Anthony Le

ROB 534: SDM

Winter 2017

03-13-2017 (4 PM)

HW #3: Planning Under Uncertainty

Questions

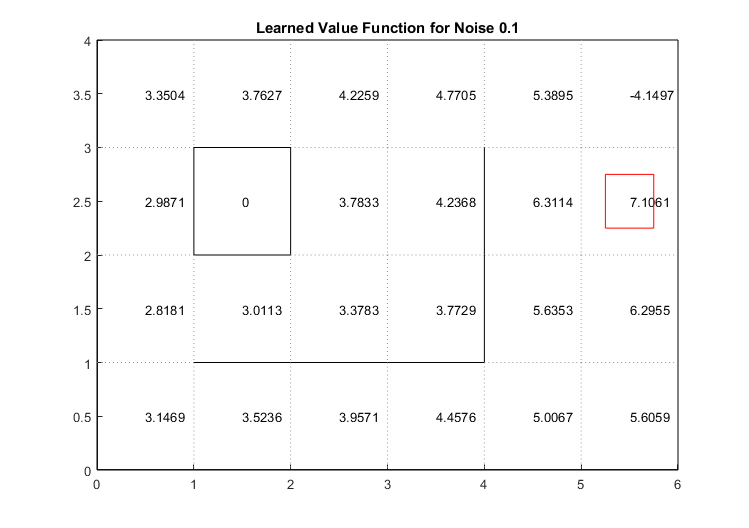
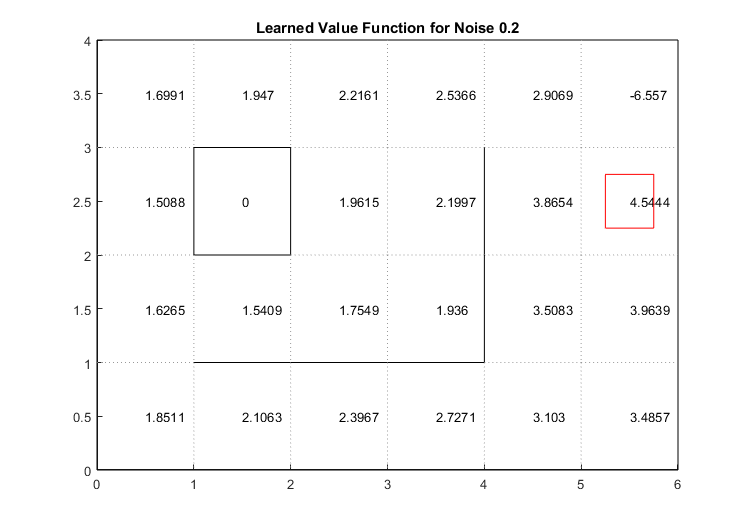
1. “Tiger and Pot-of-Gold”
   1. The state of the world is either when the tiger is behind the left door (sl) or when the tiger is behind the right door (sr). The actions are open the left door (LEFT), open the right door (RIGHT), and listen (LISTEN). The reward for opening the correct door with the pot-of gold behind it is +10 and the penalty for choosing the door with the tiger behind it is -100. The cost of listening is -1. The only two observations are to either hear the tiger behind the left door (TL) or hear the tiger behind the right door (TR). Once the agent opens a door and receives a reward or penalty, the problem resets, and the tiger is randomly relocated behind one of the two doors. The rewards are discounted by a factor (γ)of 0.9 after each action

The LISTEN action does not change the state of the world. The LEFT and RIGHT actions cause a transition to each state sl and sr with probability 0.5 each (opening a door resets the tiger to a 50-50 chance of being behind either door). When the world is in state sl, the LISTEN action results in observation TL with probability 0.75 (for hearing the tiger behind the correct door) and observation TR with probability 0.25 (for hearing the tiger behind the incorrect door) and vice versa for when the world is in state sr.

* 1. \*see attachment\*
  2. The new probability that the tiger is behind the right door is 0.1875.
     + Probability is 0.5625 if you heard the tiger on the right instead of the on the left.
     + Probability is 0.5 if you performed the RIGHT action instead of LISTEN.
     + Probability is 0.0833 if the initial belief was 0.3333 instead of 0.75 and you heard the tiger on the left.
  3. Slopes of the action values become flatter which shifts the proportions of the belief state on the x-axis and the LISTEN action became more dominate.

Programming Assignment

Step 1

1.  b)

|  |  |  |
| --- | --- | --- |
| States | Actions | Rewards |
| 1 | north (1) | 0 |
| 2 | east (2) | 0 |
| 3 | south (3) | 0 |
| 4 | west (4) | 0 |
| 5 |  | 0 |
| 6 |  | 0 |
| 7 |  | 0 |
| 8 |  | 0 |
| 9 |  | 0 |
| 10 |  | 0 |
| 11 |  | 0 |
| 12 |  | 0 |
| 13 |  | 0 |
| 14 |  | 0 |
| 15 |  | 0 |
| 16 |  | 0 |
| 17 |  | 0 |
| 18 |  | 0 |
| 19 |  | 0 |
| 20 |  | 0 |
| 21 |  | -10 |
| 22 |  | 1 |
| 23 |  | 0 |
| 24 |  | 0 |

1. Final reward found by running qNavigate for 100 iterations with navigation noise 0.1 and 0.2.

|  |  |  |
| --- | --- | --- |
| qNavigate | | |
| Noise | Discount | Final Reward |
| 0.1 | 0.9 | 83 |
| 0.2 | 0.9 | 31 |

1. Changing the noise from 0.1 to 0.2 qualitatively decreases the optimal strategy performance resulting in a decrease in the final reward. Even though the policy may be similar in terms of Q values, the intended action as more noise and is less likely to be executed by the robot when following the learned policy of Q values.

