

范例推理中面向对象的范例表示研究^{*})

耿焕同 方 瑾 蔡庆生

(中国科学技术大学计算机科学技术系 合肥 230026)

摘 要 结合范例推理的具体特点,按照面向对象的形式定义,给出范例的面向对象表示方法。通过类的层次信息,方便地表示范例复杂的结构信息;它一方面克服了属性-值对表示法无法表示复杂结构范例的缺陷,另一方面将领域知识通过约束规则与范例的表示有机结合起来。在范例相似性计算上,区分类内相似性和类间相似性,使得相似性表示尽可能准确和科学,从而有效地拓展了范例推理系统中的知识表示能力。

关键词 范例推理,面向对象,范例表示,相似性计算

Object-Oriented Case Representation in Case-Based Reasoning

GENG Huan-Tong FANG Jin CAI Qing-Sheng

(Department of Computer Science, USTC, Hefei 230026)

Abstract This paper proposes a new object-oriented case representation based on the formalize definition of object and the specification of Case-based Reasoning. It represents the complex case structural information through class hierarchy. The new method not only overcomes limitations that attribute-value case representation can not represent complex structured cases, but also combines domain knowledge and case representation through constraint rules. At the point of case similarity, this paper divides it into two parts; inter-class similarity and intra-class similarity, which improve correctness of similarity representation as high as possible. The object-oriented case representation can effectively reinforce the representing ability of knowledge in the CBR system.

Keywords Case-based reasoning, Object-oriented, Case representation, Similarity measures

1 引言

基于范例推理(Case-Based Reasoning,简称 CBR)是由 R. Schank 教授提出基于范例的推理方法^[1]。CBR 是用先前求解问题的经验和方法,通过类比和联想来解决当前相似问题的推理方法,克服了传统的基于规则推理(Rule-Based Reasoning,简称 RBR)系统的知识难于获取和推理的脆弱性等缺陷^[2,3]。所谓范例(case)是对应用领域内一个问题的描述,由问题的状态和与其相应的解决方案两部分组成,实际上就是一个从问题的状态空间到解决方案空间的一个映射^[4]。所有的范例构成范例库(case base),目前已被广泛应用于医学诊断、工业制造、法律诉讼和 Q&A 应用系统^[5]等领域,并取得很好的效果。

CBR 是通过类比和范例复用来得到新问题的解。可知范例库是 CBR 中最重要的一个部分,是范例推理的知识库,包含了应用领域中的历史经验;领域知识表示直接关系到 CBR 推理的正确性^[6]。因此范例表示是 CBR 中首要问题,一般采用属性-值对表示法。这种表示方法简单,较适用于领域知识贫瘠的问题。另外还有层次式知识表示方法^[7],但未给出具体描述。

面向对象技术是当今最为流行的程序设计方式和数据表示方法,它的突出特点是模块性、封装功能、代码共享、灵活性和易维护性^[8]。面向对象知识表示在专家系统等领域得到广泛应用^[9,10]。本文结合 CBR 的具体特点,提出面向对象的范

例表示方法,并将领域知识蕴涵于类的层次结构和类的约束规则中。另外,在范例检索中,范例间的相似性计算是 CBR 中范例检索的关键,本文首先提出了范例的面向对象表示方法,并给出在基于面向对象范例表示中的范例间相似性计算模型和方法。

本文的内容组织如下:第 2 节形式化描述范例的面向对象表示;第 3 节给出基于面向对象表示的范例间相似性计算模型;最后是结束语。

2 范例的面向对象表示

类和对象是面向对象技术中的重要概念。类是对一组相似对象的共同抽象描述,它将该组对象所具有的共同特征(操作特征和存储特征)集中起来,用来说明该组对象的性质。类是创建实例对象的模板。类可定义为一个四元组:

$CLS ::= \langle ID, DD, OP, INT \rangle$

其中 ID 是类的标识;DD 是数据结构描述;OP 是操作集合的具体实现,即方法集;INT 是统一对外接口。类和对象的这种结构,起到了“信息隐藏”的作用。

在范例推理中,范例的结构化信息是通过不同层次信息加以表示的。首先是对象(object),然后是对象类(object class),最后是类层次(class hierarchy)。这种层次信息也包含了对象间的相似性信息。

2.1 范例类的形式化定义

我们将采用 BNF 范式给出类和对象的形式化定义:

