

STAT 243 : Project

Skander Jemaa
Rivers Jenkins
Yachuan Liu
Yu Wang

December 14, 2017

Introduction

We have chosen a functional coding style. We identified 3 main parts in the implementation. The first one was the parents selection mechanism, then came the gene operator and the last one was to carry out the production of a new generation.

For each part, we have designed several approaches and the arguments can be changed by the user that can provide their own functions.

In order to ease of reading and debugging we designed utility functions for each step.

We will present each step of the genetic algorithm below.

Parent Selection Mechanism

To create each successive generation in the genetic algorithm, we have to select parents from the previous generation to be paired off. After two parents are paired off, they will create two children according to the gene selection mechanisms detailed in a later section. In order to select parents for each generation, there are a variety of methods available to us. We chose to implement the following four methods of parent selection.

1. Random: This selection mechanism is the simplest. We randomly choose two parents from the population to pair off.
2. Proportional: This selection mechanism is more discriminating in picking parents. For each individual in the current generation, the fitness or objective function is calculated. We then divide this value by the sum of the objective values for all the individuals in that generation to get the probability of selecting an individual to be a parent. This allows the algorithm to pick two parents proportionally to their fitness values. Both parents are picked in this manner.
3. Proportional/Random: This method is a hybrid of the previous two methods. One parent is selected according to the proportional method described above, and one parent is selected uniformly randomly.
4. Tournament: This selection mechanism groups the individuals of the current generation in groups of size k , and then selects parents based on the best fitness within each of these groups.

Gene operator

After the parents are selected for a new generation, we must create the new children for this generation. We can do this using a variety of gene operators on the two parents. The gene operators we implemented are detailed below.

1. Crossover: This is similar to selecting which chromosome to take from each parent in human reproduction. We select a sequence of alleles (in this case binary digits) from each parent to add to the child. So if each parent has 5 alleles with one as (1,0,0,0,0) and the other as (1,1,1,1,1), we could take the first 3 alleles from parent 1, and the last two from parent 2 giving (1,0,0,1,1). These sequences do not have to be contiguous, thought they were in that example.
2. Allele Swapping: This is similar to crossover, but instead of looking at sequences of alleles, we swap individual alleles. After the initial children are formed using crossover, we randomly pick k alleles to swap between the children. So (1,1,0,0) could become (1,1,0,1) if we pick the last allele to swap and the other child had a 1 in that position.

3. Mutation: This is the final step in constructing the new children and is performed separately on each child. We will randomly mutate an allele according to some (usually quite small) probability. This ensures that we are not totally restricted by the alleles in the parent generation.

Iterations of the genetic algorithm

Initialization

Speak about most significant variables, interactions, random selection of covariates...

Iterations

permutation gap, decrease of the population size,

select function

Describe it and how parameters can be changed saying that the use of `as.name` allows the user to enter their own function.