

## ACISS 2009 Tutorial Genetic Programming for Data Mining

Mengjie Zhang

School of Engineering and Computer Science  
Victoria University of Wellington  
New Zealand

Copyright is held by the author/owner(s).  
First Australasian Computational Intelligence  
Summer School (ACISS'09), Melbourne,  
Australia, 30 Nov - 1 Dec 2009

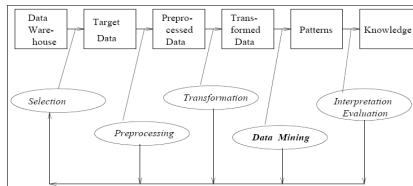
## Outline

- Data Mining
- Evolutionary Computing and Genetic Programming
- GP for Symbolic Regression
- GP for Classification
- Challenges and Issues
- Upcoming events

2

## Data Mining and Knowledge Discovery

- KDD: is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [Fayyad].
- DM: a step



3

## Data Mining Tasks

- Classification (✓)
- Regression (✓)
- Prediction
- Time Series analysis
- Clustering
- Summarisation
- Association rules
- Sequence discovery

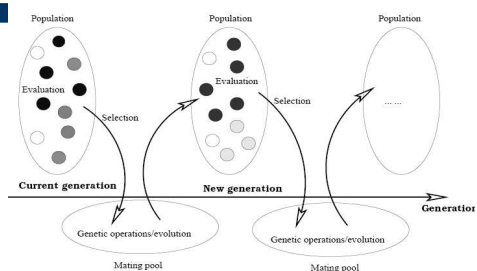
4

## Evolutionary Computation

- Population based
- Multiple solutions
- Global search
- Many paradigms:
  - GAs, GP
  - ES, EP
  - PSO
  - DE, EDA, ...

5

## EC Process



6

## Genetic Programming

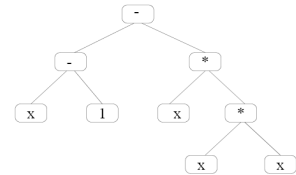
- GP inherits properties from EC techniques (GAs) and *automatic programming*
  - GP uses a similar evolutionary process to the general EC/GAs
  - bit strings chromosomes vs tree-like structures **that can represent computer programs** such as LISP (and C, Java)
  - **fixed** length representation vs **Variable** length
- Automatically learning a set of computer programs for a particular task is a dream of computer scientists
- GP is such a technique that helps achieve this goal

7

## Programs as Tree Structures

$$(x - 1) - x^3$$

$$(- (- x 1) (* x (* x x)))$$



8

## GP Programs

- Terminal set: features/attributes from a task, constants (coefficients)
- Function set:
  - Standard functions: +, -, \*, /, sin, log, exp, ...
  - Task/domain specific functions
- Sufficiency and closure
  - Selection of the functions and terminals is critical to success
  - Sufficient vs redundant

9

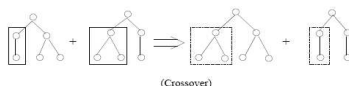
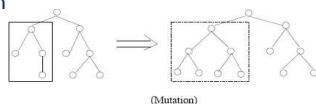
## Program Generation

- For initialising a population or mutation.
- **Maximum program depth**: the maximum size permitted for a program,
- **Program generation methods**:
  - **Full**:
  - **Grow**, and
  - **Ramped half-and-half**:

10

## Genetic Operators in GP

- Reproduction
- Crossover
- mutation



11

## Fitness Cases and Fitness Function

- Fitness cases: instances, training/test
- Fitness is the **measure** of how well a program has learnt to predict the output from the input during simulated evolution
- The fitness of a program is calculated using the **fitness function** via **program evaluation**.
- The fitness function should be designed to give **graded** and **continuous** feedback

12

## Fitness Function Examples

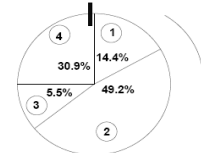
- Image matching: the number of matched pixels
- Robot learning obstacle avoidance: the number of wall hits for a robot
- Classification task: the number of correctly classified examples
- Prediction application: the deviation between prediction and reality
- GP-controlled agent in a betting game: the amount of money won
- Artificial life application: the amount of food found and eaten.

