Project 2: Space Rats

16:198:520

You are a brave bot on a lonely spaceship.

1 The Ship

The ship layout is as in Project 1. There is no fire and there is no button. For the purpose of this project, set the dimension to 30×30, and keep all outer edge cells blocked.

Two problems have occurred that the bot needs to solve.

- A blast of cosmic radiation has wiped some of the bot's memory, and it no longer knows where it is in the ship. It has access to a map, it just does not know where it currently is located in the ship.
- At the same time, a space rat has gotten into the ship and needs to be caught.

At every timestep, the bot can take one of three actions:

- The bot can sense how many of the neighboring eight cells are currently blocked.
- The bot can use its space rat detector which will ping based on how close the space rat is. The space rat detector will tell you definitively if you are in the same cell as the space rat.

  > If the bot is in cell i and the space rat is in cell j, then the probability of hearing a ping is given by $e^{-\alpha(d(i,j)-1)}$, where $d(i,j)$ is the manhattan distance between cell i and cell j, and $\alpha > 0$ is a constant specifying the sensitivity of the detector.

- The bot can attempt to move in a cardinal direction (if it runs into a wall and is blocked, it will know).

Using these actions, the bot must attempt to identify where it is, navigate through the ship, locate the space rat, and catch it. I will propose a bot design for you to implement. You will do so, collect data on its performance (average number of actions to catch the space rat), and then design and implement your own bot to improve on mine.

> **Baseline Bot:** This bot works in two phases.
>
> - Phase 1: Identify where the bot is. Keep a knowledge base of where the bot might be (open cells on the map). Alternating every timestep, first sense the number of blocked neighbors, and rule out all locations in the knowledge base that do not agree with that result. Of the remaining possibilities, identify the direction which is most commonly open, and then attempt to move in that direction. If successful, rule out all remaining locations where that direction was blocked; if unsuccessful, rule out all locations where that direction was not blocked. Repeat until you have identified where the bot is.
> - Phase 2: Track the space rat. Keep a knowledge base of where the space rat might be (how should this be represented?). Alternating every timestep, use the space rat sensor (receiving a ping or no ping with the indicated probabilities), and based on the results, update your knowledge base of where the space rat might be. Then move towards the cell the space rat is most likely to be in. Repeat, until the detector tells you that you are in the same cell as the space rat.
>
> Return the number of movements taken, the number of blocked cell sensing actions taken, and the number of space rat detector actions taken.

2 Data and Analysis

In your writeup, consider and address the following:

1) Explain how the space rat knowledge base should be updated, based on whether the bot receives a ping or does not receive a ping.

   > NOTE: There is a correct answer for this. Work out the math, and give the necessary update formulas.

2) Explain the design and algorithm for the bot that you design, being as specific as possible as to what your bot is actually doing. How does your bot make use of the information available to make informed decisions about what to do next?

3) Evaluate your bot vs the baseline bot, reporting a thorough comparison of performance. Plot your results as a function of α.

   > Note that as α gets larger, the probability of hearing a ping ever gets smaller and smaller - the sensor provides less and less information. Similarly if α is too small, you always hear a ping - which means the sensor is also providing less information. Empirically, I find the most interesting values of α to be between 0 and 0.2.

4) Previously, the space rat was assumed to be stationary. Now assume that in every timestep, after the bot takes an action, the space rat moves in a random open direction.

   – Work out the space rat knowledge base updates based on this. What changes, what probabilities do you need to calculate? Again, there is a correct answer.

   – Simulate and compare both bots in this new environment. How does the performance change? Again compare across $\alpha > 0$.

   – Try to improve on your bot design to be even more effective in this moving space rat situation. What did you need to change, if anything?