MLDS 490 Lab 9

Shuyang Wang Fall 2023

Agenda

- Assignment 4 due on December 8 by 10pm. No Joker.
- Office hours on final week on Zoom:
  - Monday 5:30-6:30pm
  - Thursday 5:30-6:30pm
- Genetic Algorithm
- Bayesian optimization

Genetic Algorithm

Genotype representation: one-hot encoding for categorical variables

Muticlass classification: use the multiclass version of macro-averaged F1 score

Fitness score of a chromosome: validation F1 by the model trained with the hyperparameters represented by this chromosome.

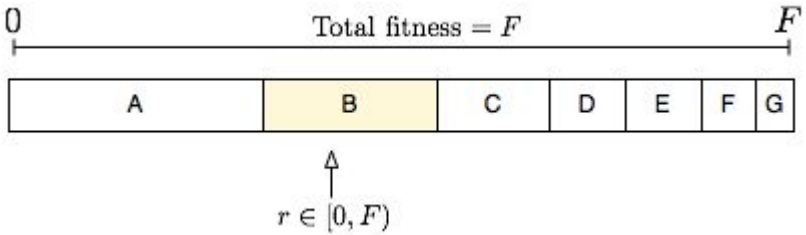
1. Initialize the population
2. Compute the fitness scores
3. Select parents: Roulette wheel
4. Produce offsprings: Crossover + mutation
5. Age based selection to update the population -> new generation
6. Repeat steps 2-5 until termination

Roulette wheel

- Select a subset from the population of the current generation as **Parents.**

  Let N be the population size, f_i is the fitness score for the i-th candidate, probabilities p_i should be proportional to f_i's: $$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j},$$

  Sample a random number r, and select according to:



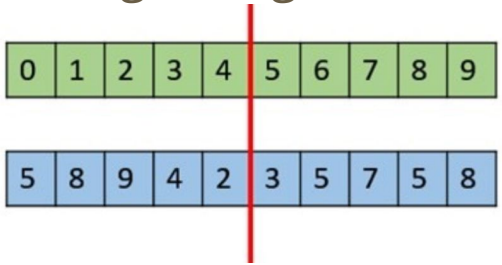
Crossover: one-point crossover

1. Sample two parent chromosomes (with replacement) from the selected set of **Parents**
2. Produce offsprings by crossover operation:

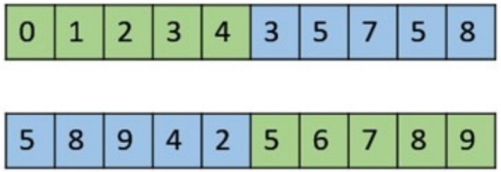
   2.1. Randomly select a crossover point

   2.2. Exchange the genes to the right of the crossover point

2 parents



=>



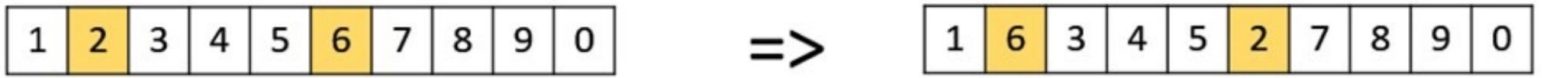
2 Children

slide 15 from Week 11 GA

Mutation

1. For each offspring chromosome, with a small probability to mutate it:

   Change the value **within the permissible range** for this gene: based on swap mutation, you can customize the mutation rule to ensure the resulting chromosome is within the permissible range.



Age based selection

- Requires you to keep track of the age (# of generation) of all the chromosomes in the current population and make increments after each generation.
- Replace the oldest chromosomes with the new offsprings, while keeping the population size unchanged.

Bayesian Optimization

Learn and use the package bayes_opt:

```
pip install bayesian-optimization
```

Tutorials:
https://github.com/bayesian-optimization/BayesianOptimization/tree/master