Assignment 1 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Which among the following are features of a reinforcement learning solution to a learning problem?

   (a) trial and error approach to learning
   (b) exploration versus exploitation dilemma
   (c) learning based on rewards
   (d) absence of any feedback or supervision

   Sol. (a), (b) & (c)
   The last option is not correct, since rewards act as feedback/supervision for RL agents.

2. Modelling a living organism as an RL agent, the environment encompasses

   (a) everything external to the organism
   (b) some portions internal to the organism as well

   Sol. (b)
   In living organisms, reward signals are generated within the brain. In the idealised RL framework this is part of the environment.

3. It was mentioned in the lecture that reinforcement learning solutions have to be able to handle delayed rewards. What exactly is/are the problems that arises due to delayed rewards?

   (a) problem in storing rewards
   (b) problem in assigning rewards to the actions that caused them
   (c) problem in mapping states to the rewards that can be achieved from those states

   Sol. (b) & (c)
   Different techniques can be used to maintain reward information, so the first option is not really an issue. The next two options do pose challenges to an RL agent. Remember that rewards, in general, depend both on the chosen action and the state in which that action was taken.

4. In the tic-tac-toe problem, it can be observed that many board positions (states) are symmetric. Assuming an imperfect opponent with a stationary policy/strategy who is also taking advantage of symmetries in the board positions, which of the following statements are true, if we take advantage of symmetries by maintaining a single (probability) value for all symmetric states?

(a) time taken to learn the values reduces

(b) space required to store the values reduces

(c) the final policy learned in this approach is better than the policy learned when not taking advantage of symmetries

Sol. (a) & (b)
Option (b) is obviously true. Option (a) is true because for each value, the experience generated from all corresponding symmetric states are being used to update that value, and hence we can expect faster convergence (when compared to the case where symmetries are being ignored). Option (c) is false because in both cases, given an opponent with a stationary policy, we expect to arrive at the same policy, the difference is only in the time it takes.

5. Considering again the question of taking advantage of symmetric board positions, is it true, that symmetrically equivalent positions should necessarily have the same value?

(a) no

(b) yes

Sol. (a)
Consider the case where the opponent does not make use of symmetries. Assuming an imperfect opponent, we can consider two states a and b which are symmetrically equivalent. Now, assume that the actions taken by the opponent when in these two states is not symmetrically equivalent. For example, in state a, the opponent may play a sound move, but in state b, the opponent may make a mistake that can be exploited by us. Thus, to be able to best respond to the opponents imperfection as in the above case, it would be beneficial for us to maintain separate values for the states a and b even though they are symmetric.

6. This question is related to the one discussed in class. Recall the temporal difference learning approach to the tic-tac-toe problem. Suppose that the probability of winning at a particular state is 0.6, the max probability value in the next set of states is 0.8, and based on our exploration policy, we choose a next state which has probability value 0.4. Should you backup the current state's probability value based on this choice of next state (i.e., move probability value 0.6 closer to 0.4) or not, given that the agent never stops exploring (i.e., the agent always makes an exploratory move some fraction of the time)?

(a) backup the value

(b) do not backup the value

Sol. (a)
There are valid arguments for both backing up, as well as not backing up values on exploration steps. The argument in favour of backing up is that by backing up values, even on exploration steps, the value function (i.e., the function specifying the value at each state - here, the probability of winning) actually represents the behaviour you are following (because according to your behaviour, you do explore some fraction of the times). The argument against backing up is that exploration is performed only for learning purposes, and that if we were to actually go by our learned policy, say when the agent is deployed in a competition, we will perform only exploitative steps to maximise reward. In this sense, we should only backup values on greedy moves.

Now, in the question, we are given that the agent never stops exploring. So, if we choose not to backup the values during exploration steps, we are actually learning a value function (and hence, policy), which is different from the actual policy the agent will follow. Thus, it makes sense to backup the values during all steps.

7. Suppose we want an RL agent to learn to play the game of golf. For training purposes, we make use of a golf simulator program. Assume that the original reward distribution gives a reward of +10 when the golf ball is hit into the hole and -1 for all other transitions. To aide the agents learning process, we propose to give an additional reward of +3 whenever the ball is within a 1 metre radius of the hole. Is this additional reward a good idea or not? Why?

   (a) yes, the additional reward will help speed-up learning
   (b) yes, getting the ball to within a metre of the hole is like a sub-goal and hence, should be rewarded
   (c) no, the additional reward may actually hinder learning
   (d) no, it violates the idea that a goal must be outside the agents direct control.

   Sol. (c)
   In this specific case, the additional reward will be detrimental to the learning process, as the agent will learn to accumulate rewards by keeping the ball within the 1 metre radius circle and not actually hitting the ball in the hole. This example highlights the importance of being cautious in setting up the reward function in order to prevent unintended consequences.

8. Suppose that we are given a 10 armed bandit problem and are also told that the rewards for each arm are deterministic. How many times should each arm be tried before we can identify the best arm with 95% probability?

   (a) 1
   (b) 2
   (c) 95
   (d) none of the above

   Sol. (a)
   Since we are told that the rewards are deterministic, it is sufficient to pull each arm once and then subsequently pick the arm for which we observed the maximum reward. Note that this is only possible because the rewards were deterministic (i.e., non-stochastic) and also because we knew before hand that the rewards were deterministic.

