CSE 6369 - *Reinforcement Learning*

Homework 2- Spring 2024

Due Date: Apr. 15 2024, 3:30pm

Simple Navigation Problem

Consider the following discretized (Grid World) navigation problem where an agent is moving on a $n \times m$ grid surrounded by a wall with the actions "forward", "backward", "turn left", and "turn right" (thus the state representation has to include the orientation of the agent). The agent has reliable actions that move the agent in the given direction except if the cell in that direction is occupied by an obstacle or a wall. The environment contains a number of obstacles as well as a goal, both of which are unknown to the agent but have fixed locations. If the agent's action would have it enter an obstacle cell (or leave the grid), the agent will stay in place (i.e. obstacles can not be traversed) and will receive a reward of -10. If the agent reaches the goal, the trial ends and the agent receives a reward of $+100$. All other rewards are 0 (i.e. there is no explicit cost to taking an action).
The objective of the agent is to learn a policy that allows the agent to reach the goal location from a random start location as efficiently as possible (i.e. with the highest reward possible), given that it initially does not know the transition probabilities and rewards.

Fully Observable Navigation

1. Assume that the learning agent has the ability to observe its position and orientation completely. Under these conditions the system is Markov in terms of its location and orientation, given fixed obstacles and goal locations.

    a) Design the state space and action space of the underlying MDP for this problem for a 10×20 world.

    b) Implement a simulation for this environment. The simulation should should allow to configure up to 5 obstacles and 1 goal location.

    c) Implement a Q-learner for this fully observable problem and show the learning curve in terms of the reward per trial for 4 different environment (i.e. obstacle and goal) configurations. Each environment should have 5 obstacles and 1 goal in different locations.

Partial Observability Using Linear Function Approximators

2. Assume that the learning agent can no longer observe its location in the environment but rather can only determine when it is hitting a wall or and obstacle and when it is sitting on top of the goal.

    a) Expand the MDP that you designed in part 1 to a POMDP by designing an observation function and an observation probability function.

b) Expand your simulator to also generate observations.

c) Implement a Q_{MDP} agent for this problem (you can use your learner from part 1 do determine the $q(s, a)$ parameters) and show its performance on the same 4 environments you used in parts 1c). Discuss what behavior the learner exhibits.

d) Implement a replicated Q-learning agent for this problem and train it again on the same 4 environments you used in parts 1c). Show the learning curves in terms of the reward per trial for the 4 environments and discuss the performance you observe.

e) Implement a linear Q-learning agent for this problem and train it again on the same 4 environments you used in parts 1c). Show the learning curves in terms of the reward per trial for the 4 environments and discuss the performance you observe. Compare the performances of the 3 POMDP learners