Genetic Programming

Charles Chilaka

Department of Computational Science

Memorial University of Newfoundland

Class Project for Bio 4241
March 27, 2014

Outline

1. Historical perspective, definitions and examples.
2. Genetic Algorithms and its mathematical formulation
   - Mathematical formulation of the Schemata equation
3. Genetic Programming
4. Application of Genetic programming in Biological Research
   - Limitations of Genetic Programming

Historical perspective, definitions and examples

Basic elements of genetic programming *GP* introduced by Stanford University Professor John R. Coza in 1987.

It is a domain dependent method that genetically breeds population of computer programs to solve problems.

Basically applies the principle of Darwinian evolution to breed a new (and often improved) population of programs.

It is a step further in Genetic Algorithms *GA*.

Prof. Wolfgang Banzhaf (our own) is a leading light in this area.

The development history:
EC(Rechenberg, 1960)= GP(Koza, 1987/1992)+ *ES (Rechenberg,1965)*+ EP (Fogel, 1962) + *GA(Holland, 1970)*

Historical perspective, definitions and examples

The mathematical foundation of genetic algorithms and genetic programming is the schemata theory proposed by J.H. Holland in 1973. It is a statement about the propagation of schemata (or building blocks) within all individuals of one generation.

Definition

A *Schemata* are sets of strings that have one or more features in common. It consists of bit strings (0,1) and a "don't care" symbol $\sharp$.
Every schema matches exactly 2^r strings, where r is the number of don't care symbols in the schema template. A schema is a generalization of (parts of) an individual.

Example

The set of the schema $\sharp 1101 \sharp 0$ are $1110110, 1110100, 0110110, 0110100$.

Definition

The length of a schema $\alpha(S)$ is the distance from the first to the last fixed symbol(1 or 0 but not $\sharp$).

Example

The individuals 10010010010000011110100010 1 and 01001001010000011111101010 1 of length 25 can be summarized by the schema $\sharp\sharp0\sharp\sharp0\sharp\sharp0100000111\sharp10\sharp0101$
where all identical positions are retained and differing positions marked with a $\sharp$.

Definition

The order of a schema $o(S)$ is the number of fixed positions (1 or 0 but not $\sharp$). The above example has order equals 19.

Basic facts from Biology

- All living organisms consist of cells containing the same set of one or more chromosomes i.e strings of DNA.
- A gene seen as an "encoder" of a characteristic, such as eye color.
- The different possibilities for a characteristic (eg, brown, green, blue, gray) are called alleles.
- Each gene is located at a particular position (locus) on the chromosome.
- Most organisms have multiple chromosomes in each cell.
- The sum of all chromosomes i.e the complete collection of genetic material is called the genome of the organism.

Basic facts from Biology cont'd

- Genotype refers to the particular set of genes contained in a genome
- Organisms whose chromosome are arranged in pairs are called diploid, else haploid.
- Most sexually producing species are haploid since after meiosis, the number of chromosomes in gametes are halved.
- For reproduction, the genes of the parents are combined to form a new diploid set of chromosomes
- Offsprings are subject to mutation where elementary parts of the DNA are changed.
- The fitness of an organism is defined as its probability to reproduce or as a function of the number of offspring the organism has produced.

Extracts from Biology to Genetic Algorithm

- Chromosome refers to a solution candidate.
- Genes are single bits or small blocks of neighboring bits that encode a particular element of the solution.
- Alleles are values 0, 1 and ♯.
- Crossover operates by exchanging genetic material between two haploid parents.
- Mutation is implemented by flipping the bit at a randomly chosen locus.

Remark

Most applications of GA employ haploid single chromosome individuals. The major genetic operators used in GA are parent selection, crossover, mutation and replacement.

Genetic Algorithms and its mathematical formulation

Definition

GA is a searching process based on the laws of natural selection. It consists of three operations: selection, genetic operation(crossover and mutation) and replacement.

Remark

The population comprises a group of chromosomes from which candidates can be selected randomly for the solution of a problem.

