增量式解释学习算法 EBG－plus*

郝继刚　　石纯一

（清华大学计算机系，北京 100084）

摘要　传统的解释学习(EBL)是通过单个实例进行学习的，学习结果往往带有实例本身的特殊性质. 知识求精能较正这一缺陷，但学习结果的效用不高. 本文结合了 EBL 方法和求精算法，提出综合多个实例的增量式解释学习算法 EBG－plus，学习质量随实例数目增加而单调上升，学习结果效用高，并能够自动改进领域知识的编码质量.

关键词　解释学习，知识求精，增量式学习.

1　问题的提出

以 EBG[1]和 EGGS[2]为代表的 EBL 算法，都是从单例出发进行学习. 虽然在学习过程中对实例的解释进行了概括，学习结果仍往往带有实例本身的特殊性质，只能得到一个描述目标概念的充分条件. 在训练实例不典型或者目标概念本身具有析取性质的时候，学习结果难于充分表达目标概念的本质特点，应用于概念的其它实例时，识别率不高.

为此，文献[3]将知识求精引入 EBL，以不断增加实例的增量学习方式来弥补单个实例的不足，从而提高学习质量. 当学到的概念描述用于识别新的实例发生错误时，可确定出错部分在原解释树中所对应的位置，并对其重新进行解释和概括，获得修正的概念描述.

这种求精在一定程度上提高了所学概念描述的质量，但文献[3]中尚有待改进之处：

(1)效用不高. 求精所得新的概念描述与原描述有两种关系：并存和取代. 新旧描述并存时，学习结果的识别率提高，但增加了选择与匹配的花费. 如原有学习结果 $A \wedge B \wedge C \wedge D \rightarrow G$，求精后又得两条新的描述规则 $A \wedge B \wedge C' \wedge D \rightarrow G$ 和 $A \wedge B \wedge C'' \wedge D \rightarrow G$. 三者并存，显然不如等价的一条描述规则 $A \wedge B \wedge (C \vee C' \vee C'')D \rightarrow G$ 的效用高. 新描述取代旧描述时，实际效果等于抛弃了原描述，不能保证原描述规则所能识别的概念实例，新的描述规则都能识别. 这样，学习质量不是单调增加，而是伴随不同特性的实例而时好时坏.

(2)一般只学到充分条件而不是充要条件，学到的概念描述的外延是目标概念外延的真子集.

(3)对于求精后所引起的知识库中相关领域知识的调整，没有给出具体办法.

本文在第 2 节给出有关概念的定义，学习算法 EBG－plus 见第 3 节，第 4 节给出实例

* 本文 1993－07－06 收到，1994－03－28 定稿

本文得到国家自然科学基金资助. 作者郝继刚，1967 年生，博士研究生，主要研究领域为人工智能，机器学习. 石纯一，1935 年生，教授，博士导师，主要研究领域为人工智能应用基础，知识工程.

本文通讯联系人：郝继刚，北京 100084，清华大学计算机系

说明,最后进行了讨论.

2 预备概念

为表达基于多例的EBL,对原来的EBL算法进行如下扩充:

(1)将EBL解释树扩充为与或树.

(2)将规则描述$C_1\wedge\cdots\wedge C_{n1}\rightarrow A$扩充为演绎关系$C_1,\cdots,C_{n1}\vdash_{Th}A$,其含义是:以逻辑程序$Th$作为推理规则集,则$A$是$C_1,\cdots,C_{n1}$的演绎结果.

(3)对解释树进行拓扑优化.

设OP为系统中的可操作谓词集合.记文献[1]中的EBG算法为过程EBG(G,I,Th,λ),其中目标概念G为一不可操作的谓词;实例I为可操作谓词集合$\{I_1,\cdots,I_{n2}\}$,$I_i\in OP$($i=1\cdots n_2$);领域理论Th为形如$A\leftarrow B_1\wedge\cdots\wedge B_{n3}$的规则的集合,$A,B_1,\cdots,B_{n3}$为谓词;可操作性准则$\lambda$要求$G$必须以$OP$中的谓词来描述.EBG过程构造$G$的解释树$T$,概括后获得规则$C_1\wedge\cdots\wedge C_{n4}\rightarrow G$,其中$C_1,\cdots,C_{n4}\in OP$.

