Duck Battle!

Math 339

January 15, 2010

Objective

For your course project, you need to devise a strategy a set of rules that specify when an individual duck should eat the bread from its current bread dispenser to the best of its ability (FORAGE), when it should slightly compromise its bread eating abilities to learn what is going on over at the other bread dispenser (OBSERVE) and when it should drastically compromise its bread eating abilities and move over to the other bread dispenser (SWITCH), in a variable environment. Being at the right bread dispenser at the right time is important, as fitness depends on how much bread is eaten each turn. However, Observing and Switching are not free, and fitness costs are imposed in terms of forgone bread eating oportunites each time a duck Observes or Switches. For the purposes of the tournament, ducks will be assumed to know their own individual histories of behaviour and the bread payoffs they received each round. Strategies will be tested in a computational simulation framework. The specification of the simulations, and detailed tournament rules are given in the technical details section below. You should ensure that you are familiar with this material, as the details given are crucial in ensuring that your strategy will be considered in the tournament.

Strategy evaluation

Strategies will take part in round-robin contests between all pairs of entered strategies. A contest, say between strategies A and B, involves exploring whether strategy A can invade a population containing only strategy B, and vice-versa. Each contest will involve several repeated simulations, with each strategy as the invader half of the time. In each simulation, after a fixed number of iterations, the frequency of each strategy (that is, the proportion of the population with that strategy) will be recorded, and the average frequency across repetitions will be the score of that strategy in that contest.

Technical Details

Matlab code of how the simulation will actually be run will be posted soon, in the mean time you have words.

1 Simulation Specifics

There is a population of 100 ducks who are at any given time either at bread dispenser 1 or bread dispenser 2. A simulation consists of 10000 rounds. Each round the following occurs

1. Each duck will submits a decision to either Forage, Observe, or Switch.
2. Payoffs to each duck are calculated using C_o as the cost of observing and C_s as the cost of switching.
3. A death-birth proccess occurs where each duck has a P_d chance of dieing and given that they were not the ducks who just died each remaining duck has a chance proportional to their average lifetime bread score of being the parent of a new duck replacing a dead duck.
4. The bread rates at each dispenser change independently of eacho other, each with probability P_c of their value being redrawn from a geometric distribution with mean λ.

1.1 Environment and Behavior

1.1.1

The environment can be thought of as bread dispenser 1 (bd1) and bread dispenser 2 (bd2). each bread dispenser has a bread rate associated with it.

bread dispenser:	0	1
bread rate:	3	7

The bread score a duck gets each round is equal to the bread rate at the dispenser she happens to be at divided by the effective number of ducks at the bread dispenser and then potentially less the costs of observing what is happening at the other bread dispenser and the cost of moving between bread dispensers. A duck's chance of being the parent of a new duck is proportional to their breadscore each round averaged over their whole life time.

1.1.2

The environment is not constant, and the bread rate at each dispenser will change between each round of the simulation with a fixed probability, $P_c = 0.1$. There is no association between payoffs for acts before and after the environment changes - the new payoff will be chosen at random from a geometric distribution with a mean of 10. The payoff at each bread dispenser will change independently of the others.

1.2 Moves

1.2.1

Participants must specify a set of rules, henceforth a strategy, detailing when ducks should perform each of three possible moves. The options are:

1. Forage (ducks do their darndest to get as much bread as they can from the bread dispenser they are currently at)

2. Switch (ducks move to the other bread dispenser and pay some cost in that they compete with the effectiveness of $\frac{1}{4}$ of a Foraging duck at the bread dispenser they move to)

3. Observe(ducks check out whats going on at the other bread dispenser and pay some cost in that they compete with the effectiveness of $\frac{3}{4}$ of a Foraging duck)

1.2.2

Observe is ducks giving 75% of their effort to getting bread, and 25% of their effort this round to figuring out what is happening at the other bread dispenser. If a duck plays Observe it knows with certainty what the bread rate is at the other patch that round.

1.2.3

Ducks remember their own history. This histroy includes past moves, the bread rate of the dispenser they were at, which dispenser they were at, the distribution of ducks between bread dispensers, and the bread rate of the other dispenser if they happened to have been observing. Strategies use this information when deciding which moves to play.

1.2.4

Switch is a duck taking some time out of the mad bread scramble and putting 75% of their effort this round into moving from one bread dispenser to the other, and thus only competing as $\frac{1}{4}$ of a foraging duck at the bread dispenser it arrives at.

1.2.5

Forage is ducks giving 100% of their effort to get as much bread as they can in the moment.

