第6卷第4期 软件学报 Vol. 6, No. 4
1995年4月 JOURNAL OF SOFTWARE Apr. 1995

基于组合—分解的机器发现方法*

李爱中 刘叙华

(吉林大学计算机科学系,长春 130023)

摘要 本文提出了一个用以刻划综合与分析的发现过程的机器发现方法,定义了组合—分解算子、组合—分解函数、函数发现和函数维护等概念,给出了组合—分解函数的发现算法和维护算法.作为应用,本文探讨了关于知识发现和知识维护的函数型知识发现过程.

关键词 组合,分解,函数,算子,发现,维护.

机器学习(Machine Learning)就是用计算来表示人类的学习过程,而机器发现(Machine Discovery)则是用计算来表示人类的发现过程.

关于计算,从理论上已经有了许多通用的计算模型(如一阶谓词演算、递归函数、Turing机和λ—转换演算等等).这些模型的计算能力是现行计算机的计算能力的极限.然而,在使用这些模型的计算来表示学习和发现的进程中却使人感到失望.作者认为其根源在于对学习和发现过程的认识并在此基础上建立恰当的计算模型来表示学习和发现.

Turing机是现代计算机的理论模型."Turing所作的工作是分解人的计算动作,得到若干简单的操作,它们在性质上是明显的、机械的,并能够证明能把它们组合起来执行任意复杂的机械操作"[1].从Turing的工作可以看出分析(或分解)和综合(或组合)在Turing机中的体现.但Turing的工作仅是建立在一次性的分析基础上的多次综合,即"分析—综合—新的综合…",而缺少对新的综合(或综合)的新的分析.

作者认为学习和发现是一个分析(或分解)—综合(或组合)—新的分析—新的综合等的不断过程.为此,作者进行了所谓的基于组合—分解的机器发现方法的研究工作.

1 基于组合—分解的机器发现方法

除非特别声明,本文所讨论的数是指自然数(0,1,…),函数是指数论函数,算子是定义的数论函数集上的函数.

1.1 组合—分解算子

定义 1. 如果二元算子OS与一元算子OL、OR之间满足下列条件(叫作组合—分解条件):对于一切全函数$l(X_1,\cdots,X_n)$,$r(Y_1,\cdots,Y_m)$,$OL(OS(l(X_1,\cdots,X_n),r(Y_1,\cdots,Y_m)))=l$

* 本文1993-03-09收到,1993-07-12定稿

本文工作得到国家自然科学基金和国家863计划的支持.作者**李爱中**,1963年生,副教授,主要研究领域为机器学习的计算理论及其应用、管理决策.**刘叙华**,1937年生,教授,主要研究领域为定理机器证明,逻辑和人工智能.

本文通讯联系人:李爱中,北京 100044,北方交通大学计算机科学系

$(X_1,\cdots,X_n)$, $OR(OS(l(X_1,\cdots,X_n),r(Y_1,\cdots,Y_m)))=r(Y_1,\cdots,Y_m)$;则$(OS,OL,OR)$叫作(二元)组合—分解算子.

例如:从原始递归式[2]中可以抽象出一个组合—分解算子(OS,OL,OR):

原始递归式:

$$\begin{cases}f(0,X_2,\cdots X_n)=A(X_2,\cdots X_n),\\ f(X_1+1,X_2,\cdots X_n)=B(X_1,\cdots,X_n,f(X_1,\cdots X_n));\end{cases}$$

$$\begin{cases}f(X_1,\cdots X_n)=OS(A(X_2,\cdots X_n),B(X_1,\cdots X_n,f(X_1,\cdots X_n));\\ OL(f(X_1,\cdots X_n))=A(X_2,\cdots X_n);\\ OR(f(X_1,\cdots X_n))=B(X_1,\cdots,X_n,f(X_1,\cdots,X_n)).\end{cases}$$

1.2 组合—分解函数集

定义 2. 由初始函数集利用有限个组合—分解算子所生成的组合—分解函数集由下列规则确定:

(1)初始函数集为形如:$a_0+a_1X_1+\cdots a_nX_n$(a_i 为常数,$i=0,1,\cdots,n$)的线性函数所构成;初始函数均是组合—分解函数;

(2)有限个组合—分解算子为$\{(OS_i,OL_i,OR_i)/i=1,\cdots,k\}$,$k$ 为一个常数;

(3)若 $l(X_1,\cdots,X_n)$,$r(Y_1,\cdots,Y_m)$是组合—分解函数,(OS_i,OL_i,OR_i) $(1\leqslant i\leqslant k)$为一个组合—分解算子,则 $OS_i(l(X_1,\cdots,X_n),r(Y_1,\cdots,Y_m))$亦为一个组合—分解函数.