将Th依G转换为与或树,树中每个结点对应一谓词(文中除非特别说明,不再对谓词A和结点A加以区分).对任一规则$A\leftarrow B_1\wedge\cdots\wedge B_{n3}$,自树中$A$结点引出$n_3$条与分枝,联系着子结点$B_1,\cdots,B_{n3}$;所有以$A$为结论的$n_5$条规则,对应的$n_5$个与分枝组构成$A$结点的$n_5$个或分枝.

当不考虑与(或)分枝间的谓词(规则)次序时,依G由Th生成的与或树,在逻辑和结构上同Th是一一对应的,称为关于G的Th与或树.从根G处开始,对树中每个结点,只保留一个或分枝,就得到一棵与树.若该与树的所有叶子谓词$L_1,\cdots,L_{n6}$都是可操作的,它就是依EBG算法得到的一棵解释树,并由此得到一可操作规则:

$$L_1\wedge\cdots\wedge L_{n6}\rightarrow G.$$

在关于G的Th与或树中,任一结点A,其任一或分枝中的所有与分枝所连结的子结点$C_1,\cdots,C_{n7}$,必对应着Th中的一条规则$C_1\wedge\cdots\wedge C_{n7}\rightarrow A$.现在放松这个约束条件,保证在$Th$中$C_1\wedge\cdots\wedge C_{n7}$能推导出$A$即可,即$C_1\wedge\cdots\wedge C_{n7}\vdash_{Th}A$.

定义1. 据此得到的与或树称为关于理论Th和目标G的证明与或树.

在下节的EBG—plus算法中,利用多个实例构成的解释树就是这种证明与或树.

定义2. 一定义在$\wedge$、$\vee$上的逻辑表达式F,化成析取范式后,利用分配律不断提取出公共合取项,直至在表达式内部再无公共的合取项可以提出.记这样的运算为规范操作,所得结果称$\prod$范式.对$\prod$范式F,其任意子表达式A在出现处的析取度μ,递归定义为:

(1)$\mu_F(A)=1$　　当$A=F$

(2)$\mu_F(A)=\mu_F(B)=\mu_F(C)$　　当$A=B\wedge C$

(3)$\mu_F(A)=\mu_F(B)-1=\mu_F(C)-1$　　当$A=B\vee C$

其中B,C为任意逻辑表达式.

例1:　　$F=(A\wedge B)\vee(A\wedge C\wedge D)\vee(A\wedge C\wedge E)\vee(A\wedge E\wedge G)$

规范化为$\prod$范式,有

$F_{\prod}=A\wedge(B\vee(C\wedge D)\vee(C\wedge E)\vee(E\wedge G))=A\wedge(B\vee(C\wedge(D\vee E))\vee(E\wedge G))$

其中 $\mu_F(A)=1,\mu_F(B)=2,\mu_F(C)=2,\mu_F(D)=3$，子表达式$(D\vee E)$中：$\mu_F(E)=3$，子表达式$(E\wedge G)$中：$\mu_F(E)=2,\mu_F(G)=2,\mu_F(E\wedge G)=2$

定义 3. 给定$\prod$范式 F，将其转换为一与或树，递归过程 $Tree(F)$定义如下：

(1)F 为根结点；

(2)对树中任一结点所对应的表达式 A，当其真子表达式(即除 A 本身以外的 A 的子表达式)$B_1,\cdots,B_{n8}$满足：$\mu_F(A)=\mu_F(B_1)=\cdots=\mu F(B_{n8})$，则自 A 结点引出 n_8 条与分枝，每一分枝连结一子结点 $B_i(i=1,\cdots,n_8)$.