^{*})本文得到国家自然科学基金项目资助(No. 70171052),No. 90104030)。耿焕同 博士生,主要研究领域:人工智能、知识发现;方 瑾 博士,主要研究领域:人工智能、专家系统;蔡庆生 教授,博导,主要研究领域:人工智能、机器学习、知识发现。

类::=CLASS <类名>(<父类>(<属性>(<约束规则>(<方法>
END
类名::=<字符串>
父类::=EXTEND <类名>
属性::=SimpleAttr <简单属性> ComplexAttr <复杂属
性> END
简单属性::=[(<属性名>,<类型>,<初始权重>)]_n
属性名::=<字符串>
类型::=<字符串>
初始权重::=<实数值>
复杂属性::=CLASS <类名>
约束规则::=RULE <规则名>(<规则体>) END
规则名::=<字符串>
规则体::=IF <规则前件> THEN <规则后件> END
规则前件::={(<前件属性>(<操作算子>(<值>)) AND/OR
<前件属性>(<操作算子>(<值>))_n
前件属性::=<字符串>
操作算子::=</>=</>=</>=</>=</>=</>=
方法::=METHOD <方法名>(<参数表>(<方法体>)) END
方法名::=<字符串>
参数表::=<字符串>
方法体::=<字符串>

类的定义主要包括类名、父类、类所包含的属性集合、属性所满足的约束规则及类的操作方法。将属性分为简单属性和复杂属性,简单属性一般是那些不可再分的属性组成,可采用属性-值对表示法;复杂属性可以进一步再分,表达较为复杂的范例属性知识,采用类的实例进行表示。

基于范例推理中,存在大量的领域知识,除了一部分存在于类的层次结构中外,另外可利用属性的约束规则加以表达。约束规则采用产生式知识表示,由规则名和规则体组成,规则体包含规则前件和规则后件。规则前件采用四联体表示法。规则可看成是类的特殊方法,在子类与父类间存在继承关系。此方法的好处在于不必为每个子类重复定义约束规则,减少了规则维护代价。规则的冲突消解策略是优先激活子类的规则,然后是父类的规则。这种消解策略一方面是因为子类中可能引进新的属性,增加了新的约束;另一方面在类的层次结构中,子类对问题的描述更为贴切和具体,父类更加抽象。因此,优先激活子类规则,更能准确表达领域知识。

下面将给出一个描述 PC 机的类的定义(假设其父类为 COMPUTER):

```
CLASS PC EXTEND COMPUTER
  SimpleAttr
    Price: Real, 0.2;
    Speed: Real, 0.3;
  END
  ComplexAttr
    Processor: CLASS PROCESSOR;
    Hard-Disk: CLASS HARD-DISK;
  END
END
```

2.2 范例对象的形式化定义

类是对象的抽象模板,对象是类的实例。对象的定义可表示为:

对象::=OBJECT <对象名>FROM CLASS <类名>
对象名::=<字符串>
类名::=<字符串>

假设计算机 A 是 PC 类的实例,则计算机 A 可以形式化地表示为:OBJECT 计算机 A From CLASS PC。

3 范例间相似性计算

范例间相似性计算是范例推理的核心问题。在采用属性-值对范例表示法的系统中一般用距离函数表示,如欧氏距离、曼哈顿距离等。然而在面向对象的系统中,对象间相似性计算变得比较复杂,简单的距离函数无法准确表达范例间的相似性。因为被查询范例与范例库中的某个范例的某些属性相同(或是同一个类的实例,或都从某个父类继承而来),范例的这部分相似性可用距离函数表示。某些属性可能根本不相同,无法简单地用距离函数刻画这部分的相似性。因此,在计算范例间相似性时需要区分两种不同形式的相似性,即类内相似性计算与类间相似性计算。

3.1 类内相似性计算

类内相似性用来描述被检索范例 q 与范例库中的某个范例 c , 具有相同属性的那部分相似性。可以形式化地表示为:

$$Sim_{Intra}(q, c) = \varphi(Sim_{A_1}(q, A_1, c, A_1), Sim_{A_2}(q, A_2, c, A_2), \dots, Sim_{A_k}(q, A_k, c, A_k))$$

其中, $Sim_{Intra}(q, c)$ 表示范例 q 与范例 c 的类内相似性; q, A_i 与 c, A_i 表示范例 q 与范例 c 的 A_i 属性的值; φ 表示聚集函数。