1.3 组合—分解函数的编号

定义 3. 组合—分解函数的编号由下列规则确定:

(1) 初始函数 $a_0+a_1X_1+\cdots a_nX_n$(a_i 为常数,$i=0,1,\cdots,n$)的编号均定义为 0;

(2) 若 $l(X_1,\cdots,X_n)$和 $r(Y_1,\cdots,Y_m)$是组合—分解函数且编号分别为 nl 和 nr,(OS_i,OL_i,OR_i) $(1\leqslant i\leqslant k)$为一组合—分解算子,则组合—分解函数 $OS_i(1(X_1,\cdots,X_n),r(Y_1,\cdots,Y_m))$的编号定义为 $pg(i,nl,nr)+1$;其中 $pg(X,Y,Z)=S(X,S(Y,Z))$,$S(X,Y)$是康托编号[3]且 $L(S(X,Y))=X$,$R(S(X,Y))=Y$.

显然,在常数确定的情况下,一个编号确定唯一的一个函数.每个函数均有编号.但并非每个自然数均是编号.为此,我们给出了判定任意一个自然数 *Number* 是否为编号的算法 *Decide* (*Number*).

```
Decide(Number): Boolean;
    if Number=0 then return(Ture)
    else begin i1=L(Number-1);
              NL1=L(R(Number-1));
              NR1=R(R(Number-1));
              return(1≤i≤k and Decide(NL) and Decide(NR))
         endbegin
    endif.
```

在本文中把关于函数的讨论代之以关于函数编号的讨论.

1.4 函数发现及其算法

定义 4. 已知关于自变量 $X_1,\cdots,X_n$ 和因变量 Y 的一组数据 $Date=\{(X_{1,i},\cdots,X_{n,i},Y_i)/i=1,\cdots,m\}$,求一个编号最小的组合—分解函数 $f(X_1,\cdots,X_n)$满足:对所有 $i=1,\cdots,m$,Y_i

$=f(X_{1,i},\cdots,X_{n,i})$的过程叫作组合—分解函数的发现(简称函数发现);并称函数 $f(X_1,\cdots,X_n)$解释数据 *Data*.

函数发现算法是建立在下列判断编号为 *Number* 的函数 $f(X_1,\cdots,X_n)$是否解释数据 *Data* 的判定算法 *Explain*(*Number*,*Data*)基础之上的.

```
Explain(Number,Data): Boolean;
    if Number=0 then
    若可以确定常数 a0,a1,…,an 使 Yi=a1+a1X1,i+…+anXn,i对所有 i=1,…,m 成立,则 return (Ture);
    否则,return(False)
    else begin i:=L(Number-1);
              DL:=OLi(Data);
              DR:=ORi(Data);
              NL:=L(R(Number-1));
              NR:=R(R(Number-1));
              return (Decide(i) and Explain (NL,DL) and Explain (NR,DR))
         endbegin
    endif.
```

函数发现算法 *Discover*(*Data*,*Number*)是在假设存在的编号为 *Number* 的函数解释数据 *Data* 的条件下给出的.

```
Discover (Data, Number);
begin       N:=-1; /* N 为整数 */
    Loop: N:=N+1;
        if Decide(N) and Explain(N,Data) then Number:=N
        else goto Loop
        endif
endbegin.
```

1.5　函数维护及其算法

定义 5. 已知关于自变量 $X_1,\cdots,X_n$ 和因变量 Y 的一组数据 $Data=\{(X_{1,i},\cdots,X_{n,i},Y_i)/i=1,\cdots,m\}$,以及发现的能解释数据 *Data* 的编号最小的组合—分解函数 $f(X_1,\cdots,X_n)$,但 $f(X_1,\cdots,X_n)$不能解释数据 $data=\{(X_{1,m+1},\cdots,X_{n,m+1})\}$;求出一个编号最小的组合—分解函数 $f'(X_1,\cdots,X_n)$解释数据 *Data* 和 *data* 的过程叫作组合—分解函数的维护(简称函数维护).