13

## Selection

- Fitness selection determines which evolved program will be used by the genetic operators for evolution

- *Proportional selection*
- *Tournament selection*



14

## Basic GP Algorithm

1. Initialise the population
2. Evaluate the individual programs in the current population. Assign a fitness to each program.
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.

15

## Tackling a Problem with GP

If a GP package is available:

- Determine the terminal set, function set
- Determine the fitness function
- Determine the parameter values
  - Pop size, program size/depth, max generations, crossover/mutation/reproduction rates, etc.
- Determine the stopping criteria

16

## Regression Analysis and Modelling

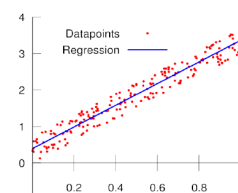
- In statistics, regression analysis examines the relation of a **dependent variable** (response variable) to specified **independent variables** (explanatory variables)
- The **mathematical model** of their relationship is **regression equation**
- A regression equation contains estimates of one or more hypothesized regression parameters
- The estimates measure the relationship between the dependent variable and each of the independent variables

17

## Statistical Regression Example

- Simple Linear Regression
- Process:

- Given data points
- Assume linear model
- $y_i = \alpha + \beta x_i + \varepsilon_i$
- $\alpha$  is the intercept
- $\beta$  is the slope
- $\varepsilon$  is the error term
- The error term is usually taken to be normally distributed

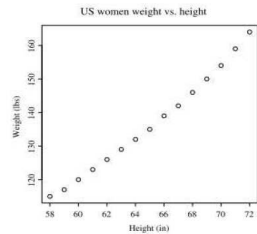


- Use some methods to estimate  $\alpha$  and  $\beta$

18

## Statistical Regression Example

- Prediction of future observations
- Data set describing the average heights and weights for American women aged 30-39
- Assume:  $Y = \eta(X) + \varepsilon$
- $Y = \beta_0 + \beta_1 X + \beta_2 X^3 + \varepsilon$
- Let  $\beta_0 = 1$
- Estimate  $\beta_1, \beta_2$



Height (in) 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72  
 Weight (lbs) 115 117 120 123 126 129 132 135 139 142 146 150 154 159 164

19

## 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 in a pre-specified model.
  - the model structure, with
  - the corresponding coefficients/parameters

20

## GP for Symbolic Regression

- Objective: Find a program that produces the correct value of  $x^6 - 2x^4 + x^2$  when given the value of  $x$
- Terminal Set:  $X$  and a random number  $R$  in  $[-1.0, 1.0]$
- Function Set:  $\{+, -, *, \%, \}$
- Fitness Cases: 50 random  $x$  values in  $[-1.0, 1.0]$
- 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 for each of the 50 points is less than 0.01
- Termination criteria: satisfactory solutions found, or at generation 51.

21

## Symbolic Regression Example

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

22

## GP for Symbolic Regression: 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!

23

## GP for Regression 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
- Electronic circuit design (GP IV, Koza)

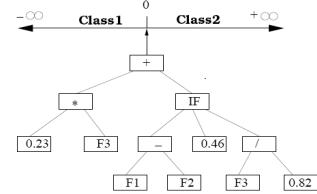
24

## GP for Binary Classification

- Terminal set: features, constants
- Function set: standard + specific
- Fitness function: classification accuracy or error rate on the training example
- Program class translation rule: how to translate/convert the single program output to one of class labels

25

## Tree-based GP for Classification

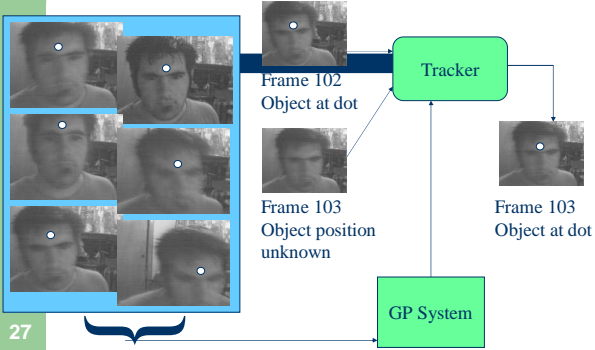


Genetic Program: (+ (\* 0.23 F3)  
(IF (- F1 F2) 0.46 (/ F3 0.82))  
)

