

93-94

遗传程序设计

赋值函数

节点分类 (25)

计算机科学 1998 Vol. 25 No. 3

遗传程序设计中的赋值函数和节点分类

The Assignment Function and Node Classification in Genetic Programming

齐越胜 王保中 康立山

(武汉大学软件工程国家重点实验室 武汉 430072)

TP311 0242.23

摘 要 To make sure that the crossover operator produces meaningful offsprings, it is always a must that every function node of the tree has the property of closeness. This paper points out the problems of this closeness requirement, especially when assignment function is needed in the function set. In this paper, we propose a node classification method based on the return data type of the node so as to cancel this closeness requirement. Using this method we evolved a program to model a multioutput problem. The computational result is quite satisfactory.

关键词 Genetic programming, Assignment function, Node classification

一、遗传程序设计

遗传程序设计(GP)是用演化方法让计算机自己发现解决问题的算法,生成可以解决实际问题的程序。一个程序可以用树结构很好地表示出来,因此,演化一个程序也就是演化一个合适的树的过程。通常,树的节点分为两类:函数节点和终端节点^[1],函数节点取自函数集(根据具体问题进行定义),用向下的分支表示其参数,终端节点取自终端集(不同的问题也有不同的终端集),不再有分支。比如,在 $+(a,b)$ (即 $a+b$)这个表达式中(也是一个简单的程序), $+$ 是函数节点,有两个参数,而 a,b 是终端节点。遗传程序设计的演化过程是这样的^[2](同遗传算

法):

1. 设定最大代数 MaxGeneration, 随机生成一群个体, 记当前代 Generation 为 0;
 2. 判定当前代的每一个体的性能(计算适应值);
 3. 若已经得到一合符要求的程序, 则该次演化成功, 结束运行, 否则做 4;
 4. 若当前代已经达到最大代数, 则该次演化失败, 结束运行, 否则做 5;
 5. 通过杂交, 变异和复制, 产生新一代个体, 当前代 Generation 增加 1, 返回 2。
- 杂交是两个树通过交换它们的子树而得到两个

相似和语言近似。在这里隶属函数显得很重要。Zadeh 在定义“年龄”这一概念时, 曾简单地给出了几个隶属函数。根据人类的思维习惯, 在考虑隶属函数时, 一般倾向于逼近或拟合二次函数, 如果大部分隶属函数能基于此考虑的话, 对模糊的演算将变得容易些。然而对于人类来说, 一个概念不仅与其自身的含义有关, 也与对象所处的环境、人们的心理等相关因素有联系。因而, 与其说概念本身模糊还不如说人们用以建立该概念的标准是模糊的^[3]。如上所述, 要解决带有模糊的一系列问题, 比如模糊知识表示、模糊推理、模糊集合的语言值等问题, 由于模糊逻辑的理论基础还不完善, 也还只处于探索应用阶段。其根本的解决办法只能从揭示人类思维的奥秘, 了解其本质入手, 才能解决机器理解概念、命题以及进行演绎、归纳推理的非清晰性和不明确性。

参 考 文 献

- [1] L. N. Kanal, and J. F. Lemmer, Uncertainty in Artificial Intelligence, North-Holland, 1986
- [2] P. Z. Wang and H. M. Zhang, Truth-Valued Flow Inference and its Dynamic Analysis, Jour. of Beijing Normal Univ., No. 1, 1989
- [3] 赵沁平译, 李未等校, 人工智能中的逻辑, 北京大学出版社, 1990年
- [4] 陆汝钊编著, 人工智能, 上、下册, 科学出版社, 1996年
- [5] 王雨田, 现代逻辑科学导引, 下册, 中国人民大学出版社, 1987年
- [6] 魏长华, 人类自然语言不确定性及粗集理论的表示, 计算机科学, 24(5), 1997
- [7] 吴望名, 模糊推理的原理和方法, 贵州科技出版社, 1994年
- [8] 冯嘉礼著, 思维与智能科学中的性质论方法, 原子能出版社, 1991年

新的树,杂交节点是随机选取的,为保证杂交后产生的两个新个体仍有意义,一般总是要求函数节点有运算封闭性,即任何终端节点可以做什么函数的参数,且任何函数的返回值也可以做什么函数的参数。这样处理当然方便,但也带来一些问题,一是可能使搜索空间无意义地增加了许多,影响搜索速度。比如会有这样的子树出现,prog($x+1, y+x$),含义是先计算出 $x+1$ 的值,再计算出 $y+x$ 的值,而实际上,因为没有存储命令(赋值语句),这两个值计算出以后又立即被抛弃了;二是有些函数可能根本就不具备封闭性。比如,赋值函数 $=$,它要求第一个参数为变量,而两个树的杂交会产生出语义为赋值7给 $x+y$ 的语句(子树) $(x+y=7)$,显然,这是毫无意义的。