Remark

The fitness values of all the chromosomes are evaluated by calculating the objective function in a decoded form(phenotype).

Fitness techniques

Two broadbased methods of fitness techniques:

1. Windowing:

$$f_i = c \pm (v_i - v_m) \tag{1}$$

where v_i and v_m are the objective values of chromosome i and worst chromosome m respectively.

2. Linear normalization

$$f_i = f_{best} - (i - 1) \times d \tag{2}$$

where d is the decrement rate.

Top level description of a Genetic Algorithm.

A simple Genetic Algorithm follows these steps:

1. Randomly generate an initial population $X(0) := (x_1, x_2, ..., x_N)$
2. Compute the fitness $F(x_i)$ of each chromosome x_i in the current population.
3. Create new chromosomes $X_r(t)$ by mating current chromosomes, applying mutation and recombination as the parent chromosomes mate.
4. Delete numbers of the population to make room for the new chromosomes.
5. Compute the fitness of $X_r(t)$ and insert this into population
6. $t := t + 1$, if not (end-test) go to step 3, or else stop and return the best chromosome.

Mathematical formulation of the schemata equation

Remark

The mathematics of the schemata theory done by using the effects of selection and genetic operation(crossover and mutation).

Remark

Since a schema represents a set of strings, associate a fitness value f(S,t) with schema S, and the average fitness of the schema. f(S,t) is determined by all the matched strings in the population.

Remark

Usage of proportional extimation in the reproduction phase enables us to extimate the number of matched strings of a schema S in the next generation.

Effect of selection

Let M(S,t) be the number of occurrences of a particular schema S in a population of n individuals at time t.
The bit string A_i of individual i gets selected for reproduction with the probability

$$P_i = \frac{f(S,t)}{F(t)} \tag{3}$$

where $F(t)$ is the average fitness value of the current generation.
The expected number of occurrences of S in the next generation is

$$M(S,t+1) = M(S,t) \times \frac{f(S,t)}{F(t)}. \tag{4}$$

Effect of selection cont'd

By change of variable

$$\epsilon = \frac{f(S,t) - F(t)}{F(t)} \tag{5}$$

and substituting equation (5) into equation (4), we have

$$M(S,t) = M(S,0)(1+\epsilon)^t. \tag{6}$$

Equation (6) shows that an "above average" schema receives an exponential increasing number of strings in the next generations.

Effect of crossover and mutation

The effect of crossover and mutation can be incorporated in the formulation of the schemata equation.

Remark

During evolution of a genetic algorithm, interferences come from genetic operations which affect the current schemata.

Definition

The defining length of a schema $\alpha(S)$ is the distance between the outermost fixed positions.

Example

The defining lengths of $\sharp 000\sharp$ and $1\sharp 00\sharp$ are 2 and 3 respectively.

Effect of crossover and mutation cont'd

Assuming the length of the chromosome is L and a point crossover is applied.
In general, a crossover point is selected uniformly among L-1 possible positions.
The probability of destruction of a schema S is given by

$$P_d(S) = \frac{\alpha(S)}{L-1}. \tag{7}$$

The probability of a schema survival is

$$\begin{aligned} P_s(S) &= 1 - P_d(S) \\ &= 1 - \frac{\alpha(S)}{L-1}. \end{aligned} \tag{8}$$

Assuming an operation rate of crossover of P_c, the new probability of survival becomes

$$P_s(S) \geq 1 - P_c \frac{\alpha(S)}{L-1} \tag{9}$$

The probability effect of selection(reproduction) and crossover been independent events is the product of the respective probabilities.
Hence the equation for the expected occurence of a schema S at time t+1 is

$$M(S, t+1) = M(S, t) \times \frac{f(S, t)}{F(t)} \times (1 - P_c \frac{\alpha(S)}{L-1}) \tag{10}$$

Equation (10) tells us the rate of increase of the schemata over time is proportional to their relative fitness and inversely proportional to their length.

