Function Skeleton Design(need to be updated)

Skander/Yu/Yachuan

December 4, 2017

The point marked by ? is additional, which we haven't thought thoroughly about how to design.

Input

1. Dataset
2. Type of regression
   - lm()
   - glm()
3. Criterion
   - AIC(default)
   - BIC
   - . . . (Other common criteria)
   - User can provide objective function?
4. Parent Selection mechanism
   - Chosen with probability proportional to the fitness value.
   - One proportional. One Random
   - generation gap, G
   - tournament selection
5. Gene selection mechanisim
   - Operator choice
     - Crossover
     - Randonm loci selection and chromosomes swap
     - Mutation
     - Additional operators provided by user
   - Rate
6. Maximum number of iterations
7. Other arguments, similar to those of lm() and glm(), including some choices on the output format.

Funtion Design

1. Initialization
   - Normalize the data
   - Calculate the covariance matrix between the covariates
   - Choose the interaction terms
   - Generate the first generation
     - It is recommended to use a heuristic approach to select starting chromosome instead of random selection.
2. Parent selection in n-th generation
   - Transform objective function $f(\theta)$ to fitness function $\Phi(\theta)$
   - Selection mechanisom. Mutiple methods are provided in the website mentioned in the input part. Generation gap G is also considered.
3. Gene selection. Use specific genetic operator to generate the n+1-th generation.
4. Stopping critetia:
   - Whether the number of iterations exceeds the max number.

- Whether the error(objective function value) is small enough

Other thoughts about the function design

1. Good data structure design to improve computing efficiency.
   - For example, if using partial updating(only update a proportion of the n-th generation in the n+1-th generation), the fitness value of a given chromosome may be used more than once. And if calculating the fitness value of one chromosome is complicated, it may be better to store the value. I think hashtable can solve the probelm.
2. Avoid colinear problem
3. Utils functions

Output

1. Result (?prediction of the parameters)
2. Objective funtion value
3. The number of iterations
4. ?The corelation of the variables selected.
5. Other outputs which can make our function perfect.
6. Convergency analysis

Implementation on real example

Test

1. Simulation tests.

The skeleton we designed is based on the material provided by Chris. There are a lot details in the design that we haven't considered yet. Further work is needed. Feel free t correct and improve it.