

反应块及其在类比推理中的应用*

李红 徐立本 张世伟

(吉林大学商学院管理科学系 长春 130023)

摘要 本文讨论了类比在问题求解中的应用问题,给出了应用反应块识别类比源以及自动生成反应块的算法.本文还给出了一种解法序列分割方法,用于类比源的获取及存储.这些思想和方法已在符号积分求解与学习系统 ISLS(integration solving and learning system)中实现.

关键词 机器学习,类比学习,问题求解.

近年来,类比推理这种人类最常采用的思维方式,以其独具的魅力吸引了越来越多的AI研究者的关注,做了大量的工作,取得了一定的进展.

类比推理的主要思路是:由于目标和源在某些方面具有一定的相似性,因而我们有理由把关于源的某些知识经过一定的转换处理过程映射到目标中去.围绕这一思路还有许多工作要做.例如:如何在记忆中搜索到一个最贴切的类比源;转换源中的哪些知识到目标中去,如何转换;如何判断类比是成功的还是失败的;类比之后如何学习新的知识等等.

Hall 在文献[1]中曾将类比框架归纳成4个过程:(1)识别:寻找一个合适的类比源.(2)建立:在目标及源之间建立类比映射,实现知识的传递.(3)评价:检查所应用的类比,所传递的知识是否是合理的、真实的、正确的.(4)充实:类比之后的学习问题.其中建立阶段是类比问题研究者关注最多的部分,在其上做的工作也最多,这是因为在不同的领域间进行类比时,要选择源域中的部分知识(而非全部)映射到目标域中,如太阳系和原子相似,传递太阳系的哪些性质给原子对机器来说是一个很难确定的问题.但是当我们关注的是同一领域内的类比,特别是在同一领域内应用类比方法进行问题求解时,第1阶段——识别过程及最后阶段——充实过程的困难及重要性似乎远远超过了建立阶段,对于数学领域中的问题求解及学习系统来说更是如此.

本文将符号积分求解与学习系统 ISLS(integration solving and learning system)为例,讨论如何将类比应用到同一领域内的问题求解过程中去.虽然我们讨论的背景是符号积分,但是所给出的理论、方法也适用于其它一些问题求解领域.

* 本文研究得到国家高技术发展计划的资助.作者李红,女,1972年生,硕士研究生,主要研究领域为机器学习,专家系统,管理科学.徐立本,1935年生,教授,博士导师,主要研究领域为机器学习,专家系统,经济管理软件.张世伟,1962年生,副教授,主要研究领域为机器学习,专家系统,经济管理软件.

本文通讯联系人:徐立本,长春 130021,长春市朝阳区清华胡同8号201

本文 1995-07-07 收到修改稿

1 类比源的获取与存储

当 ISLS 系统遇到一个不会解的问题时,它将进入求解模块,请求专家输入该问题的解法序列.对该解法序列我们采用根据不同情况做不同分割再存储的处理方法.下面在具体给出我们的方法之前,首先给出在我们类比库中每一个类比源的知识表示形式.

每一个类比源采用框架形式表示,它包括以下各槽:

No: 类比源编号.

Block: 反应块.它是处于特殊重要地位的问题状态的子结构,以前缀表的形式给出.该槽最初为空,类比之后将由类比的充实过程动态地产生.

apstr: 附属子结构集.它们是问题状态的子结构,但是不能体现问题状态的本质特征,处于附属地位.以表的形式给出.同 Block 一样动态地生成.

substr: 一般子结构集.问题状态的一般子结构,其重要性介于前面二者之间,最初,问题状态的所有子结构都存储在 substr 槽内.随着学习的深入,有些转入 Block 槽,有些转入 apstr 槽.

opname: 针对该源问题状态所实施的操作名称.

opmess: 关于如何实施操作的具体信息.

属于分部积分操作时给出 $u=f(x), v=g(x)$;

属于结构替换操作时给出 $y=f(x), x=g(y)$;

example: 类比实例槽

在积分领域里,我们把所有的操作归纳成 3 大类.重写表达式操作:对表达式进行恒等变形;结构替换操作:用一个结构式代替问题的某个子结构式,达到简化问题状态的目的;分部积分操作.