9. Suppose that you have been given a number of different drug formulations to treat a particular disease and you job is to identify one among them that best meets certain criteria with regards to its efficacy in treating the disease. Before you run the experiments, you need to provision for the samples that would be required. Treating this as a multi-armed bandit problem, which kind of solution method would you prefer for identifying the best option?

   (a) asymptotic correctness
   (b) regret optimality
   (c) PAC optimality

Sol. (c)
A PAC optimal solution allows us estimate the number of samples that are required to achieve the required level of performance for the chosen arm.

10. Suppose we have a 10-armed bandit problem where the rewards for each of the 10 arms is deterministic and in the range (0, 10). Which among the following methods will allow us to accumulate maximum reward in the long term?

    (a) ϵ-greedy with $\epsilon = 0.1$

    (b) ϵ-greedy with $\epsilon = 0.01$

    (c) greedy with initial reward estimates set to 0

    (d) greedy with initial reward estimates set to 10

    Sol. (d)
    Since the rewards are deterministic, we only need to select each arm once to identify an optimal arm. The greedy method with initial reward higher than all possible rewards ensures that each arm is selected at least once, since on selecting any arm, the resultant reward estimate will necessarily be lower than the initial estimates of the other arms. Once each arm has been selected, the greedy method will settle on the arm with the maximum reward.

Assignment 2 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. In the UCB 1 algorithm, how is it ensured that the confidence bounds around the estimated values for each arm shrink?

   (a) the intervals do not shrink

   (b) the confidence intervals of each arm are made to shrink after every arm selection

   (c) in calculating the bounds we use the number of times an arm is selected in the denominator

   (d) on selecting an arm its interval is shrunk by half every time

   Sol. (c)
   The number of times arm j is selected, denoted by n_j appears in the denominator of the upper confidence bound for each arm. As n_j increases, the bound decreases.

2. After 12 iterations of the UCB 1 algorithm applied on a 4-arm bandit problem, we have $n_1 = 3$, $n_2 = 4$, $n_3 = 3$, $n_4 = 2$ and $Q_{12}(1) = 0.55$, $Q_{12}(2) = 0.63$, $Q_{12}(3) = 0.61$, $Q_{12}(4) = 0.40$. Which arm should be played next?

   (a) 1

   (b) 2

   (c) 3

   (d) 4

   Sol. (d)
   The next action, A_{13}, will be the action with the maximum upper confidence bound among the four arms. Calculating these values, we have

$$Q_{12}(1) + \sqrt{\frac{2ln12}{n_1}} = 0.55 + \sqrt{\frac{2ln12}{3}} = 1.837$$

$$Q_{12}(2) + \sqrt{\frac{2ln12}{n_2}} = 0.63 + \sqrt{\frac{2ln12}{4}} = 1.745$$

$$Q_{12}(3) + \sqrt{\frac{2ln12}{n_3}} = 0.61 + \sqrt{\frac{2ln12}{3}} = 1.897$$

$$Q_{12}(4) + \sqrt{\frac{2ln12}{n_4}} = 0.40 + \sqrt{\frac{2ln12}{2}} = 1.976$$

   Clearly, arm 4 has the highest upper confidence bound and hence will be selected by the UCB 1 algorithm.

3. In the proof of the UCB 1 algorithm, we had the following expression:

$$\{min_{0<s<m}\left(Q_s(a^*) + C_{m-1,T_{a^*}(s)}\right) \leq max_{l \leq s_i \leq m}(Q_{s_i}(i) + C_{m-1,T_i(s_i)})\}$$

What does this expression stand for?

(a) the number of instances where the minimum value among the upper confidence bound values of an optimal arm up till the m^{th} time step is less than or equal to the maximum value among the upper confidence bound values of the i^{th} arm between the times steps l and m

(b) the condition that the minimum value among the upper confidence bound values of an optimal arm up till the m^{th} time step is less than or equal to the maximum value among the upper confidence bound values of the i^{th} arm between the times steps l and m

(c) the condition that there exists an upper confidence bound value of an optimal arm that is smaller than the corresponding upper confidence value of the chosen arm i

(d) the number of instances where the upper confidence bound value of an optimal arm is smaller than the corresponding upper confidence value of the chosen arm i

Sol. (b)

4. In the initial stage of the UCB 1 algorithm, each arm is selected one time. Thereafter, the arm that is selected depends on the arm with the maximum upper confidence bound. Suppose that one of the arms has not been selected after its initial selection in the first stage of the algorithm. What happens to the upper confidence bound of this arm?

(a) it increases

(b) it decreases

(c) it remains constant until it is selected

Sol. (a)
Presence of n, the number of time steps in the numerator ensures that at each step the upper confidence bounds of each unselected arm increases.

5. Suppose that we apply the naive PAC algorithm on a 10-arm bandit problem where the required (ϵ, δ) values are: $\epsilon = 0.5$ and $\delta = 0.05$. How many iterations would the algorithm take to output an arm selection? (Note: each arm is sampled $\frac{2}{\epsilon^2} ln\left(\frac{2k}{\delta}\right)$ times).

