· U C ·

Evolutionary Computation

2016/2017

Notes for the PL 7

Ernesto J. F. Costa

21 de Março de 2017

Class PL 7

Genetic Programming: a few experiments

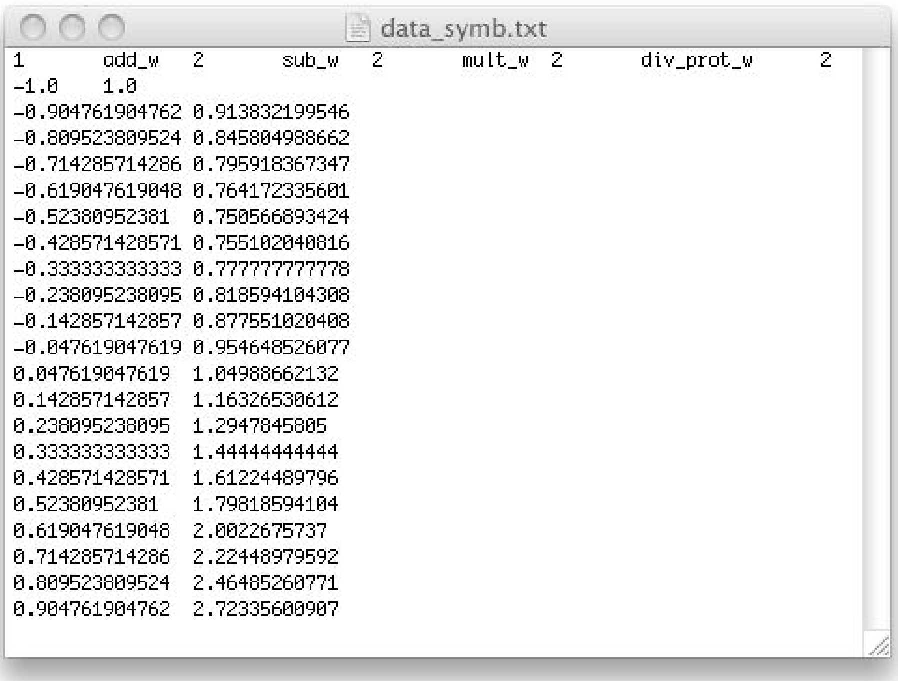
6.1 Introduction

In the lectures we introduced the basics of genetic programming (GP). The most relevant aspect of this approach is the fact the the algorithm evolves programs which, when executed, solve a particular problem. In this lab we are going to do some experiments using a simple implementation of GP, in order to grasp the way it works. We will also try to do some modifications to the provided algorithm, and test the best configuration for the several parameters. We will use classical benchmark problems.

You must download from InforEstudante the following files, provided in a one single compressed file:

1. sgp_alunos.py
2. gen_data.py

In order to succeed you must study previously the given code to understand how it works. The data to be used are generated by a program and stored into a file. The file has an header that defines the number of input variables of the problem, followed by pairs indicating a function's name and arity. After the header we have the examples, one by line, of pairs inputs-output. The figure 6.2 show the example of symbolic regression.



```
1       add_w   2       sub_w   2       mult_w  2       div_prot_w      2
-1.0    1.0
-0.904761904762 0.913832199546
-0.809523809524 0.845804988662
-0.714285714286 0.795918367347
-0.619047619048 0.764172335601
-0.52380952381  0.750566893424
-0.428571428571 0.755102040816
-0.333333333333 0.777777777778
-0.238095238095 0.818594104308
-0.142857142857 0.877551020408
-0.047619047619 0.954648526077
0.047619047619  1.04988662132
0.142857142857  1.16326530612
0.238095238095  1.2947845805
0.333333333333  1.44444444444
0.428571428571  1.61224489796
0.52380952381   1.79818594104
0.619047619048  2.0022675737
0.714285714286  2.22448979592
0.809523809524  2.46485260771
0.904761904762  2.72335600907
```

Figura 6.1: Symbolic Regression

6.2 The problems

6.2.1 Symbolic Regression

The problem of symbolic regression consists in finding the best polynomial that approximates a given set of points. In our case we will deal with the case of the polynomial

$$x^2 + x + 1$$

with $x \in [-1, 1]$. You should use the programs provided in file **gen_data.py** to generate the training set of data. The figure 6.2 show the polynome when you use a set of 21 points.

