

# 一种类比匹配原理及其实现\*

李波 罗玉龙 赵沁平

(北京航空航天大学计算机系,北京 100083)

**摘要** 本文给出了一种类比匹配原理及其计算模型,并在此基础上实现了类比匹配器 AME. 其中类比匹配原理包括一致性、谓词相似和支持性原则;计算模型采用以命题为中心,以命题映射为基本单位,从而 AME 的复杂性低于同类系统. 另外,证明了本文的实现算法具备完备性和正确性.

**关键词** 类比匹配,类比推理,自动推理,机器学习.

类比推理是由于认识到当前新情况(称靶(target))与已知熟悉情况(称基(base))在某些方面相似,从而推出它们在其它相关方面也相似. 类比推理包括联想、求精、匹配和转换四个子过程<sup>[1]</sup>. 经过联想和求精得到相似情况中与当前类比相关的描述,即基. 匹配的任务就是详细比较基和靶,识别它们组成元素间的全面对应关系. 转换以此为指导将基中知识引入到靶. 因此,识别合适的对应直接关系到最终能否在靶中生成有用、正确的新知识.

类比匹配要解决两个主要问题:(1)由于类比匹配(简称匹配)中的对应不同于模式匹配的合一,故需要建立匹配原理.(2)由于匹配不可避免地具有子图同构复杂性,故如何利用匹配的特点寻找有效的实现算法对建造实用的类比推理系统有重要意义.

在类比诸子过程中对匹配的研究较多,但许多工作只讨论特殊情况或具体问题的匹配,从而缺乏一般的原理和实现技术. 例如,情况描述具层次结构(即树)限制下的匹配<sup>[2]</sup>,严格受限表示模式下的匹配<sup>[3]</sup>,或用合一代替对应<sup>[4]</sup>. 虽然这样可以简化匹配,但适用范围极小.

本文给出了一种匹配原理及其相应的认知实验支持,与目前工作相比,该原理形式化程度更高、内容更全面. 然后介绍了基于这一原理设计的计算模型及其实现算法. 由于计算模型以命题映射为基本单元,并引入映射的对应名集和冲突名集,从而使得其实现算法具有较低复杂性. 此外,还证明了该实现算法具有正确性和完备性.

在本文中,用 $\langle B, O_b, P_b, S_b, G_b \rangle$ 表示基  $B$  的对象集、谓词集、命题集和目标,也简称  $B$ . 类似地,用 $\langle T, O_t, P_t, S_t, G_t \rangle$ 表示靶  $T$  的对象集、谓词集、命题集和目标,也简称  $T$ . 情况的对象、谓词和命题统称为情况的元素. 由于基是已知情况,其 $S_b$ 描述 $B$ 的整个处理过

\* 本文 1992-10-06 收到,1993-01-06 定稿

本研究受 863 计划和国家自然科学基金资助. 作者李波,1966 年生,副教授,主要研究领域为机器学习,专家系统和联想记忆. 罗玉龙,1967 年生,1992 年硕士毕业于北京航空航天大学,主要研究领域为机器学习与内涵逻辑. 赵沁平,1948 年生,教授,主要研究领域为计算机软件,人工智能.

本文通讯联系人:李波,北京 100083,北京航空航天大学计算机系

程,故当  $G_b \neq \Phi$  (空) 时,  $G_b \subseteq S_b$ . 对靶, 当  $G_i \neq \Phi$  时, 由于  $G_i$  是待推出目标, 故  $G_i \subseteq S_i$ ; 为统一起见, 令  $SG_i = S_i \cup G_i$ . 另外,  $G_i$  中可能有存在变量, 用于查询待定对象、谓词或基本命题.

## 1 匹配原理

抽象地讲, 类比发生于相似的两个因果体系之间<sup>[5,6]</sup>, 显然这种推理建立在相似元素比较的基础上. 因此需要按一定原则建立两个体系中元素的联系, 本节将讨论这些原则.

首先, 语法相容约束是比较的基础. 用  $e \rightarrow e'$  表示基元素  $e$  和靶元素  $e'$  间的对应, 并且满足语法相容约束.

