VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Engineering and Computer Science

COMP 307 — Lecture 12

Evolutionary Computing 3 (ML 9)

GP for Symbolic Regression and Classification

Prof. Mengjie Zhang (Meng) and Dr Bing Xue

mengjie.zhang@ecs.vuw.ac.nz and bing.xue@ecs.vuw.ac.nz

Outline

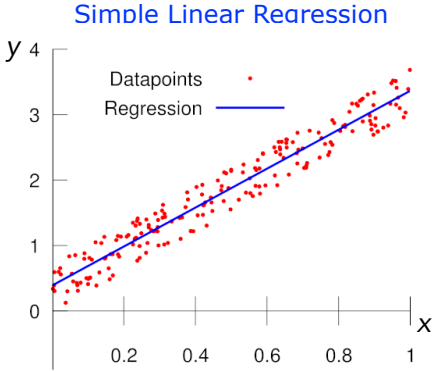
- Statistical parameter regression
- Symbolic regression
- GP for symbolic regression
- GP for binary classification

(*Statistical*) Regression Analysis

- In statistics, regression analysis examines the relation of a dependent variable (response variable) to specified independent variables (explanatory variables)
  - The *mathematical model* of their relationship is the *regression equation*
  - *estimates of one or more hypothesized regression parameters ("constants")*
- Allow predicting the value of the dependent variable for given value(s) of independent variable(s)
- e.g. *curve fitting, prediction, modelling of causal relationships, and testing scientific hypotheses about relationships between variables*

(*Statistical*) Regression Analysis

- Process
  - Given data points
  - Assume linear model
  - function:
    $$y = \alpha + \beta x + \epsilon$$
  - α is the intercept
  - β is the slop
  - ϵ is the error term
  - The error term is usually taken to be normally distributed
  - Use some methods to estimate α and β



Symbolic Regression

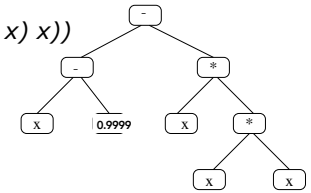
- Problems of statistical (parameter) regression
  - Need domain expertise to assume certain distribution of the given data, which is usually unknown in advance
  - Need statistical expertise to find an "appropriate" model, which is usually very hard
- Symbolic regression: the object to be found is a symbolic description of a model, not just a set of coefficients/ parameters in a pre-specified model.
- To find both:
  - the model structure, and
  - the corresponding coefficients/parameters

GP for Symbolic Regression

- Objective: Find a program/model that produces the correct value of the dependent variable y when given the value of independent variable x
- Terminal Set: *x* and a random number *r*
- Function Set: {+ , -, *, %}
- Fitness Cases: 50 sets of *x* and the corresponding *y* values (e.g. *50 instances/patterns/cases*)
- Fitness Measure: Sum of the errors for the 50 cases
- Parameters: Population = 100. Generations = 51, ProgSize = 6? reproduction rate: 5%, crossover rate: 90%, mutation rate: 5%
- Success: The error measure for each of the 50 points is less than 0.011
- Termination criteria: satisfactory solutions found, or at generation 51.

GP for Symbolic Regression

- One run gave: $y = (x - 0.9999) - x^3$ which can be written as *(- (x-0.9999) (*(* x x) x))*
  - Successful ? if the "*true*" model is
  - $y = (x - 1) - x^3$



- Sometimes:
  (% (% (* (* X 0.571) (* (- (* (+ (% 0.634094 0.68469) (+ (+ X X) -0.5992))(* (* (+ (% 0.634094 0.68469) (+ X -0.5992)) (* (% 0.354904 - 0.7549) (* X 0.571))) (- X 0.395493))) - 0.4665) ....another 15 lines)
- This example: one input variable (x), training set only
- Real-world applications: usually multiple variables, can have a separate test set, but use the same principle

GP for Symbolic Regression Problems: Properties

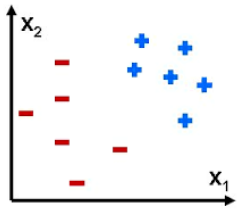
- Compared with statistical parameter regression methods, GP method has the following properties:
  - Does NOT need to assume any distribution of data set,
  - Does NOT need to assume the independence of the input variables
  - Does NOT need to use any statistical background knowledge to assume any model
  - Can automatically learn/***evolve*** both the model structure and the model parameters at the same time!
  - System input: just the data with a black box model/parameters
  - System output: a white box model structure with appropriate parameters!