Effect of mutation

Mutation can affect a schema S at each of it o(S) positions with mutation probability P_m. The probability of survival of a single constant position in a schema is

$$P_s(S) = 1 - P_m. \tag{11}$$

The probability of survival of the entire schema is

$$P_s(S) = (1 - P_m)^{o(S)}. \tag{12}$$

For small P_m, equation (12) can be approximated by

$$P_s \approx 1 - o(S) \times P_m. \tag{13}$$

Total effects

By combining equation (10) and (13), we have the formula for the expected count of a schema:

$$M(S, t+1) = M(S, t) \times \frac{f(S, t)}{F(t)} \times (1 - P_c \frac{\alpha(S)}{L-1} - o(S) \times P_m). \quad (14)$$

Recall the schema outperformance factor ϵ i.e equation (5), equation (14) can be rewritten as

$$M(S, t+1) = M(S, t) \times (1+\epsilon) \times (1 - P_c \frac{\alpha(S)}{L-1} - o(S) \times P_m). \quad (15)$$

Equation (15) is of the form

$$M_t = M_0 \times (1+\epsilon)^t \times f(P_c, P_m, L, \alpha(S)). \quad (16)$$

- Equation (16) says that the number of schemata better than average will exponentially increase over time.
- Basic rationale behind Genetic Algorithms
- If the linear representation of a problem allows the formation of a schemata, then the genetic algorithm can efficiently produce individuals that continuously improve in terms of fitness function.

Genetic Programming

Remark

Genetic programming is an extension of Genetic algorithm.
The main concepts here are also those concepts from biology vis a vis selection, crossover and mutation, and replacement.
It is the main function to control the genetic algorithm.

Definition

Genetic programming is a machine learning algorithm. It uses a fixed representation for programs and creates a recombination operation that could combine programs to produce offspring programs.

Genetic Programming preparatory steps.

Two important steps towards solving a problem using Genetic programming.

- Define a set or series of functions.
  1. Some functions may return a variable or input.
  2. Other functions may perform an operation:
     - $+$, $-$, $>$, $<$, **if-then-else**, **do**, **for** are all functions
- Define the fitness of the program:
  1. How many events does it classify correctly?
  2. Does it provide the correct output for some cases?
  3. Does it fit the data?

Genetic programming process.

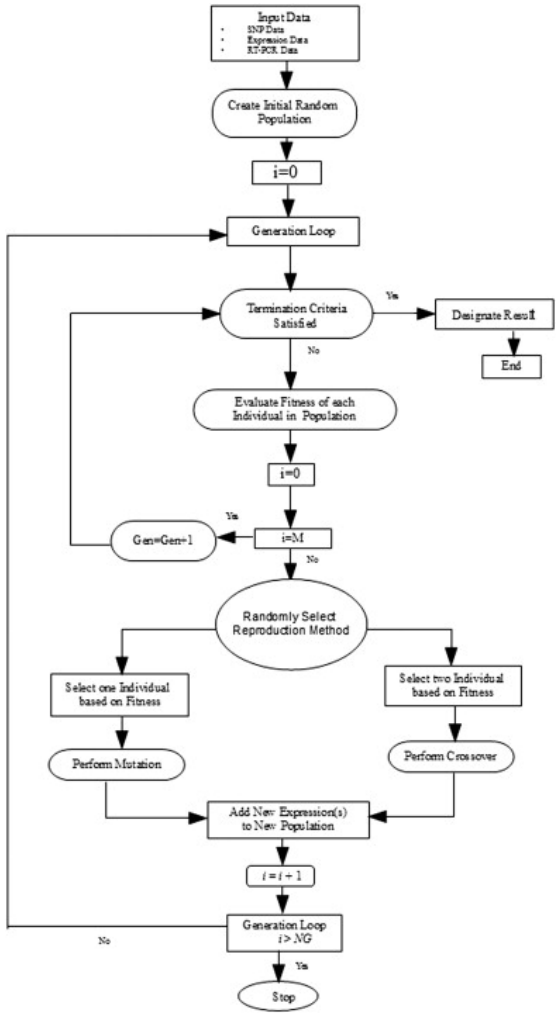


Figure: Schematics of a typical genetic programming process

Representation via Trees.