`if ProgOut < 0 then Class1 else Class2;`

26

## Example: Object Tracking Task



27

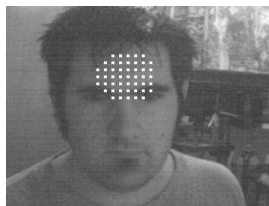
## Approach

- Use GP to track an object in low-quality webcam footage, at a real-time speed.
- Test the GP method on two object tracking problems of varying difficulty.

28

## Training

- Specify target object position
- Evaluate tracker program at a set of training points around target producing refined estimates.
- Fitness of program = avg. distance from target



29

## Tracking the left eye



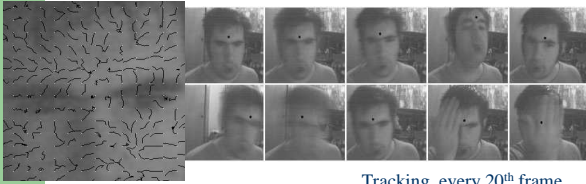
Trails of tracker convergence

Tracking, every 20<sup>th</sup> frame

- Tracks well, even when the face was quite blurry due to fast movement.

30

## Tracking the head



Tracking, every 20<sup>th</sup> frame

Trails of tracker convergence

- Tracks well, even when the face was quite blurry due to fast movement and when the head looks up.

31

## GP for English Stress Detection

- English becomes more and more important as a communication tool in the world.
- Provide P2P training to ESL students is very expensive. Therefore, software is desirable.
- Correct *rhythmic* stress in ESL students' speech is a key point to make the speech sound like native. Therefore, to accurately detect rhythmic stress in spoken English becomes an important functionality in this kind of software.

32

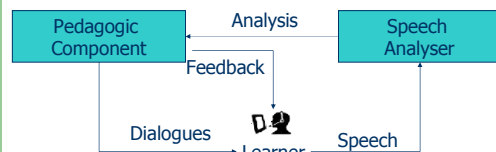
## Known stress classifiers

- Bayesian classifier
- Support vector machine classifier
- Decision tree classifier
- Neural networks classifier

The best accuracy is around 85%. It is not high enough for a commercial use.

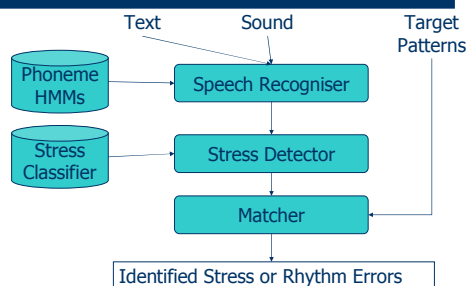
33

## Overview of the whole project



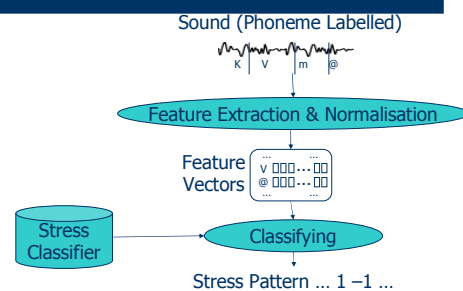
34

## The Speech Analyser



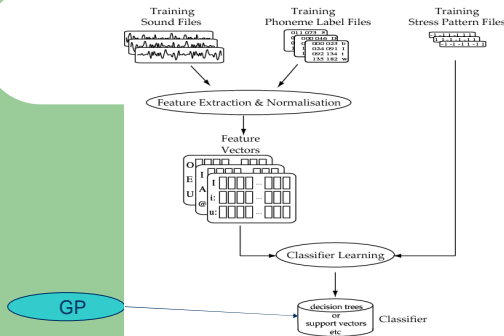
35

## The Stress Detector



36

## Classifier Learning Procedure



## Experiments

- Data set: 703 vowels in 60 utterances of ten distinct sentences produced by 6 female speakers – 340 stressed and 363 unstressed
- Scaled feature values in the range [-1,1] are also used.
- Three experiments are conducted on the three terminal sets respectively.
- 10 times 10-fold cross validation for training and testing
- Comparing with
  - DT -- C4.5
  - SVM -- LibSVM (with Radial Basis Function kernel and C = 1)
  - GP: Discipulus