下面给出解法序列分割及存储算法,它根据不同情况,对解法序列做不同分割,然后再存储.

输入: $State(i) (i=0, 1, \dots, t)$ $OP(i) = (i=1, 2, \dots, t)$

(对 $State(i)$ 实施 $OP(i+1)$ 操作得 $State(i+1)$)

步骤: ① $j=1$.

② 若 $j>t$ 则返回;

③ 若 $OP(j)$ 是结构替换或分部积分操作,则把 $State(j-1)$ 及 $OP(j)$ 的有关信息写人类比库,生成一个新的类比源.

$j=j+1$ GOTO ②

④ 若 $OP(j)$ 是重写表达式操作,则寻找一个满足下列条件的最小整数.记为 $flag, flag>j$ 且 $OP(flag)$ 不是重写表达式操作,若找不到这样的 $flag$,则 $flag=t+1$.于是我们得到重写表达式操作子序列 $OP(j), OP(j+1), \dots, OP(flag-1)$ 及相应的问题状态子序列 $State(j-1), State(j), \dots, State(flag-1)$.运用解释方法将之抽象成宏操作存储起来(其具体办法超出了本文的讨论范围).

$j=j+1$ GOTO ②

例子:

问题状态 $\int x^2 \ln \sqrt{x} dx$

将结构替换 $y = \sqrt{x}, x = y^2$ 存入类比库中,其形式如下:

No: 4

```

Block: Nil
apstr: Nil
substr: (( $\wedge x$  2)( $\wedge x$ (/1 2))
         (ln( $\wedge x$ (/1 2))
          (* ( $\wedge x$  2)(ln( $\wedge x$ (/1 2))))
opname: 结构替换 jgth
opmess: (( $\wedge x$ (/1 2))( $\wedge y$  2))
examples: Nil

```

2 类比源的识别和类比映射的建立

对于类比最初究竟是由什么激发的这一问题, Thargard 等人^[2]以及国内的一些研究者是这样看的: 类比最初是由语义激发的, 即受目标中对象及谓词的语义信息的启发, 激活了记忆中类似的对象和谓词, 进而激活了对象和谓词所在的源, 在所有被激发的源中选一个证据最多的作为类比源。

我们认为上面的方法更多地适合于不同的领域间的类比。如水流和电流, 流这个词把它们联系起来, 但是对于某些领域内的类比来说, 情况却不是这样, 在我们的研究领域——数学领域内, 类比无法由语义激发, 这是因为问题状态——被积表达式是由常量、变量、运算符、函数符组成的, 它们做不同的组合, 生成不同的表达式, 如果类比由语义激发, 那么所有合乎语法要求的数学表达式都将被激活。

我们认为类比是由反应块引动的。什么叫反应块? 人们在解决若干个相似的问题之后, 对这类问题在头脑中会逐渐形成一种相似的“子状态”。这些子状态的记忆强度较高, 处于比较活跃的状态。当人们面临新的问题状态时, 问题状态的各个子状态在人们头脑中处于同等重要层次上。但是当人们在解决一些相似问题时, 某个子状态的重要性会变得突出起来, 并使解题者的注意力集中在该子状态上, 使之成为处理的核心。我们称之为反应块。例如求积分 $\int x^2 \ln \sqrt{x} dx$ 时, 由于记忆中有 \sqrt{x} 这个反应块, 从而使 \sqrt{x} 的重要性变得突出起来, 进而做出结构替换 $y = \sqrt{x}$ 这样的决策, 解决了问题的一个难点。我们称上面的方法为“反应块识别法”。

如上所述, 反应块不是最初就有的, 它是在积累解题经验的过程中逐渐生成的。那么在反应块生成之前类比是由什么诱发的呢? 我们认为此时类比是上语法结构诱发的, 即认识到目标和某个源之间具有相似的子结构, 从而为二者相似积累了证据, 当这种证据积累到一定程度之后, 该源被激活成为候选源。在所有候选源中挑选证据最多的一个。我们称这种思想为“语法结构激活原理”。

下面给出具体的激活算法:

