


Machine Learning

Genetic Programming



"DNA is like a computer program, but far, far more advanced than any software every created."

- Bill Gates

What is Genetic Programming?

Matt Johnson, Ph.D.

Genetic Programming

Genetic Programming (GP) applies the approach of the genetic algorithm to the search space of possible computer programs.

The chromosomes of GAs become programs in GPs.

A wide variety of seemingly different problems from many different fields can be reformulated as a search for a computer program to solve the problem.

Matt Johnson, Ph.D.

Development of GP

- 1950, Alan Turing made first proposal to evolve computer programs.
- 1981, Richard Forsyth demonstrated the successful evolution of small programs, represented as trees, to perform classification of crime scene evidence for the UK Home Office.
- 1985, Michael Cramer published evolved programs in two specially designed languages, which included the first statement of modern "tree-based" GPs.

Matt Johnson, Ph.D.

4

Development of GP (2)

- 1992, John R. Koza at Stanford University published his treatise "Genetic Programming. On the Programming of Computers by Means of Natural Selection." That same year Koza begins a series of four books with accompanying videos that established the field of GP.
- David Goldberg, a student of John Holland's, coined the name "Genetic Programming".

Matt Johnson, Ph.D.

5

Tree Structures

Genetic Programming operates upon a population of **tree structures** (or graphs).

Genetic programming typically uses functional languages such as Lisp which are naturally tree-structured.

Matt Johnson, Ph.D.

6

LISP

- Lisp stands for **LIS**t **P**rocessor.
- All programs and data in LISP are in the form of fully parenthesized lists called symbolic expressions (or **S-expressions**) which can be of any length and have a nested structure.
- Lisp uses **prefix** notation:
 - The first item in the list is the function name.
 - The remaining members of the list are its arguments.

Matt Johnson, Ph.D.

7

Lisp S-Expressions

The first elements of each (sub)list can be seen as its root, the rest of the (sub)list as its leafs.

Examples:

- $(+ 4 2 3)$
- $(* (\min 5 2 3) (/ 6 (- 4 2)))$
- $(\cos x)$
- $(\text{defun power-j } (x y)$
 $(\text{if } (> y 1) (* x (\text{power-j } x (- y 1))) x))$

Matt Johnson, Ph.D.

8

Other Forms as Trees

Trees are a universal structure and, with care, other forms can be represented as trees:

Consider the following:

- $(2\pi + (x + 3) - (y / (5 + 1)))$
- $(x \wedge \text{true}) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y)))$
- ```
i = 1;
while (i < 20)
{
 i = i + 1
}
```

Matt Johnson, Ph.D.

9

---

---

---

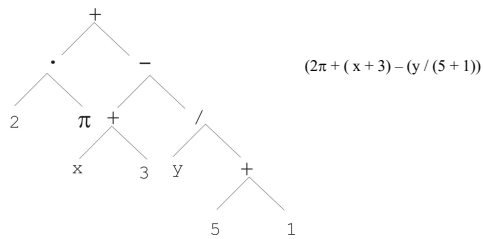
---

---

---

---

## Arithmetic Formula as Tree



Matt Johnson, Ph.D.

10

---

---

---

---

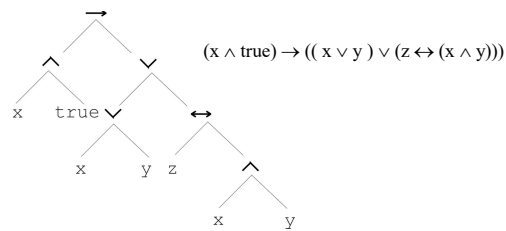
---

---

---

---

## Logical Formula as Tree



Matt Johnson, Ph.D.

11

---

---

---

---

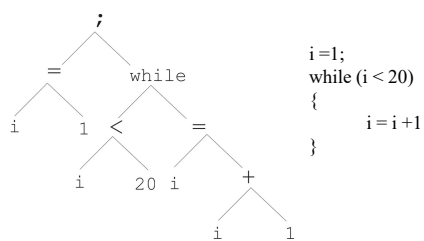
---

---

---

---

## Computer Program as Tree



Matt Johnson, Ph.D.

12

---

---

---

---

---

---

---

---

## GP Overview

|                     |                          |
|---------------------|--------------------------|
| Representation:     | Tree structures          |
| Recombination:      | Exchange of subtrees     |
| Mutation:           | Random change in tree    |
| Parent selection:   | Fitness proportional     |
| Survivor selection: | Generational replacement |

Matt Johnson, Ph.D.

13

“Who do you think made the first stone spears? The Asperger guy. If you were to get rid of all of the autism genetics, there would be no more Silicon Valley.”

