

CS341 SYSTEMS BIOLOGY I

LECTURE 21

GENETIC ALGORITHMS FOR PARAMETER-FITTING

Linear Ranking Selection. Recall that we have implemented one useless selection operator (uniform) and two useful selection operators (truncation and tournament). Truncation selection results in a uniform distribution of costs, with a small range (if $\text{numParents} \times 5$ is numChildren , then it is the first $1/5$ th of cost). Tournament selection results in a triangular distribution in which lower cost individuals are more likely to be chosen, so we have more of them, then see a roughly linear decrease as the costs increase. Tournament selection operates by randomly choosing k individuals from the previous generation and adding the best-cost individual of that tournament to the breeding pool. The slope of the triangle top is indirectly controlled by k . For small k , we are more likely to be stuck with more higher cost individuals, so the slope is more gradual. For large k , every tournament has a large number of individuals, so we are more likely to have a low-cost individual in every tournament. Therefore, the breeding pool has more low-cost individuals and the slope is steeper.

Today, we are going to talk about linear ranking selection, which has a distribution similar to that of tournament selection. We control the slope of the distribution directly, by specifying how much more likely the best cost individual is likely to be chosen than the worst cost individual. Figure 1 show the distributions of costs after applying linear ranking selection to the population we have been using as our example - one with costs uniformly distributed from 0.1 to 500. Figure 2 shows the probability distribution we construct to carry out the linear ranking selection. Individual 1 is the lowest-cost individual, and its probability for being selected into the breeding pool is p_1 . Individual 50 is the highest-cost individual, and its probability for being selected into the breeding pool is p_{50} , but let's call it p_n when there are n individuals to choose from. In our example, the ratio of probabilities between individual 1 and individual 50 is 4, so $p_1 = 4p_n$.

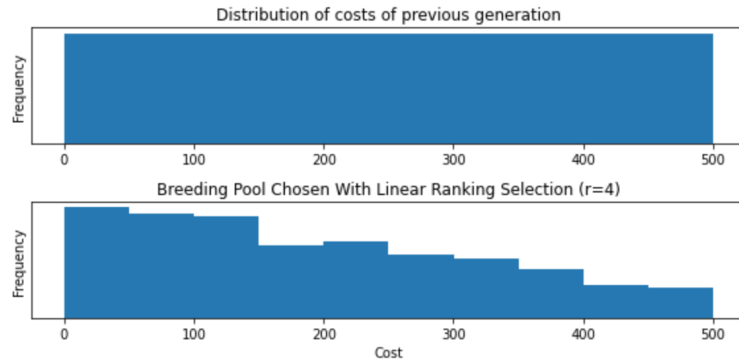


FIGURE 1. Histograms of costs of previous generation (top) and breeding pool (bottom). The previous generation has evenly spaced costs ranging from 0.1 to 500. Using linear ranking selection with a ratio of 4, our distribution of costs favors low-cost individuals and the lowest cost individuals appear with 4 times the frequency as the highest cost.

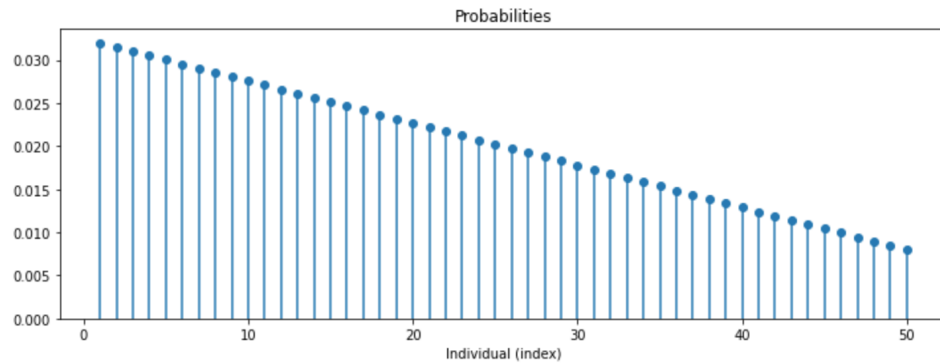


FIGURE 2. Discrete probability distribution used by linear ranking selection with a ratio of 4:1 for best:worst cost individuals. The previous generation has 50 individuals, so there are 50 discrete probabilities. As in any good probability distribution, the probabilities sum to 1.

How to Construct the Discrete Probability Distribution for Linear Ranking Selection. We need r , the ratio of probability between the best and worst cost individuals and the number of individuals n in the previous generation.