(a) 48

(b) 21

(c) 210

(d) 480

Sol. (d)
We know that in the naive algorithm, each arm is sampled

$$l = \frac{2}{\epsilon^2} ln\left(\frac{2k}{\delta}\right)$$

times, where k is the number of arms. Thus, we have

$$l = \frac{2}{0.5^2} ln\left(\frac{2 * 10}{0.05}\right) = 8ln(400) = 47.93 = 48$$

Thus, the total number of iterations = 10 * l = 480.

6. In the proof of the Naive PAC algorithm, we have the expression

$$P(Q(a') > Q(a^*)) \leq P(Q(a') > q_*(a') + \epsilon/2 \text{ OR } Q(a^*) < q_*(a^*) - \epsilon/2)$$

Why is the LHS "less than or equal to" the RHS here?

(a) because the occurrence of at least one of the events on the RHS is a necessary but not sufficient condition for the event on the LHS to occur

(b) because the event on the LHS does not require that the events on the RHS occur

(c) because the event on the LHS requires that both the events on the RHS occur

Sol. (a)
It should be clear that for $Q(a')$ to be greater than $Q(a^*)$, at least one of the events on the RHS must occur, but even then, $Q(a')$ may not be greater than $Q(a^*)$. This implies that the probability of the LHS is less than or equal to the probability of the RHS.

7. Suppose we have a 64-arm bandit problem. We apply the median elimination algorithm where the (ϵ, δ) values are: $\epsilon = 0.5$ and $\delta = 0.01$. How many times do we sample arms in the first round? (Note: in each round, each arm is sampled $\frac{1}{\epsilon_l^2/2} ln\left(\frac{3}{\delta_l}\right)$ times).

(a) 52404

(b) 818

(c) 52416

(d) 819

Sol. (c)
We know that in the median elimination algorithm, each arm is sampled

$$\frac{1}{\epsilon_l^2/2} ln\left(\frac{3}{\delta_l}\right)$$

times. In the first round, $\epsilon_1 = \epsilon/4$ and $\delta_1 = \delta/2$. Substituting the values of ϵ and δ, we have number of samples

$$\frac{32}{0.5^2} ln\left(\frac{6}{0.01}\right) = 818.81 = 819$$

This is the number of times each arm is pulled in the first round. Thus, the total number of sampled arms in the first round is 64 * 819 = 52416.

8. Continuing with the previous example, what is the number of samples in the third round? Also, what is the total number of rounds required to identify the (ϵ, δ)-optimal arm?

(a) 50384, 6

(b) 50384, 7
(c) 50378, 6
(d) 201514, 7

Sol. (a)
From the initial and update conditions, we know that $\epsilon_l = (\frac{3}{4})^{l-1}\frac{\epsilon}{4}$ and $\delta_l = \frac{\delta}{2^l}$. Now, in any given round, each arm is sampled

$$\frac{1}{\epsilon_l^2/2} ln\left(\frac{3}{\delta_l}\right) = \frac{2}{((\frac{3}{4})^{l-1}\frac{\epsilon}{4})^2} ln\left(\frac{3*2^l}{\delta}\right)$$

times. Substituting the values of ϵ and δ, with $l = 3$, we have

$$\frac{2}{((\frac{3}{4})^2\frac{0.5}{4})^2} ln\left(\frac{3*2^3}{0.01}\right) = 3148.65 = 3149$$

This is the number of times each arm is pulled in the third round. The number of arms in the third round is 16 (32 of 64 eliminated at the end of the first round and 16 eliminated at the end of the second round). Thus, the total number of sampled arms in the third round is 16 * 3149 = 50384.

The number of rounds required to identify the (ϵ, δ)-optimal arm is $log_2(64) = 6$.

9. Consider a bandit problem in which there is a single optimal arm, a^*. Applying the median elimination algorithm to this problem, is it possible that arm a^* is eliminated in the first round? In case it is possible, does this mean that the algorithm cannot output an arm that is ϵ-close to a^*?

(a) no
(b) yes, no
(c) yes, yes

Sol. (b)
It is possible that after the first round of sampling, the estimated payoff of the optimal arm is below the median estimated payoff and hence is eliminated. However, the algorithm can still output an arm that is ϵ-close to a^*. This is because in each round the probability that the best arm is more than ϵ_l away from the best arm in the previous round is less than δ_l. Thus, even if a^* has been eliminated in the first round, it is highly probable (since δ and consequently δ_l are typically small) that an arm that is ϵ_l-close to a^* remains in the second round. Also, the use of ϵ_l and δ_l in each round (with each of these values getting smaller as the rounds progress) implies that over all rounds, the probability of eliminating all arms that are ϵ-close to a^* is less than δ, which is again, typically a small value. Hence, even if the best arm is eliminated in the first round, it is still possible that the algorithm outputs an arm that is ϵ-close to a^*.

10. We know that there is an exploration/exploitation dilemma in reinforcement learning problems. Considering the Thompson sampling algorithm for solving bandit problems, which step of the algorithm ensures that we perform exploration?