GP for Symbolic Regression Applications

- GP for symbolic regression has many real-world applications, including:
  - Economic prediction, e.g. stock market prediction, GDP prediction,
  - Industrial prediction, e.g. prediction of containers handling capacity at a particular sea port; short-term, medium-term and long-term prediction of power load at a region
  - Experiential formula modelling in Engineering, e.g. formulating the amount of Gas emitted from Coal surface
  - Time series projection, e.g. CPI projection for a country or a region
  - Selection/Choice of Equipments, e.g. equipment choice for work platform in mine industry
  - Fault diagnosis, e.g. find optimal strategy in fault isolation, fault analysis in combustion system for diesel engine
  - Robot self-adaptive behaviour
  - GIS systems, e.g. projection transformation

Binary Classification

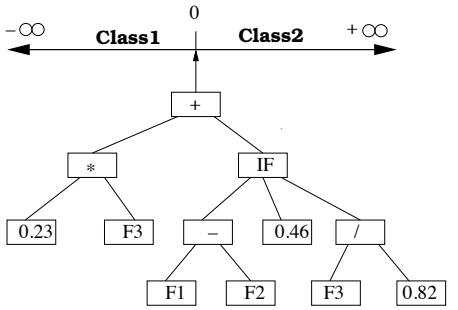
- Binary classification is the task of classifying the instances of a given set into two categories on the basis of whether they have some property or not
- Two target classes, e.g.
  - Disease vs non-disease
  - normal vs abnormal
  - grant loan or not
  - fault vs non-fault/normal
  - X vs O
  - object vs non-object



GP for Binary Classification

- Compared with GP for symbolic regression problems, the terminal set and function set can be the same or very similar, but the fitness function is normally very different
- In fitness function, we can simply use classification accuracy or error rate, which need to determine which class a training example belongs to
- This is called Classification Strategy or Program Class Translation Rule
- For binary classification problems, this is quite easy: we can use the value "zero" in the real number space for the program output to separate the two classes

GP for BC — Program Class Translation Rule




```
Genetic Program: (+ (* 0.23 F3)
                    (IF (- F1 F2) 0.46 (/ F3 0.82))
                 )
```

```
if ProgOut < 0  then  Class1  else  Class2;
```

GP for Classification Example

- Task: Object classification: *objects vs non-objects*
- Objective: Find a program which can successfully *split* the instances into *two* classes
- Terminal Set: Object *attributes*: pixels, pixel statistics, or specific features, and *random numbers*.
- Function Set: {+ , -, *, %, ABS, EXP, LOG, SIN, COS, RAND}
- Fitness Cases: Build a training set of *patterns* (*feature vectors*), some are objects, some not.
- Fitness Measure: classification accuracy/ error rate
- Classification strategy: ProgOut > 0 for objects, otherwise non-objects
- Parameters: Population = 200? Generations = 50? Crossover rate =? Mutation rate = ? Program size =?

Basic GP Algorithm

This GP algorithm is based on the proportional selection model — (check Slide 4)

1. Initialise the population
2. Evaluate the fitness of each individual program in the current population.
3. Until the new population is fully created, repeat the following:
   - Select programs in the current generation.
   - Perform genetic operators on the selected programs.
   - Insert the result of the genetic operations into the new generation.
4. If the termination criterion is not fulfilled, repeat steps 2-4 with the new generation.
5. Present the best individual in the population as the output.

Tackling a Problem with GP

- What is the set of terminals used in the program trees?
- What kind of functions can be used to form the function set to represent the program tree?
- What is the fitness measure?
- What values can be given for the parameters and variables for controlling the evolutionary process, for example, population size and number of generations?
- When to terminate a run?
- How do we know the result is good enough?
- What genetic operators, at what frequencies, are going to be applied?

Summary

- GP for symbolic regression
- Properties of GP for symbolic regression
- GP for binary classification

- How do you use GP for multi-class classification? Can we get better translation rules? COMP422

- Next Lectures: Quantifying uncertainty and probabilistic reasoning