(3)否则，若 $B_1,\cdots,B_{n8}$满足：$\mu_F(A)=\mu_F(B_1)-1=\cdots=\mu_F(B_{n8})-1$ 成立，则自 A 结点引出 n_8 条或分枝，每一分枝连结一子结点 $B_i(i=1,\cdots,n_8)$.

(4)当所有叶结点都不具有真子表达式，结束.

对任意定义在$\wedge$、$\vee$上的逻辑表达式 F，进行规范操作后得$\prod$范式 $F_\prod$，再施加 $Tree$ 过程，得与或树 T. 现在我们把 $F=F_\prod$作为领域理论(这里是 F 谓词符号，$F_\prod$是$\prod$范式)，把符号 F 作为目标概念，可以得到如下结论：

命题该与或树 T 是关于理论 $F=F_\prod$和目标 F(此处 F 为符号)的一棵证明与或树.

证明：(1)符号 F 为 T 的根结点.

(2)对 T 中任一结点 A，若由它引出了 n_9 条与分枝，连结着子结点 $B_1,\cdots,B_{n9}$，则 $B_1,\cdots,B_{n9}$是 A 的真子表达式，且满足：

$$\mu_F(A)=\mu_F(B_1)=\cdots=\mu F(B_{n9}).$$

由定义 2，$A=B_1\wedge\cdots\wedge B_{n9}$，则 $B_1,\cdots,B_{n9}\models_{F\prod}A$(即根据 $F_\prod$，A 是 $B_1,\cdots,B_{n9}$的逻辑结果). 根据一阶谓词演算的完全性定理，$B_1,\cdots,B_n\vdash_{F\prod}A$ 成立. 这是证明与或树中结点所连或分枝数为 1 的特殊情形.

(3)若由 A 引出的是 n_9 条或分枝，亦同理可证. 这时对应的是结点所连各或分枝都只含一个与分枝的特殊情形.

对 T 的树结构自根到叶进行递归论证，即得 T 是关于理论 $F_\prod$和目标 F 的证明与或树的结论. 证毕

一个概念可以有不同的定义和编码方式. 领域理论给出的目标概念的定义，是不可操作的. EBL 方法的思想即将目标概念可操作化(Operationalization)[4,5]. 由单例学得的可操作定义，在逻辑上与目标概念不等价. 通过求精得到的多条可操作定义规则，效用又不高. 等价的概念定义，不同的编码方式，对应着不同结构的与或树，对系统求解效率有着显著的影响. 将$\prod$范式对应的证明与或树还原为规则形式，树根作为目标概念，叶子作为可操作谓词. 我们发现，越具有典型性(覆盖更多例子)的可操作谓词，析取度越小(如析取度为 1 的谓词，必定出现于全部例子中)，在证明与或树中层次越高，用于例子识别时，首先被匹配. 所以，以$\prod$范式作为概念表示方式，有最高的整体效用. 与此相对照的是析取式：在$\prod$范式中析取度为 1 的原子谓词，只出现在对应的一条规则中；在析取式中，则出现于全部析取项，也出现在对应的所有规则中，匹配的花费自然高了. 第 1 节中给的例子，就是很好的体现. 基于这一结论，我们提出了提高学习效用的 EBG—plus 算法.

3　增量解释学习算法 EBG－plus

3.1　算法描述

(1)初始化概念描述 $D=\varnothing$.

(2)输入实例 I.

(3)若 D 正确识别 I,转(2);否则,得 D 中无法满足的谓词集 E(当 $D=\varnothing$ 时,令 $E=\{G\}$).

(4)选择谓词 $P\in E$,在解释树中确定对应结点.

(5)运行过程 EBG(P,I,Th,λ),若成功,转(8).

(6)若 $P=G$,解释失败,现有 Th 不完善,无法构造 I 的识别规则,退出.

(7)$P=parent(P)$,转(5).