(a) initialisation of each arm's reward distribution

(b) updating the reward distribution based on observing actual reward for an arm

(c) sampling the expected payoff of each arm from the corresponding reward distribution

(d) identifying the correct arm given the set of expected payoffs for each arm sampled from each arm's reward distribution

Sol. (c)
At each step, we sample the expected payoffs of each arm from their corresponding reward distributions. This introduces an element of exploration since even an arm for which the mean of the distribution is less than the mean of the distribution of another arm can be selected by the algorithm if the expected payoff sampled for the former arm is larger than the expected payoff sampled for the latter arm.

Assignment 3 (Sol.)
Reinforcement Learning
Prof. B. Ravindran

1. In solving a multi-arm bandit problem using the policy gradient method, are we assured of converging to the optimal solution?

   (a) no
   (b) yes

   Sol. (a)
   Depending upon the properties of the function whose gradient is being ascended, the policy gradient approach may converge to a local optimum.

2. In many supervised machine learning algorithms, such as neural networks, we rely on the gradient descent technique. However, in the policy gradient approach to bandit problems, we made use of gradient ascent. This discrepancy can mainly be attributed to the differences in

   (a) the objectives of the learning tasks
   (b) the parameters of the functions whose gradient are being calculated
   (c) the nature of the feedback received by the algorithms

   Sol. (c)
   The feedback in most supervised learning algorithms is an error signal which we wish to minimise. Hence, we would look to perform gradient descent. In policy gradient, we are trying to maximise the reward signal, hence gradient ascent.

3. Consider a bandit problem in which the parameters on which the policy depends are the preferences of the actions and the action selection probabilities are determined by the softmax relationship as $\pi(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$, where k is the total number of actions and θ_i is the preference value of action a_i. Derive the parameter update conditions according to the REINFORCE procedure considering the above described parameters and where the baseline is the reference reward defined as the average of the rewards received for all arms.

   (a) $\Delta\theta_i = \alpha(R - b)(1 - \pi(a_i; \theta))$
   (b) $\Delta\theta_i = \alpha(R - b)\frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$
   (c) $\Delta\theta_i = \alpha(R - b)(\pi(a_i; \theta) - 1)$
   (d) $\Delta\theta_i = \alpha(R - b)\frac{1-e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$

Sol. (a)
According to the REINFORCE method, at each step, we update the parameter as

$$\Delta\theta_n = \alpha(R_n - b_n)\frac{\partial ln\pi(a_n;\theta)}{\partial\theta_n}$$

Considering the softmax preferences, the characteristic eligibility is

$$\frac{\partial ln\pi(a_n;\theta)}{\partial\theta_n} = \frac{\partial}{\partial\theta_n} ln\frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}} = \frac{\partial}{\partial\theta_n}(\theta_n - ln(\sum_{j=1}^{k} e^{\theta_j})) = 1 - \pi(a_i;\theta)$$

Considering the baseline, b as the average of all rewards seen so far, and α as the step size parameter, the update conditions are

$$\Delta\theta_i = \alpha(R - b)(1 - \pi(a_i;\theta))$$

4. Repeat the above problem for the case where the parameters are the mean and variance of the normal distribution according to which the actions are selected and the baseline is zero.

   (a) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n - \mu_n}{\sigma_n}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2}{\sigma_n^2} - 1\right\}$
   (b) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n - \mu_n}{\sigma_n^2}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2 - \sigma_n^2}{\sigma_n^2}\right\}$
   (c) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n - \mu_n}{\sigma_n^2}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2 - \sigma_n^2}{\sigma_n^3}\right\}$
   (d) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n - \mu_n}{\sigma_n}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2 - \sigma}{\sigma_n^2}\right\}$

   Sol. (c)
   Assuming parameters to be μ and σ, we have policy,

$$\pi(a;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(a-\mu)^2}{2\sigma}}$$

For the mean:

$$\frac{\partial \ln \pi(a_n;\mu_n,\sigma_n)}{\partial\mu_n} = \frac{\partial}{\partial\mu_n}\left\{-\frac{(a_n-\mu_n)^2}{2\sigma_n^2}\right\} = \frac{a_n - \mu_n}{\sigma_n^2}$$

For the variance:

$$\frac{\partial \ln \pi(a_n;\mu_n,\sigma_n)}{\partial\sigma_n} = \frac{\partial}{\partial\sigma_n}\left\{-\ln(\sqrt{2\pi\sigma_n})\right\} + \frac{\partial}{\partial\sigma_n}\left\{-\frac{(a_n-\mu_n)^2}{2\sigma_n^2}\right\}$$

Solving, we have:

$$\frac{\partial \ln \pi(a_n;\mu_n,\sigma_n)}{\partial\sigma_n} = -\frac{1}{\sigma_n} + \frac{(a_n-\mu_n)^2}{\sigma_n^3} = \frac{1}{\sigma_n}\{\left(\frac{a_n-\mu_n}{\sigma_n}\right)^2 - 1\}$$

Thus, the updates are:

$$\mu_{n+1} = \mu_n + \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n^2}\right)$$

$$\sigma_{n+1} = \sigma_n + \alpha_n r_n \frac{1}{\sigma_n}\left\{\left(\frac{a_n-\mu_n}{\sigma_n}\right)^2 - 1\right\}$$

5. Which among the following is/are differences between contextual bandits and full RL problems?

   (a) the actions and states in contextual bandits share features, but not in full RL problems
   (b) the actions in contextual bandits do not determine the next state, but typically do in full RL problems
   (c) full RL problems can be modelled as MDPs whereas contextual bandit problems cannot
   (d) no difference

   Sol. (b)
   Option (a) is not true because we can think of representations where actions and states share features. Similarly, option (c) is false because contextual bandit problems can also be modelled as MDPs.