### 语法相容约束

若  $e \rightarrow e'$  是对应, 则  $e$  与  $e'$  具相同语法类型, 即: 对象  $\rightarrow$  对象, 谓词  $\rightarrow$  谓词, 命题  $\rightarrow$  命题.

即是说只能建立相同语法类型元素间的对应. 显然, 两个语法类型不同的元素不可能在相似的因果关系体系中起相似的作用.

**定义 1.** 对应  $(e_b \rightarrow e_i)$  与  $(e'_b \rightarrow e'_i)$  是一致的 (记为  $(e_b \rightarrow e_i) \sim (e'_b \rightarrow e'_i)$ ) iff

- (1) 若  $e_b = e'_b$ , 则  $e_i = e'_i$  或者  $e'_i$  是存在变量; 或
- (2) 若  $e_b \neq e'_b$ , 则  $e_i \neq e'_i$ .

否则称  $(e_b \rightarrow e_i)$  与  $(e'_b \rightarrow e'_i)$  冲突, 记为  $(e_b \rightarrow e_i) \not\sim (e'_b \rightarrow e'_i)$ .

定义 1 中对存在变量的处理本着经赋值后能保证一对一, 即  $(e_b \rightarrow e_i) \sim (e'_b \rightarrow e'_i)$  iff 若  $e_b = e'_b$ , 则  $e_i = e'_i$ ; 否则  $e_i \neq e'_i$ .

映射是满足匹配原理的对应集合, 习惯用  $M, M_i$  表示映射.

### 一致性原则 ( $M$ 是一致的)

若  $(e_b \rightarrow e_i) \in M$  且  $(e'_b \rightarrow e'_i) \in M$ , 则  $(e_b \rightarrow e_i) \sim (e'_b \rightarrow e'_i)$ .

尽管处理方式上的差别, 一致性原则受到当前类比推理的普遍公认<sup>[7,8]</sup>. 当然我们并不否认在人类类比中存在个别违背一致性的现象, 例如, 在基中同一对象起着两种不同作用, 而在靶中有两个不同对象起着相应的作用. 但是在类比推理研究中, 一旦允许不一致对应, 就难以区别是由于错误对应引起还是由于上面现象引起, 并且会给转换带来极大的混淆; 因此, 我们仍要求映射必须满足一致性.

### 谓词相似原则

若  $e_b \rightarrow e_i$  是谓词对应, 则  $e_b$  与  $e_i$  是相似谓词.

只有由相似谓词组成的命题才可能有相似的含义, 也才可能在因果关系体系中起相似的作用. 显然人们认为 help(Tom John) 相似于 assist(Tom John). 许多实验表明, 人们除建立相同谓词间的对应外, 还建立相似谓词间的对应<sup>[3,5,6]</sup>, Holyoak & Thagard 也认为谓词相似在匹配中有重要作用<sup>[8]</sup>. Genter 的相同谓词对应已受到了许多批评<sup>[3,9]</sup>, 因为这种约束太强, 既不符合实际也大大限制了类比推理的应用范围.

由于两命题间的对应将引起它们的组成元素间的对应, 用  $\langle\langle s, s' \rangle\rangle$  表示按照语法相容约束、一致性和谓词相似原则建立命题  $s$  和  $s'$  的相同位置元素间的对应所构成的集合. 若  $\langle\langle s, s' \rangle\rangle \neq \Phi$ , 则称  $\langle\langle s, s' \rangle\rangle$  是命题映射, 当不必具体指明内容时, 用  $pc, pc_i$  表示命题映射. 基和靶间的所有命题映射集合

$$PCS = \bigcup_{e_b \in S_b, e_t \in S_t} \{ \langle \langle e_b, e_t \rangle \rangle \mid \langle \langle e_b, e_t \rangle \rangle \neq \Phi \}$$

**支持性原则**

- (1) 若  $(e_b \rightarrow e_t) \in M$  且  $e_b$  和  $e_t$  是命题, 则  $\langle \langle e_b, e_t \rangle \rangle \subseteq M$ ;
- (2) 若  $(e_b \rightarrow e_t) \in M$ , 则必存在某个命题对应  $s_1 \rightarrow s_2 \in M$  使得  $(e_b \rightarrow e_t) \in \langle \langle s_1, s_2 \rangle \rangle$ .

