Introduction to Artificial Intelligence

COMP307

Evolutionary Computing 3: Genetic Programming for Regression and Classification

Yi Mei

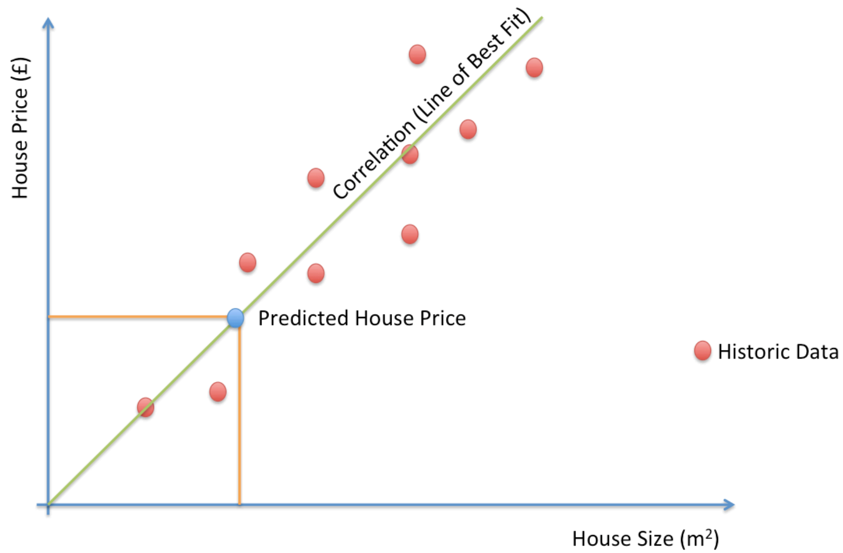
yi.mei@ecs.vuw.ac.nz

Outline

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

House Price Prediction

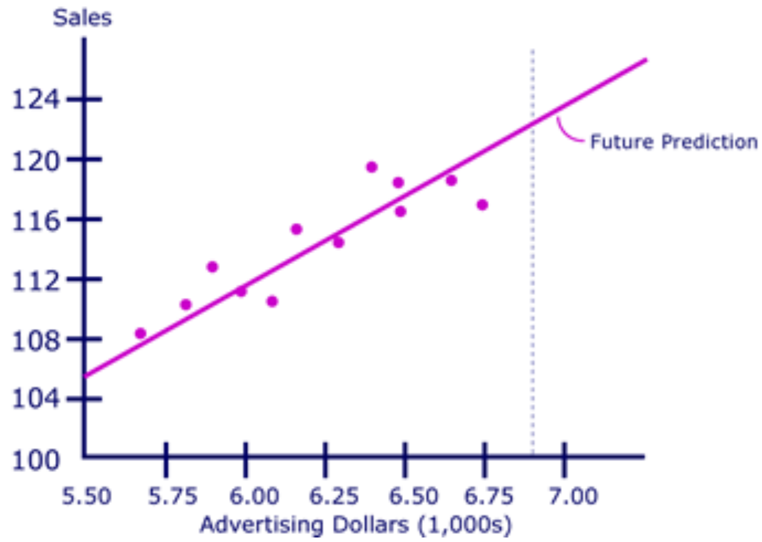
- How much to put in your tender?



Price	Floor space	Rooms	Lot size	Appartment	Row house	Corner house	Detached
250000	71	4	92	0	1	0	0
209500	98	5	123	0	1	0	0
349500	128	6	114	0	1	0	0
250000	86	4	98	0	1	0	0
419000	173	6	99	0	1	0	0
225000	83	4	67	0	1	0	0
549500	165	6	110	0	1	0	0
240000	71	4	78	0	1	0	0
340000	116	6	115	0	1	0	0

(Statistical) Regression Analysis

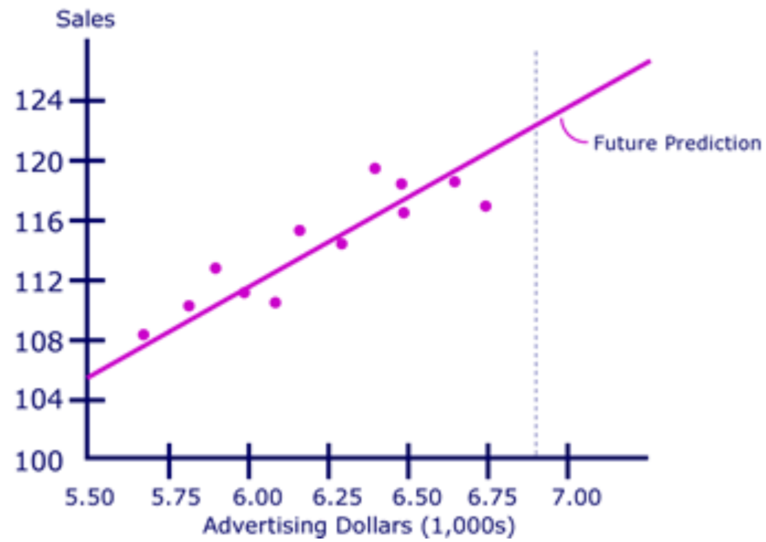
- In statistics, regression analysis examines the relation of a dependent variable (response variable) to specified independent variables (explanatory variables)
  - The mathematical model of their relationship is the regression equation (e.g. $f(x, y) = 0$)
  - estimates of one or more hypothesized regression parameters ("constants")
- Examples
  - Financial prediction
  - Saving prediction
  - Ad cost vs sales
  - Natural law discovery