输入: 问题状态, 即被积表达式 $f(x)$ 及其子结构 $S = \{S_1, S_2, \dots, S_n\}$; 类比库

步骤: ① $H = NULL$, H 为候选类比源集。

② 把类比库中的类比源分成 2 类,

有反应块的, $A = \{A_1, A_2, \dots, A_i\}$

无反应块的, $B = \{B_1, B_2, \dots, B_j\}$

③ 对 $A_i \in A (i=1, 2, \dots, t)$ 做如下处理:

若存在 j 使 $S_j \in s$, 且 S_j 与 A_i 的反应块是相似的子结构, 则调用打分算法 $F(S, A_i)$, 返回匹配度 d_i , 同时生成匹配信息表 *bridge*, 记录匹配信息, *bridge* 以配对表的形式表示.

$H = HU\{\langle A_i, d_i \rangle\}$

否则, 否定 A_i .

④ 对 $B_i \in B (i=1, \dots, s)$ 做如下处理:

直接调用打分算法 $F(S, B_i)$, 返回匹配度 d_i , 同时生匹配信息表 *bridge*, 以配对表的形式表示.

⑤ 若 $d_i > \epsilon$, (ϵ 为阈值, 是一个经验参数) 则 $H = HU\{\langle B_i, d_i \rangle\}$

否则否定 B_i .

⑥ 在 H 中挑选 d 值最高的候选源, 作为最后选定的类比源输出, 同时输出相应的匹配信息表.

在 ISLS 系统里设计了一个打分子程序 F , 用以计算 2 个问题状态的匹配度, 在这个子程序中首先寻找 2 个状态的各个子结构间的最佳配对, 并为该配对打分. 其中反应块间的配对分值高于一般子结构间的配对分值, 一般子结构间的配对分值高于附属子结构间的配对分值. 这种打分方法体现了不同类型的子结构重要程度不同的思想. 此外配对分值的高低还依赖于 2 个子结构的相似程度——是相同还是同属一超类等等.

由以上的分析可知: 如果类比源生成了反应块, 那么激活它或者否定它变得容易起来, 同时也使激活算法效率提高. 因此反应块在类比检索中起着不同寻常的作用.

关于反应块的生成请参见第 4 节.

3 类比的评价

ISLS 求解一个问题时, 它可能应用了类比且不止 1 个, 整个积分求解过程可能以成功结束——求出了被积函数的原函数, 也可能以失败返回. 无论哪种情况, 我们都需要知道它所应用过的每一个类比是成功的, 还是失败的. 这些信息将被存入人类比源的类比实例槽内, 以供类比充实过程所用.

如何识别类比的成功或失败是类比评价过程的任务. 在 ISLS 系统中评价部分不是一个单独的模块, 它融于类比求解过程之中. 这里只是为了讨论的方便而将其独立出来.

在类比求解过程中, 我们使用了回溯技术, 根据求解过程所经过的路径不同, 我们将求解过程分为 3 类:

(1) 未经过任何回溯, 求解过程以成功结束.

在这种情况下, 所使用的所有类比均被标记为成功, 存入相应类比源的类比实例槽内.

(2) 经过回溯, 求解过程以成功结束.

系统沿着某一路径向下搜索, 如果走到叶结点, 仍未能找到问题的解, 那么系统自底向上回溯, 每回溯掉一个类比, 系统就改用弱方法求解. 这种向上回溯的过程不断继续下去, 直到用弱方法在一定资源限制条件下搜索到问题的解为止. 在回溯过程中, 每回溯掉一个类比, ISLS 就将其标记为“失败”, 并存储在相应类比源的类比实例槽内. 回溯过程结束后, 未被回溯掉的类比被记为“成功”, 存入相应类比源的类比实例槽内.

(3) 回溯掉了所有类比仍不能用弱方法在指定的资源限制条件下将问题解出来, 求解过程以失败返回.

