COMPUTER LAB

Evolution of communication

Evolution of Language and Music

October 23, 2018

In this lab, we will model the evolution of a communication system. A possible way of representing a communication system is by using matrices that describe a mapping from a set of meanings to a set of forms (or signals). For instance, the well known alarm call system of Vervet monkeys **seyfarth1980monkey** in its usual idealisation, can be described as follows:

$$S = \left(\begin{array}{c|ccc} & \text{chirp} & \text{grunt} & \text{chutter} \\ \hline \text{leopard} & 0.8 & 0.2 & 0 \\ \text{eagle} & 0.1 & 0.9 & 0 \\ \text{snake} & 0.05 & 0.1 & 0.85 \end{array}\right)$$

$$R = \left(\begin{array}{c|ccc} & \text{leopard} & \text{eagle} & \text{snake} \\ \hline \text{chirp} & 0.9 & 0 & 0.1 \\ \text{grunt} & 0 & 1 & 0 \\ \text{chutter} & 0.2 & 0 & 0.8 \end{array}\right)$$

The S matrix represents the sender: the first column contains the meanings (or situations) that the sender may want to express, the first row the signals that it can use to express these meanings. The numbers in the matrix represent the probabilities that the sender will use a certain signal to express a certain meaning. The matrix R describes the behaviour of the receiver in a similar way: the numbers in the matrix are the probabilities that the receiver will interpret a certain signal (first column) as having a certain meaning (first row).

EXERCISE 1

- ⋆ What are the optimal S and R, for maximal communicative success in a population?
- ⋆ How is ambiguity (i.e., one signal with multiple meanings) reflected in S and R matrices? And synonymity (two signals that have the same meaning).

By using a bit of a trick, we can study the evolution of such a communication system using the same protocol as in the first part of this assignment. The S and R matrices of an individual are uniquely defined by 18 numbers. Assume that we model this by saying that every individual is characterized by a genome of length 18, where each nucleotide codes for one value in S and R. Let's say

$A = 3, G = 2, C = 1$ and $U = 0$. To construct the S and R matrices, we put the numbers corresponding to the nucleotides in two matrices and normalise the rows, such that the probabilities add up to 1.

EXERCISE 2

? What would a genome corresponding to the S and R matrix depicted above look like?
? Can you think of two strings that have a different genotype but the same phenotype?

Of course our previous fitness function — the count of the substring "CAC" — does not make much sense in this case. We will have to define a new one. We can compute the chance of successful communication between two agents by summing up the chance of success for each individual meaning-signal combination. For instance, let's assume the sender wants to convey the meaning "leopard". We multiply the probabilities for all signals the sender could use for this meaning (the row in S starting with "leopard") with the chance that the receiver will interpret this signal as having the meaning "leopard" (the "leopard" column in R). In this case as the sender only uses the signal chirp to express the meaning leopard, and the receiver interprets this signal as having the meaning leopard with probability 1, the chance of successfully conveying the meaning "leopard" in this system is 1. Due to the fact that agents have both a sender and receiver matrix, it is possible that the communication in one direction runs flawlessly, but any communication in the other direction is unsuccesfull. We define the fitness as the sum of the chances of success for all meanings in both directions.

EXERCISE 3

? Is it possible to compute the fitness of one individual without taking into account who he is communicating with? Why (not)?

We implemented some fitness functions that you can find in the file `auxiliary_functions.R`:

- `CAC_count`: This is the fitness function you used before, that counts the number of occurrences of the substring "CAC" in the genome;
- `communication_fixed_target`: This fitness function captures how well the population member can communicate with a fixed target with S and R matrices that allow perfect communication (i.e., this other population member does not use the same signal for different meanings, or assign different meanings to the same signal).
- `communication_random_target`: This fitness function describes the more realistic situation, in which the fitness of a population member is determined based on its communication with a random other member of the population.

EXERCISE 4

- Load the auxiliary functions library by typing

  ```
  source('auxiliary_functions.R')
  ```

  in the terminal. Leave the file `auxiliary_functions.R` untouched, you don't have to change anything there.

You can change the fitness function - like the rest of the parameters - at the top of the file `lab-2.R`, by uncommenting the line with the preferred fitness function (and commenting out all other fitness function lines). As you may have guessed, you can (un)comment a line in an R script by placing (removing) a '#' at the beginning.

EXERCISE 5

? What is the maximal fitness that an individual can have?

- Change the fitness function in the script to `communication_fixed_target`. Run an evolutionary simulation with a low mutation rate for 100 iterations. What is the average fitness and most frequent communication system at the end of it? You can check the population at the end by typing `population` in your command line.

? Can the members of the resulting population also communicate with each other or only with the preset fixed target?

? What would happen if the target was fixed, but not perfect? You can test your assumption by changing the target matrices in the `auxiliary_functions file`.

A more realistic situation is the one in which the members of the population do not all communicate with the same fixed target, but with other members of the population, that have their own (evolved) S and R matrix.

EXERCISE 6

- Run some evolutionary simulations for this scenario (compute the fitness by using the function `communication_random_target`). What is the average fitness and most frequent communication system at the end of it? Experiment with the learning rate.

⋆ This is frequency dependent selection. Why does it not always evolve to the optimal communication system?

⋆ What do you expect to happen if only successfully *receiving* but not *sending* contributes to fitness?

- Test your assumption by using the fitness function `sending_random_target`.