Step 2



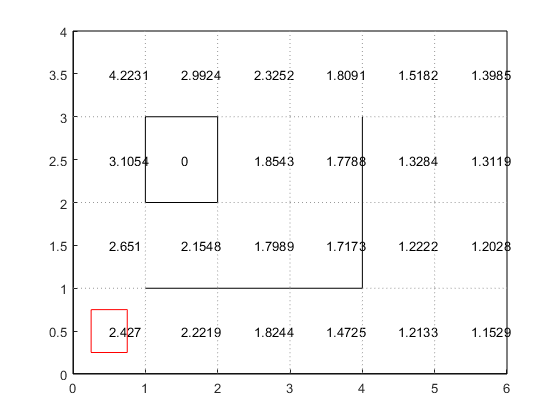
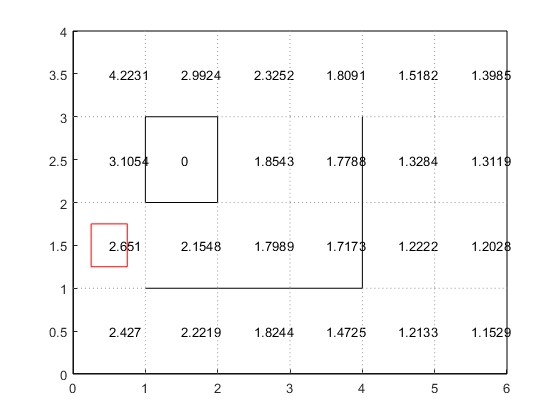
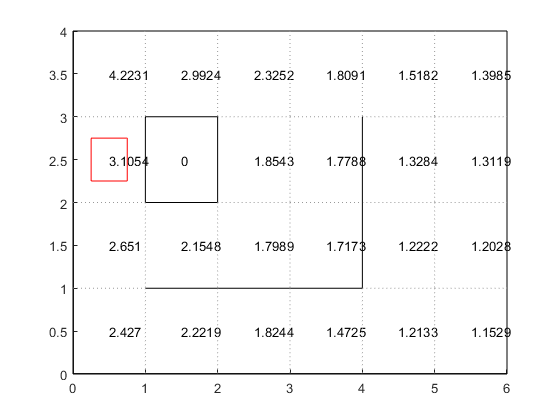
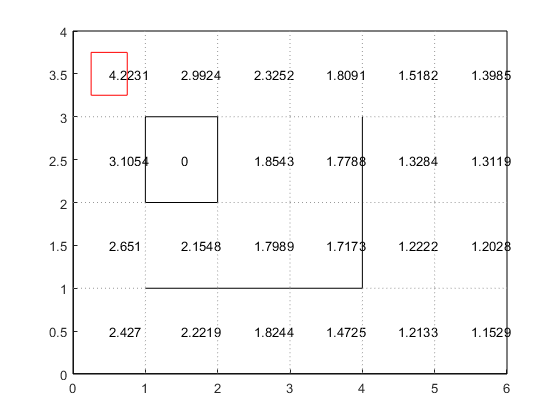
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Robot States | Target States | Actions | Observations | Rewards |
| 1 | 1 | north (1) | 1 | 1 |
| 2 | 2 | east (2) | 0 | 0 |
| 3 | 3 | south (3) |  |  |
| 4 | 4 | west (4) |  |  |
| 5 | 5 |  |  |  |
| 6 | 6 |  |  |  |
| 7 | 7 |  |  |  |
| 8 | 8 |  |  |  |
| 9 | 9 |  |  |  |
| 10 | 10 |  |  |  |
| 11 | 11 |  |  |  |
| 12 | 12 |  |  |  |
| 13 | 13 |  |  |  |
| 14 | 14 |  |  |  |
| 15 | 15 |  |  |  |
| 16 | 16 |  |  |  |
| 17 | 17 |  |  |  |
| 18 | 18 |  |  |  |
| 19 | 19 |  |  |  |
| 20 | 20 |  |  |  |
| 21 | 21 |  |  |  |
| 22 | 22 |  |  |  |
| 23 | 23 |  |  |  |
| 24 | 24 |  |  |  |

