15-281 Archives — Practice Midterm 2a

INSTRUCTIONS

- Exam length: 80 minutes
- You are permitted to have one handwritten page of notes, double-sided
- No calculators or other electronic devices allowed

Name	
Andrew ID	

Q1. [24 pts] Package Delivery Scheduling

Robbie the robot is tasked with picking up and dropping off items in an office hallway shown below. As AI experts, you are asked to plan its daily routes. You are given a list of packages to deliver from one location to another as a start state and the goal of delivering all objects.

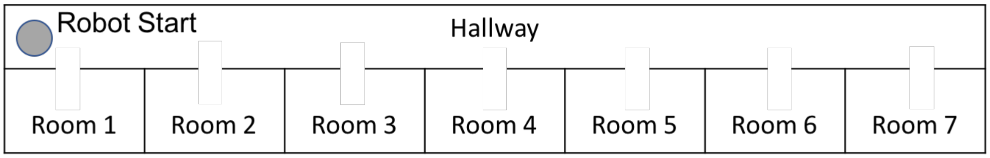


Figure 1: The robot's hallway where it navigates

You choose to implement a classical planning approach. Additionally, the tool you use for solving the classical planning problem has the ability to track the cost of the plan. You implement two operators - `pickup(object)` and `dropoff(object)`.

```
pickup(object):
Preconditions:  [At(room), Task(object,pickroom,droproom), ¬Has(object) & ¬Delivered(object)]
Add:  [Has(object), At(pickroom)]
Delete:  [¬Has(object), At(room)]
Cost += dist(room, pickroom)

dropoff(object):
Preconditions:  [At(room), Task(object, pickroom, droproom), Has(object), ¬Delivered(object)]
Add:  [Delivered(object), At(droproom)]
Delete:  [Task(object,pickroom,droproom), Has(object), ¬Delivered(object), At(room)]
Cost += dist(room, droproom)
```

(a) [12 pts] Suppose you receive delivery requests for a pencil and pen in rooms noted below. You create the following start state:
`At(Room1) & Task(pencil,Room1,Room5) & Task(pen,Room4,Room6) & ¬Has(pencil) & ¬Has(pen) & ¬Delivered(pencil) & ¬Delivered(pen)`

Write the goal state.

Goal:
`Delivered(pencil) & Delivered(pen)`

Write the shortest cost plan to achieve the goals, assuming the distance function subtracts the room numbers (i.e., dist(Room2,Room6) = 4).

Plan:
pickup(pencil), pickup(pen), dropoff(pencil), dropoff(pen)

What is the cost of the plan? Show your work.

cost of pickup(pencil) from Room1 = 0
cost of pickup(pen) from Room4 = 3
cost of dropoff(pencil) from Room5 = 1
cost of dropoff(pen) from Room6 = 1
Total cost = 5.

(b) Instead, you decided to use a linear planner.

(i) [6 pts] Is a linear planner sound, complete, and/or shortest-path optimal for this application?

Sound?	● Yes	○ No
Complete?	● Yes	○ No

Optimal? ○ Yes ● No

(ii) [6 pts] Write the plan that would be generated using a linear planner assuming the goals are tested in the order above. Be sure to number the actions so we know what order they would be executed.

Plan:
1) pickup(pencil), 2) dropoff(pencil), 3) pickup(pen), 4) dropoff(pen)

Q2. [34 pts] MDPs/RL

(a) [15 pts] **Multiple Choice.** Select the single best answer for each question. We are given an MDP (S, A, T, γ, R), where R is only a function of the current state s. We are also given an arbitrary policy π.

i) If $f(s) = R(s) + \sum_{s'} \gamma T(s, \pi(s), s') f(s')$, then f computes

○ V^* ○ Q^* ○ π^* ● V^π ○ Q^π ○ None of these

ii) If $g(s) = \max_a \sum_{s'} T(s, a, s')[R(s) + \gamma \max_{a'} Q^*(s', a')]$, then g computes

● V^* ○ Q^* ○ π^* ○ V^π ○ Q^π ○ None of these

iii) If $h(s, a) = \sum_{s'} T(s, \pi(s), s')[R(s) + \gamma h(s', a)]$, then h computes

○ V^* ○ Q^* ○ π^* ○ V^π ○ Q^π ● None of these

iv) Which of the following iterative MDP-solving techniques typically converges in the fewest number of iterations?

○ Value Iteration ○ Asynchronous Value Iteration ● Policy Iteration

v) Which of the following reinforcement learning techniques sometimes diverges?