常见简单属性一般有三类,即数值型属性、符号型属性和模糊型属性。下面给出基于属性-值对的范例类内相似性的计算公式。定义如下:设 q, c 为两范例,它们的类内相似性 $Sim_{Intra}(q, c)$ 可用下面公式计算:

$$Sim_{Intra}(q, c) = 1 - \sqrt[n]{\sum_{i=1}^n w_i \times d(q_i, c_i)^2}$$

其中: $\sqrt[n]{\sum_{i=1}^n w_i \times d(q_i, c_i)^2}$ 为 q, c 间的欧氏距离。记 f_i 为范例的第 i 个属性特征字段; q_i, c_i 为范例 q, c 中属性 f_i 的值; w_i 为第 i 个属性 f_i 的权重, $d(q_i, c_i)$ 为两属性的距离(规格化)。分几种情况讨论:

1) 若该属性为数值型,则 $d(q_i, c_i) = \frac{|q_i - c_i|}{ValueRange(f_i)}$;

2) 若该属性为符号型,一般采用按属性值的相似性大小建立层次关系,且为每个相邻的属性-值对分配一个相似值,不相邻属性值间的相似性定义一定的函数约束。若属性 f 的不同属性值的相似性层次为 $V_1 \rightarrow V_2 \rightarrow V_3$, 且设 $sim(V_1, V_2) = s_1, sim(V_2, V_3) = s_2$, 则 $sim(V_1, V_3) = s_1 \times s_2$; 即 $d(q_i, c_i) = 1 - sim(q_i, c_i)$;

3) 若该属性为模糊型数据,设该属性的模糊等级有 m 级,记 $grades = \{g_1, \dots, g_m\}$, 且设范例模糊属性在各等级中的隶属度的值分别表示为: $gradvalue = \{v_1, \dots, v_m\}$; 其中 v_i 为该属性属于等级 g_i 的隶属度的值,则 $d(q_i, c_i) = \frac{\sum_{i=1}^m |v_{q_i} - v_{c_i}|}{m}$, 其中: v_{q_i}, v_{c_i} 分别为范例 q, c 的 f_i 属性在该属性模糊等级 j 中的隶属度的值。

3.2 类间相似性计算

在范例类的层次结构中(如图 1),有两种类型节点:内部结点和叶子结点。其中叶子结点表示现实世界中的具体对象;内部结点为类层次树中的非叶子结点,表示一类抽象对象的集合。为了更清楚地叙述类间相似度计算模型,先做如下定义:

1) 若结点 K 为内部结点,用 $Leaf(K)$ 表示 K 的叶子结点;

(下转第 178 页)

- chastic Language Processing. Sikkel K, Nijholt A, eds. Parsing Natural Language, TWLT6, Twente University, 1993
- 5 Bod R. Using an Annotated Corpus as a Stochastic Grammar. Proceedings of EACL'93, Utrecht, The Netherlands, 1993
 - 6 Bod R. Mathematical Properties of the Data Oriented Parsing Model. Preprints Third Meeting on Mathematics of Language, Austin, Texas, 1992
 - 7 Sima'an K. Computational Complexity of Probabilistic Disambiguation by Means of Tree-Grammars. In: Proc. COLING-96, Copenhagen, 1996
 - 8 Bod R. Monte Carlo Parsing. Recent Advances in Parsing Technology. Kluwer Academic Publishers
 - 9 Sima'an K. An Optimized algorithm for Data Oriented Parsing. In: Proc. International Conference on Recent Advances in Natural Language Processing, Tzigrav Chark, Bulgaria, 1994
 - 10 Goodman J. Parsing Algorithms and Metrics. In: Proc. of the 34th

Annual Meeting of the ACL, June, 1996

- 11 Green P J. Overview; Monte Carlo methods. In: Complex Stochastic Systems and Engineering. Titterton D M, ed. IMA Conference Series, No. 54, Oxford Univ. Press, 1995, 183~190
- 12 Hammersley J, Handscomb D. Monte Carlo Methods. Chapman and Hall, 1964
- 13 Robert C, Casella G. Monte Carlo Statistical Methods. New York: Springer-Verlag, 1999
- 14 Mayer H. Symposium on Monte Carlo Methods. New York Wiley, 1956
- 15 Nicol J, Pickering M. Processing Syntactically Ambiguous Sentences: Evidence from Semantic Priming. Journal of Psycholinguistic Research, 1993
- 16 MacDonald M, Pearlmuter N, Seidenberg M. Lexical Nature of Syntactic Ambiguity Resolution. Psychological Review, 1994, 101, 676~703