- Temple Grandin

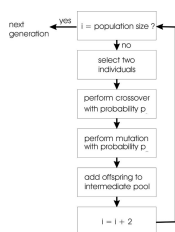
## GP Operators

Matt Johnson, Ph.D.

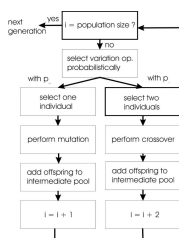
14

## GA vs. GP Process

### GA Flowchart



### GP Flowchart



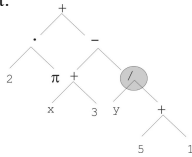
Matt Johnson, Ph.D.

15

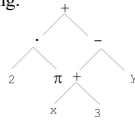
## Mutation

The most common mutation operator is to replace a randomly chosen subtree by randomly generated tree.

Parent:



Offspring:



Matt Johnson, Ph.D.

16

---

---

---

---

---

---

---

---

## Mutation (2)

Mutation has two parameters:

- Probability  $p_m$  to choose mutation vs. recombination.
- Probability to choose an internal point as the root of the subtree to be replaced.

Remarkably,  $p_m$  is advised to be 0 (Koza'92) or very small, like 0.05 (Banzhaf et al. '98)

Note: the size of the offspring can exceed the size of the parent.

Matt Johnson, Ph.D.

17

---

---

---

---

---

---

---

---

## Recombination

The most common recombination operator is to exchange two randomly chosen subtrees among the parents.

Recombination has two parameters:

- Probability  $p_c$  to choose recombination vs. mutation.
- Probability to choose an internal point within each parent as the crossover point.

Note: The size of the offspring can exceed that of the parents.

Matt Johnson, Ph.D.

18

---

---

---

---

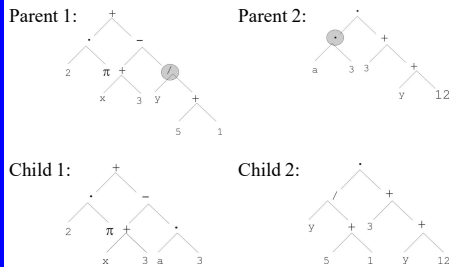
---

---

---

---

## Recombination (2)



## Selection

**Over-selection** is often used in very large populations:

- rank population by fitness and divide it into two groups: group 1 is the best  $x\%$  of population, group 2 is the other  $(100-x)\%$ .
- 80% of selection operations choose from group 1, only 20% from group 2.
- for pop. size = 1000, 2000, 4000, 8000,  $x = 32\%$ , 16%, 8%, and 4% respectively.

Motivation: To increase efficiency. The percentages come from rule of thumb experience.

## Initializing Population

How do you create random programs?

- Define a function set **F**.
- Define a terminal set **T**.

An example:

- $F = \{+, -, /, *, \%\}$
- $T = \{X, Y, \text{Random\_Constants}\}$

Random\_Constants would be further defined in terms of type and range.

## Initializing Population (2)

A maximum initial depth of trees  $D_{\max}$  is set.

There are two common methods of tree creation:

- Full method
- Grow method

A Common GP initialisation: ramped half-and-half, where both Grow and Full method each deliver half of the initial population.

Matt Johnson, Ph.D.

22

---

---

---

---

---

---

---

## Initializing Population (3)

Full method:

- Each branch has depth =  $D_{\max}$ .
- Nodes at depth  $d < D_{\max}$  randomly chosen from function set  $F$ .
- Nodes at depth  $d = D_{\max}$  randomly chosen from terminal set  $T$ .

Matt Johnson, Ph.D.

23

---

---

---

---

---

---

---

## Initializing Population (4)

Grow method:

- Each branch has depth  $\leq D_{\max}$ .
- Nodes at depth  $d < D_{\max}$  randomly chosen from  $F \cup T$ .
- Nodes at depth  $d = D_{\max}$  randomly chosen from terminal set  $T$ .

Matt Johnson, Ph.D.

24

---

---

---

---

---

---

---



## Code Bloat

Code Bloat is the “survival of the fittest”; that is, the tree sizes in the population keep increasing over time.

This is a big issue in genetic programming. There is ongoing research and debate about appropriate countermeasures. These include:

- Prohibiting variation operators that would deliver offspring that are too big.
- Parsimony pressure: including a fitness penalty for programs that are oversized.

Matt Johnson, Ph.D.

25

---

---

---

---

---

---

---

## Other Issues

- Trees for data fitting are not always the same as programs that are truly executable.
- Execution can change the environment and therefore the calculation of the fitness. For example, a robot controller.
- Fitness calculations are done mostly by simulation, and are super time intensive.