6. Given a stationary policy, is it possible that if the agent is in the same state at two different time steps, it can choose two different actions?

   (a) no
   (b) yes

   Sol. (b)
   A stationary policy does not mean that the policy is deterministic. For example, for state s_1 and actions a_1 and a_2, a stationary policy π may have $\pi(a_1|s_1) = 0.4$ and $\pi(a_2|s_1) = 0.6$. Thus, at different time steps, if the agent is in the same state s_1, it may perform either of the two actions.

7. In class we saw that it is possible to learn via a sequence of stationary policies, i.e., during an episode, the policy does not change, but we move to a different stationary policy before the next episode begins. Does the temporal difference method encountered when discussing the tic-tac-toe example follow this pattern of learning?

   (a) no
   (b) yes

   Sol. (a)
   Recall that within an episode, at each step, after observing a reward, we modify the value function (the probability values in the tic-tac-toe example). Since the policy used to select actions is derived from the value function, in effect we are modifying the policy at each step.

8. We saw the following definition of the action-value function for policy π: $q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$. Suppose that the action selected according to the policy π in state s is a_1. For the same state, will the function be defined for actions other than a_1?

   (a) no
   (b) yes

   Sol. (b)
   The interpretation of the action-value function $q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$ is the expectation of the return given that we take action a in state s at time t and thereafter select actions based on policy π. Thus, the only constraint for the actions is that they must be applicable in state s.

9. Consider a 100x100 grid world domain where the agent starts each episode in the bottom-left corner, and the goal is to reach the top-right corner in the least number of steps. To learn an optimal policy to solve this problem you decide on a reward formulation in which the agent receives a reward of +1 on reaching the goal state and 0 for all other transitions. Suppose you try two variants of this reward formulation, (P_1), where you use discounted returns with $\gamma \in (0, 1)$, and (P_2), where no discounting is used. Which among the following would you expect to observe?

   (a) the same policy is learned in (P_1) and (P_2)

   (b) no learning in (P_1)

   (c) no learning in (P_2)

   (d) policy learned in (P_2) is better than the policy learned in (P_1)

   Sol. (c)
   In (P_2), since there is no discounting, the return for each episode regardless of the number of steps is +1. This prevents the agent from learning a policy which tries to minimise the number of steps to reach the goal state. In (P_1), the discount factor ensures that the longer the agent takes to reach the goal state, the lesser reward it gets. This motivates the agent to find take the shortest path to the goal state.

10. Given an MDP with finite state set, S, and an arbitrary function, f, that maps S to the real numbers, the MDP has a policy π such that $f = V^{\pi}$.

   (a) false

   (b) true

   Sol. (a)
   Consider an MDP where all rewards are 0.

Assignment 4 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. You receive the following letter:

   Dear Friend,

   Some time ago, I bought this old house, but found it to be haunted by ghostly sardonic laughter. As a result it is hardly habitable. There is hope, however, for by actual testing I have found that this haunting is subject to certain laws, obscure but infallible, and that the laughter can be affected by my playing the organ or burning incense. In each minute, the laughter occurs or not, it shows no degree. What it will do during the ensuing minute depends, in the following exact way, on what has been happening during the preceding minute:

   Whenever there is laughter, it will continue in the succeeding minute unless I play the organ, in which case it will stop. But continuing to play the organ does not keep the house quiet. I notice, however, that whenever I burn incense when the house is quiet and do not play the organ it remains quiet for the next minute.

   At this minute of writing, the laughter is going on. Please tell me what manipulations of incense and organ I should make to get that house quiet, and to keep it so.

   Sincerely, At Wits End

   Assume that we make the following decisions in formulating this problem as an MDP:

   State set: $\{L, Q\}$, where L indicates that there is laughter in the room, and Q indicates that the room is quiet.

   Action set: $\{O \wedge I, O \wedge \neg I, \neg O \wedge I, \neg O \wedge \neg I\}$, where O corresponds to playing the organ, and I corresponds to burning incense.

   We consider this as a continuing discounted problem with $\gamma = 0.9$ and we let the reward be +1 on any transition into the silent state, and -1 on any transition into the laughing state.

   Assuming deterministic state transitions and rewards based upon current state and action, which among the following 4-tuples (current state, action, next state, reward) represent correct state transitions and rewards?