○ Exact (not approximate) Q-Learning ● Q-Learning with linear function approximations
○ Exact (not approximate) TD-Learning

(b) [6 pts] Consider policy evaluation in a setting where the reward R is a function of s, a, s', instead of just s. Suppose we have n states, s_1 through s_n. Then for any s, we have the following policy evaluation equation:

$$V^\pi(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi(s')].$$

Now, suppose the policy $\pi(s)$ that we are evaluating behaves as follows. At each timestep, it picks one out of m different "local" policies $\pi_1(s), \pi_2(s), ..., \pi_m(s)$ with corresponding probabilities $p_1, p_2, ..., p_m$ of being picked. (Note that $p_1 + p_2 + ... + p_m = 1$.) For this timestep, it acts according to the chosen policy. Write down the policy evaluation equation for $V^\pi(s)$ in terms of the local policies $\pi_1(s), \pi_2(s), ..., \pi_m(s)$.

Answer:

$$V^\pi(s) = \sum_{i=1}^{m} p_i \sum_{s'} T(s, \pi_i(s), s')[R(s, \pi_i(s), s') + \gamma V^\pi(s')]$$

(c) [13 pts] Baymax has found the unique optimal policy for a specific healthcare reinforcement learning / MDP problem. He has given this optimal policy to you. Neither of you have access to the MDP reward or transition functions.

Specify what you should set the following values to in order for your epsilon greedy q-learning agent to always act according to this optimal policy. (Not approximate q-learning.) Baymax's settings have been given to you. Briefly explain each.

Learning rate (Baymax $\alpha = 0.5$):

$\alpha =$
0

Explain:
Stop changing Q values

Epsilon (Baymax $\epsilon = 0.5$):

$\epsilon =$
0

Explain:
Stop exploring

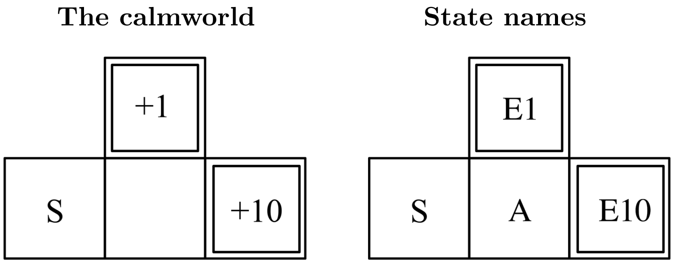
What specifically would you need to do to confirm that Baymax's policy is indeed optimal?

Answer:
Run Q-learning for a sufficiently long time with exploration (so that all states are visited sufficient times) and see if the policies match.

Q3. [16 pts] Infinite Time to Study

Pacman lives in a calm gridworld. S is the start state and double-squares are exit states. In exits, the only action available is exit, which earns the associated reward and transitions to a terminal state X (not shown). In normal states, the actions are to move to neighboring squares (for example, S has the single action →) and they always succeed. There is no living reward, so all non-exit actions have reward 0.

Throughout the problem the discount $\gamma = 1$.



The Q-learning update equation is $Q'(s,a) = (1-\alpha)Q(s,a) + \alpha[R(s,a,s') + \max_{a'} Q(s',a')]$. However, this problem can be solved without manually computing any Q-value updates.

(a) [2 pts] What are the optimal values of S and A?

$V^*(S) =$
10

$V^*(A) =$
10. In a deterministic undiscounted ($\gamma = 1$) MDP, the optimal value is the maximum return from the state.

Pacman doesn't know the details of this gridworld so he does Q-learning with a learning rate of 0.5 and all Q-values initialized to 0 to figure it out.

Consider the following sequence of transitions in the calmworld:

s	a	s'	r
S	→	A	0
A	↑	E1	0
E1	exit	X	1
S	→	A	0
A	→	E10	0
E10	exit	X	10

(b) [2 pts] Circle the Q-values that are non-zero after these episodes.

$Q(S,\rightarrow)$ $Q(A,\uparrow)$ $Q(A,\rightarrow)$ [✓ $Q(E1, exit)$] [✓ $Q(E10, exit)$]

Q-values are only updated when a transition is experienced. $Q(E1, exit), Q(E10, exit)$ are updated to the reward earned, but the other states were updated when all the Qs were still zero.

(c) [2 pts] What do the Q-values converge to if these episodes are repeated infinitely with a constant learning rate of 0.5? Write none if they do not converge. The MDP is undiscounted and deterministic, so Q-learning converges even though the learning rate is constant. With infinite visits the Q-values will converge to the true values.

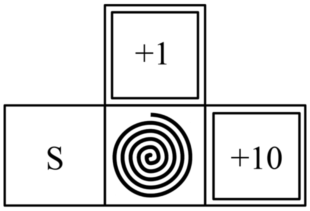
$Q(S,\rightarrow) =$
10. $Q(S,\rightarrow)$ is the only state-action for S, so it converges to the optimal value $V^*(S)$.

$Q(A,\leftarrow) =$
0. The episode $A,\leftarrow$ is never experienced so it is unchanged after initialization.