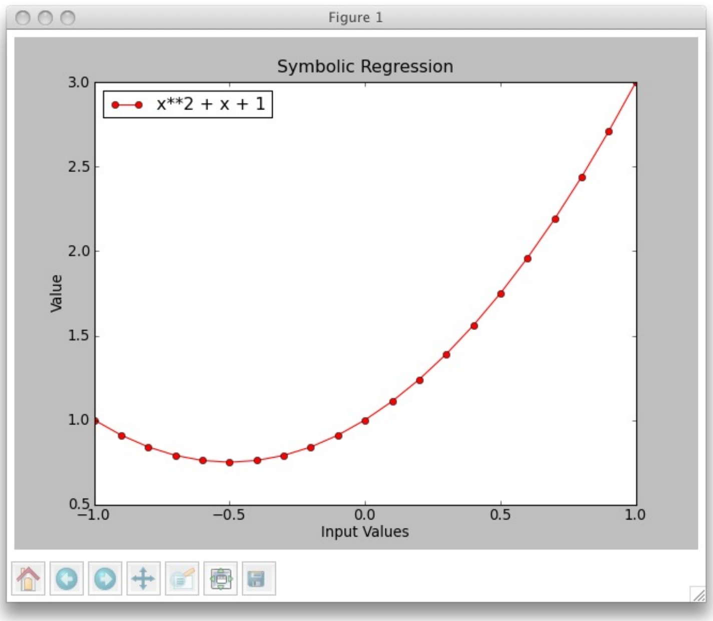


Figura 6.2: Symbolic Regression

6.2.2 The `sin` function

The `sin` function is just the ... the `sin` function. We will work with a training set of 62 points in the interval $x \in [-3.14, 3.14]$. The classical figure in shown in 6.3.

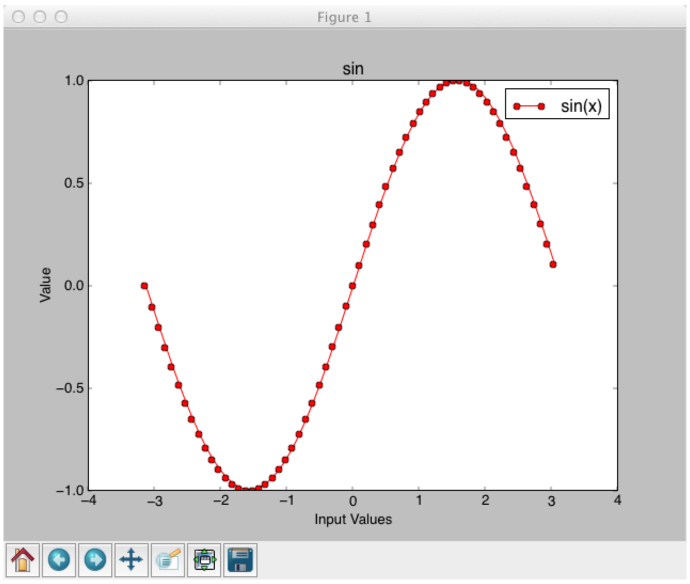


Figura 6.3: The `sin` function

6.3 Exercices

Problema 6.1 **MF**

Use the code given to generate the data for the two problems that we are going to use: symbolic regression and the sin function.

Problema 6.2 **F**

The given code of GP uses the following:

- ramped-half-and-half to initialize the population
- tournament selection of the parents
- sub-tree crossover
- point-mutation
- selection of survivors of type generational

Test the code for different values of the parameters, such as the population and tournament sizes and the probabilities of mutation and crossover. Change each parameter at a time keeping the others fixed. What conclusions can you draw?

Problema 6.3 F

Change the code provided so you can visualize the evolution over generations of the fitness of the best individual and of the average fitness of the population. Try do to the modifications as modular as possible, i.e., defining the a **display function** that plot the data provided by the genetic programming algorithm.

Problema 6.4 F

Define a new function to **run** the algorithm several times and then plotting the results of the best over all and of the average best by generation.

Problema 6.5 M

Implement two **variants** of the survivors's selection mechanism:

a) Survivors' selection of type elitist

b) Survivors' selection of type merge and select

For the selection of the survivors of type elitist try several percentage of individual that go unchanged to the next generation, and determine a good value this parameter. In the case of merge and select just merge the parents and the offspring populations, and choose the best individuals. Devise an experiment to compare the three types of selecting the survivors implemented(these two plus the generational). Any conclusion?