   (a) $(L, O \wedge I, Q, +1)$
   (b) $(L, O \wedge \neg I, L, -1)$
   (c) $(L, \neg O \wedge I, Q, +1)$
   (d) $(L, \neg O \wedge \neg I, L, -1)$
   (e) $(Q, O \wedge I, Q, +1)$

(f) $(Q, O \wedge \neg I, L, -1)$

(g) $(Q, \neg O \wedge I, Q, +1)$

(h) $(Q, \neg O \wedge \neg I, L, -1)$

Sol. (a), (d), (f), (g), (h)
We know that if there is laughter and the organ is played, then in the next step laughter will stop. This contradicts option (b). Similarly, option (c) indicates that by burning incense alone laughter can be made to stop, which is incorrect. Option (e) is also not correct because we know that playing the organ when the house is quiet does not result in the house staying quiet.

2. Based on the above problem description, what advice will you give to At Wit's End?

   (a) if there is laughter, play the organ and do not burn incense; if room is quite, play the organ and burn incense

   (b) never play the organ, always burn incense

   (c) always play the organ, never burn incense

   (d) if there is laughter, play the organ; if room is quite, do not play the organ and burn incense

   Sol. (d)

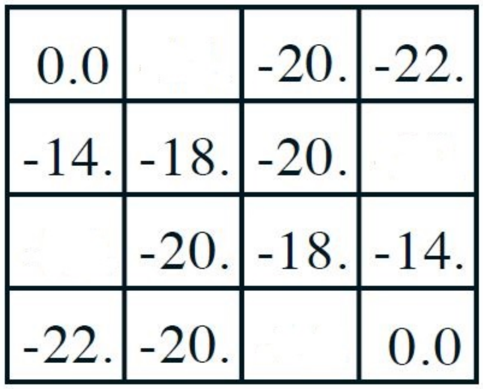
3. If a policy is greedy with respect to its own value function, then it is an optimal policy.

   (a) false

   (b) true

   Sol. (b)
   Consider the value function corresponding to an arbitrary policy π. If we derive a policy that is greedy with respect to this value function, by the policy improvement theorem, we are guaranteed to get a policy which is at least as good as the policy π. This derived policy will be equivalent to the policy π if and only if π is optimal. Hence, if a policy is greedy with respect to its own value function, then it is optimal.

4. Consider a 4 X 4 grid world problem where the goal is to reach either the top left corner or the bottom right corner. The agent can choose from four actions {up, down, left, right} which deterministically cause the corresponding state transitions, except that actions that would take the agent off the grid leave the state unchanged. We model this as an undiscounted, episodic task, where the reward is -1 for all transitions. Suppose that the agent follows the equiprobable random policy. Given below is the partial value function for this problem. Calculate respectively, the missing values in the first and second row? (Hint: the Bellman equation must hold for every state.)



0.0		-20.	-22.
-14.	-18.	-20.	
	-20.	-18.	-14.
-22.	-20.		0.0

(a) -20, -14
(b) -14, -20
(c) -14, -18
(d) -20, -18

Sol. (b)
For the value in the first row, we have

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)[r + v_\pi(s')]$$

$$v_\pi(s) = 0.25 * (-1 + v_\pi(s) - 1 - 21 - 19)$$
$$v_\pi(s) = 0.25 v_\pi(s) - 10.5$$
$$0.75 v_\pi(s) = -10.5$$
$$v_\pi(s) = -14$$

Similarly, for the value in the second row, we have

$$v_\pi(s) = 0.25 * (-23 + v_\pi(s) - 1 - 21 - 15)$$
$$v_\pi(s) = 0.25 v_\pi(s) - 15$$
$$0.75 v_\pi(s) = -15$$
$$v_\pi(s) = -20$$

5. If π is the equiprobable random policy, what are the respective values of $q_\pi(s_1, down)$ and $q_\pi(s_2, down)$ given that s_1 is the last cell in the third row (value -14) and s_2 is the last cell in the second row?

(a) -1, -15
(b) -15, -21
(c) 0, -14
(d) -13, -19

Sol. (a)
For s_1, we have

$$q_\pi(s_1, down) = \sum_{s'} p(s'|s_1, down)[r + v_\pi(s')]$$

$$q_\pi(s_1, down) = -1 + 0 = -1$$

Similarly, for s_2, we have

$$q_\pi(s_2, down) = -1 - 14 = -15$$

6. In a particular grid-world example, rewards are positive for goals, negative for running into the edge of the world, and zero the rest of the time. Are the signs of these rewards important, or only the intervals between them? Prove, using the discounted return equation

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

that adding a constant C to all the rewards adds a constant, K, to the values of all states, and thus does not affect the relative values of any states under any policies. What is K in terms of C and γ?

(a) $K = \frac{1}{C(1-\gamma)}$

(b) $K = C(\frac{1}{1-\gamma} - 1)$

(c) $K = C(\frac{1}{1-\gamma} + 1)$

(d) $K = \frac{C}{1-\gamma}$

Sol. (d)
Assume that the grid-world problem is a continuing task. For some policy π and state s, the value function can be give as

$$v_\pi(s) = E_\pi\{G_t|s_t = s\}.$$

Using the discounted reward equation, we have

$$v_\pi(s) = E_\pi\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|s_t = s\}.$$

Adding a constant C to all rewards, we have

$$\begin{aligned} v'_\pi(s) &= E_\pi\{\sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + C)|s_t = s\} \\ &= E_\pi\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + C\sum_{k=0}^{\infty} \gamma^k|s_t = s\} \\ &= v_\pi(s) + \frac{C}{1-\gamma}. \end{aligned}$$

We see that adding a constant C to all rewards does not affect the relative values of any states under any policies. Here $K = \frac{C}{1-\gamma}$.