可以给支持性原则这样的直观解释: 若映射建立两个命题间的对应, 则该映射必须同时建立组成它们的谓词和对象间的分别对应; 反之, 任一对应都必有支持它的命题对应. 我们把这种根据命题间的对应建立参量对应的方式称以命题为中心的匹配.

支持性原则保证了对应的命题具相同的形式, 也体现了命题对应及其组成元素间对应的相互作用. Gentner 的认知实验表明在建立映射时, 人们强调共享的系统性关系结构, 并根据对象在共享关系结构中的类似作用建立对应, 而趋于忽视这些对象是否相似. 由于结构相似是建立映射的重要约束, 故当前类比推理系统普遍采用以命题为中心的匹配. 虽然也有少数系统以其它方式构成对应集, 但最终仍以是否有命题对应支持来判断对应集的优劣.

**匹配原理**

已知  $M$  是从基到靶的对应集.  $M$  是一个映射 iff  $M$  是一致的, 并且满足谓词相似和支持性原则.

Keane 的实验证实可借助下列两种信息建立对象对应<sup>[3]</sup>: (1) 对象在两情况中的格关系. 例如, 若两对象都是 actor, 则它们可能对应. (2) 功能相关属性(FRA). 他把“功能相关属性”定义为经某些命题的描述而变得突出的对象属性. 匹配原理概述了这两点. 首先, 按照上述方式建立的映射中, 相对应的两对象有相同的格. 其次, 一个对象在具体上下文的 FRA 隐含于描述它的命题之中, 例如, “医生用射线消除瘤”隐含了射线作为“工具”, 谓词相似原则保证了相对应的两对象有相似的功能(或作用).

**2 计算模型**

如果用带标志结点表示对象、谓词和命题, 用边表示命题与其谓词和参量间的联系, 则每个情况就可用一个图表示. 由匹配原理知, 建立映射相当于寻找子图同构<sup>[2]</sup>, 这就决定了匹配算法的固有复杂性很高. 因此, 如何利用匹配的特点、设计有效的算法是匹配的主要任务.

**定理 1.** 已知  $M$  是从基到靶的映射, 则存在一组命题映射  $PCS' \subseteq PCS$  使得

$$M = \bigcup_{pc \in PCS'} pc$$

支持性原则表明命题映射是构成映射的基本单元——任何映射至少包括一个命题映射, 因此可以通过合并命题映射构成更大的映射. 这样定理 1 显然.

由于基和靶间常常有多个映射, 例如, 每个命题映射就是一个映射; 匹配的任务决定了只需寻找能全面比较基和靶的映射, 为此引入“极大映射”.

**定义 2.** 已知从基到靶的映射  $M$ . 若对任意命题映射  $\langle \langle e_b, e_t \rangle \rangle \in PCS$  和  $e_b \rightarrow e_t \in M$  都有  $M \cup \langle \langle e_b, e_t \rangle \rangle$  不是一致的, 则称  $M$  是极大映射.

极大映射即不能在满足一致性原则下加入任何其它命题映射的映射. 在寻找好的极大映射过程中, 有如下四个影响算法复杂性的问题:

问题 1: 由于极大映射来源于逐步组合命题映射, 故同一命题映射需多次使用.

问题 2: 由于各映射经历的合并路径不一样, 因而可能生成重复映射.

问题 3: 需要多次检查映射间的一致性、包含关系, 而且随着映射的元素增加, 这种检查越复杂、费时.

问题 4: 有的极大映射明显不符合转换的要求.

对问题 1 和 2 的处理很简单. 问题 1 的解决方法是首先生成所有命题映射  $PCS$ , 供整个合并过程使用. 对问题 2, 只需消除每次合并生成的映射集中的重复映射即可.

