

## Project 2: The Circle of Life

16:198:520

The purpose of this project is to build a probabilistic model of an environment in the presence of uncertainty, and use it to inform and direct decision making. You are pursuing a prey object, while simultaneously being pursued by a predator object, and want to capture your target and evade capture yourself. This will be complicated by not necessarily being able to see where the predator and prey are - but you must use the information you have to make the most informed decisions you can about what to do.

### 1 The Environment

The environment for this project is a graph of nodes connected by edges. The agent, the prey, and the predator, can move between the nodes along the edges. There are 50 nodes, numbered 1 to 50, connected in a large circle. Additionally, add edges at random to increase connectivity across the circle, in the following way:

- Picking a random node with degree less than 3,
- add an edge between it and one node within 5 steps forward or backward along the primary loop. (So node 10 might get connected to node 7 or node 15, but not node 16.)
- Do this until no more edges can be added.

**Graph Theory Question:** With this setup, you can add at most 25 additional edges (*why?*). Are you always able to add this many? If not, what's the smallest number of edges you're always able to add?

### 2 The Predator, The Prey, and You

This environment is occupied by three entities, the Predator, the Prey, and the Agent, who can move from node to node along the edges. The Agent wants to catch the Prey; the Predator wants to catch the Agent. If the Agent and the Prey occupy the same node, the Agent wins. If the Agent and the Predator occupy the same node, the Agent loses. For simplicity, the Predator and the Prey can occupy the same node with no issue. The three players move in rounds, starting with the Agent, followed by the Prey, followed by the Predator.

**The Prey:** The rules for the Prey are simple - every time the Prey moves, it selects among its neighbors or its current cell, uniformly at random (*i.e., if it has 3 neighbors, there is a 1/4 probability of it staying where it is.*) This continues, regardless of the actions or locations of the others, until the game concludes.

**The Predator:** Every time the Predator moves, it looks at its available neighbors, and calculates the shortest distance to the Agent for each neighbor it can move to. It then moves to the neighbor with shortest distance remaining to the Agent. If multiple neighbors have the same distance to the Agent, the Predator selects uniformly at random between them.

**The Agent:** The motion of the Agent is dictated by the specific strategy of the Agent. In partial information settings, the Agent may choose to survey a node at a distance to determine what if anything is in that node before deciding where to move. (Imagine, for instance, the Agent sending out a drone to spy on a given location.) Note, the Agent is aware of how the Predator and Prey choose the actions they are going to take, though is unaware of the specific actions chosen.

### 3 Agents and Environments

Implement the following, and collect data to analyze performance between them. In each case, the Predator, the Prey, and the Agent start at randomly selected nodes (though the Agent should never start on an occupied node).

- **The Complete Information Setting:** In this setting, the Agent always knows exactly where the Predator is and where the Prey is.
  - *Agent 1:* Whenever it is this Agent's turn to move, it will examine each of its available neighbors and select from them in the following order (breaking ties at random).
    - \* Neighbors that are closer to the Prey and farther from the Predator.
    - \* Neighbors that are closer to the Prey and not closer to the Predator.
    - \* Neighbors that are not farther from the Prey and farther from the Predator.
    - \* Neighbors that are not farther from the Prey and not closer to the Predator.
    - \* Neighbors that are farther from the Predator.
    - \* Neighbors that are not closer to the Predator.
    - \* Sit still and pray.
  - *Agent 2:* An Agent of your own design that outperforms Agent 1.
- **The Partial Prey Information Setting:** In this setting, the Agent always knows where the Predator is, but does not necessarily know where the Prey is. Every time the Agent moves, the Agent can first choose a node to survey (anywhere in the graph) to determine whether or not the Prey is there. Additionally, the Agent gains information about where the Prey isn't every time it enters a node and the Prey isn't there. In this setting, the Agent needs to track a belief state for where the Prey is, a collection of probabilities for each node that the Prey is there. Every time the Agent learns something about the Prey, these probabilities need to be updated. Every time the Prey is known to move, these probabilities need to be updated.

In these cases, how should you be updating the probabilities of where the Prey is? Be explicit and clear. There is a right answer to this, and there is definitely a wrong answer that every year proves very popular.