(8)概括关于 I 的解释,得概念描述 D'. 令 $D=D\vee D'$,利用规范操作将 D 化为 $\prod$ 范式,转(3).

3.2　算法分析

从第 2 节的定义可知,一目标概念 G,给定领域理论 Th 后,与依据 Th 构造的、以 G 为根结点的与或树一一对应. 不妨设树中或分枝数为 m,自 G 开始到树叶,含或分枝的结点只保留一个或分枝,从而将其分解为 n 棵与树. EBG 过程中成功构造的解释树,必为这 n 棵与树之一,可看作是对应着 G 的一个"子"概念. 这样,在 Th 完善的前提下,我们将 G 划分成了互不相同的 n 个子概念(注意,不能说对 G 对应的实例空间进行了等价划分. 因为对于一个实例,可能构造出一个以上的解释树,从而同时成为多个子概念对应的实例,这样的例子集划分不满足分块的不相交性).

EBG 过程根据一个实例构造一解释树,概括后得到识别该树对应的子概念的充要条件描述,但仅为识别目标概念 G 的一充分条件描述. 输入新的实例,若隶属于同一子概念,系统能够成功地进行识别;若不对应该子概念,但能够对应其它子概念,系统将拒绝例子,导致判断错误,即 EBG 不考虑可能的选言项[6],学习结果不全面. EBG 这样做的原因是,在实际情况下,概念的全部例子往往是不会都出现的,出现的例子都是有代表性的,他们构成了系统关于目标概念的"经验". 经验过的例子是倾向于再度出现的,这一思想在类比学习(Analogical Learning)[7]和基于范例的学习(Examplar－Based Learning)中也都得到运用. 使用借助于实例学得的可操作的目标概念描述,能减少实例识别的花费,提高系统的效用.

但是,片面强调单个实例的使用,会产生负效果. 当目标概念本质上具有析取特性时,使用单个实例只能学到其中一个特性;而且学习效果还要受实例质量的影响. 即使后续的例子提供了校正和完善的机会,系统也不加考虑. 这是我们不希望的. 多个实例的 EBL 方法,可以消除单个实例本身的特殊性,使学习系统对实例不敏感,而且可以学习析取关系. 当由输入实例集概括得到上述全部 n 个概念描述并合并化简,便得到满足可操作性准则的、构成识别目标概念 G 充要条件的概念描述. 也就是说,EBG－plus 算法提供了获得与目标概念等价的可操作概念描述的可能性. 这时,用该可操作的定义取代原 Th,也就优化了 Th. 即使实例集合无法对应全部子概念,或因领域理论 Th 或可操作性准则 λ 等条件所限,达不到充分

必要，也可在时空代价不显著增加的前提下，提高学习结果的质量. 本算法可以视为在学习效果的质量和计算代价(如实例数量，时空消耗等)之间，即学习的效用和效率之间的折衷优化.

在文献[5]中，给出了谓词的可操作性计算公式. 但在算法实现时，因涉及谓词的析取关系，公式的计算仍是不可操作的，还必须借助于领域理论. 使用 EBG－plus 算法后，解释结构本身是与或树形式，这样无须领域理论，即可直接算得可操作性集合和可操作性界限.

为提高学习效率，在算法第(4)步中，在构造与或证明树 Tr 后，选择欲解释谓词的方式为：将 E 中各元素按析取度由大到小次序排列，然后循环：在第 i 次循环中，置前 i 个谓词 $P_1,\cdots,P_i$ 为可满足，改写 D；若 D 能识别 I，则谓词 P_i 被解释，否则恢复 D.

因为算法是增量式的，在学习结果识别实例发生错误时才继续学习，所以并不需要许多实例. 令 EBG 的时间复杂度为 $O(\mathrm{EBG})$，则 EBG－plus 算法的复杂度为 $N*O(\mathrm{EBG})$，N 为增量学习过程中识别实例发生错误的次数.

4 算法应用举例

给出一个简单例子的算法运行过程，体现其增量式学习和对领域理论自动调整的特点.