下面讨论问题 3 的处理方法.

在定义 1 的基础上定义两映射是一致的.

**定义 3.** 已知映射  $M_i$  与  $M_j$ , 若对任意  $(e_b \rightarrow e_i) \in M_i$ , 任意  $(e'_b \rightarrow e'_i) \in M_j$  都有  $(e_b \rightarrow e_i) \sim (e'_b \rightarrow e'_i)$ ; 则称  $M_i$  与  $M_j$  是一致的, 记为  $M_i \sim M_j$ . 否则称  $M_i$  与  $M_j$  冲突, 记为  $M_i \not\sim M_j$ .

命题映射是映射的特殊情况, 因此定义 3 也适合判断命题映射之间, 命题映射与映射之间的一致和冲突关系.

**定理 2.** 已知映射  $M_i$  和  $M_j$ ,  $M_i \sim M_j$  iff  $M_i \cup M_j$  是映射.

容易证明定义 3 符合一致性原则, 而合并过程仍保持  $M_i \cup M_j$  遵守谓词相似和支持性原则, 从而可证定理 2.

支持性原则和定理 1 表明, 命题映射是构成极大映射的基本单元, 而每个命题映射正好有一个命题对应. 因此, 用一个名称代替一个命题对应, 从而可以引入映射的对应名集和冲突名集.

**定义 4.** 映射  $M$  的对应名集  $N_{pc}(M)$  递归定义如下:

(1) 若  $M \in PCS$ , 设  $M = \langle\langle s_1, s_2 \rangle\rangle$  且  $s_1 \rightarrow s_2$  的名称为  $name$ , 则  $N_{pc}(M) = \{name\}$ ;

(2) 若  $M = M_1 \cup M_2$  ( $M_1, M_2$  是映射), 则  $N_{pc}(M) = N_{pc}(M_1) \cup N_{pc}(M_2)$ .

$N_{pc}(M)$  包括了构成  $M$  的所有命题映射相应的命题对应的名称, 并且若  $|N_{pc}(M)| = n$ , 则称  $M$  是长度为  $n$  的映射. 当不必显式指出命题映射  $pc_i$  的对应名集时, 用  $N_{pc}(pc_i)$  表示.

**定义 5.** 映射  $M$  的冲突名集  $N_{cf}(M)$  递归定义如下:

(1) 若  $M \in PCS$  (设  $M = pc$ ), 则  $N_{cf}(M) = N_{cf}(pc) = \bigcup_{pc_i \in PCS \text{ 且 } pc_i \neq pc} N_{pc}(pc_i)$ ;

(2) 若  $M = M_1 \cup M_2$  ( $M_1, M_2$  是映射), 则  $N_{cf}(M) = N_{cf}(M_1) \cup N_{cf}(M_2)$ .

命题映射的冲突名集就是与它冲突的所有命题映射相应的命题对应的名称的集合, 映射的冲突名集就是构成它的各命题映射的冲突名集的并集.

**定理 3.\*** 已知映射  $M_i$  和  $M_j$ ,  $M_i \sim M_j$  iff  $N_{pc}(M_i) \cap N_{cf}(M_j) = \Phi$ .

证明: (略)

**推理 1.** 已知命题映射  $pc_i$  和映射  $M_j$ ,  $pc_i \sim M_j$  iff  $N_{pc}(pc_i) \cap N_{cf}(M_j) = \Phi$ .

**定理 4.** 已知命题映射  $pc_i$  和映射  $M_j$ ,  $pc_i \subseteq M_j$  iff  $N_{pc}(pc_i) \subseteq N_{pc}(M_j)$ .

\* 文献[7]中 P. 21 有类似定理, 即  $\text{Consist}(\text{GMap}_i, \text{GMap}_j)$  iff  $(\text{Element}(\text{GMap}_i) \cap \text{NoGood}(\text{GMap}_j) = \Phi) \wedge (\text{NoGood}(\text{GMap}_i) \cap \text{Element}(\text{GMap}_j) = \Phi)$ . 由于能证明  $\text{Element}(\text{GMap}_i) \cap \text{NoGood}(\text{GMap}_j) = \Phi$  等价于  $\text{NoGood}(\text{GMap}_i) \cap \text{Element}(\text{GMap}_j) = \Phi$ , 故只需其中一半即可, 这有助简化算法的操作.

