STAT 243:
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

We took the S3 approach to object-oriented programming for our package and created functions that were as modular as possible to facilitate code creation, maintenance and testing.

For a given data set, e.g. the built-in `airquality` data set in `R`, the user can carry out model selection based on the Akaike Information Criterion for an ordinary linear regression of the variable `Ozone` on other variables in the data set with the following command:

```
ga <- select_model(data = airquality, yvar = "Ozone")
```

Finer control of other parameters in the genetic algorithm

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

The output of the results can be viewed using the following commands:

```
summary(ga)
model(ga)
```

The result of this function is an object of `ga` class that contains the settings, model data, final population of chromosomes/models, model evaluation values of this population and results of model selection using the genetic algorithm.

2.1 reproduce function

2.2 reproduce function

3 Testing

4 Contributions

4.1 Code writing

General structure: Jin Rou New Functions:

4.2 Code testing

Function testing: Eddie Buehler Overall function tests: Yang Hu

4.3 Documentation

Manual creation: Eddie Buehler Project write-up: All

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
)
```

`select_model` *Carry out model selection with a genetic algorithm.*

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

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

`model`	Character; "lm" (default) or "glm"; Linear model or generalized linear model.
`glm_family`	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.
`criterion`	"AIC" (default) or "BIC"; Criterion to be minimized.
`pop_size`	Integer; Default is 100; Number of chromosomes per generation.
`method_select`	String; "rank" (linear rank selection) (default) or "tournament"; Method to select chromosomes for inclusion in mating pool.
`method_recombine`	String; "onepoint" (default), "twopoint", "uniform"; Type of crossover, at one point, at two points or uniformly (at all possible points).
`prob_recombine`	Numeric, between 0 and 1; Default is 0.6; Probability of recombination.
`prob_mutate`	Numeric, between 0 and 1; Default is 0.01; Probability of mutation.
`num_max_iterations`	Non-negative integer; Default is 100; Maximum number of iterations before algorithm is stopped.
`seed`	Non-negative integer; Default is 123; Random seed for reproducibility.
`do_parallel`	Logical; Default is FALSE; Do in parallel?

`evaluate_once` *Do evaluation once.*

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

`model`	Character; "lm" (default) or "glm"; Linear model or generalized linear model.
`glm_family`	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.
`criterion`	"AIC" (default) or "BIC"; AIC or BIC.
`model_data;`	Object of class model_data.
`xvars_select;`	Logical vector;

Value

Numeric; Value of criterion.

`evaluate` *Do evaluation.*

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

`pop`	Matrix of population of chromosomes.
`model_data`	Object of class model_data.
`model`	Character; "lm" (default) or "glm"; Linear model or generalized linear model.

`glm_family`	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.
`criterion`	"AIC" (default) or "BIC"; Criterion to be minimized.
`do_parallel`	Logical; Default FALSE; Do in parallel?

Value

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

`initialize` *Initialize first generation of chromosomes.*

Description

Initialize first generation of chromosomes completely randomly.

Usage

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

Arguments

`pop_size`	Non-negative integer; Number of chromosomes in population.
`num_vars`	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.

`mutate` *Mutate genes in the population.*

Description

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

Usage

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

Arguments

`pop`	Matrix; Population of chromosomes.
`prob_mutate`	Numeric, between 0 and 1; Default is 0.01; Probability of mutation.

Value

Matrix of population of chromosomes that have undergone mutation.

`plot.ga` *Plots results from the genetic algorithim.*

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

`ga`	Object of class ga.
`num_view`	Number of top models to display.

Value

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

`process_data` *Process data for input into genetic algorithm.*

Description

Process data for input into genetic algorithm.

Usage

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

Arguments

`data`	Data frame
`yvar`	Character; Name of column containing response variable.
`xvars`	Character vector; Default is all column names that are not yvar; 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.

`recombine_once` *Recombine once.*

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 = "onepoint", prob_recombine = 0.6,
  do_parallel = FALSE)
```

Arguments

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

`ga`	Object of class ga.
`iteration`	Iteration number.

Value

Updated ga list object.

`select` *Select chromosomes for recombination.*

Description

Select chromosomes for recombination based on fitness.

Usage

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

Arguments

`pop`	Matrix; Population of chromosomes.
`evaluation`	Numeric vector; Evaluation values of all chromosomes in population.
`method`	String; "rank" (linear rank selection) (default) or "tournament"; Method to select chromosomes for inclusion in mating pool.
`do_parallel`	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 algorithim.*

Description

Outputs the top models selected from the genetic algorithm.

Usage

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

Arguments

ga	Object of class ga.
num_view	Number of top models to display.

Value

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