1. I created a transition matrix for the belief space where the target moves perfectly and each action at a given state has a probability 0.25. Initially, the belief spaces were equal and normalized while excluding the obstacle of state 6. The belief was updated by multiplying the transition matrix by the belief space matrix. For every action the robot made, an observation was taken at the current state. If the reward was present at the current state of the robot, then an observation equal to 1 was taken at the state and all the other states were zeroed out (this was the updated belief space for finding the target reward). If the reward was not present at the current state of the robot, then an observation equal 0 was taken at the state and all the other states’ beliefs were normalized with the new probability 0 at the current state of the robot.
2. Final reward achieved after 100 iterations

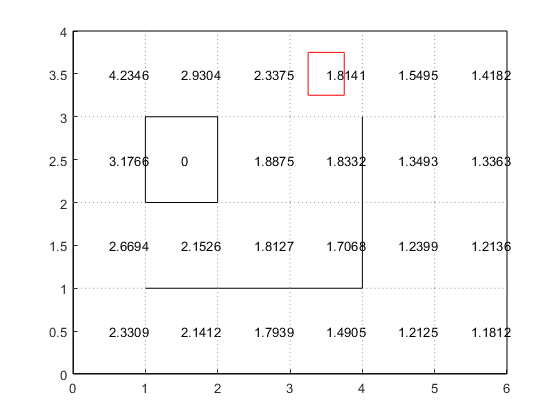
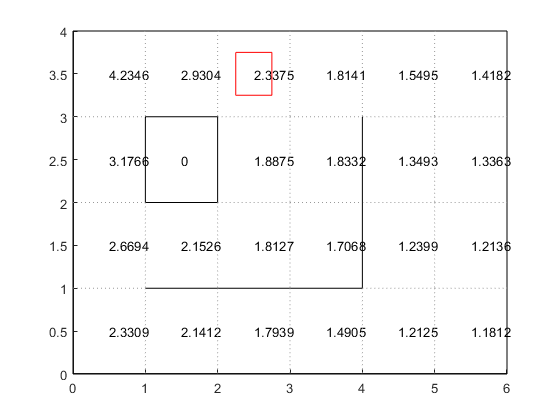
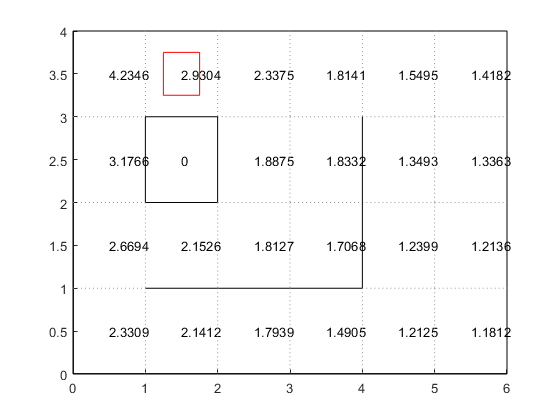
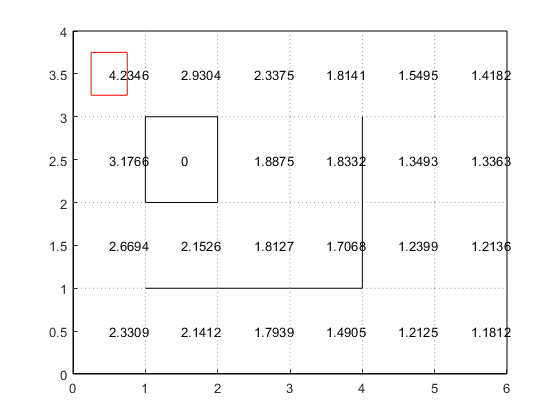
|  |  |  |
| --- | --- | --- |
| mostLikelySearch | | |
| Noise | Discount | Final Reward |
| 0 | 0.9 | 11 |
| 0.3 | 0.9 | 4 |

|  |  |  |
| --- | --- | --- |
| qmdpSearch | | |
| Noise | Discount | Final Reward |
| 0 | 0.9 | 5 |
| 0.3 | 0.9 | 2 |

1. mostLikelySearch



qmdpSearch



Discussion

For the most likely heuristic, there next intended action of the robot is determined by the maximum belief state hence by the algorithm is called the “most likely” search because the robot picks the belief state it thinks the reward is most likely in. So, the POMDP search policy is based on the state-action that uses the belief state the robot thinks the reward is most likely in. On the other hand, the QMDP heuristic takes the belief state of a given state and multiples it with the Q values of the given state and action, then sums over all the belief-Q-values multiplication and finds the maximum to determine that intended action should be taken. The belief and Q-values becomes the learned policy of the POMDP search. They are similar in that to essentially act optimally, the POMDP becomes observable after the next action that is chosen based on the given belief state. Overall, they learn a policy based the belief state or both the Q values and belief state where the most likely heuristic picks the most likely belief to choose an action and develop the policy and the QMDP uses the belief states to give weights to the Q value tables to choose an action given a belief state.

Another scenario that could be formulated as a POMDP is object-based search in a cluttered refrigerator (already formulated as a POMDP). I believe the QMDP would perform slight better than the most likely heuristic because the target object occluded is stationary and by using updated belief states correlated to occlusion ratio of the target object is beneficial for weight Q value tables and determining the optimal policy to find the target object.