证明:(略)

我们引入对应名集(定义 4)和冲突名集(定义 5)是为了利用映射的特点,将直接判断命题映射与映射间“一致关系”(定义 3)、“包含关系”的操作,转化为简单集合的运算(推论 1 和定理 4).显然,随着映射长度的增加,这将大大减少算法的实际操作.

对问题 4,极大映射只是一个定性说明,命题映射也可能是极大映射,因此极大映射不足以描述对类比推理有用的映射.为便于全面比较基和靶,应排除包括命题映射少的映射.另外,当  $G_i \neq \Phi$  时类比推理的目的就是仿效基推出  $G_i$ ;而求精阶段已保证  $|G_b| = |G_i|$ ,并且  $\ll g_{bi}, g_{ii} \gg \neq \Phi$ ,这里  $g_{bi}$  和  $g_{ii}$  分别是  $G_b$  和  $G_i$  的第  $i$  个元素,  $1 \leq i \leq |G_i|$ .因此,应建立包括所有  $\ll g_{bi}, g_{ii} \gg$  的映射.另外,由于所有映射具相同基准——同一基和靶,故在比较映射时只需考虑绝对数值即可,不必考虑相对值.

**定义 6.** 已知  $M$  是从  $\langle B, O_b, P_b, S_b, G_b \rangle$  到  $\langle T, O_t, P_t, S_t, G_t \rangle$  的极大映射.若  $M$  满足下列两个条件:

- (1) 若  $G_i \neq \Phi$ , 设  $G_b = \{g_{b1}, \dots, g_{bk}\}, G_i = \{g_{i1}, \dots, g_{ik}\}$ , 则  $(g_{bi} \rightarrow g_{ii}) \in M$ , 其中  $1 \leq i \leq k$ ;
- (2)  $|N_{pc}(M)| \leq \frac{1}{2} \min\{|S_b|, |S_{G_i}|\}$ .

则称  $M$  是候选映射.

### 3 实现算法

基于上节的计算模型,我们实现了类比匹配器 AME. 在 AME 的匹配算法  $Match(B, T)$  中,输入是基  $\langle B, O_b, P_b, S_b, G_b \rangle$  和靶  $\langle T, O_t, P_t, S_t, G_t \rangle$ ,返回所有候选映射  $Mext$ ;若无则返回  $\Phi$ .为描述  $G_b$  与  $G_i$  的对应,约定当  $G_i \neq \Phi$  时 PCS 的第一个元素是\*

$$\ll G_b, G_i \gg = \bigcup_{\substack{g_{bi} \text{ 和 } g_{ii} \text{ 分别是 } G_b \text{ 和 } G_i \\ \text{的第 } i \text{ 个元素, } 1 \leq i \leq |G_i|}} \ll g_{bi}, g_{ii} \gg$$

否则 PCS 的第一个元素是  $NIL$ .

#### 匹配算法 $Match(B, T)$

```

1.  $S \leftarrow S_b - \{G_b\}; S' \leftarrow S_t$ 
2. # 建立 B 和 T 间所有命题映射的集合 PCS #
   IF  $G_i \neq \Phi$  THEN  $PCS \leftarrow \{\ll G_b, G_i \gg\}$  ELSE  $PCS \leftarrow \{NIL\}$ 
   FOR 每个  $s \in S$  DO
   BEGIN
   FOR 每个  $s' \in S'$  DO
   IF  $\ll s, s' \gg \neq \Phi$  THEN  $PCS \leftarrow PCS \cup \{\ll s, s' \gg\}$ 
   END
3. # 建立极大映射 #
    $Mext \leftarrow merge(PCS)$ 
4. # 评估极大映射 #
   FOR 每个  $M \in Mext$  判断  $M$  是否是候选映射

```

\* 为统一称呼,仍把  $\ll G_b, G_i \gg$  看作一个命题映射.

