

STAT 243: Software Manual for Model Selection with Genetic Algorithms using **ga**

Eddie Buehler, Yang Hu & Jin Rou New
University of California, Berkeley

Version 1.0, December 9, 2014

1 Introduction

A genetic algorithm has the following steps:

1. Calculate fitness of chromosomes.
2. Select chromosomes to form a mating pool based on their fitness.
3. Recombine parent chromosomes from the mating pool.
4. Apply mutation to produce the resulting generation of chromosomes.

2 Code

```
ga <- select_model(data = data, yvar = "y", xvars = NULL,  
  model = "lm", glm_family = NULL, criterion = "AIC",  
  pop_size = 100, method_select = "rank",  
  method_recombine = "onpoint", prob_recombine = 0.6,  
  prob_mutate = 0.01, num_max_iterations = 100,  
  seed = 123, do_parallel = FALSE)
```

The result of this function is an object of **ga** class that contains the results

2.1 reproduce function

2.2 reproduce function

3 Testing

4 Contributions

5 Appendix

GA-package

Genetic algorithm for model variable selection

Description

Final project for Statistics 243. An R package that implements a genetic algorithm for variable selection in linear and GLM problems.

Details

Package: GA-package
Type: Package
Version: 1.0
Date: 2014-12-13

Author(s)

Eddie Buehler, Yang Hu, JR New

References

G. Givens and J. Hoeting. **Computational Statistics**, 2nd ed. (2012).

See Also

<https://github.com/jrnew/genetic-algo>

Examples

```
# Select regression variables for airquality data using lm model and AIC criterion
select_model(
  data = airquality,
  yvar = "Ozone",
  xvars = NULL,
  model = "lm",
  criterion = "AIC",
  pop_size = 100L,
  method_select = "rank",
  method_recombine = "onepoint",
  prob_recombine = 0.6,
  prob_mutate = 0.01,
  num_max_iterations = 100L,
  seed = 123,
  do_parallel = FALSE
)
```

<code>select_model</code>	<i>Carry out model selection with a genetic algorithm.</i>
---------------------------	--

Description

Main function for carrying out model selection with a genetic algorithm.

Usage

```
select_model(data, yvar, xvars = NULL, model = "lm", glm_family = NULL,
  criterion = "AIC", pop_size = 100L, method_select = "rank",
  method_recombine = "onepoint", prob_recombine = 0.6, prob_mutate = 0.01,
  num_max_iterations = 100L, seed = 123, do_parallel = FALSE)
```

Arguments

<code>data</code>	Data frame
<code>yvar</code>	Character; Name of column containing response variable
<code>xvars</code>	Character vector; Default is all column names that are not <code>yvar</code> ; Name(s) of column(s) containing set of explanatory variables to select on.
<code>model</code>	Character; "lm" (default) or "glm"; Linear model or generalized linear model.
<code>glm_family</code>	Character if model is "glm", NULL otherwise; "binomial", "gaussian" (default), "Gamma", "inverse.gaussian", "poisson", "quasi", "quasibinomial", "quasipoisson"; A family function that gives the error distribution and link function to be used in the model.
<code>criterion</code>	"AIC" (default) or "BIC"; Criterion to be minimized.
<code>pop_size</code>	Integer; Default is 100; Number of chromosomes per generation.
<code>method_select</code>	String; "rank" (linear rank selection) (default) or "tournament"; Method to select chromosomes for inclusion in mating pool.
<code>method_recombine</code>	String; "onepoint" (default), "twopoint", "uniform"; Type of crossover, at one point, at two points or uniformly (at all possible points).
<code>prob_recombine</code>	Numeric, between 0 and 1; Default is 0.6; Probability of recombination.
<code>prob_mutate</code>	Numeric, between 0 and 1; Default is 0.01; Probability of mutation.
<code>num_max_iterations</code>	Non-negative integer; Default is 100; Maximum number of iterations before algorithm is stopped.
<code>seed</code>	Non-negative integer; Default is 123; Random seed for reproducibility.
<code>do_parallel</code>	Logical; Default is FALSE; Do in parallel?

<code>evaluate_once</code>	<i>Do evaluation once.</i>
----------------------------	----------------------------

Description

Do evaluation for a chromosome by calculating model selection criterion.

Usage

```
evaluate_once(model_data, xvars_select, model = "lm", glm_family = NULL,  
              criterion = "AIC")
```

Arguments

<code>model</code>	Character; "lm" (default) or "glm"; Linear model or generalized linear model.
<code>glm_family</code>	Character if model is "glm", NULL otherwise; "binomial", "gaussian" (default), "Gamma", "inverse.gaussian", "poisson", "quasi", "quasibinomial", "quasipoisson"; A family function that gives the error distribution and link function to be used in the model.
<code>criterion</code>	"AIC" (default) or "BIC"; AIC or BIC.
<code>model_data</code> ;	Object of class <code>model_data</code> .
<code>xvars_select</code> ;	Logical vector;

Value

Numeric; Value of criterion.

<code>evaluate</code>	<i>Do evaluation.</i>
-----------------------	-----------------------

Description

Do evaluation for chromosomes in population by calculating model selection criterion.