1.2.6

Bread payoffs for a duck are calculated as follows. If a duck forages then its payoff is:

$$\frac{\text{bread rate}}{(\#\text{ forage}) + 0.75(\#\text{ observe}) + 0.25(\#\text{ switch})}$$

If a duck observes then its payoff is:

$$\frac{0.75(\text{bread rate})}{(\#\text{ forage}) + 0.75(\#\text{ observe}) + 0.25(\#\text{ switch})}$$

If a duck switches then its payoff is:

$$\frac{0.25(\text{bread rate})}{(\#\text{ forage}) + 0.75(\#\text{ observe}) + 0.25(\#\text{ switch})}$$

These payoffs will usually be decimals which will make a ducks history more annoying to look at so to make our lives simpler we will multiply each of these fractions by 100 and then round down to the nearest integer.

1.3 Evolutionary Dynamics: Lifespan, fitness, and reproduction

1.3.1

Evolutionary change will occur through a death-birth process. Individuals die at random, with probability of 0.02 per simulation round giving an expected lifespan of 50 rounds, and are replaced by the offspring of ducks selected to reproduce with probability proportional to their mean lifetime bread payoffs. For duck z,

$$Pr(\text{reproduction}) = \frac{B_z}{\sum_i B_i}$$

where B_z is the mean life time bread points of a duck. (i.e. the sum of all its bread payoffs from each round divided by the number of rounds it has been alive.)

1.3.2

Unless mutation occurs, offspring inherit the strategy of their parents. Mutation will occur with probability 1/50, and when it does, the offspring will be of the other strategy type(s) in that simulation. These mutations are how other strategies will first arise in a population initially containing only a single strategy.

2 Running the Simulations

Strategies will take part in round-robin contests against all other strategies. A contest involves each strategy invading a population of the other strategy. In a given simulation, a population of the dominant strategy will be introduced, and run for 100 rounds to establish itself. At this point, mutation will be introduced, providing the second strategy the opportunity to invade. Simulations will then run for up to a further 10,000 rounds. Each pairwise contest will be repeated 10 times with strategy A as the invader and 10 times with strategy B as the invader. The mean frequencies of each strategy in the last quarter of each run (i.e. the last 2,500 rounds in a 10,000 round run) will be averaged over the 20 repetitions. This average will then be recorded as the score of that strategy in that contest. Strategies will be assessed on their total score once every strategy has been tested against every other strategy.

3 How to enter

3.1

Strategies will take the form of computer code functions that take the data specified below as arguments and return a decision on which move to play. An example strategy will be given soon. Strategies must be submitted as a Matlab function (using only those commands available in the base installation, excluding toolboxes). Matlab is avaliable on the computers in the lab here, and if you don't know matlab yet now is a good time to learn since coding is the way of the future and matlab is really useful for people who want to do math and science. (R is also really good for some things)

3.2

The strategy function should return an integer number representing the individuals move in this round (here termed move). To switch it should return -1 to observe it should return 0 and to forage it should return 1.

3.3

Strategies will receive all the information they need in the form of a matrix entitled myhistory. The first row of the this matrix will be an index of the turns the duck in question has been alive. The second row will give the bread dispenser,(1 or 2) that that duck has been at for each round of its life, the third row will give the number of ducks at bread dispenser zero for each round of the duck's life, the forth row will give the move that the duck played, and the fifth row will give the payoff that the duck received that round, the sixth row will give the bread rate at bread dispenser 1 at that round if the duck was at that dispenser or Observed that round, and the seventh row will give the bread rate at dispenser 1 at that round if the duck was at that dispenser or Observed that round. If a duck wasn't at a bread dispenser and didn't observe then a -1 will show up in row six or seven for that round.

Here's an example myhistory:

$$\text{myhistory} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \text{rounds} \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & \text{mypatch} \\ 50 & 53 & 52 & 41 & 38 & 32 & 35 & \text{ducks at 0} \\ 1 & 1 & 1 & 0 & -1 & 0 & -1 & \text{what I did} \\ 21 & 20 & 20 & 17 & 9 & 19 & 8 & \text{my bread payoff} \\ 11 & 11 & 10 & 10 & -1 & 10 & 10 & \text{rate at 0} \\ -1 & -1 & -1 & 15 & 15 & 15 & -1 & \text{rate at 1} \end{pmatrix} \tag{1}$$

So the simulation will pass a duck this history and then the duck will have to pass the simulation back a move of -1 for Switch, 0 for Observe and 1 for Forage.

4 Rules

1. Entrants may be single individuals, or collaborative groups. Entrants may submit only one strategy, and individuals may only participate in one group entry.

2. Your strategy cannot access the disk or memory storage of the computer in any way beyond the information provided as input. I reserve the right to disqualify strategies that are deemed not in the spirit of the contest.

3. Any strategy that has been designed so as in any way to recognise and specifically help other entered strategies at their own expense will be disqualified and the authors of the strategy will be given no further oppor-

tunity to enter a modified strategy. This rule is essential to preserve the evolutionary validity of the tournament.

4. In the event of a tie, the tied strategies will be submitted to further tests under varied simulation conditions as deemed appropriate by me. If I judge that the tied strategies do indeed have equal merit, then they may decide at their discretion to share the prize between the tied entrants.

5. The winning entrant will receive a cash prize of $200 (CAD).