  - *Agent 3:* Whenever it is this Agent's turn to move, if it is not currently certain where the Prey is, it will survey the node with the highest probability of containing the Prey (breaking ties at random), and update the probabilities based on the result. After this, it will assume that the Prey is located in the node with the now highest probability of containing the Prey (breaking ties at random), and will act in accordance with the rules for Agent 1. Note, if it is certain where the Prey is, there is no need to survey any node.
  - *Agent 4:* An Agent of your own design that outperforms Agent 3.
- **The Partial Predator Information Setting:** In this setting, the Agent always knows where the Prey is, but does not necessarily know where the Predator is. Every time the Agent moves, the Agent can first choose a node to survey (anywhere in the graph) to determine whether or not the Predator is there. Additionally, the Agent gains information about where the Predator isn't every time it enters a node and the Predator isn't there. In this setting, the Agent needs to track a belief state for where the Predator is, a collection of probabilities for each node that the Predator is there. Every time the Agent learns something about the Predator, these probabilities need to be updated. Every time the Predator is known to move, these probabilities need to be updated. For this environment, the Agent starts by knowing the location of the Predator.

How do the probability updates for the Predator differ from the probability updates for the Prey? Be explicit and clear.

- *Agent 5:* Whenever it is this Agent's turn to move, if it is not currently certain where the Predator is, it will survey the node with the highest probability of containing the Predator (breaking ties first based on proximity to the Agent, then at random), and update the probabilities based on the result. After this, it will assume the Predator is located at the node with the now highest probability of containing the Predator (breaking ties first based on proximity to the Agent, then at random), and will act in accordance with the rules for Agent 1. Note, if it is certain where the Prey is, there is no need to survey any node.
- *Agent 6:* An Agent of your own design that outperforms Agent 5.
- **The Combined Partial Information Setting:** In this setting, the Agent does not necessarily know where the Predator or Prey are. Every time the Agent moves, the Agent can first choose a node to survey (anywhere in the graph) to determine who occupies that node. In this setting, the Agent needs to keep track of belief states for both the Predator and the Prey, updating them based on information collected and knowledge of the actions of the two players. For this environment, the Agent starts by knowing the location of the Predator.
  - *Agent 7:* Whenever it is this Agent's turn to move, if it is not currently certain where the predator is, it will survey in accordance with Agent 5. If it knows where the Predator is, but not the Prey, it will survey in accordance with Agent 3. As before, however, this Agent only surveys once per round. Once probabilities are updated based on the results of the survey, the Agent acts by assuming the Prey is at the node of highest probability of containing the Prey (breaking ties at random) and assuming the Predator is at the node of highest probability of containing the Predator (breaking ties by proximity to the Agent, then at random). It then applies the actions of Agent 1.
  - *Agent 8:* An Agent of your own design that outperforms Agent 7.

Some questions to consider, when contemplating your own Agents:

- What node should the Agent move towards, given its current information?
- What node should the Agent survey, given its current information?
- How best can the Agent use all the knowledge it has available to it to make a decision?

## 4 Analysis and Report

As usual, your report needs to detail your Agents, how they track information, and how they make decisions based on that information. The math should be explicit and clear, as should be your decision strategies. For each of the environments, you need to collect enough simulations (re-generating the graph environment) to get a good estimate of the performance of these strategies.

For each of the above, you should keep track of the following:

- How often the Predator catches the Agent.
- How often the Agent catches the Prey.

- How often the simulation hangs (no one has caught anything past a certain large time threshold).
- How often the Agent knows exactly where the Prey is during a simulation where that information is partial.
- How often the Agent knows exactly where the Predator is during a simulation where that information is partial.

When do your Agents outperform the specified Agents in the above, and why? Do you think that your Agents utilize the available information as effectively as possible? Why or why not?

For the Combined Partial Information Setting: imagine that your survey drone is defective, and that if something is actually occupying a node being surveyed, there is a 0.1 probability that it gets reported as unoccupied (a false negative).

- How do Agents 7 and 8 compare in this setting, if you do not update your belief update rules to account for this?
- How should you update your belief update rules to account for this?
- How do Agents 7 and 8 compare in this setting, once belief update rules have been updated to account for this?
- Can you build an Agent 9 to do better?

**Bonus:** In the Partial Information Settings - suppose that whenever it is the Agent's turn, the Agent must make a choice to move or to survey, instead of doing both. How should the Agent decide which to do, and once the decision is made, how should the Agent decide what to do (move to take or node to survey)? Implement this and compare its performance to the above.