Genetic programming basics easier shown using the tree structure.

Definition

A tree is a connected graph that contains no circuits.

Example

Consider the python code

```
def main ():
sumadd =float(sumadd)
f = float(f)
g = float(g)
if f > g:
sumadd = f * f + g
else:
sumadd = g * g + f
main()
```

The tree Structure

Another example of tree representation:

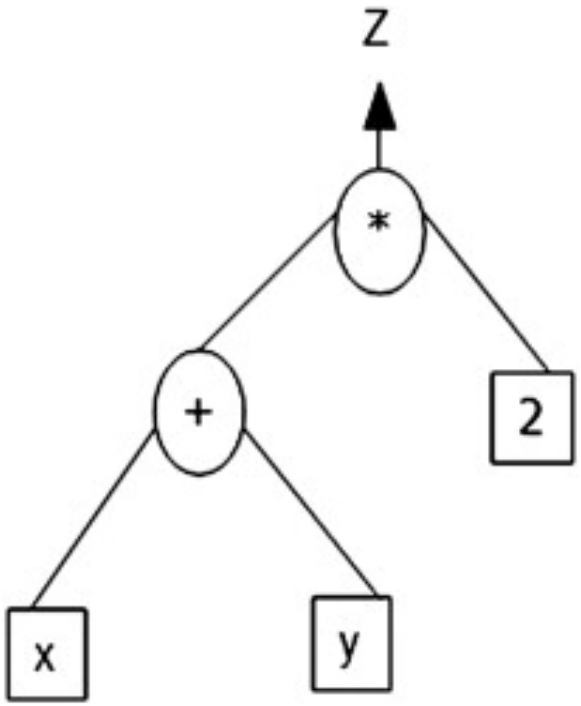


Figure: Schematics of a tree structure

Crossover schematics

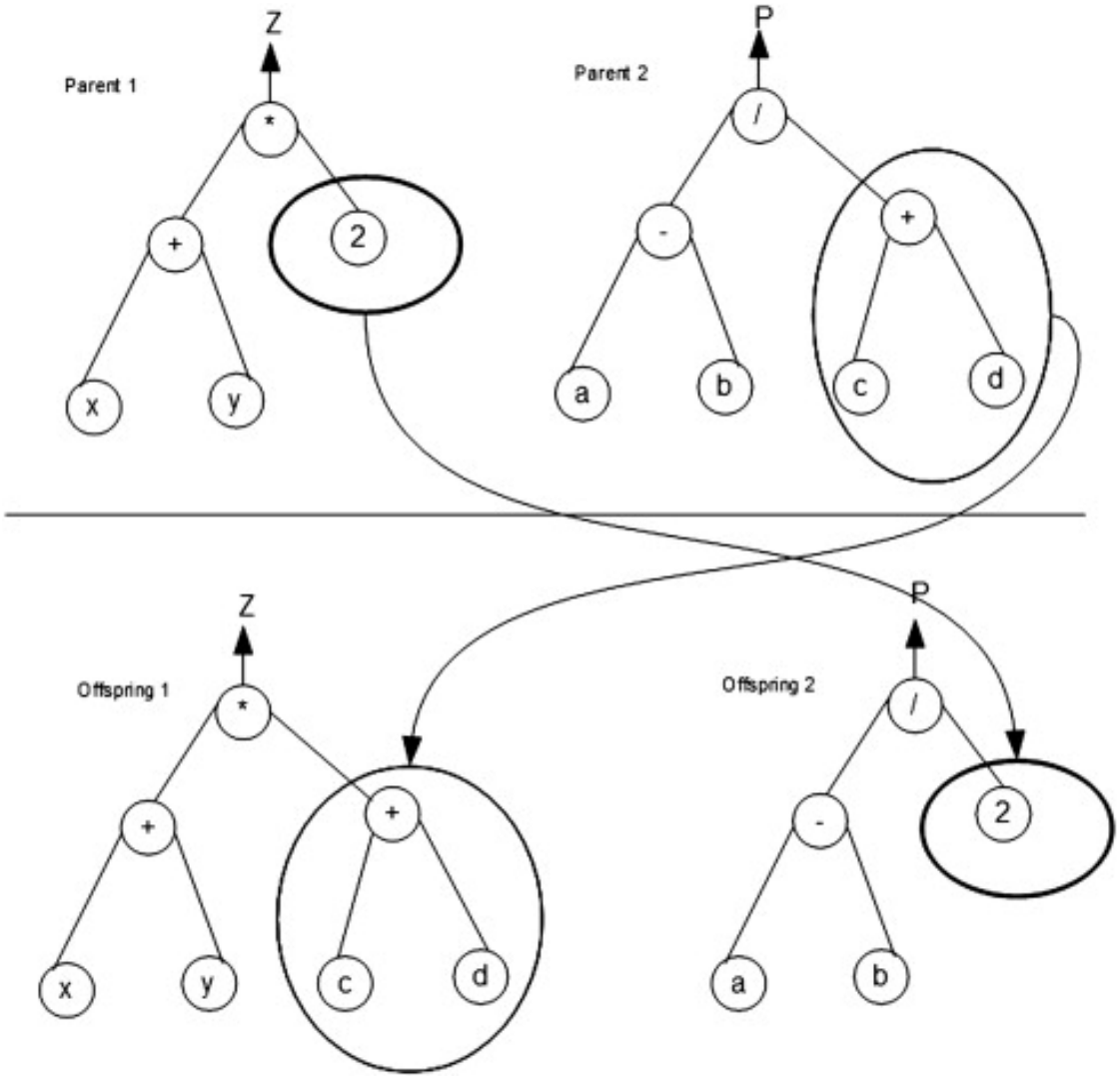
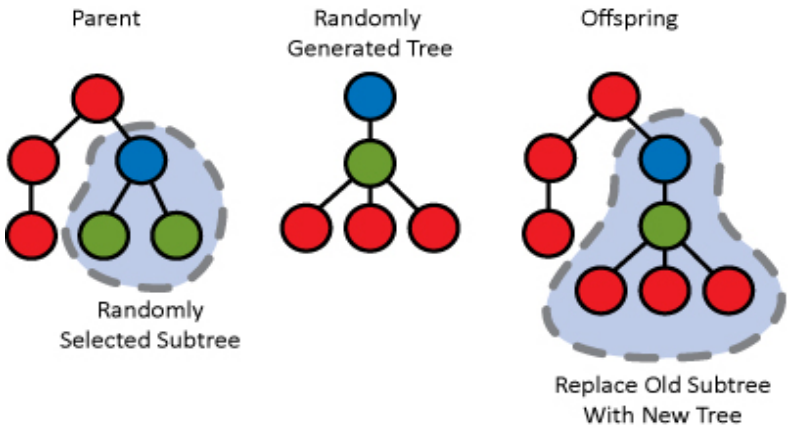
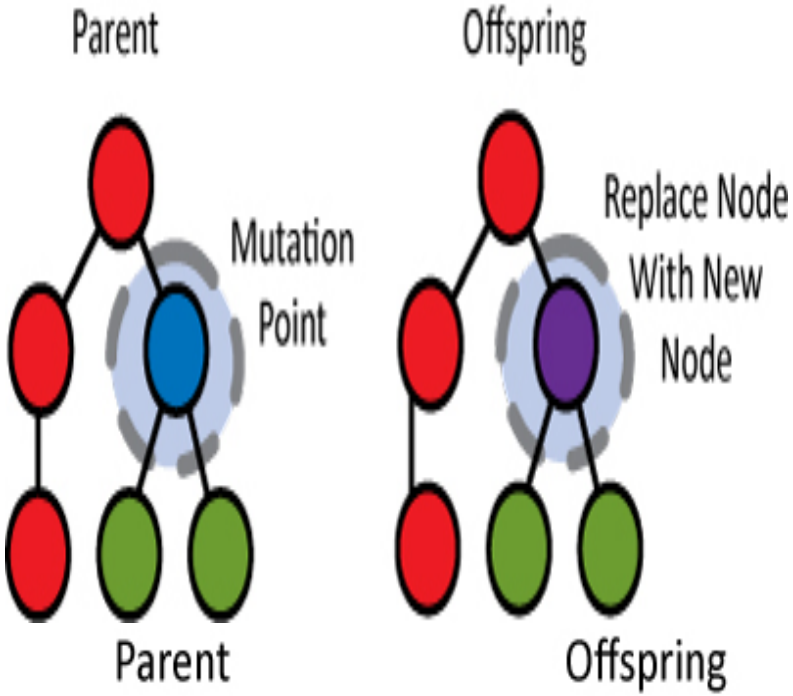
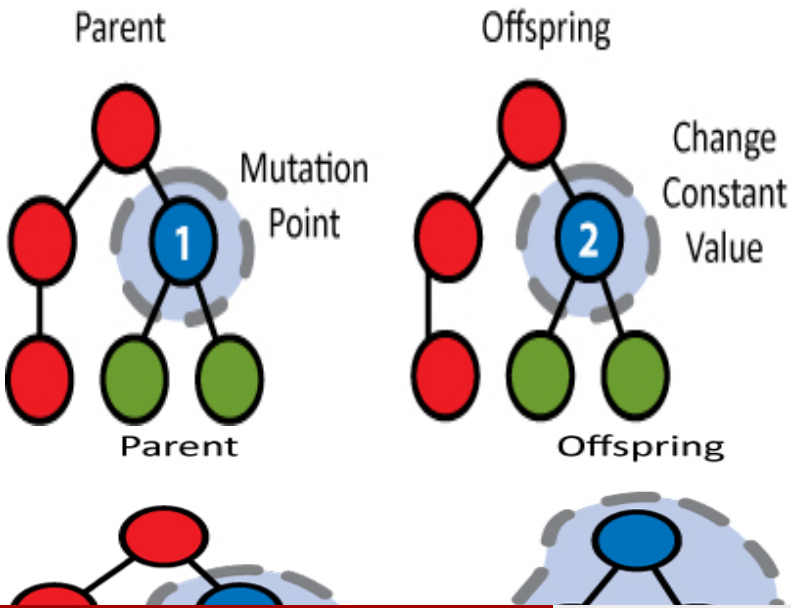