We begin with the ratio of probabilities:

$$p_1 = rp_n$$

We need a formula for p_i that takes into account the fact that

$$1 = \sum_{i=1}^n p_i$$

That formula is

$$p_i = p_n + (p_1 - p_n) \frac{n-i}{n-1}$$

where

$$p_n = \frac{2}{n} \left(\frac{1}{r+1} \right)$$

Mathematical diversion. Let's derive the formula for p_n given the formula for p_i and the fact that all the p_i 's must sum to 1.

$$\begin{aligned}
1 &= \sum_{i=1}^n p_i \\
&= \sum_{i=1}^n \left(p_n + (p_1 - p_n) \frac{n-i}{n-1} \right) \\
&= \sum_{i=1}^n p_n + \sum_{i=1}^n \left((p_1 - p_n) \frac{n-i}{n-1} \right) \\
&= \sum_{i=1}^n p_n + \sum_{i=1}^n \left(\left(\frac{p_1 - p_n}{n-1} \right) (n-i) \right) \\
&= n \cdot p_n + \frac{p_1 - p_n}{n-1} \sum_{i=1}^n (n-i) \\
&= n \cdot p_n + \frac{p_1 - p_n}{n-1} \left(\sum_{i=1}^n n - \sum_{i=1}^n i \right) \\
&= n \cdot p_n + \frac{p_1 - p_n}{n-1} \left(n^2 - \frac{n(n+1)}{2} \right) \\
&= n \cdot p_n + \frac{r \cdot p_n - p_n}{n-1} \left(n^2 - \frac{n(n+1)}{2} \right) \\
&= p_n \left(n + \frac{r-1}{n-1} \left(n^2 - \frac{n(n+1)}{2} \right) \right) \\
p_n &= \frac{1}{n + \frac{r-1}{n-1} \left(n^2 - \frac{n(n+1)}{2} \right)} \\
&= \frac{1}{n + \frac{r-1}{n-1} \left(\frac{n^2}{2} - \frac{n^2}{2} - \frac{n}{2} \right)} \\
&= \frac{1}{n + \left(\frac{r-1}{n-1} \right) \left(\frac{n}{2} \right) (n-1)} = \frac{1}{n + n \frac{r-1}{2}} \\
&= \left(\frac{1}{n} \right) \frac{1}{1 + \frac{r-1}{2}} = \left(\frac{1}{n} \right) \frac{2}{2 + r - 1} \\
&= \left(\frac{2}{n} \right) \frac{1}{1 + r}
\end{aligned}$$

How to use our own probability distribution to generate random numbers. We want to generate random numbers from the probability distribution we just defined. We can do that using the `random.random`, which returns a random number from the uniform distribution in the range 0 to 1. To do so, we need to use the cumulative probability distribution. In the cumulative distribution, the first element is p_1 , the second is $p_1 + p_2$, third is $p_1 + p_2 + p_3$, and the n^{th} element is 1 (from $\sum_{i=1}^n p_i$). This is most easily explained with a smaller example (see Figure 3).

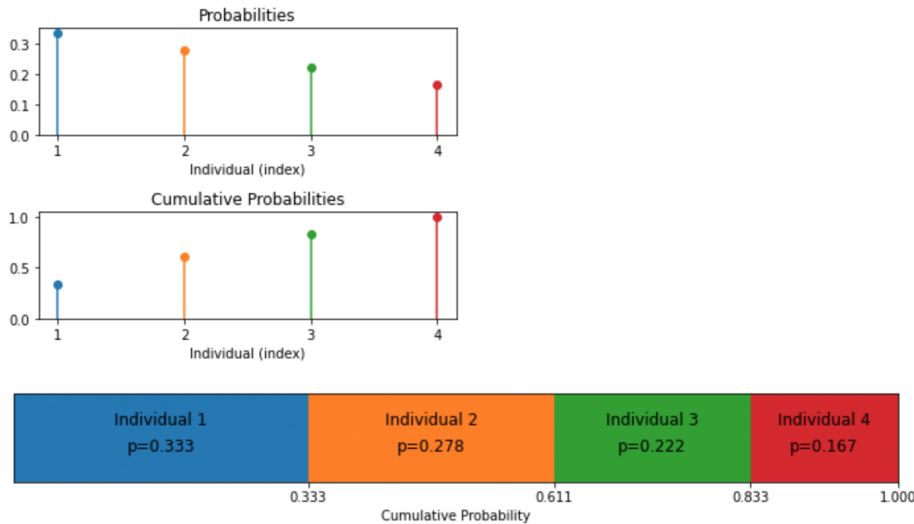


FIGURE 3. Linear ranking selection from tiny population. (Top) Discrete probability distribution used by linear ranking selection with a ratio of 2:1 for best:worst cost individuals. The previous generation has 4 individuals, so there are 4 discrete probabilities. As in any good probability distribution, the probabilities sum to 1. (Middle) We show the cumulative probability distribution (and note that the final value is 1, because the probabilities sum to 1.) (Bottom) We layout the cumulative probabilities so that the area is proportional to the probability. This makes it easy to see that if we choose a random number between 0 and 1 (from a uniform distribution), then 33.3% of the time, the number will be in the blue area, 27.8% of the time, the number will be in the orange area, 22.2% of the time, the number will be in the green area, and 16.7% of the time, the number will be in the red area. To choose an individual, we generate a random number from the uniform distribution from 0 to 1, then search the cumulative probabilities for the first element in the cumulative probabilities list that is larger than the random number. That index tells us which individual has been chosen.