5. IF 无候选映射 THEN 返回  $\Phi$  ELSE 返回候选映射集

算法  $merge(PCS)$  返回由命题映射集合  $PCS$  构造的所有极大映射, 用到下列变量:

$MS$ : 记录合并长度为  $i$  的映射与命题映射所生成的长度为  $i+1$  的所有映射.

$label$ : 用于判断一个映射是否是极大映射. 当某映射与  $PCS$  中的所有其它(即不包括在该映射之中的)命题映射冲突时, 则该映射是极大映射,  $label$  值保持为 0.

$Mext$ : 记录合并过程中所生成的极大映射.

**合并算法  $Mergc(PCS)$**

1.  $N \leftarrow PCS$  中命题映射数

IF  $N=0$  THEN 返回  $\Phi$

IF  $N=1$  THEN 返回  $car(PCS)$

命名  $PCS$  中各命题对应:  $MS \leftarrow \Phi, Mext \leftarrow \Phi$

2. # 生成长度为 2 的所有映射 #

IF  $car(PCS) \neq NIL$  THEN  $MS_1 \leftarrow car(PCS)$  且  $PCS \leftarrow cdr(PCS)$

ELSE  $MS_1 \leftarrow cdr(PCS)$  且  $PCS \leftarrow cdr(PCS)$

FOR 每个  $pc_1 \in MS_1$  DO

BEGIN

$PCS_1 \leftarrow PCS - \{pc_1\}, label \leftarrow 0, N_{cf}(pc_1) \leftarrow \Phi$

FOR 每个  $pc_2 \in PCS_1$  DO

BEGIN

IF  $(pc_1 \sim pc_2)$  THEN

BEGIN

$label \leftarrow 1, MS \leftarrow MS \cup \{pc_1 \cup pc_2\}$

$N_{pc}(pc_1 \cup pc_2) \leftarrow N_{pc}(pc_1) \cup N_{pc}(pc_2)$

END

ELSE  $(pc_1 \not\sim pc_2) N_{cf}(pc_1) \leftarrow N_{cf}(pc_1) \cup N_{pc}(pc_2)$

END

IF  $(label=0)$  THEN  $Mext \leftarrow Mext \cup \{pc_1\}$

END

消除  $MS$  中重复元素, 计算  $MS$  中各元素的冲突各集

3. # 建立所有极大映射 #

WHILE  $(MS \neq \Phi)$  DO

BEGIN

$MS_1 \leftarrow MS, MS \leftarrow \Phi$

# 合并  $MS$  中长度为  $i$  的映射与  $PCS$  中的命题映射, 生成所有长度为  $i+1$  的映射 #

FOR 每个  $M \in MS_1$  DO

BEGIN

$label \leftarrow 0$

FOR 每个  $pc \in PCS$  DO

BEGIN

IF  $(N_{pc}(pc) \subseteq N_{pc}(M))$  (即  $pc \subseteq M$ ) THEN  $label \leftarrow label$

IF  $(N_{pc}(pc) \cap N_{cf}(M) = \Phi)$  (即  $pc \sim M$ ) THEN

BEGIN

$MS \leftarrow MS \cup \{M \cup pc\}, N_{pc}(M \cup pc) \leftarrow N_{pc}(M) \cup N_{pc}(pc)$

$N_{cf}(M \cup pc) \leftarrow N_{cf}(M) \cup N_{cf}(pc); label \leftarrow 1$

END

END

IF ( $label = 0$ ) THEN  $Mext \leftarrow Mext \cup \{M\}$

END

消除 MS 中重复元素

END

4. 返回 Mext

### 4 算法分析

由于目前尚无能寻找所有好映射(如定义6的候选映射)的启发式准则<sup>[2,7]</sup>,所以我们采用穷举法——生成所有极大映射\*,然后从中选择候选映射.这就要求合并算法能生成所有极大映射.另外,算法 Match 除第3步调用 merge 外,很容易看出其余各步具有多项式时间复杂性.本节只讨论算法 merge 的完备性和正确性,并分析它的时间复杂性.

