Genetic Algorithm for Variable Selection in Regression Problems

November 27, 2017

1 Object

Your task is to make an R package to implement a genetic algorithm for variable selection in regression problems, including both linear regression and GLMs.

2 Genetic Algorithm

2.1 Overall

Analogy:

- One Chromosome → One candidate model.
- number of alleles C → number of variables.
- i-th allele → i-th variable.
- i-th allele is 0 → not include i-th variable.
- i-th allele is 1 → include i-th variable.

Say $C = 4$. For a single chromosome, 1010, the suggested linear regression model is

$$y = \beta_0 + \beta_1 x_1 + \beta_3 x_2 + \varepsilon \tag{2.1}$$

while β_2, β_4 are zero.

1. Initialization
2. Iteration:
    (a) Evaluate all the initial points:
    evaluate AIC or other fitness function.
    (b) Selection:
    continue on the next generations and use them to create the subsequent generation.
    (c) Crossover:
    Crossover at all the points.
    (d) Mutation:
    mutation.
    (e) Next Generation and Redo the iteration
3. Until convergence

2.2 Primary Issues

- crossover in model selection? how many crossover point? Provide an option for the function, $crossover = TRUE/FALSE$, $crossover\_point =$ integer input, randomly distribute the points given the number or input a vector, customize the position of crossover. Another issue: should the number and position of crossover points be random?

2.3 Initialization and Parameter values

- Equal sizes for subsequent generations are not required.
- large generation size P for early generations to discourage premature convergence and promote search diversity.
- P can be decreased progressively as iterations continue
- variable mutation rate that is inversely proportional to the population diversity. Purpose: provides a stimulus to promote search diversity as generations become less diverse.

2.4 Evaluation

Evaluate the fitness function/Criteria:

- Akaike information criterion.
  From $\{stats\}$ package, the $AIC()$ function.
- Bayesian information criterion.
- Cross-validation.
- Deviance information criterion.
- False discovery rate.
- Focused information criterion.

For other fitness functions, user can supply the function defined themselves.

2.5 Selection

Select a few parents from the $i-th$ generation to continue on the next generations and use them to create the subsequent generation. The selection criteria:

- Based on the rank of the fitness function (AIC of the models).

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

  where P is the size of the generation, r_i is the rank of $f(\theta_i^{(t)})$ among generation t.

2.6 Crossover

2.7 Mutation

3 Requirement

3.1 Generality

Allow reasonable inputs, in terms of specifying a dataset and regression model formula, as well as the type of regressions. Much of this is information you should just be able to pass along to lm() or glm().

3.2 Coding Style

- modular code
- functions or OOP methods that implement discrete tasks.
- overall design and style that is consistent across the components, in terms of functions vs. OOP methods, naming of objects, etc.

3.3 Efficiency

- Try to vectorize as much as possible to speed up.
- allow for shared memory parallel processing when working with the population in a given generation, in particular the evaluation of the fitness function,

3.4 Validation and Testing

- implementation on one or more real examples.
- Formal testing is required, with a set of tests where results are compared to some known truth.

3.5 Criterion

By default AIC as your objective criterion/fitness function. Allow specific fitness functions.

3.6 Operator

use the genetic operators described in Givens and Hoeting for variable selection. Allow additional operators.