Regression

- Two main tasks
  - **Regression equation**: relationship between a dependent variable (response variable) to specified independent variables (explanatory variables)
  - **Parameters/Coefficients**

- Example: linear regression
  - Regression equation is a linear model: $y = \alpha \cdot x + \beta + \epsilon$
  - Coefficients: α and β
  - α is the slope, β is the intercept
  - ϵ is the error term (assume normally distributed)
  - Estimate α and β to minimise ϵ

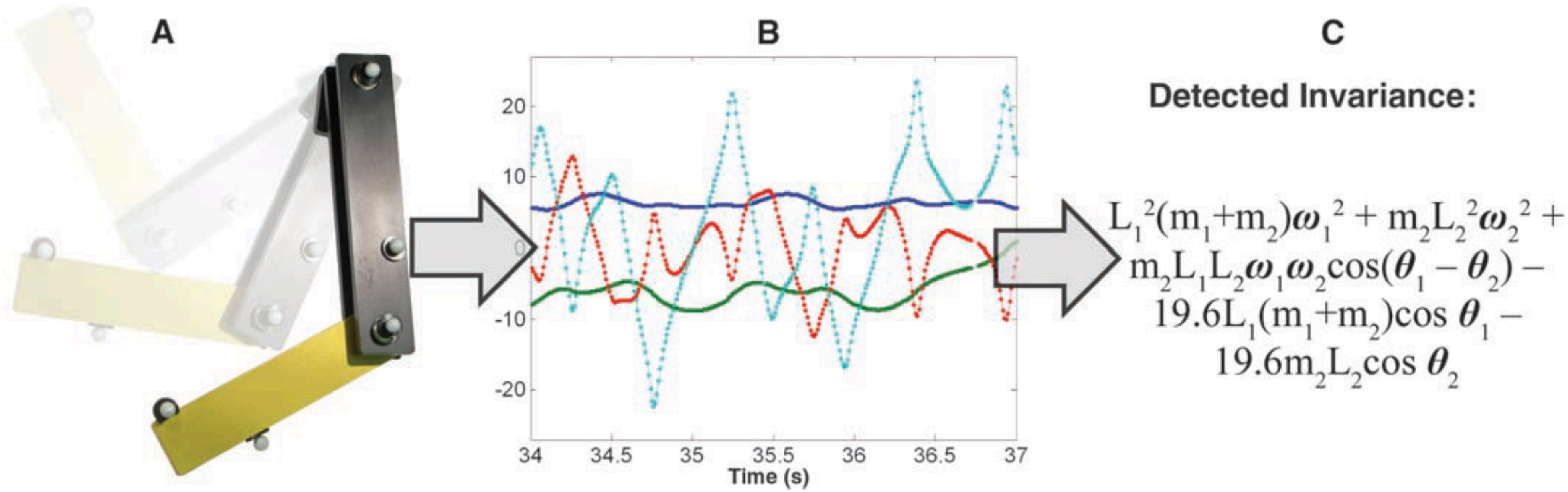


Assume model structure, estimate model parameters

Symbolic regression

- However, linear model is too simple for many real-world data

- Hard to find out the proper regression equation



Schmidt, Michael, and Hod Lipson. "Distilling free-form natural laws from experimental data." *Science* 324, no. 5923 (2009): 81-85.