定理5(完备性和正确性).

已知 PCS 是基和靶间的全部命题映射的集合.

(1)当  $G_i = \Phi$ , merge(PCS)生成而且只生成所有极大映射.

(2)当  $G_i \neq \Phi$ , merge(PCS)生成而且只生成包括  $\langle\langle G_b, G_i \rangle\rangle$  的所有极大映射.

证明:对  $G_i = \Phi$ ,用  $MS^i$  表示所有长度为  $i$  的映射的集合,  $Mext^i$  表示所有长度为  $i$  的极大映射的集合,  $Mext$  表示所有极大映射的集合.

根据映射、极大映射的定义,我们有:

$$MS^i = \begin{cases} PCS & \text{当 } i=1 \\ \bigcup_{M \in MS^{i-1}, pc \in PCS} \{M \cup pc \mid (pc \not\subseteq M) \wedge (pc \sim M)\} & \text{当 } i>1 \end{cases}$$

$$Mext^i = \bigcup_{M \in MS^i} \{M \mid \forall pc ((pc \in PCS) \wedge (pc \not\subseteq M) \wedge (pc \not\sim M))\}$$

$$Mext = \bigcup_{i=1}^l Mext^i$$

其中映射的最大长度  $l \leq \min\{m, n\}$ , 基中命题数  $m = |S_b|$ , 靶中命题数  $n = |SG_i|$ .

把 merge 的执行过程作如下划分:第2步的双重循环称作 merge 的第1次循环,第3步中第  $i$  次 WHILE 循环称作 merge 的第  $i+1$  次循环.

这样,merge 的第  $i$  次循环生成并且只生成  $MS^{i+1}$  和  $Mext^i$ . 当 merge 执行第  $l$  次循环时,生成的  $MS^{l+1} = \Phi$ ,故循环结束.

merge 的第4步返回各次循环生成的极大映射的并集,正好是 Mext.

类似地,可证明对  $G_i \neq \Phi$ , merge(PCS)生成而且只生成包括  $\langle\langle G_b, G_i \rangle\rangle$  的所有极大映射.

因此定理5得证. □

下面分析 merge 的时间复杂性,这里借用定理5证明中的表示和术语. 首先引入几个新

• 当  $G_i \neq \Phi$  时生成所有包括  $\langle\langle G_b, G_i \rangle\rangle$  的极大映射

的表示:

$PCS_b$ : 包括基中命题  $b$  的所有命题映射的集合

$$PCS_b = \{\langle\langle b, s \rangle\rangle \mid s \in SG, \wedge \langle\langle b, s \rangle\rangle \neq \Phi\}.$$

$k$ :  $PCS_b$  的平均元素个数, 这样每个靶命题在一个  $PCS_b$  中构成命题映射的平均数为  $\frac{k}{n}$ .

$d$ :  $MS^i$  中极大映射的比例.

把单位时间看作合并一个映射与一个命题映射所花的时间, 这样一个单位时间内只进行一组基本操作, 是与  $m, n$  无关的时间常量. 为讨论方便, 令  $l = m = n$ ,  $x = \frac{(1-d)k}{n}$

(1) 当  $G_i = \Phi$  时, 经计算总时间

$$T = \frac{nk \times (n!)^2}{1-d} \sum_{i=1}^n \frac{x^i}{i! [(n-i)!]^2}$$

$$\text{令 } h = \left\lfloor \frac{2n + \frac{1}{x} - \sqrt{\frac{4n}{x} + \frac{1}{x^2} + 4}}{2} \right\rfloor \quad (\text{这里 } \lfloor \text{value} \rfloor \text{ 表示对 value 向下取整})$$

$$\text{则有 } T \approx \frac{nk \times (n!)^2}{1-d} \times \frac{2x^h}{h! [(n-h-1)!]^2}$$

最坏情况:  $d=0, k=n$ ; 即  $x=1$ . 经估算当  $n=10 \sim 30$  时,  $h \approx 0.865n$ , 则有  $T \approx n^5 \times n!$ . Gentner 的 SME 在最坏情况的总时间为  $T' = N'!$ , 其中  $N'$  是基(或靶)中命题数和对象数之和, 即  $N' = |S_b| + |O_b|$ ; 经计算有  $T' \approx 10^3 T$ , 显然  $T'$  远远大于  $T$ .