38

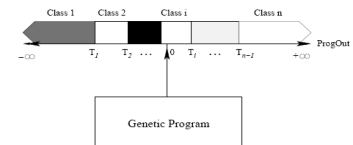
## Detection Accuracy (%)

Terminal Set I (prosodic features)			
	GP	DT	SVM
Unscaled	91.9	80.4	79.7
Scaled	91.6	80.6	83.2
Terminal Set II (vowel quality features)			
	GP	DT	SVM
Unscaled	85.4	79.7	79.1
Scaled	84.6	78.9	80.5
Terminal Set III (combination)			
	GP	DT	SVM
Unscaled	92.0	79.9	81.3
Scaled	92.6	80.1	82.0

39

## GP for Multi-class Classification

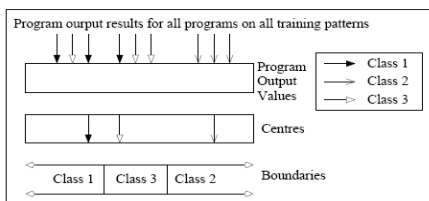
- **Classification map: Static method**
- Boundaries are **fixed**
- These boundaries are **predefined**
- A class is determined from the **fixed regions**
- Classes are in a **fixed order**



40

## GP for Multi-class Classification

- **Dynamic methods:**



41

## Linear GP for Multi-class Classification

```

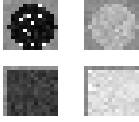
r[0] = 0.453 - cf[1];
r[1] = r[0] * 0.9;
if (cf[6] < cf[1])
    r[2] = 0.453 - cf[3];
r[3] = cf[4] - 0.87;
// instrs, src regs, des regs, ops
    
```

- We use **multiple destination registers** each corresponding to one class.
- The winner-takes-all strategy is used for classification: the class represented by the register with the largest value is considered the class of the input object.

42

## Experiment Design

- Data sets: (Shape: 600 objects; digit15 and digit30: 1000)



4 5  
7 8

- Terminal set: 8 features (shape), and 49 pixels (digits)
- Function set: {+, -, x, /, if}
- TGP length heuristic based on LGP
- Repeat 50 runs and mean/standard deviation of the results are reported

43

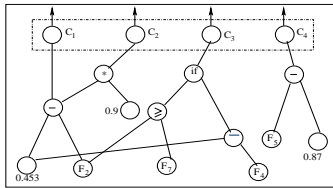
## Classification Results

Data set	Method	Training Accuracy % ( $\mu \pm \sigma$ )	Test Accuracy % ( $\mu \pm \sigma$ )
shape	LGP	100.00 $\pm$ 0.00	99.91 $\pm$ 0.17
	TGP	85.04 $\pm$ 16.49	84.41 $\pm$ 17.17
digit15	LGP	68.62 $\pm$ 4.67	65.78 $\pm$ 5.25
	TGP	52.60 $\pm$ 6.65	51.80 $\pm$ 6.85
digit30	LGP	55.22 $\pm$ 3.49	51.04 $\pm$ 4.26
	TGP	41.15 $\pm$ 5.03	35.00 $\pm$ 6.17

44

## Program Comprehensibility

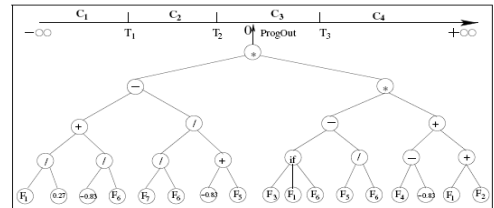
```
//x[1] = x[1] / x[1];
//x[3] = cf[0] + cf[5];
//if (x[3] < 0.86539)
//x[3] = x[3] - x[1];
x[0] = 0.453 - cf[1];
//x[3] = x[2] * cf[5];
x[1] = x[0] * 0.9;
if (cf[6] < cf[1])
x[2] = 0.453 - cf[3];
x[3] = cf[4] - 0.87;
```



