GA Documentation

Mary Combs, Heejung Kim, Mohammad Soheilypour, Linqing(Waverly) Wei

December 15, 2017

1 Github

User Name: WaverlyWei
Link: https://github.com/WaverlyWei/GA

2 Contributions

Group Memebers: Mary Combs, Heejung Kim, Mohammad Soheilypour, Linqing(Waverly) Wei

Mary: Implemented selection function; Implemented doc strings in select and auxiliary functions; Wrote and compiled README and DESCRIPTION using roxygen2; Coded for Select function.

Heejung: Implemented initiation function; Designed and Implemented testing.

Mohammad: Implemented mutation function; Implemented doc strings in Select and auxiliary functions; Designed the structure of Select function; Wrote and compiled README and DESCRIPTION using roxygen2.

Waverly:Implemented crossover function; Wrote pdf documentation; Created Git Repo; Created initial R package; created GA.tar.gz

3 Approach

The solution is composed of two levels: Main function and Modular functions(initiation, selection, crossover, mutation)

4 Main function:Select

Input: User defined dataset(data.frame), Model(lm/glm), Convergence Criterion, Maximum number of steps
Output: Selected variabels(list)
Stopping Criteria: AIC value converges $\Delta\text{AIC}- > 0$

Select function first calls Initiation function to get starting data. The starting data are separated into two components: parents and interncept. Since converge happens aymptotically in this solution, the limiting steps of converge is preset within Select function. Then it continues to the next step: breeding the next generation. To generate the next generation, according to Genetic Algorithm, four modular functions are involved. Parents are passed into Selection function, ranked based on objective function(fitness) and parent matrix is updtaed for the next steps. Children are then passed into Crossover and Mutation function to be randomized. The outcome is then set to be the next generation of parents and will go into another round.

5 Modular Functions

5.1 Initiation

Input: The number of independent variables and population size for each generation
Output: Initial matrix of parents

Initiation function creates a subset of original dataset as the starting generation for Select function. The starting matrix passed onto Select function contains both the initial matrxi and intercepts.

5.2 Selection

Input: Model matrix object with intercept and column for each independent variable specified in model, formula object Parents matrix of P rows, Population size for each generation

Selection function assigns initital fitness probablity to each parent and then uses AIC as the objective function to rank parents. After sorting, fitness probablity is updated for each parent. Meanwhile, updated intercept and AIC value is also recorded together with selected parents.

5.3 Crossover

Input:P1, P2 (parent strings) and C as the number of variables
Output:P3,P4 (crossover strings, a list of two components)

Crossover function takes two parents strings and randomly choose a crossover site. Parents strings are cut at the site and ligated to produce children strings.

5.4 Mutation

Input:Parent, MutationProb, number of variables
Output: Mutated Parent

Mutation function takes a parent and mutates one or more sites according to the mutation probability.

6 Additional Notes

6.1 Fitness Probability

Assign fitness probablity to each parent based on the formula:

$$\phi({v_i}^{(t)}) = \frac{2r_i}{P(P+1)}$$

This assignment gives the best inidvdual a probability of $2/(P+1)$

7 Testing

Testing is performed on four modular functions in two aspects: (1) Valid input (2)Expected return results

7.1 Example 1: Simulated Data

1. Simulated Data Input:
Data <- matrix(rnorm(2500 , sd = c(1, 5, 7 , 100 , 40)) , ncol = 5 , byrow = TRUE)
Outcome <- 10 - 15 * initData[, 1] + 2 * initData[, 3] + 1.1 * initData[, 5]
Expected Output: c("(Intercept)","X1","X3","X5"))
Actual Output: c("X1","X2", "X3","X5")) or c("X1","X2", "X3","X5"))

7.2 Example 2: Whitewine Quality Data

2. Wine Data
Input: White Wine Quality Dataset
Output: (Intercept) X1 X2 X3 X4 X5 X6 X8 X9 X10 X11
AIC using variables from output is -5,992.7, and it is better than AIC of full model which is -5,983.
expected output: c(1,2,4,6)
actual output: c(1,6)
(1: intercept, 2: X1, 3:X2, etc.)