一般情况, 取  $d=0.15, k=3$ , 则  $x = \frac{2.25}{n}$ . 当  $n=10 \sim 30$  时, 得  $h \approx 0.5n$ ; 近似计算可得随着  $n$  的增大  $T < 4^n$ . 显然, 相对 merge 所解决的问题——相当于寻找所有极大子图同构, 这个复杂性是很低的.

(2) 当  $G_i \neq \Phi$  时, 可得总时间为:

$$T \approx \frac{nk \times [(n-1)!]^2}{1-d} \times \frac{2x^h}{h! [(n-h-2)!]^2}$$

$$\text{其中 } h = \left\lfloor \frac{2(n-1) + \frac{1}{x} - \sqrt{\frac{4(n-1)}{x} + \frac{1}{x^2} + 4}}{2} \right\rfloor$$

显然此时的时间复杂性比  $G_i = \Phi$  时更小.

## 5 结束语

我们给出了一种反映人类比较相似情况的匹配原理. 该原理的一致性和支持性原则保证基和靶的描述具相同形式——相同因果关系网, 可看作语法原则; 谓词相似从内容上保证对应命题有相似的功能, 可看作语义原则. 此外, 候选映射要求目标相似, 可看作语用准则.

计算模型采用以命题为中心的匹配, 并根据匹配的特点, 通过引入对应名集和冲突名集实现以命题映射为基本处理单位的合并. 从而使 AME 的实现算法具有很低的时间复杂性



(相对其任务),而且,该实现算法具有完备性和正确性.

### 参考文献

- 1 赵沁平,李波,罗玉龙.关于类比推理若干基本问题的研究.计算机科学,1993(2).
- 2 Welsh K, Jones M. Computational analogy. In: Boulay B D, Hogg D, Steels L eds. *Advances in Artificial Intelligence I*, 1987:93-102.
- 3 Keane M T. *Analogical problem solving*. Ellis Horwood Limited, England, 1988.
- 4 Greiner R. Learning by understanding analogies. *Artificial Intelligence*, 1988(35):81-125.
- 5 Gentner D. The mechanisms of analogical learning. In: Vosniadou S, Ortony A eds. *Similarity and Analogical Reasoning*, Cambridge University Press, 1989:199-241.
- 6 Vosniadou S. Analogical reasoning as mechanism in knowledge acquisition: a developmental perspective. In: Vosniadou S, Ortony A eds. *Similarity and Analogical Reasoning*, Cambridge University Press, 413-437.
- 7 Falkenhainer B, Forbus K D, Gentner D. The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 1990(41):1-63.
- 8 Holyoak K J, Thagard P. Analogical mapping by constraint satisfaction. *Cognitive Science*, 1989;13:295-355.
- 9 Burstein M H. Concept formation by incremental analogical reasoning and debugging. In: Michalski R S, Carbonell J G, Mitchell T M eds. *Machine Learning: An Artificial Intelligence Approach*, 1986:351-370.

## AN THEORY OF ANALOGICAL MATCH AND ITS IMPLEMENTATION

Li Bo Luo Yulong Zhao Qinqing

(Department of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing 100083)

**Abstract** This paper presents a theory and a computational model of analogical match, and an implementation of this model, named AME. The theory includes consistency, predicate similarity and support principles. Because the computational model constructs mapping in proposition-oriented match, and takes proposition mapping as primitive units, the complexity of AME is lower than the same kind of system. The paper also proves the implementation algorithms to be complete and reasonable.

**Key words** Analogical match, analogical reasoning, automatic reasoning, machine learning.