BUT genetic programming has a wide range of applications!!

Matt Johnson, Ph.D.

26

---

---

---

---

---

---

---

“Your genetics load the gun. Your lifestyle pulls the trigger.”  
- Mehmet Oz

## GP Examples

Matt Johnson, Ph.D.

27

---

---

---

---

---

---

---

## Symbolic Regression

**Symbolic Regression (SR)** searches the space of mathematical expressions to find a model that best fits a given dataset, both in terms of accuracy and simplicity.

In other words, SR learns a function that fits some data.

Matt Johnson, Ph.D.

28

---

---

---

---

---

---

---

## Symbolic Regression (2)

|   |               |                                                                                                                                                                                                   |
|---|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Objective:    | Find a computer program with one input (independent variable $x$ ) whose output equals the given data                                                                                             |
| 1 | Terminal set: | $T = \{X, \text{Random-Constants}\}$                                                                                                                                                              |
| 2 | Function set: | $F = \{+, -, *, \div\}$                                                                                                                                                                           |
| 3 | Fitness:      | The sum of the absolute value of the differences between the candidate program's output and the given data (computed over numerous values of the independent variable $x$ from $-1.0$ to $+1.0$ ) |
| 4 | Parameters:   | Population size $M = 4$                                                                                                                                                                           |
| 5 | Termination:  | An individual emerges whose sum of absolute errors is less than $0.1$                                                                                                                             |

---

---

---

---

---

---

---

## Symbolic Regression (3)

$x^2 + x + 1$ :

| Independent Variable $X$ | Dependent Variable $Y$ |
|--------------------------|------------------------|
| -1.00                    | 1.00                   |
| -0.80                    | 0.84                   |
| -0.60                    | 0.76                   |
| -0.40                    | 0.76                   |
| -0.20                    | 0.84                   |
| 0.00                     | 1.00                   |
| 0.20                     | 1.24                   |
| 0.40                     | 1.56                   |
| 0.60                     | 1.96                   |
| 0.80                     | 2.44                   |
| 1.00                     | 3.00                   |

Matt Johnson, Ph.D.

30

---

---

---

---

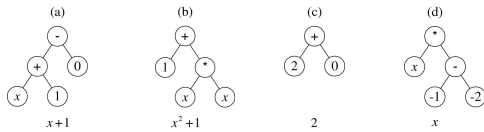
---

---

---

## Symbolic Regression (4)

A population of four randomly created individuals is created for generation 0:

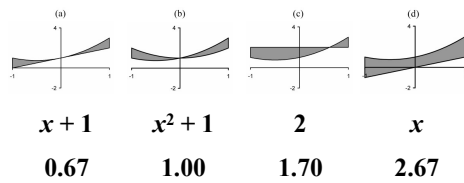


Matt Johnson, Ph.D.

31

## Symbolic Regression (5)

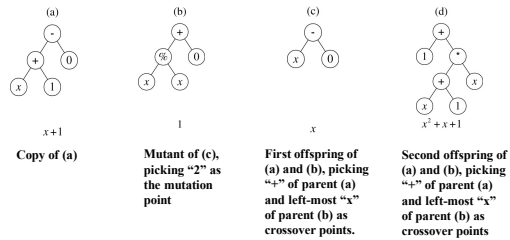
The fitness of the four individuals is calculated:



Matt Johnson, Ph.D.

32

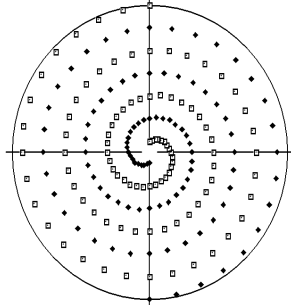
## Symbolic Regression (6)



Matt Johnson, Ph.D.

33

## Classification: Intertwined Spirals



Matt Johnson, Ph.D.

34

---

---

---

---

---

---

---

## Classification: Intertwined Spirals (2)

|   |               |                                                                                                                 |
|---|---------------|-----------------------------------------------------------------------------------------------------------------|
|   | Objective:    | Create a program to classify a given point in the $x$ - $y$ plane to the red (square) or blue (diamond) spiral. |
| 1 | Terminal set: | $T = \{X, Y, \text{Random-Constants}\}$                                                                         |
| 2 | Function set: | $F = \{+, -, *, \%, \text{IFLTE}, \text{SIN}, \text{COS}\}$                                                     |
| 3 | Fitness:      | The number of correctly classified points (0 – 194)                                                             |
| 4 | Parameters:   | $M = 10,000$ . $G = 51$                                                                                         |
| 5 | Termination:  | An individual program scores 194                                                                                |

---

---

---

---

---

---

---