二、节点分类

用 GP 建模的文献,大多是单值输出的例子^[3,4],即建立一由多个输入得到一个输出的数学模型,因而,演化的结果也大多是运算表达式。对于要求多值输出的建模,变量和赋值函数的引入几乎是不可回避的,因此,上面提到的最后一个问题,也必须彻底解决。我们提出了节点分类的思想。象一般程序设计语言中有不同的数据类型一样,我们把节点简单分为如下四个类型:VOID, BOOLEAN, VAR, NUMBER。VOID 是空型,主要指无返回值的函数;BOOLEAN 是布尔型,指逻辑值;VAR 是变量型;NUMBER 是数值型,终端节点主要有变量型和数值型(对于终端节点,数值型相当于常数),它们都可以表示整数和实数值,函数节点的类型是指它的返回值的类型,主要有空型,布尔型和数值型。函数节点的类型在分支处注明,只允许同种类型的节点连接在该分支上,我们没有处理字符型数据,而且也没有区分整数型和实数型。这样处理以后,上面所提到的问题就可以得到很好的解决了。不过,下面有两个细节需要注意:

1) 当 A, B 两树杂交时,从 A 随机挑选的子树的类型(一般指该子树的根节点的类型,但有例外,见下段落),可能 B 中无相应类型的节点,这时,不能杂交。也可以从 A 中选取其它类型的子树,并保证 B 中也有该类型的节点。

2) VAR 类型的节点可以做为 NUMBER 类型的节点,但相反的替代是不允许的。比如,赋值函数要求第二个参数是 NUMBER 型,但可以用 VAR 型的节点,赋值函数的第一个参数是 VAR 型,就必须接 VAR 型节点,而不能接其它类型的节点。因此,在确定子树的类型时,应该是看上层节点参数所要

求的类型,而不一定是该子树根节点的类型。

三、运算实例

有以下数据,希望找出从两个输入值 H, W 得到两个输出值 C, S 的模型。

输入值 H	输入值 W	输出值 C	输出值 S
12.4	27.1	79	336.04
204.5	74.6	558.2	15255.7
13.25	38	102.5	503.5
256	86.9	685.8	22646.6
832.2	16.5	1697.4	13731.3

取函数集为 $\{+, -, *, /, =, \text{prog}\}$,终端集为 $\{W, H, C, S, X\}$,各节点意义和类型是:

H, W: 输入常量, NUMBER 型;

C, S: 输出变量, VAR 型;

X: 中间变量, VAR 型;

$+, -, *, /$: 四则运算函数,是 NUMBER 类型(因为返回值是 NUMBER 型),要求两个 NUMBER 型的参数,当 $/$ 的第二个参数为0时,令其返回值为1;

prog: VOID 型的函数,要求两个 VOID 型的参数,表示依次执行两个 VOID 参数所表示的操作;

$=$: 赋值函数, VOID 型,要求两个参数,第一个参数为 VAR 型,第二个参数是 NUMBER 型。

在规模为500,最大代数1500的一次演化中,得到一满意解,prog($= (C, + (+ (H, W), + (H, W))), = (S, * (H, W)))$,见图1。其含义是 $C = (H + W) + (H + W)$, $S = H * W$ 。

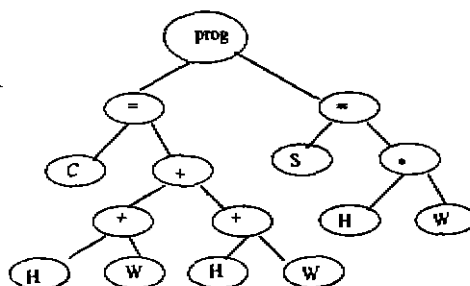


图1

参考文献

- [1] J. R. Koza, Genetic Programming II, Automatic Discovery of Reusable Programs, Cambridge, MA: MIT Press, 1994
- [2] 刘勇、康立山、陈毓屏, 非数值并行算法(第二册)遗传算法, 科学出版社, 1995
- [3] 吴少岩、陈火旺, 自动程序设计——模拟进化的途径, 计算机学报, 1997, 2
- [4] Melanie Mitchell, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, 1996