$Q(A,\uparrow) =$
1. The only return possible after $A,\uparrow$ is 1.

(Q-learning details reminder: assume $\alpha = 0.5$ and the Q-values are initialized to 0.)

It's vortex season in the gridworld. In the vortex state the only action is escape, which delivers Pacman to a neighboring state uniformly at random.



The vortexworld

(d) [2 pts] What are the optimal values of S and A in the vortex gridworld?

The optimal value is the mean of the end returns 1 and 10 because the exit states have equal probability. The value of S is the same as A since the discount $\gamma = 1$ and the transition $S, \rightarrow, A$ is deterministic. The transition $A, escape, S$ has no impact on the value because the MDP is undiscounted / infinite horizon.

$V^*(S) =$
5.5

$V^*(A) =$
5.5

Consider the following sequences of transitions in the vortexworld:

S1

s	a	s'	r
S	→	A	0
A	escape	E1	0
E1	exit	X	1
S	→	A	0
A	escape	E10	0
E10	exit	X	10

S2

s	a	s'	r
S	→	A	0
A	escape	E1	0
E1	exit	X	1
S	→	A	0
A	escape	E10	0
E10	exit	X	10
S	→	A	0
A	escape	E10	0
E10	exit	X	10

(e) [2 pts] What do the Q-values converge to if the sequence S1 is repeated infinitely with appropriately decreasing learning rate? Write never if they do not converge.

$Q^{S1}(S, \rightarrow) =$
5.5

$Q^{S1}(A, escape) =$
5.5

The conditions for convergence are satisfied and the Q-values converge to the expected return. The expectation of returns is $\frac{1}{2} \times 1 + \frac{1}{2} \times 10$.

(f) [2 pts] What if the sequence S2 is repeated instead?

$Q^{S2}(S, \rightarrow) =$
7

$Q^{S2}(A, escape) =$
7

(g) [2 pts] Which is the true optimum $Q^*(S, \rightarrow)$ in the vortex gridworld? Circle the answer.

[✓ $Q^{S1}(S, \rightarrow)$] $Q^{S2}(S, \rightarrow)$ other

The sequence $S1$ has the same distribution of returns as the true distribution, even though all of the possible transitions are not experienced.

(h) [2 pts] Q-learning with constant $\alpha = 1$ and visiting state-actions infinitely often converges

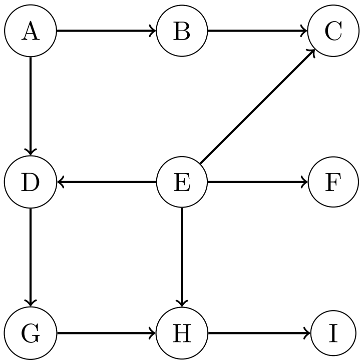
[✓ in calmworld]   in vortexworld   in neither world

For learning rate $\alpha = 1$ the Q-learning update sets $Q(s, a)$ to the sample $[R(s, a, s') + \max_{a'} Q(s', a')]$ with no regard for the previous value of $Q(s, a)$.

In deterministic MDPs (like calmworld), even with constant learning rate $\alpha = 1$, Q-learning converges. In fact, this learning rate is optimal for deterministic MDPs in the sense that it converges fastest.

In stochastic MDPs (like vortexworld), with constant learning rate $\alpha = 1$, the $Q(s, a)s$ are always equal to the most recent sample for the state-action (s, a). The $Q(s, a)$s will cycle among the possible samples and never converge.

Q4. [21 pts] Bayes' Nets: Independence



Given the above Bayes' Net, select all true statements below. ($\emptyset$ means that no variables are observed.)

- ● $\mathbf{A} \perp\!\!\!\perp \mathbf{F} \mid \emptyset$ is guaranteed to be true
- ○ $\mathbf{A} \perp\!\!\!\perp \mathbf{D} \mid \emptyset$ is guaranteed to be true
- ○ $\mathbf{A} \perp\!\!\!\perp \mathbf{I} \mid \mathbf{E}$ is guaranteed to be true
- ○ $\mathbf{B} \perp\!\!\!\perp \mathbf{H} \mid \mathbf{G}$ is guaranteed to be true
- ● $\mathbf{B} \perp\!\!\!\perp \mathbf{E} \mid \mathbf{F}$ is guaranteed to be true
- ○ $\mathbf{C} \perp\!\!\!\perp \mathbf{G} \mid \mathbf{A}, \mathbf{I}$ is guaranteed to be true
- ○ $\mathbf{D} \perp\!\!\!\perp \mathbf{H} \mid \mathbf{G}$ is guaranteed to be true

d-Separation rules (Bayes' ball).

THIS PAGE INTENTIONALLY LEFT BLANK