实际上这种情况是(2)的一种特例. 此时不仅所有的类比被标记为“失败”存储起来, 系

统还将进入求教模块,请求专家输入该问题的解法序列,进而开始了新一轮的类比源的获取与存储。

4 类比的充实

前几个阶段在源与目标之间建立了映射,并在该映射的支持下做了类比推理.接下来的工作是如何把刚才的解题的经验充实到知识库中,以获得更高层次上的知识,这是本节要讨论的问题。

系统在问题求解过程中使用的每一个类比无论成功或失败,都将被存储在类比源的类比实例槽内.给定一个阈值,当槽内实例个数超过该阈值时,开始类比的充实过程。

其步骤如下:

对于类比源 S_0 做如下处理:

(1) 把 S_0 的类比实例槽内的类比实例分为 2 类:

成功的类比实例集 $succ = \{S_1, S_2, \dots, S_m\}$

失败的类比实例集 $fail = \{S'_1, S'_2, \dots, S'_n\}$

(2) 对 S_0 及集合 $succ$ 做如下处理:

① 寻找相似子结构集 $subset$, 满足若 $str_0 \in S_0$ 的一般子结构槽, $str_i \in S_i$ 的一般子结构槽 ($i=1, 2, \dots, m$) str_0 分别与 str_i 对应相似, 则对 $str_0, str_1, \dots, str_m$ 做尽可能保守的抽象, 使之仅囊括 $str_0, str_1, \dots, str_m$ 即可, 设抽象后的结果为 str , 则将 str 压入 $subset$.

如此继续, 直到找出所有满足条件的 str 为止。

② 在 S_0 的反应块槽内加入 $subset$ 的所有元素。

③ 从 S_0 的一般子结构槽内删除已被抽象到 $subset$ 中的所有子结构。

(3) 对 S_0 及集合 $fail$ 做如下处理:

① 寻找相似子结构集 $subset$, 满足:

若 $str_0 \in S_0$ 的一般子结构槽 $str_i \in S'_i$ 的一般子结构槽 ($i=1, 2, \dots, n$),

则对 $str_0, str_1, \dots, str_n$ 做如下抽象:

a. 常量抽象成 C , 不同位置出现的不同常量分别用 $C(i)$ 代入;

b. 三角函数符抽象成 $trig$, 不同函数名分别用 $tgrig(i)$ 代入;

c. 指数对数抽象为 lex , 不同函数名分别用 $lex(i)$ 代入;

d. 反三角函数抽象成 $atrig$, 不同函数名分别用 $atrig(i)$ 代入;

设抽象后的结果为 str , 则将 str 压入 $subset$ 中;

如此继续, 直到找出所有满足条件的 str 为止。

② 在 S_0 的附属子结构槽内压入 $subset$ 中所有元素。

③ 从 S_0 的一般子结构槽内删除已被抽象入 $subset$ 中的所有子结构。

(4) 删除 S_0 类比实例槽内中的所有类比实例, 给该槽置空。

由以上的算法可以看出: 在类比的应用过程中产生了很多有用的信息, 其中有些信息被用来产生反应块, 有些信息被用来生成附属子结构集, 反应块及附属子结构集的生成将有助于提高类比检索算法的运行效率及效果, 使之能更快地发现最合适的类比源。

5 结 论

上面从类比源的获取及存储、类比的检索、映射的建立、类比的评价和类比的充实等几个方面讨论了如何将类比技术应用于同一领域内的问题求解问题。我们认为,在类比检索中,不同的子结构在相似性匹配中的贡献不同。因此,我们给出“反应块识别法”,并在不定积分求解的系统中实现了该方法。同时认识到类比库中的知识体系对于类比检索的效率也有很大影响,如何管理类比库,如何消除冗余,如何消解冲突是我们进一步研究的课题。

参考文献

- 1 Hall R P. Computational approaches to analogical reasoning: a comparative analysis. *Artificial Intelligence*, 1989, 39:29~120.
- 2 Thagard P, Holyoak K J, Nelson G. Analog retrieval by constraint satisfaction. *Artificial Intelligence*, 1990, 46: 259~310.

REACTING—BLOCK AND ITS APPLICATION IN ANALOGY

Li Hong Xu Liben Zhang Shiwei

(*Department of Management Science College of Business Jilin University Changchun 130023*)

Abstract This paper discusses the problem of how to use analogy in automatic formation algorithms of reacting-block. The authors also give a new solution sequence separating method. These ideas and methods have been implemented in ISLS(integration solving and learning system).

Key words Machine learning, learning by analogy, problem solving.