7. Given a reinforcement learning problem, algorithm A will return the optimal state value function for that problem and algorithm B will return the optimal action value function. Your aim is to use the value function so obtained to behave optimally in the environment. Assuming that you know the expected rewards but not the transition probabilities corresponding to the problem in question, which algorithm would you prefer to use for your control task?

   (a) algorithm A
   (b) algorithm B

   Sol. (b)
   Since algorithm B returns the optimal action value function, we can use the information provided by the optimal action value function to control the behaviour of the agent without knowledge of the transition probabilities of the underlying MDP.

8. In proving that L_π is a contraction, we had the expression

$$\gamma \sum_j p(j|s)[v(j) - u(j)] \le \gamma ||v - u|| \sum_j p(j|s)$$

   This inequality holds because

   (a) $v(j) - u(j)$ is a component of $||v - u||$
   (b) the max norm of the difference on the LHS is less than the max norm of the difference on the RHS
   (c) the difference in the LHS can be negative but the norm in the RHS is non-negative
   (d) the max norm on the RHS can at worst be equal to the difference in the LHS

   Sol. (d)

9. We defined the operator $L_\pi : V \to V$ as $L_\pi v = r_\pi + \gamma P_\pi v$. Having seen the proof of the Banach fixed point theorem and assuming that v^π and v^* have their usual meanings, which among the following are implications of showing that L_π is a contraction?

   (a) v^π is a fixed point of L_π
   (b) v^π is a unique fixed point of L_π
   (c) repeatedly applying L_π starting with an arbitrary $v \in V$ results in convergence to v^π
   (d) repeatedly applying L_π starting with an arbitrary $v \in V$ results in convergence to v^*

   Sol. (b), (c)
   Note that while the statement of option (a) is true, it is a result of the Bellman equation and the definition of the L_π operator. Option (d) is not true since repeated application of the operator guarantees convergence only to v^π and not the optimal v^*.

10. Given a value $v \in V$, suppose $L_\pi v = v'$. Then we can conclude that

   (a) $v = v'$
   (b) $v \neq v'$
   (c) $||L_\pi v - L_\pi v'|| \le \lambda ||v - v'||$, $0 \le \lambda < 1$

(d) none of the above

Sol. (c)
The first option may not hold if $v \neq v_\pi$. Similarly, the second option may not hold if $v = v_\pi$. The third option is true because L_π is a contraction and in all three possible scenarios ($v \neq v' \neq v_\pi$, $v \neq v' = v_\pi$, and $v = v' = v_\pi$), the statement holds.

Assignment 5 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. For a particular finite MDP with bounded rewards, let V be the space of bounded functions on S, the state space of the MDP. Let Π be the set of all policies, and let v_π be the value function corresponding to policy π where $\pi \in \Pi$. Is it true that $v_\pi \in V$, $\forall \pi \in \Pi$?

   (a) no

   (b) yes

   Sol. (b)
   The question essentially asks whether V contains all value functions, which we have seen in the lectures to be true.

2. In the proof of the value iteration theorem, we saw that $Lv^{n+1} = L_\pi v^{n+1}$. Is it true, in general, that for an arbitrary bounded function v, $Lv = L_\pi v$ (disregarding any special conditions that may be existing in the aforementioned proof)?

   (a) no

   (b) yes

   Sol. (a)

3. Continuing with the previous question, why is it the case that $Lv^{n+1} = L_\pi v^{n+1}$ in the proof of the value iteration theorem?

   (a) because the equality holds in general

   (b) because v^{n+1} is the optimal value function

   (c) because we are considering only deterministic policies choosing a max valued action in each state

   (d) because v^{n+1} is not a value function

   Sol. (c)

4. Given that $q_\pi(s, a) > v_\pi(s)$, we can conclude

   (a) action a is the best action that can be taken in state s

   (b) π may be an optimal policy

   (c) π is not an optimal policy

(d) none of the above

Sol. (c)
The inequality indicates that there exists an action that if taken in state s, the expected return would be higher than the expected return of taking actions in state s as per policy π. While this indicates that π is not an optimal policy, it does not indicate that a is the best action that can be taken in state s, since there may exist another action a' such that $q_\pi(s, a') > q_\pi(s, a)$.

5. Recall the problem described in the first question of the previous assignment. Use the MDP formulation arrived at in that question and starting with policy $\pi(\text{laughing}) = \pi(\text{silent}) = (\text{incense, no organ})$, perform a couple of policy iterations or value iterations (by hand!) until you find an optimal policy (if you are taking a lot of iterations, stop and reconsider your formulation!). What are the resulting optimal state-action values for all state-action pairs?

   (a) $q_*(s, a) = 8$, $\forall a$
   (b) $q_*(s, a) = 10$, $\forall a$
   (c) $q_*(s, a^*) = 10$, $q_*(s, a) = -10$, $\forall a \neq a^*$
   (d) $q_*(s, a^*) = 10$, $q_*(s, a) = 8$, $\forall a \neq a^*$

   Sol. (d)
   First consider policy iteration.