例 2： $G=\{A\}$，$OP=\{D,E,F\}$

λ：A 必须用 OP 中的谓词来描述

Th：$A(X)\leftarrow B(X)$

$B(X)\leftarrow C(Y)\wedge D(X,Y)$

$C(X)\leftarrow E(X)$

$C(X)\leftarrow F(X)$

Th 对应的与或树见图 1.

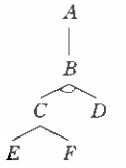

图1 初始Th

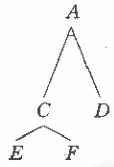

图2 优化的Th

首先输入 $I_1=\{D,E\}$，调用 $\mathrm{EBG}(A,I_1,Th,\lambda)$

学习结果：$A\leftarrow D\wedge E$

再输入 $I_2=\{D,F\}$，描述中 E 不满足，调用 $\mathrm{EBG}(E,I_2,Th,\lambda)$，失败；

再调 $\mathrm{EBG}(C,I_2,Th,\lambda)$，概括得：$A\leftarrow D\wedge F$；

与原描述合并，规范化，新的概念描述为：

$$A\leftarrow(D\wedge E)\vee(D\wedge F)=D\wedge(E\vee F)$$

新概念描述与目标概念逻辑等价，可得优化的 Th：

$A\leftarrow C\wedge D$　　$C\leftarrow E$　　$C\leftarrow F$

对应的与或证明树，见图 2.

5 讨 论

本文将 EBG 算法和求精算法进行合并，给出增量式的多例解释学习算法 EBG－plus，具有如下特点：

(1)对解释结构进行扩充，在与或树上进行解释和概括，合并可能的选言项，从而提高学

习的质量和效用，逼近识别目标概念的充要条件描述.

(2)综合 EBL 与 SBL 的优点. 仅由一个实例便可学到目标的概念描述；而随着训练例数目增大，识别出错时，又及时改善学习结果，提高学习质量，是增量式学习.

(3)利用学习结果优化领域理论，调整编码质量不高、存在冗余和层次紊乱的领域理论，可视为对领域理论的优化和"编译"，如第 4 节中例子所示.

(4)对实例自身所具有的特殊性不敏感.

致谢　成文过程中，得到李膺春老师、张旗、于津、王彤、罗毅同志的帮助，与柳常青同志的讨论也受益非浅，在此表示感谢!

参考文献

1 Mitchell T M *et al*. Explanation_based generalization: a unifying view. Machine Learning, 1986,**1**(1):47－80.
2 Mooney R J, Bennett S. A domain independent explanation－based generalizer. AAAI－86, 1986.
3 石纯一，黄毅青. 解释学习中的求精算法及其实现. 知识工程进展(1991)，北京：中国地质大学出版社，1991.
4 Keller R M. Defining operationality for explanation－based learning. AI, 1988,**35**:227－241.
5 邹晨东，石纯一. 解释学习的可操作性. 第二届中国人工智能联合学术会议论文集(CJCAI'92), 1992. 301－308.
6 石纯一等. 基于解释的机器学习方法. AI 基础理论研讨会，1992.
7 Michalski R, Carbonell J, Mitchell T M eds. Machine learning, 1986.

INCREMENTAL EXPLANATION－BASED LEARNING ALGORITHM EBG－PLUS

Hao Jigang　Shi Chunyi

(*Department of Computer Science, Tsinghua University, Beijing* 100084)

Abstract　Explanation－based learning (EBL) methods learn from single training example. The learning result often bears the example's own speciality. The knowledge refinement algorithm can rectify the speciality in EBL, but with rather low utility. This paper combines EBG and refinement algorithm, gives an incremental learning algorithm——EBG－plus, which can take advantage of many examples. While maintaining high utility, the authors get better result as new instances are met. By the way, the quality of domain knowledge can be automatically improved.

Key word　EBL, knowledge refinement, incremental learning.