UCD CASL

Complex & Adaptive Systems Laboratory

Genetic Programming: Syntax & Semantics

Michael O'Neill

Computational Toolbox Winterschool

Engelberg 8-12 Feb 2014

Overview

Genetic Programming: Syntax & Semantics

1. Setting the Stage
    - What is Natural Computing?
    - What is Evolutionary Computation?
    - An Introduction to Genetic Programming (GP)
2. Grammar-based GP
3. Semantic methods & Open Issues in GP

Semantic Methods & Open Issues in GP

Semantic Methods

- Over dependence on fitness (single point)
- Credit Assignment
- Semantic analysis of evolving populations
- Semantic-aware program construction
- Semantic-aware search operators

Attribute Grammars - Adding Semantics to Solution Construction

$$\text{maximise} \sum_{j=1}^{n} p_j x_j \quad (1)$$

$$\text{subject to} \sum_{j=1}^{n} w_{ij} x_j \leq c_i, \qquad i = 1 \ldots m \quad (2)$$

$$x_j \in \{0,1\}, \qquad j = 1 \ldots n \quad (3)$$

1

$S \rightarrow K \qquad lim(K) = lim(S)$

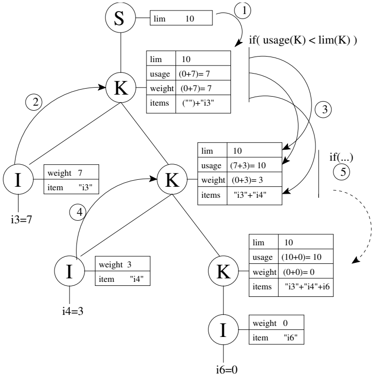
$K \rightarrow I \qquad weight(K) = weight(K) + weight(I)$
Condition : $if(usage(K) + weight(I) <= lim(K))$
$items(K) = items(K) + item(I)$

$K_1 \rightarrow I K_2 \qquad weight(K_1) = weight(K_1) + weight(I)$
$items(K_1) = items(K_1) + item(I)$
$usage(K_1) = usage(K_1) + weight(I)$
Condition : $if(usage(K_1) < lim(K_1))$
$lim(K_2) = lim(K_1)$
$usage(K_2) = usage(K_1)$
$items(K_2) = items(K_1)$

$I \rightarrow i_1 \qquad item(I) = \text{“}i_1\text{”}$
Condition : $if(notinknapsack?(i_1))$
$weight(I) = weight(i_1)$

$\vdots$

$I \rightarrow i_n \qquad item(I) = \text{“}i_n\text{”}$
Condition : $if(notinknapsack?(i_n))$
$weight(I) = weight(i_n)$



[1] Where, p_j refers to the profit, or worth of item j, x_j refers to the item j, w_{ij} refers to the relative-weight of item j, with respect to knapsack i, and c_i refers to the capacity, or weight-constraint of knapsack i. There are present $j = 1 \ldots n$ items, and $i = 1 \ldots m$ knapsacks.

http://ncra.ucd.ie

Semantic-aware search operators

- Crossover
- Mutation
- Semantic Locality & Diversity

Semantic Similarity Crossover

Algorithm 1: Semantic Similarity based Crossover

select Parent 1 P_1;
select Parent 2 P_2;
Count=0;
while *Count<Max_Trial* **do**
  choose a random crossover point $Subtree_1$ in P_1;
  choose a random crossover point $Subtree_2$ in P_2;
  generate a number of random points (P) on the problem domain;
  calculate the SSD between $Subtree_1$ and $Subtree_2$ on P
  if $Subtree_1$ *is similar to* $Subtree_2$ **then**
    execute crossover;
    add the children to the new population;
    return true;
  else
    Count=Count+1;
if *Count=Max_Trail* **then**
  choose a random crossover point $Subtree_1$ in P_1;
  choose a random crossover point $Subtree_2$ in P_2;
  execute crossover;
  return true;

Sampling Semantic Distance

Based on SS, we define a *Sampling Semantic Distance* (SSD) between two subtrees. It differs from that in [24] in using the mean absolute difference in SS values, rather than (as before) the sum of absolute differences. Let $U = (u_1, u_2, ..., u_N)$ and $V = (v_1, v_2, ..., v_N)$ represent the SSs of two subtrees, S_1 and S_2; then the SSD between S_1 and S_2 is defined in equation 1:

$$\mathrm{SSD}(S_1, S_2) = \frac{\sum_{i=1}^{N} |u_i - v_i|}{N} \tag{1}$$

We follow [24] in defining a semantic relationship, *Semantic Similarity* (SSi), on the basis that the exchange of subtrees is most likely to be beneficial if they are not semantically identical, but also not too different. Two subtrees are semantically similar if their SSD lies within a positive interval. The formal definition of SSi between subtrees S_1 and S_2 is given in the following equation:

$$\mathrm{SSi}(S_1, S_2) = \textbf{TruthValue}(\alpha < \mathrm{SSD}(S_1, S_2) < \beta)$$

where α and β are two predefined constants, the *lower* and *upper* bounds for semantics sensitivity. In general, the best values for these semantic sensitivity bounds are problem dependent. In this work we set $\alpha = 10^{-4}$ and several values of β were tested.

