Function Documentation

Florian Papion
Qiuhan Wu
Brianna Noland

12/12/2014

Primary Function :

select Carries out variable selection optimization using auxiliary functions

---

Description

Parameters

1. *initialPopulationSize*: Integer, size of beginning population (initial number of models).
2. *numberGenerations*: Integer, number of iterations.
3. *regressionType*: String, type of regression: LM or GLM.
4. *criterion*: optional: String, defaults to AIC. If you would like to use a BIC, enter it as a parameter.
5. *mutationProbability*: Double, probability that any given chromosome will be mutated randomly.
6. *replacementPercentage*: Double, percent of parent population to replace with children.

Output Returns optimal model and its associated fitness score.

Examples

1. AICGAtest ← select(initialPopulationSize=25, numberGenerations=10, regressionType = ”LM”, replacementPercentage=.5, mutationProbability = .01)
2. BICGAtest ← select(initialPopulationSize=25, numberGenerations=30, regressionType = ”GLM”, criterion = ”BIC”, replacementPercentage=.6)

Auxiliary Function 2 :

readDataFile Reads in pre-cleaned datafile

Description

Parameters

1. *fileName*: String, name of file to be read in.
2. *sep*: Seperator type used in data file (i.e. ",", ":", " ", etc). Optional - defaults to ",".
3. *dependentVariable*: String, variable from data file on which to regress other variables.

Output Returns data loaded to variable "dataFile"

Examples

1. dataFile ← readDataFile(fileName, sep=":", dependentVariable)
2. dataFile ← readDataFile(fileName, dependentVariable)

Auxiliary Function 1 :

initiate Generate initial bolean matrix for parameter selection

---

Description

Parameters

1. *data*: Dataframe, input data containing all independent variables of interest.
2. *populationSize*: Integer, number of models in first generation (default = 50)

Output Returns matrix PopulationSize as data frame

Note All zero rows are excluded, since the purpose of the function is for variable selection.

Examples

1. parentPopulation ← initiate(dataFileName, 25)
2. parentPopulation ← initiate(dataFileName)

Auxiliary Function 3 :

runModel Evaluates the fitness of each set of variable combinations (model) in the population

---

Description

Parameters

1. *model*: Matrix, created by initiation function
2. *dependentVariable*: String, name of dependent variable in model
3. *regressionType*: String, regression type (LM or GLM)
4. *criterion*: String, optimization technique (optional- defaults to AIC, but BIC may be used)

Output Returns a results matrix containing a column with the fitness score for each model

Note Uses helper functions disjointSets and fitness.

Examples

1. modelFit ← runModel(populationMatrix, dependentVariable, regregressionType=”LM”)
2. modelFit ← runModel(populationMatrix, dependentVariable, regressionType=”LM”, criterion=”BIC”)

Auxiliary Function 4 :

crossover Realizes the crossover between two chromosomes

Description

Parameters

1. *chromosome1*: Vector, The first parent chromosome
2. *chromosome2*: Vector, the second parent chromosome

Output One chromosome containing the starting sequence of chromosome1 (before the splitting point) and the finishing sequence of chromosome2 (after the splitting point)

Note Two chromosomes produce one chromosome after the crossover.

Examples

1. crossedChromosome ← crossover(chromosome1, chromosome2)

Auxiliary Function 6 :

Matching Matches two parents to breed within a population

Description

Parameters

1. *parentPopulation*: Matrix, the parent population
2. *selectionProbability*: Double, the probability for each parent to be chosen to breed

Output The two parents chosen to breed

Examples

1. parents ← matching(parentPopulation, selectionProbability)

Auxiliary Function 8 :

Breeding Generates an offspring population by breeding a parent population

---

Description

Parameters

1. *parentPopulation*: Matrix, the parent population

Output A new offspring population resulting from the breeding of the parent population

Note The offspring population is of the same size as the parent population

Examples

1. offspringPopulation ← breeding(parentPopulation)

Auxiliary Function 5 :

mutation Realizes the mutation operation of resulting crossed-over chromosomes

---

Description

Parameters

1. *offspringPopulation*: Matrix, output generated by crossover function
2. *mutationProbability*: Double, probability of mutation (default = 0.01, must be between 0 and 1)

Output Mutated chromosomes (matrix)

Note Locus in which mutation occurs is random with probability of mutation determined by user

Examples

1. mutatedOffspring ← mutation(offspringPopulation)
2. mutatedOffspring ← mutation(offspringPopulation, mutationProbability=0.05)

Auxiliary Function 7 :

replacement Replaces current population with most fit of parent and offspring populations

---

Description

Parameters

1. *sortedParent*: Dataframe, ranked and sorted parent population chromosomes
2. *sortedOffspring*: Dataframe, ranked and sorted 1st generation chromosomes
3. *replacementPercentage*: Double, percentage of replacement by offspring population (default = 0.4, must be between 0 and 1).

Output Defaults to a dataframe containing 60

Examples

1. newGeneration ← replacement(sortedParent, sortedOffspring, replacementPercentage)
2. newGeneration ← replacement(sortedParent, sortedOffspring, replacementPercentage)