buy Xobj1 => kill X X)). 具有把推广固定到obj1上的作用。相应的证明与图2近似, 只是删除了证明树的最右分支。

通过把训练实例转化为域论, 使得推广更加细化。然而, 因推广依赖于(box(gun obj1))的有效性, 这却又不安全的。如果obj1不是gun, 那么导出规则将成为错误的。

在实现操作性方面, 已经从另一方面说明了子句的转换, 亦即, 从域论到实例(图3)。由于导出规则不会因改变训练实例而变得无效, 所以说此转换是良性的。

操作性的传统含义提供了允许更加局部删节推广证明的优点, 通过说明一子目标是操作性的, 则子目标下的整个分枝便可删去。在我们的构架中, 实现同样的删节需要把每一个在分支中应用的程序语句从域论中删去。推广证明树的内部结点子句可被抽象, 故“基于子句”的操作性准则更为有效。例如, 考察“自杀”例的一个新公式, 其第一个子句认为是训练实例,

pi A \ (pi B \ (pi C \ (kill A B
: -hate A B, possess A C, weapon
C))).

这就推导出推广的高阶证明。如图6所示, 得到了:

boxpi (GG, -depressed W, buy U V,
gun Z,

(hate W W, possess U V, weapon
Z => GG)).

其中, boxpi 是 (box(pi GG \ (pi W \ ...))) 的缩略, 非 λ Prolog成分。

高阶EBG允许从证明中抽象目标、函数和一阶对象, 得到更加一般的导出规则。由于GG中可有多种例化, 故导出规则更具一般性。通过对 λ Prolog提供GG的高阶量化功

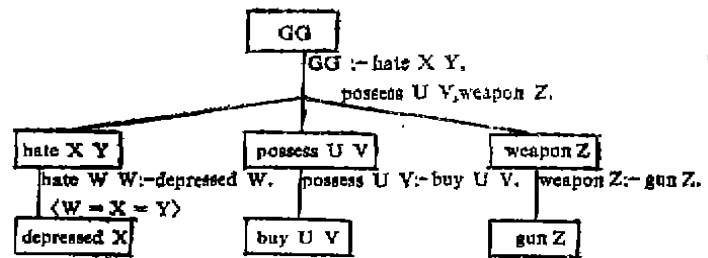


图6 改进的高阶推广证明

能, 以及在导出规则的前置条件中提供蕴含表达式; λ Prolog的表达已能力已超过目前的EBG语言。

另外, 认识论模态逻辑把模态算子解释为“知道”和“认可”。认识论模态逻辑系统, 对于解决像著名的S, P先生谜题的形式化问题是较完美的。同时, 期望它能够在AI中, 特别是在机器学习领域中得到应用。用其形式化“知道”和“认可”描述的转变问题, 也许具有较大潜力。在EBL中, 它究竟是否支持EBG也值得进一步研究。

四、结束语

1. 一阶和高阶

高阶EBG为解释学习提供了一种方法, 借助该方法, 就能在高阶语言描述的问题域中实现EBL。本文之例, 也许能够用一阶EBG实现, 但会变得十分复杂。为了处理高阶问题, 规则库的设计者们需要完成高阶语言所提供的表达能力。

2. 交互与EBG

人们常把解释推广誉为“加速学习”。在域论的演绎闭包中, EBG通过构造新规则来扩充域论, 也就是说, 这并没有增加新的知识, 但解决规则覆盖问题的能力却有希望改善, 而关于EBG的这一说法也非完全准确。当问题难处理 (intractable) 时, 则要求用户与系统交互。只有通过用户选择规则, 从而控制并减少搜索, 问题才有可能解决。在该过程中, EBG把知识从用户转化到学习系统中。从整体上看, 学习系统与用户的结合

一种基于面向对象环境的因果推理方法^{*)}

金 芝 李德意 胡守仁

(长沙工学院计算机系)

摘 要

It is clear that expert system's technology is one of AI's greatest successes so far. Currently we see an ever increasing application of expert systems. Yet there are also a number of well-recognized problems associated with the new technology. To alleviate the limitations of the current generation expert systems, the key is to develop the deep knowledge of the problem domain. This paper proposes an approach to the object-oriented causal reasoning about the implication of causal event stream, in arbitrarily complex causal network. The aim of our work is to develop an environment for capturing the deep knowledge in a special application domain and to support deep reasoning. This system has been prototyped in an object-oriented extension to Prolog. The impact that these two paradigms, logic and objects, have had on the design is discussed. Several example applications have also been presented to illustrate the extensibility of the environment.

一、引 言

当今专家系统已逐步成为人工智能中影

响最大、应用最广泛的领域之一。然而，尽管专家系统技术在具有良结构的狭窄领域内

完成了“加速”学习，但交互之后，学习系统就无需用户的协助，便可解决类似的问题了。

3. 进一步的工作

在本文介绍的构架中，有待于进一步探索动态操作性的引入，亦即，目标的操作性允许定义和再定义，以期得到更加一般的推广。同时，正如文中第三部分所提及的那样，新模态算子的引入，还值得进一步研究。

在本文完成过程中，作者自始至终得到徐家福先生的指导和鼓励。他仔细审阅并修改了全文，特此致谢！

主要参考文献

[1] 徐家福，软件笔谈，计算机科学，1990年第3

期

- [2] 费宗铭，吕建，徐家福等，机器学习，计算机科学，1991年第1期
- [3] Scott Dietzen, Frank Pfenning, High-order and Modal Logic as A Framework for Explanation-based Generalization, Tech. Rep. CMU-CS-89-160, Oct. 16, 1989
- [4] Gopalan Nadathur, Dale Miuer, An Overview of λ Prolog, Proc. of 15th Inter. Conf. & Symp. Vol.1, 1989
- [5] 王元元，计算机科学中的逻辑学，科学出版社，1989年
- [6] H. Hirsh, Explanation-based generalization in a logic programming environment, IJCAI-87, pp. 221-227

•) 国家高技术发展计划资助课题