Usage

```
evaluate(pop, model_data, model = "lm", glm_family = NULL,  
         criterion = "AIC", do_parallel = FALSE)
```

Arguments

<code>pop</code>	Matrix of population of chromosomes.
<code>model_data</code>	Object of class <code>model_data</code> .
<code>model</code>	Character; "lm" (default) or "glm"; Linear model or generalized linear model.
<code>glm_family</code>	Character if model is "glm", NULL otherwise; "binomial", "gaussian" (default), "Gamma", "inverse.gaussian", "poisson", "quasi", "quasibinomial", "quasipoisson"; A family function that gives the error distribution and link function to be used in the model.
<code>criterion</code>	"AIC" (default) or "BIC"; Criterion to be minimized.
<code>do_parallel</code>	Logical; Default FALSE; Do in parallel?

Value

Numeric vector; Evaluation values for all chromosomes in the current generation.

<code>initialize</code>	<i>Initialize first generation of chromosomes.</i>
-------------------------	--

Description

Initialize first generation of chromosomes completely randomly.

Usage

```
initialize(pop_size, num_vars)
```

Arguments

<code>pop_size</code>	Non-negative integer; Number of chromosomes in population.
<code>num_vars</code>	Non-negative integer; Number of variables in model under consideration/ number of genes in each chromosome.

Value

A matrix of size pop_size x num_vars with 1's and 0's.

<code>mutate</code>	<i>Mutate genes in the population.</i>
---------------------	--

Description

Mutate each gene in the population at a pre-defined rate.

Usage

```
mutate(pop, probab_mutate = 0.01)
```

Arguments

<code>pop</code>	Matrix; Population of chromosomes.
<code>probab_mutate</code>	Numeric, between 0 and 1; Default is 0.01; Probability of mutation.

Value

Matrix of population of chromosomes that have undergone mutation.

<code>plot.ga</code>	<i>Plots results from the genetic algorithm.</i>
----------------------	--

Description

Plots the best model evaluation criterion in each generation against the generation iteration.

Usage

```
## S3 method for class 'ga'  
plot(ga, num_view = 3)
```

Arguments

<code>ga</code>	Object of class <code>ga</code> .
<code>num_view</code>	Number of top models to display.

Value

Prints summary of top models and associated value of model selection criterion.

<code>process_data</code>	<i>Process data for input into genetic algorithm.</i>
---------------------------	---

Description

Process data for input into genetic algorithm.

Usage

```
process_data(data, yvar, xvars = NULL)
```

Arguments

<code>data</code>	Data frame
<code>yvar</code>	Character; Name of column containing response variable.
<code>xvars</code>	Character vector; Default is all column names that are not <code>yvar</code> ; Name(s) of column(s) containing set of explanatory variables to select on.

Value

A list object named `model_data` containing:

data Data frame; Processed data with only relevant columns.

yvar Character; Name of column containing response variable.

xvars Character vector; Name(s) of column(s) containing set of explanatory variables to select on.

num_vars Integer; Length of `xvars`.

<code>recombine_once</code>	<i>Recombine once.</i>
-----------------------------	------------------------

Description

Carry out crossover of two parent chromosomes to produce one child chromosome.

Usage

```
recombine_once(parent1, parent2, method = "onepoint")
```

Arguments

parent1 Integer vector of 1st parent chromosome containing 1's and 0's.

parent2 Integer vector of 2nd parent chromosome containing 1's and 0's.

method String; "onepoint" (default), "twopoint", "uniform"; Type of crossover, at one point, at two points or uniformly (at all possible points).

Value

Integer vector of child chromosome containing 1's and 0's.

recombine

Recombine.

Description

Carry out crossover of parent chromosomes in a mating pool.

Usage

```
recombine(pop_mating, pop_size, method = "onpoint", prob_recombine = 0.6,  
          do_parallel = FALSE)
```

Arguments

<code>pop_mating</code>	Matrix of population of chromosomes that form the mating pool.
<code>pop_size</code>	Integer; Number of chromosomes in a generation.
<code>method</code>	String; "onpoint" (default), "twopoint", "uniform"; Type of crossover, at one point, at two points or uniformly (at all possible points).
<code>prob_recombine</code>	Numeric, between 0 and 1; Default is 0.6; Probability of recombination.
<code>do_parallel</code>	Logical; Default FALSE; Do in parallel?

Value

Matrix of population of chromosomes resulting from recombination.

reproduce

Wrapper function for reproduction stage.

Description

Wrapper function for reproduction stage.

Usage

```
reproduce(ga, iteration, do_parallel = FALSE)
```

Arguments

<code>ga</code>	Object of class <code>ga</code> .
<code>iteration</code>	Iteration number.

Value

Updated `ga` list object.

<code>select</code>	<i>Select chromosomes for recombination.</i>
---------------------	--

Description

Select chromosomes for recombination based on fitness.

Usage

```
select(pop, evaluation, method = "rank", do_parallel = FALSE)
```

Arguments

<code>pop</code>	Matrix; Population of chromosomes.
<code>evaluation</code>	Numeric vector; Evaluation values of all chromosomes in population.
<code>method</code>	String; "rank" (linear rank selection) (default) or "tournament"; Method to select chromosomes for inclusion in mating pool.
<code>do_parallel</code>	Logical; Default FALSE; Do in parallel?

Value

Matrix of population of chromosomes that form the mating pool.

`summary.ga`

Display summary of results from the genetic algorithm.

Description

Outputs the top models selected from the genetic algorithm.

Usage

```
## S3 method for class 'ga'  
summary(ga, num_view = 5)
```

Arguments

<code>ga</code>	Object of class <code>ga</code> .
<code>num_view</code>	Number of top models to display.

Value

Prints summary of top models and associated value of model selection criterion.