45

## Program Comprehensibility

```
(* (- (+ (/ E1 -0.268213) (/ -0.828695 E6))
(/ / E7 E6) (+ -0.828695 E5)))
(* (- (if E3 E1 E5) (/ E5 E6)) //TGP program example)
(+ (- E4 -0.828695) (+ E1 E2)))
```



46

## Major Challenges

- Program structures and representations
- Operators and search techniques
- High dimensional data
- Unbalanced data
- Conflict objectives
- Computation cost
- Understanding of evolved programs

47

## Bibliography

- Mengjie Zhang and Victor Ciesielski. Genetic programming for multiple class object detection. In Norman Foo, editor, *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence*, volume 1747, Lecture Notes in Artificial Intelligence, pages 180-191. Springer, Heidelberg, Dec 1999.
- Thomas Loveard and Vic Ciesielski. Representing classification problems in genetic programming. In Jong-Hwan Kim, editor, *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 1070-1077. Seoul, South Korea, May 2001. IEEE.
- Mengjie Zhang, Stefano Cagnoni, Gustavo Olague. "GECCO 2009 Tutorial: Evolutionary Computer Vision". *Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*. ACM Press, 2009. pp. 3355-3380.
- Mengjie Zhang, Mark Johnston. "A Variant Program Structure in Tree-Based Genetic Programming for Multiclass Object Classification". Book chapter of *Evolutionary Image Analysis and Signal Processing, Studies in Computational Intelligence, Vol. 213*. Chapter 3. Springer, 2009. pp.55-72.
- Mengjie Zhang, Will Smart. "Multiple Object Classification Using Genetic Programming". *Evolutionary Computation in Image Analysis and Signal Processing, Lecture Notes in Computer Science, Vol. 3005*, 2004. pp. 367-376.

48

## Bibliography

- Mengjie Zhang, Christo Fogelburg, Yuejin Ma. "A Linear Structured Approach and A Refined Fitness Function in Genetic Programming for Multi-class Object Classification". *Connection Science*. Vol. 19. No. 4, 2007. pp. 339-359.
- Will Smart, Mengjie Zhang. "Tracking Object Positions in Real-time Video using Genetic Programming". In *Proceeding of Image and Vision Computing International Conference*, 2004. pp. 113-118.
- Huayang Xie, Mengjie Zhang, Peter Andrae. "Genetic Programming for Automatic Stress Detection in Spoken English". Proceedings of EvoWorkshops 2006 (EvoASP 2006). *Lecture Notes in Computer Science*, Vol. 3907. Springer. 2006. pp.460-471.
- Peter Andrae, Huayang Xie, Mengjie Zhang. "Genetic Programming for Detecting Rhythmic Stress in Spoken English". *International Journal of Knowledge-Based and Intelligent Engineering Systems (KES Journal)*. Special Issue on Genetic Programming. Vol. 12, No. 1, 2008. pp. 15-28.

49

## Upcoming Conferences/Workshops

- Special Session on Evolutionary Computer Vision, CEC 2010: IEEE Congress on Evolutionary Computation
  - Organisers: Victor Ciesielski, Mario Koeppen, Mengjie Zhang
  - July 18-23, 2010, Barcelona
  - Paper Submission deadline: 31 Jan 2010
- Genetic and Evolutionary Computation Conference (GECCO 2010)
  - Time/Venue: Portland, 7-11 July 2010
  - Paper Submission deadline: 13 Jan 2010

50

## Acknowledgements

- University Research Fund under the number of URF09 2399/85608/85808 at Victoria University of Wellington, New Zealand
- Marsden Fund council from the government funding (VUW0806), administrated by the Royal Society of New Zealand.
- Research students Jason Xie, Will Smart, Christo Fogelburg at Victoria University of Wellington, New Zealand
- ACISS 2009 organisers

51