Symbolic Regression Applications

- Symbolic regression has many real-world applications:
  - 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

Symbolic regression

- Symbolic regression: to find 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

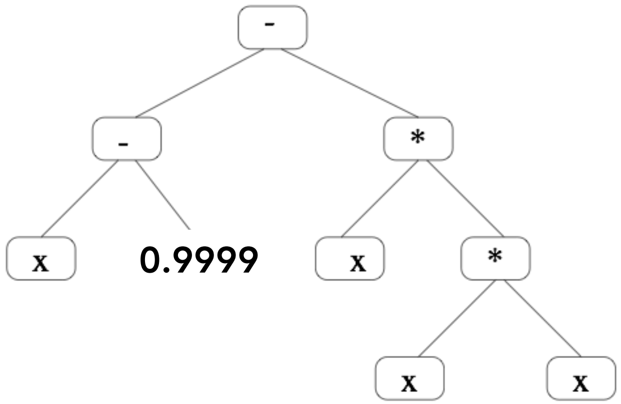
GP for Symbolic Regression (Example)

- Objective: Find a program/model that produces the correct value of the dependent variable y when given the value of an independent variable x
- Terminal Set: *x*, random number *r*
- Function Set: {+ , -, *, %}
- Fitness Cases: 50 cases of *x* and the corresponding *y* values (e.g. *50 instances/patterns/cases*)
- Fitness Measure: Sum of the *absolute* errors for the 50 cases
- Parameters: Population = 100, Generations = 51, MaxDepth = 17
  reproduction rate: 5%, crossover rate: 90%, mutation rate: 5%
- Success: The fitness value is smaller than a pre-defined value, e.g. 0.01
- Termination criteria: satisfactory solutions found, or at generation 51.

GP for Symbolic Regression (Example)

- One GP run gave: $y = (x - 0.9999) - x^3$
- 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

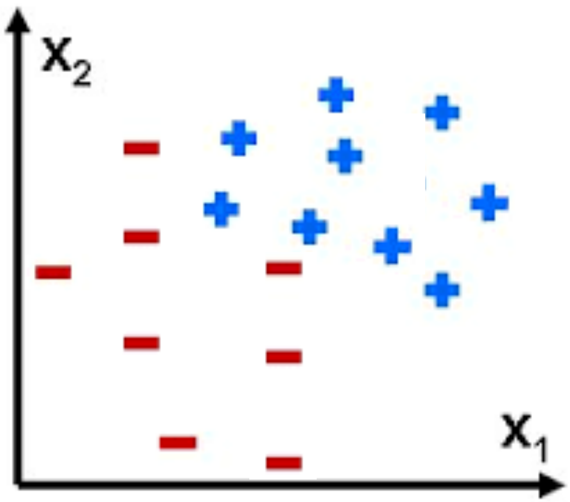
- Compared with statistical parameter regression methods, GP method has the following properties:
- Does **NOT** need to assume any distribution of data set,
- Does **NOT** need to assume the independence of the input variables
- Does **NOT** need to use any statistical background knowledge to assume 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

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
  - object vs non-object



GP for Binary Classification

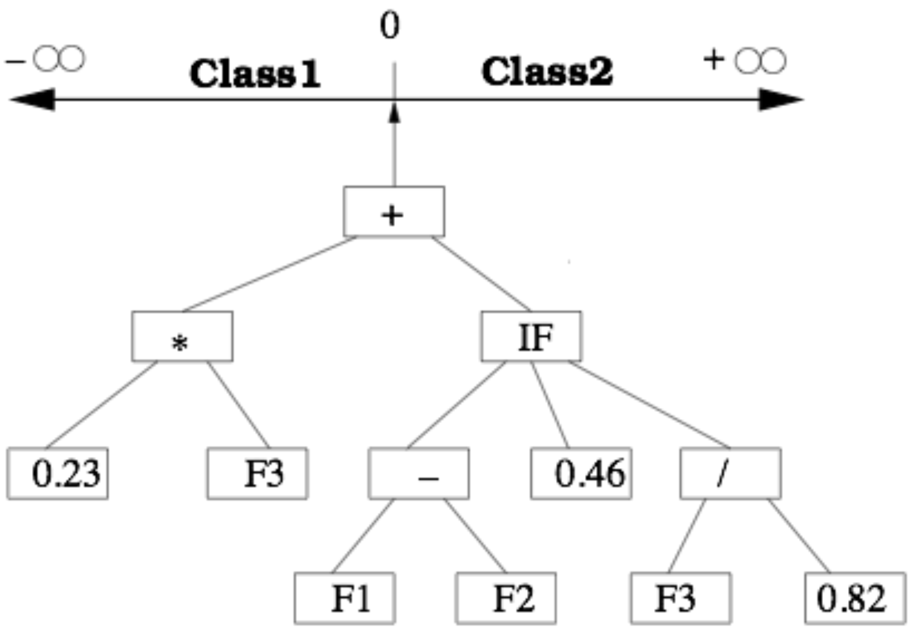
- Compared with GP for symbolic regression problems, the terminal set and function set can be the same or very similar, but the fitness function is normally very different
- In fitness function, we can simply use classification accuracy or error rate, which need to determine which class a training example 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

Program Class Translation Rule

- Is zero the best threshold?




```
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

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?