

CS 520 : Project 3 - Probability in Gridworld

16:198:520

Due: Nov 14 by 11:55 PM, Submitted via Canvas

We return to Gridworld, but not the Gridworld that you are used to.

As usual, the world is a **dim** by **dim** grid, where cells are blocked with probability $p = 0.3$. Among the unblocked cells, however, there are three kinds of **terrain**. Flat terrain, hilly terrain, and thick forests. An unblocked cell has an equal likelihood of being each of these three terrains.

A robot is located in the gridworld, and wants to find a hidden *target*. The robot can move between unblocked cells (**up/down/left/right**), and can *examine* the cell it is **currently in** to try to determine if the target is there. However, examination can potentially yield **false negatives**. If the target is in the cell being examined, the examination fails with different probabilities based on the terrain. If the terrain is flat, the false negative rate is 0.2 (these cells are relatively easy to search). If the terrain is hilly, the false negative rate is 0.5 (easy to hide in the hills, potentially). If the terrain is a thick forest, the false negative rate is 0.8 (it is very easy to miss the target). Hence if you examine a cell and **fail** to find the target there, it does not mean that the target isn't there - but it does reduce the likelihood that the target is there. If however you examine the cell and find the target, you are done - there are no false positives.

The robot initially does not know what cells are blocked, or the terrain type of each cell. The robot is 'blindfolded' as before, so it can only tell a cell is blocked if it attempts to move into that cell and fails. When it successfully enters (or starts in) a cell, however, the robot can sense the terrain type of that cell. *At every point in time, the agents you build will need to track a belief state, describing the probability of the target being in each cell - this will need to be updated as you learn more about the environment.*

- **Question 1:** (2 points) Prior to any interaction with the environment, what is the probability of the target being in a given cell?
- **Question 2:** (10 points) Let $P_{i,j}(t)$ be the probability that cell (i, j) contains the target, given the observations collected up to time t . At time $t + 1$, suppose you learn new information about cell (x, y) . Depending on what information you learn, the probability for each cell needs to be updated. What should the new $P_{i,j}(t + 1)$ be for each cell (i, j) under the following circumstances:
 - At time $t + 1$ you attempt to enter (x, y) and find it is blocked?
 - At time $t + 1$ you attempt to enter (x, y) , find it unblocked, and also learn its terrain type?
 - At time $t + 1$ you examine cell (x, y) of terrain type flat, and fail to find the target?
 - At time $t + 1$ you examine cell (x, y) of terrain type hilly, and fail to find the target?
 - At time $t + 1$ you examine cell (x, y) of terrain type forest, and fail to find the target?
 - At time $t + 1$ you examine cell (x, y) and find the target?

You may take the density of blocked cells p to be known by the robot, as well as the frequencies of each terrain type, and the false negative rates for each terrain type.

- **Question 3:** (8 points) At time t , with probability $P_{i,j}(t)$ of cell (i, j) containing the target, what is the probability of *finding* the target in cell (x, y) :
 - If (x, y) is hilly?
 - If (x, y) is flat?
 - If (x, y) is forest?

- If (x, y) has never been visited?

Note: If you get these problems incorrect, you will be unable to correctly build the necessary agents.

As in the previous projects, the robot starts knowing nothing about the cell contents or terrains, then learns about its environment by traveling through it and taking measurements (moving and sensing/examining), updating its probabilities as it goes.

For this project, you will need to code three agents:

- **Agent 6:** At any given time, the agent identifies the cell **with the highest probability of containing the target, given what the agent has observed**. If there are multiple cells that have equal (maximum) probability, ties should be broken by distance; if there are equidistant cells with equal (maximum) probability, ties should be broken uniformly at random. The agent then plans a route to that square (Repeated A*). As the agent moves along the planned route, it is both sensing the terrain of the cells it moves into and whether or not the cell it is trying to move into is blocked. If a blocked cell is encountered, or the cell with the highest probability of containing the target changes based on information collected while traveling, the agent must re-plan a new route. Once it arrives at the intended cell, it will examine the cell, and then repeat the process as necessary.
- **Agent 7:** At any time, the agent identifies the cell **with the highest probability of successfully finding the target, given what the agent has observed**. If there are multiple cells that have equal (maximum) probability, ties should be broken by distance; if there are equidistant cells with equal (maximum) probability, ties should be broken uniformly at random. The agent then plans a route to that square (Repeated A*). As the agent moves along the planned route, it is both sensing the terrain of the cells it moves into and whether or not the cell it is trying to move into is blocked. If a blocked cell is encountered, or the cell with the highest probability of finding the target changes based on information collected while traveling, the agent must re-plan a new route. Once it arrives at the intended cell, it will examine the cell then repeat the process as necessary.

You will need to repeatedly run each agent on a variety of randomly generated boards (at constant dimension) to estimate the number of actions (movement + examinations) each agent needs to find the target on average.

- **Question 4:** (30 points) Implement Agent 6 and 7. For both agents, repeatedly run each agent on a variety of randomly generated boards (at constant dimension) to estimate the number of actions (movement + examinations) each agent needs on average to find the target. You will need to collect enough data to determine which of these agents is superior. Do you notice anything about the movement/examinations distribution for each agent? Note, boards where the target is unreachable from the initial agent position should be discarded.

The third agent, **Agent 8** needs to be an agent of your own design. It has the exact same information as Agents 6 and 7, and can move and sense in the exact same way. You may only modify how it decides where to go / what to do next. Agent 8 needs to have a better performance than Agents 6 and 7, in terms of reducing the number of needed actions.

- **Question 5:** (20 points) Describe your algorithm, be explicit as to what decisions it is making, how, and why. How does the belief state ($P_{i,j}(t)$) enter into the decision making? Do you need to calculate anything new that you didn't already have available?
- **Question 6:** (25 points) Implement Agent 8, run it sufficiently many times to give a valid comparison to Agents 6 and 7, and verify that Agent 8 is superior.

- **Question 7:** (5 points) How could you improve Agent 8 even further? Be explicit as to what you could do, how, and what you would need.

Bonus: (30 points) For Agent 9, we let the target move each time step, but only to one of its immediate neighbors (uniformly at random among unblocked neighbors). The Agent can additionally sense in each cell whether the target is in one of its 8 immediate neighbors, but not which one. Build an Agent that adapts to this moving target and extra partial sensing. How does it decide where to go next, and what to do? How is the belief state updated? Implement, and generate enough data for a good comparison.