下列维护算法 *Maintain*(*Number*,*Data*,*data*,*Number'*)将根据 *data* 对由 *Data* 所发现的函数编号 *Number* 进行修正而得到的新的编号 *Number'*.

```
Maintain(Number,Data,data,Number');
    if Explain(Number,data) then Number':=Number
    else
      if Number=0 then Discover(Data ∪ data, Number')
      else
        begin
          i:=L(Number-1);
          DL:=OLi(Data);
          dl:=OLi(data);
```

```
        DR_i:=OR_i(Data);
        dr_i:=OR_i(data);
        NL_i:=L(R(Number-1));
        NR_i:=R(R(Number-1));
        Maintain(NL, DL, dl, NL');
        Maintain(NR, DR, dr, NR');
        Number=pg(i, NL', NR')+1
      endbegin
    endif
  endif
```

1.6 函数发现

定义 6. 函数发现是一次初始性函数发现和多次不断的函数维护的综合过程.

至此,我们已经初步给出了基于组合—分解的机器发现方法的概要.

2 基于组合—分解的机器发现方法的应用

本文仅探讨了基于组合—分解的机器发现方法在刻划函数型知识发现过程中的应用.

首先,人们经过大量的实践,积累了一些关于某个问题的初始经验.而这些初始经验表现为大量事实而不是知识.然后,人们试图从这些经验中归纳出抽象的知识,以此构成初始假说.我们称初始假说的构成过程为初始知识发现.初始假说仅是初步的并允许有错误.知识是相对的,其完整性和正确性亦是相对的.假说一旦被建立,既对实践具有指导作用又要经受实践的检验.若假说能够解释新的事实,则被继续接受;否是,此假说将被一个能解释迄今所有事实的新假说所替代.我们称这个用新假说来替代旧假说的过程为知识维护.知识维护是一个不断进行的反复的过程[4,5].知识永远是假说.

若我们把初始经验表示为关于自变量 $X_1,\cdots,X_n$ 和因变量 Y 的一组数据 $Data=\{(X_{1,i},\cdots,X_{n,i},Y_i)/i=1,\cdots,m\}$,知识表示为关于自变量和因变量的函数;知识发现即为发现能解释数据 $Data$ 的函数 $f(X_1,\cdots,X_n)$ 的函数发现;而当函数 $f(X_1,\cdots,X_n)$ 不能解释新事实 $data=(X_{1,m+1},\cdots,X_{n,m+1},Y_{m+1})$ 的时候,需求一新函数 $f'(X_1,\cdots,X_n)$ 以解释 $Data$ 和 $data$ 的知识维护过程即为函数维护.这样函数发现和不断的函数维护则可以刻划知识发现的过程.

3 结 论

本文从对综合与分析的发现过程的讨论出发,定义了组合—分解算子、组合—分解函数、函数发现和函数维护等概念,给出了组合—分解函数的发现算法和维护算法;从而提出了基于组合—分解的机器发现方法.作为该方法的一个应用,本文讨论了知识发现和知识维护的函数型知识发现过程.

需要指出的是:本文仅是基于组合—分解的机器发现方法的研究的开始,有许多问题有待进一步研究,例如组合—分解函数集的表达能力及其与递归函数集等的关系问题等等.最后,作者相信本文给出的基于组合—分解的机器发现方法将在机器发现、机器学习、知识获取、数据库、软件工程等方面有所应用.

参考文献

1 王浩.数理逻辑通俗讲话.北京:科学出版社,1981:198—199.

2 莫绍揆,王元元.可计算性理论.北京:科学出版社,1987.

3 罗莎·培特.递归函数论.北京:科学出版社,1958.

4 李爱中,洪家荣,黄梯云.机器发现的递归函数法.计算机学报,1993,**16**(**8**):577—582.

5 李爱中,统计信息,递归和机器发现.计算机科学,1993,**20**(**1**):25—28.

A COMPOSITION—DECOMPOSITION BASED APPROACH TO MACHINE DISCOVERY

Li Aizhong　Liu Xuhua

(*Department of Computer Science, Jilin University, Changchun* 130023)

Abstract　In order to describe the discovery process of synthesis and analysis, a composition-decomposition based approach to machine discovery is presented in the paper, which defines the concepts of composition-decomposition operator, composition-decomposition function, function discovery, function maintenance and so on, and this paper also gives a function discovery algorithm and a function maintenance algorithm. As an application of the approach, the process of functional knowledge discovery is studied in terms of function discovery and function maintenance.

Key words　Composition, decomposition, function, operator, discovery, maintenance.