(上接第 166 页)

2) $MinSup(K_1, K_2)$ 表示结点 K_1, K_2 的最近公共父结点;

3) 若 K_i 是 K_j 的子结点, 则记 $K_i < K_j$ 。

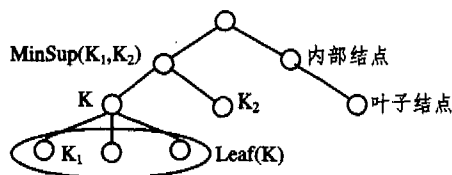


图 1 范例类的层次图

为计算类间相似性,需为类层次树中的每个内部结点 K_i 分配一个数值 $s_i \in [0 \cdots 1]$, 用来表示以该结点为根的子树中所有子结点相似度的最小值;并且要求满足如下的条件:若 $K_i < K_j$, 则要求 $s_i \geq s_j$ 。

进一步规定,若 x, y 为内部结点 K_i 的叶子结点, 则 x, y 结点的类间相似性需满足如下的条件: $\forall x, y \in Leaf(K_i) \Rightarrow Sim_{inner}(x, y) \geq s_i$ 。

3.2.1 类间相似性计算模型

假设有两个结点 K_i, K_j , 类间相似性的计算主要有三种情形:

- 1) 两个结点 K_i, K_j 都为叶子结点;
- 2) 结点 K_i 为叶子结点, 结点 K_j 为内部结点;
- 3) 结点 K_i, K_j 都为内部结点。

为简便起见,可将这三种情形统一起来,利用其公共父结点的相似值加以表示。可表示为:

如果 $K_i = K_j$, 则 $Sim_{inner}(K_i, K_j) = 1$;

如果 $K_i \neq K_j$, 则 $Sim_{inner}(K_i, K_j) = s_{MinSup(K_i, K_j)}$ 。

3.2.2 范例间相似性计算

在计算类内相似性和类间相似性之后,就可计算两范例的相似性,通过采用类内相似性和类间相似性的乘积加以计算。即:

$$Sim(q, c) = Sim_{inner}(q, c) \times Sim_{inner}(q, c)$$

结束语 本文结合 CBR 的具体特点,先给出范例推理中面向对象的范例表示一般方法。这种表示方法,可通过类的层次信息,表示范例间复杂的结构信息。它一方面克服了属性值对表示法无法表示复杂范例的缺陷,另一方面通过约束规则将领域知识与范例的表示结合起来。同时在范例的相似性计算上,区分两类不同的相似性即类内相似性和类间相似性,使得范例相似性计算尽可能地准确和科学。

参考文献

- 1 Schank R. Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge, UK: Cambridge University Press, 1982
- 2 Juell P, Paulson P. Case-based systems. Intelligent Systems, IEEE, 2003, 18(4): 60~67
- 3 Watson I, Marir F. Case-Based Reasoning: A Review. The Knowledge Engineering Review, 1994, 9(4): 327~54
- 4 Zheng Quan, Wang Junpu, Wu Xiuqing, et al. Intelligent Control and Automation. In: Proceedings of the 3rd World Congress on, 2000, 1: 242~245
- 5 Fu Yonggang, Shen Ruimin. GA based CBR approach in Q&A system. Expert Systems with Applications, 2004, 26(2): 167~170
- 6 Smyth B, Keane M T. Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems. In: International Joint Conference on Artificial Intelligence (IJCAI), 1995, 377~383
- 7 Watson I, Perera S. A hierarchical case representation using context guided retrieval. Knowledge-Based Systems, 1998, 11: 285~292
- 8 Graham I. 面向对象的方法: 原理与实践. 北京: 机械工业出版社, 2003
- 9 曹元大, 徐漫江. 面向对象知识表示在专家系统开发工具中的应用. 北京理工大学学报, 2000, 20(6): 688~692
- 10 钱梅, 汪斌, 吴耿锋, 等. 面向对象模型对复杂结构知识的信息表示方法. 计算机工程与科学, 2002, 24(3): 62~65
- 11 Free W, Gamma E. Design Patterns for Object-Oriented Software Development. Addison-Wesley Pub Co, 1995