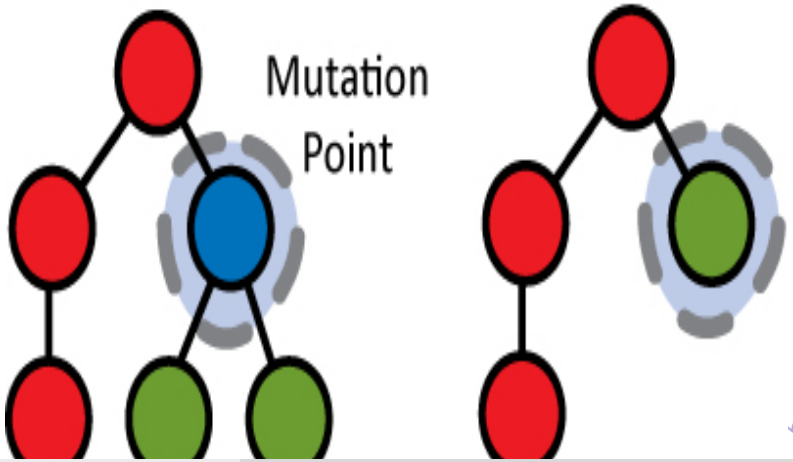
Figure: Schematics of crossover

Many mutation schematics









Genetic programming application in genomics

Genetic programming have been applied in the following areas of genomics

- Genetic network inference
- Gene expression data classification
- SNP analysis
- Epistasis analysis
- Gene annotation

Breast Cancer Research

The analysis of SNP's unveils the relationships between the genotypic and phenotypic information as well as the identification of SNP's related to a disease.

