Natural Computing 2021/2022: Assignment

Genetic programming

--- Preliminary Version ---

This is an exercise worth 40% of the course mark. The maximal score you can get for the assignment is 100 points. You are asked not to discuss or share your solution with other students. The assignment requires you to write a program in Python, and to use it to carry out a number of investigations. You are to submit your program code and a concise report. It will be helpful if you include some comments in your code, but there will be no marks for well-written code, if it does not produce good results.

Task description

Problem 1: Analysis of Particle Swarm Optimisation (40 points/100 points)

Analyse the behaviour of the standard PSO algorithm for minimisation of these two functions:

$f_1(x)=\sum_i x_i^2$, where $x=(x_1, ..., x_d)$, and $x_i \in [-5.12, 5.12]$, sphere function

$f_2(x)=10d+\sum_i (x_i^2-10\cos(2\pi x_i))$, where $x=(x_1, ..., x_d)$, and $x_i \in [-5.12, 5.12]$, Rastrigin function

1.1 Find parameters α_1, α_2, and ω with which good results can be obtained in a short time. It is useful to fix the search space dimension *d* to a reasonable value, and to define (and to justify) a termination criterion that is appropriate for this task. (20%/40%)

1.2 How do your results for the two functions depend on the number of particles *N* and on the search space dimension *d*? (10%/40%)

1.3 Modify the PSO and show that it improves the results for both functions. You may use one of the modifications discussed in class, e.g. repulsion, or from a paper or one of your own. (10%/40%)

Question 2: Genetic Programming (20%/100%)

Generate datasets from the two functions above (for d=2) and use a GP to reproduce the two functions from the previous from the respective data set.

Question 3: PSO + GP (20%/100%)

Combine your implementations of GP and PSO, and demonstrate for a suitably chosen example the improvement of this combined algorithm over the GP that you were using in the previous questions.

Question 4: Symbolic regression (20%/100%)

Improve your GP program code such that it is able to solve an unknown symbolic regression problem. Submit it as Python source code in a single file with filename "question4.py". Your program for this part should use basic libraries (such as numpy and pandas) only. It should read-in a file "datafile.txt" that contains 400 triples "x y f(x,y)", one triple per line. A sample file for this format

will be given, but your code will be tested on a different data file in the same format. The data will be obtained from a simple function with depth 2 or 3, and another function with depth ≥ 3. The two functions to be inferred can be represented as trees with unary and binary from the following elements

constants: `1, 2, 3, 4, 10, 100,` π

variables: `x, y`

operations: `+` (plus), `*` (times), `-` (minus), `/` (divide)

functions: `sin, cos, exp, abs, power(.,n), power(.,1/n) with` n=1, 2, 3, 4

Your code should not run longer than 5 mins on a standard single processor machine (there will be some grace period before timeout). The output of your program should be a single string to stdout that represents the inferred function.

You may find the following questions useful in your solution of the assignment

Try to keep the number of fitness cases and numerical constants within reasonable bounds, but consider also to add a local search method to adjust numerical factors. You can also try to add some good guesses, e.g. the constant π is here more likely to occur inside a trigonometric function and direct combinations of more than two constants (such as "1+1+1") are unlikely.

You can use a GP program that produces only polynomials with integer coefficients, but you may consider also either more general form or restrictions to polynomials of a certain form.

How does your algorithm deal with ‘bloat’ i.e. an unnecessary growth of the size of the program?

How do you decide about the termination of a single run, and how on repetitions of runs?

If you need to choose values of parameters of the GP algorithm, can you explain and justify how you came to this choice?

[This section may be expanded in updates of this Assignment]

Structure of the report

In the report on your work on this assignment use a separate page for each of the four questions. If you choose to add a front page or a general introduction, restart the page numbering such that Question 1 is on page 1, Question 2 is on page 2 etc. If you need to include more material, please use an appendix, but consider that marks may be deducted, if your report is not concise. Although the space is limited, try to give a justification and explanation of your work, rather than just a list of the steps that you have performed.

It is possible to get full marks for question 4 for the submitted code alone, but if you are not completely sure about your code, please use the opportunity to explain your working and possible limitations. Also mention any particular features or tricks that you have used in your implementation.

Wherever appropriate, use graphical representations of your numerical results.

Feel free to use exciting resources and literature to prepare your work and to support your argument, but make sure to cite all the papers and resources that you have used for this assignment. Place this list of references after the last text page (page 4), these will not count towards the page limit.

If you were using a Notebook for programming tasks towards this assignment, please extract the Python code from the notebook, e.g. by

```
jupyter nbconvert --no-prompt --to script YourNotebook.ipynb
```

and submit only the Python code. You may also like to check whether the output of the converted file is meaningful. See also the remarks in the first paragraph above.

Submit your report as a PDF file and your code as a zipped file (use zip or gzip) via Learn.

Deadline for submissions is **4pm of Thursday, 18. November 2021.**

In the tutorial session and in the classroom Q&A sessions will be time to clarify your questions related to the assignment. For critical questions, please contact michael.herrmann@ed.ac.uk directly.