Functions	Training Data	Testing Data
$F_1 = x^4 + x^3 + x^2 + x$	20 random points $\subseteq$ [-1,1]	30 points $\subseteq$[0:0.05:1.5]
$F_2 = x^3 - x^2 - x - 1$	20 random points $\subseteq$ [-1,1]	30 points $\subseteq$[0:0.05:1.5]
$F_3 = arcsin(x)$	20 random points $\subseteq$ [-1,0]	30 points $\subseteq$[-1:0.67:1]
$F_4 = \sqrt{x}$	20 random points $\subseteq$ [0,2]	30 points $\subseteq$[0:0.1:3]
$F_5 = 0.3sin(2\pi x)$	20 random points $\subseteq$ [-1,1]	30 points $\subseteq$[0:0.05:1.5]
$F_6 = cos(3x)$	20 random points $\subseteq$ [-1,1]	30 points $\subseteq$[0:0.05:1.5]

Semantic vs. Syntactic Locality

Table 3. Comparison of the effects of SC, SSC and SySC on GP performance (mean of the best fitness). The values are scaled by 10^2.

Xovers	F_1	F_2	F_3	F_4	F_5	F_6
SC	1.51	3.07	0.37	0.96	4.36	1.48
SySC6	1.63	3.20	0.46	1.06	4.42	1.46
SySC8	1.49	3.50	0.43	0.99	4.36	1.98
SySC10	1.56	3.08	0.39	1.18	4.41	2.04
SSC04	0.78	1.30	0.20	0.58	3.36	0.67
SSC05	0.85	1.40	0.21	0.61	3.28	0.81
SSC06	0.87	1.70	0.22	0.38	3.44	0.92

Open Issues

1. Identifying appropriate representations for GP
2. Fitness landscapes & problem difficulty in GP
3. Static vs. Dynamic Problems
4. The influence of biology on GP
5. Open-ended evolution in GP
6. Generalisation in GP
7. GP Benchmarks
8. GP and Modularity
9. The Complexity of GP

Open Issues

10 Miscellaneous issues:

- The Halting Problem
- How much Domain Knowledge?
- GP Theory
- Constants in GP
- Bloat
- Distributed GP
- The Elephant in the Room!

Sample Literature

Semantic Methods & Open Issues

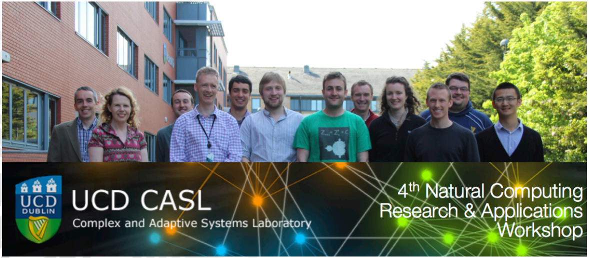
- Johnson C. (2002) Deriving Genetic Programming Fitness Properties by Static Analysis. Proceedings of the 4th European Conference on Genetic Programming (EuroGP 2002), pp.299-308. Springer.
- McPhee, Ohs, Hutchison (2008) Semantic building blocks in GP. Proceedings of 11th European Conference on Genetic Programming, pp.134-145. Springer.
- Beadle L, Johnson C G (2008) Semantically Driven Crossover in Genetic Programming. In: Proceedings of the IEEE World Congress on Computational Intelligence (CEC 2008), pp 111–116, IEEE Press
- Beadle L, Johnson C G (2009) Semantic Analysis of Program Initialisation in Genetic Programming. Genetic Programming and Evolvable Machines 10(3):307–337
- Beadle L, Johnson C G (2009) Semantically Driven Mutation in Genetic Programming. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009), pp 1336–1342, IEEE Press
- Krawiec K (2011) Semantically embedded genetic programming: automated design of abstract program representations. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO 2011), pp 1379–1386, ACM Press
- Krawiec K, Lichocki P (2009) Approximating geometric crossover in semantic space. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO 2009), pp 987–994, ACM Press

Sample Literature (continued)

Semantic Methods & Open Issues

- Krawiec K, Wieloch B (2009) Analysis of Semantic Modularity for Genetic Programming. Foundations of Computing and Decision Sciences 34(4):265–285
- Nguyen Q U (2011) Examining Semantic Diversity and Semantic Locality of Operators in Genetic Programming. PhD Thesis. University College Dublin
- Nguyen Q U, Nguyen T H, Nguyen X H, O'Neill M (2010) Improving the Generalisation Ability of Genetic Programming with Semantic Similarity based Crossover. In: Proc. of EuroGP 2010, pp.184–195, Springer
- Nguyen Q U, Nguyen X H, O'Neill M, McKay B (2010) The Role of Syntactic and Semantic Locality of Crossover in Genetic Programming. In: Proc. of PPSN 2010, LNCS 6239, pp 533–542, Springer
- Nguyen Q U, Nguyen X H, O'Neill M, McKay R I, Galvan-Lopez E (2011) Semantically-based Crossover in Genetic Programming: Application to Real-valued Symbolic Regression. Genetic Programming and Evolvable Machines 12(2):91–119
- Moraglio A., Krawiec K., Johnson C. (2012) Geometric Semantic Genetic Programming. In PPSN 2012, p21-31. Springer.
- D. R. White et al., (2013). "Better GP benchmarks: community survey results and proposals," Genetic Programming and Evolvable Machines, 14(1):3–29.
- ONeill M., Vanneschi L., Gustafson S., Banzhaf W. (2010). Open Issues in Genetic Programming. Genetic Programming & Evolvable Machines. 11(3-4):339-363.

Thank You



sfi
Science
Foundation
Ireland