- Feature extraction is one of the most important research areas in pattern recognition challenges.
- Two major steps in feature extraction:
  - Extraction of relevant information from raw data to an original feature vector with m dimensions.
  - Creation of Extracted feature vector with n dimensions ($m > n$) from the parameter vector.
- Principal Component Analysis (PCA) and Fisher Linear Discriminant Analysis(FLDA) are the two major linear feature extractors.

Principal Component Analysis

- Good for feature extraction and dimensionality reduction in terms of linear transformation.
- The $m \times m$ covariance matrix is given by

$$S = \sum_{i=1}^{N} (x_i - \eta)(x_i - \eta)^T \tag{17}$$

where x_i is the ith observation, $1 \leq i \leq N$, N is the number of observations, and η is the global mean $\eta = \frac{\sum_{i=1}^{N} x_i}{N}$.

PCA Cont'd

- By solving the eigenvalue problem,

$$SW_i^T = \lambda_i W_i^T \quad 1 \leq i \leq m \tag{18}$$

where λ is a set of eigenvalues and W is a set of eigenvectors corresponding to the eigenvalues, the n components of a given observation vector is obtained.

- The n principal components of the projected feature vector Y are obtained by the largest n eigenvalues and corresponding eigenvectors.

Fisher Linear Discriminant Analysis

- This is the classical method for real valued feature extraction using a linear transformation.
- Maximizes the ratio of between-class scatter and within-class scatter of projected features.
- For k classes, the ith observation vector from class j is defined by x_{ij} where $1 \leq j \leq k$, $1 \leq i \leq N_j$, and N_j is the number of observations from class j.
- The within-class covariance matrix is defined by

$$S_W = \sum_{j=1}^{k} S_j \tag{19}$$

where

$$S_j = \sum_{i=1}^{N_j} (x_{ji} - \eta_j)(x_{ji} - \eta_j)^T \tag{20}$$

FLDA cont'd

- The between-class covariance matrix is defined by

$$S_B = \sum_{j=1}^{k} N_j(\eta_j - \eta)(\eta_j - \eta)^T \tag{21}$$

where η_j is the mean of class j and η is the global mean.

- The Fisher discrimant criterion is defined to maximize the objective function

$$J(\rho) = \frac{|W^T S_B W|}{|W^T S_W W|} \tag{22}$$

- The optimal transformation matrix W is obtained by

$$S_B W_i^T = \lambda_i S_W W_i^T, \quad 1 \leq i \leq m \tag{23}$$

It is worth the salt

- Usage in the analysis of large, complex data concerning cancer research and better clinical undestanding.
- Ability to find useful combination of features from a molecular data set in an unbiased manner.
- Production of realistic and understandable non linear models to describe diseases and their states.
- Creation of human readable results in contrast to the "black box" solution.
- Characterization of the diseases.

Limitations of Genetic Programming

- It is not an optimization algorithm : Finding the best solution to a problem
- Finding solutions that are overfit to data in consideration.
- Find the most parsimonious solution that meets our halting criteria.
- It is computationally intensive process that require tens or hundreds of thousands of programs to be tested.
- It should not be seen as an "all in all" solution that can be run on a desktop machine to analyse biological data.

Open questions and future research:

Beagle deck and Darwins insights have formed some strong basis for solving some intricate problems even in this modern days.

Open questions and future research:

Beagle deck and Darwins insights have formed some strong basis for solving some intricate problems even in this modern days.

- How far can Genetic programming go?

Open questions and future research:

Beagle deck and Darwins insights have formed some strong basis for solving some intricate problems even in this modern days.

- How far can Genetic programming go?
- Can we make it less expensive?

THANK YOU!...