Evolution of Language

Computerlab I

February 12, 2014

1 Senders and receivers

A commonly used representation for communication systems involves two matrices that describe the mapping from a set of "meanings" (or "situations") to a set of "signals". Each value (m, s) in the matrix gives the probability that signal s is used to express meaning m (for senders), or the probability that interpretation m is given to received signal s (for receivers). For instance, the well-known alarm call system of Vervet-monkeys (Seyfarth *et al.*, 1980), in its usual idealization, can be described as follows:

$$\mathbf{S} = \left(\begin{array}{r|ccc} & \text{chirp} & \text{grunt} & \text{chutter} \\ \hline \text{leopard} & 1 & 0 & 0 \\ \text{eagle} & 0 & 1 & 0 \\ \text{snake} & 0 & 0 & 1 \end{array}\right) \qquad \mathbf{R} = \left(\begin{array}{r|ccc} & \text{leopard} & \text{eagle} & \text{snake} \\ \hline \text{chirp} & 1 & 0 & 0 \\ \text{grunt} & 0 & 1 & 0 \\ \text{chutter} & 0 & 0 & 1 \end{array}\right)$$

In today's computer lab we will do a few calculations with such matrix in the program R, with the dual goals of familiarizing ourselves a little bit with R and in sharpening our intuitions about the mechanisms of evolution and the evolution of communication.

First let's try to enter these matrices into R. Start R and type the following commands into the console (you may cut and paste). Make type the commands exactly as given, including spaces and symbols like `"` (which is not the same as " or ").

```
-------------------------------------------------------------------
signals <- c("chirp","grunt","chutter")
meanings <- c("leopard","eagle","snake")

# Create a 3x3 matrix describing the sender behaviour
sender <- diag(3)
rownames(sender) <- meanings
colnames(sender) <- signals
sender

#Create a 3x3 matrix describing the receiver behaviour
receiver = matrix(c(0,1,0,0,1,0,0,0,1),3,3)
rownames(receiver) <- signals
colnames(receiver) <- meanings
receiver
-------------------------------------------------------------------
```

The advantage of R is that it includes many mathematical operations. For instance, we can multiply those matrices with `sender*receiver`.

Question 1 *What do the values in the resulting matrix mean?*

The sum of the values of `sender*receiver` give a measure for how many meanings will be successfully communicated between sender and receiver. We will call this measure 'payoff'.

Question 2 *What is the sum of those values? Hint: the function* `sum(mat)` *sums the values from matrix* `mat`*.*

Question 3 *Which sender and receiver matrices will give you the most successful communication?*

The values in the matrices can also be between 0 and 1. This models ambiguity: for the same received signal (e.g., chutter), the monkeys might sometimes think "leopard!" and sometimes "eagle!". Let's imagine a population of monkeys where everybody's receiver-behavior is slightly different. Let's leave the sender's behavior (`sender <- diag(3)`) fixed as before, and compute the payoff's of the various receivers:

```
-------------------------------------------------------------------
# define a function "normalizerows": a row should add up to 1.
normalizerows <- function (mat) {
 for (r in 1:3) {thesum=sum(mat[r,]);
   for (c in 1:3) {mat[r,c]<-mat[r,c]/thesum}};
 mat[which(mat<0)]<-0
 mat[which(mat>1)]<-1
 return(mat)}

# create a population of 10 receivers and initialize them at random
popsize = 10
population <- array(0, dim=c(3,3,popsize))
for (i in 1:popsize) population[,,i]<-normalizerows(matrix(runif(9),3,3))
population
payoff <- matrix(0,1,popsize)

# calculate the payoff of each receiver communicating with sender
for (i in 1:popsize) payoff[i] <- sum(sender*population[,,i])
payoff
-------------------------------------------------------------------
```

Now, we are very close to running an evolutionary simulation. All we need to add is procedures for natural selection and random mutation:

```
-------------------------------------------------------------------
# apply natural selection: low payoff receivers have no offspring
# max payoff receiver has most offspring.
population[,,which(payoff<mean(payoff))]<-population[,,which.max(payoff)]

# apply mutations
population<-population+0.01*array(runif(90)-0.5,c(3,3,10))
for (i in 1:10) population[,,i]<-normalizerows(population[,,i])

#check what population now look like
population
-------------------------------------------------------------------
```

To demonstrate the power of natural selection, let repeat those steps (calculate payoffs, apply natural selection, apply mutation) for 100 times:

```
-------------------------------------------------------------------
for (generations in 1:100) {
 population<-population+0.01*array(runif(9*popsize)-0.5,c(3,3,popsize))
 for (i in 1:popsize) population[,,i]<-normalizerows(population[,,i])
 for (i in 1:popsize) payoff[i] <- sum(sender*population[,,i])
 print(mean(payoff))
 population[,,which(payoff<mean(payoff))]<-population[,,which.max(payoff)]
 }
-------------------------------------------------------------------
```

Question 4 *Describes what happens to the average payoff in the population? How many generations does it take before the payoff approaches the maximum?*

2 Simulation

In the small simulation in R we assumed the sender behavior remained fixed. Under those circumstances, we predict the evolution of an optimal communication system. But what if the sender behavior is also evolving? If only receivers benefit, why would senders continue to send truthful signals? This is an instance of what is called the "Tragedy of the Commons".

To experience this tragedy, we now turn to a more advanced simulation, available by loading the following page in your browser (you may have to switch to Explorer and add `http://homepages.inf.ed.ac.uk` as an exception in your Configure Java settings): `http://homepages.inf.ed.ac.uk/fsangati/language_evolution.html`

Question 5 *Select randomize, optimal filling, set 200 iterations and 3 meanings. Click 'go'. Describe the results you see.*

Question 6 *Select 'iterated' and run the simulation again. Do you observe different results?*

Question 7 *Deselect 'randomize' and 'iterated' and run the simulation again. Do you observe different results?*

References

Seyfarth, R., Cheney, D. & Marler, P. (1980). Monkey responses to three different alarm calls: evidence of predator classification and semantic communication. *Science* **210**, 801–803.