   Initialisation: $V(L) = V(Q) = 0$; $\pi(L) = \pi(Q) = \neg O \wedge I$

   Assuming $\theta = 0.7$ and given, $\gamma = 0.9$

   Evaluation:
   $\Delta = 0$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = 1 * (-1 + 0.9 * 0) + 0 = -1$
   $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = 0 + 1 * (1 + 0.9 * 0) = +1$
   $\Delta = 1$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = -1.9$
   $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = +1.9$
   $\Delta = 0.9$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = -2.71$
   $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = +2.71$
   $\Delta = 0.81$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = -3.439$
   $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = +3.439$
   $\Delta = 0.729$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = -4.0951$
   $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = +4.0951$
   $\Delta = 0.6561$
   Improvement:
   $\pi(L) = O \wedge I$ (or $O \wedge \neg I$)
   $\pi(Q) = \neg O \wedge I$
   Evaluation:
   $\Delta = 0$
   $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + \gamma V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + \gamma V(Q)] = +4.6856$

$V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + \gamma V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + \gamma V(Q)] = +4.6856$
$\Delta = 0.5905$
Improvement:
$\pi(L) = O \wedge I$ (or $O \wedge \neg I$)
$\pi(Q) = \neg O \wedge I$
No change in policy. This is an optimal policy.

Now consider value iteration.
Initialisation: $V(L) = V(Q) = 0$; Assuming $\theta = 0.9$
$\Delta = 0$
$V(L) = max_a\{P_{LL}^a * [R_{LL}^a + \gamma V(L)] + P_{LQ}^a * [R_{LQ}^a + \gamma V(Q)]\}$
For $a = O \wedge I$ or $O \wedge \neg I$:
$V(L) = +1$
$V(Q) = max_a\{P_{QL}^a * [R_{QL}^a + \gamma V(L)] + P_{QQ}^a * [R_{QQ}^a + \gamma V(Q)]\}$
For $a = \neg O \wedge I$: $V(Q) = +1$
$\Delta = 1$
$V(L) = +1.9$ for $a = O \wedge I$ or $O \wedge \neg I$
$V(Q) = +1.9$ for $a = \neg O \wedge I$
$\Delta = 0.9$
$V(L) = +2.71$ for $a = O \wedge I$ or $O \wedge \neg I$
$V(Q) = +2.71$ for $a = \neg O \wedge I$
$\Delta = 0.81$

Deterministic policy:
$\pi(L) = O \wedge I$ (or $O \wedge \neg I$)
$\pi(Q) = \neg O \wedge I$
Continuing evaluation will lead to convergence values of ± 10 ($\pm 1(1+\gamma+\gamma^2+...) = \frac{\pm 1}{1-\gamma} = \frac{\pm 1}{.1}$).
Optimal state-action values:

Current state	Action	Next state	$q_\star(s,a)$
L	$O \wedge I$	Q	+10
L	$O \wedge \neg I$	Q	+10
L	$\neg O \wedge I$	L	+8
L	$\neg O \wedge \neg I$	L	+8
Q	$O \wedge I$	L	+8
Q	$O \wedge \neg I$	L	+8
Q	$\neg O \wedge I$	Q	+10
Q	$\neg O \wedge \neg I$	L	+8

Note: $q_\star(s,a) = +10$ when an optimal action is taken in any state, whereas $+8 (= -1 + 0.9 * 10)$ results when an initial sub-optimal action is followed by actions taken according to an optimal policy.

6. In the previous question, what does the state value function converge to for the policy we started off with?

   (a) $v_\pi(laughing) = v_\pi(silent) = 10$
   (b) $v_\pi(laughing) = 8$, $v_\pi(silent) = 10$
   (c) $v_\pi(laughing) = -10$, $v_\pi(silent) = 10$
   (d) $v_\pi(laughing) = -8$, $v_\pi(silent) = 10$

Sol. (c)
Refer to the solution of the previous question.

7. In solving an episodic problem we observe that all trajectories from the start state to the goal state pass through a particular state exactly twice. In such a scenario, is it preferable to use first-visit or every-visit MC for evaluating the policy?

   (a) first-visit MC
   (b) every-visit MC
   (c) every-visit MC with exploring starts
   (d) neither, as there are issues with the problem itself

   Sol. (d)
   A state having to be visited exactly twice in any trajectory from the start state to the goal state indicates that the problem environment does not follow the Markov property.

8. Which of the following are advantages of Monte Carlo methods over dynamic programming techniques?

   (a) the ability to learn from actual experience
   (b) the ability to learn from simulated experience
   (c) the ability to estimate the value of a single state independent of the number of states
   (d) the ability to show guaranteed convergence to an optimal policy

   Sol. (a), (b), (c)
   Option (c) is an advantage, since it allows us to estimate values of only specific states of an MDP if those are the only states we care to know about.

9. For a specific MDP, suppose we have a policy that we want to evaluate through the use of actual experience in the environment alone and using Monte Carlo methods. We decide to use the first-visit approach along with the technique of always picking the start state at random from the available set of states. Will this approach ensure complete evaluation of the action value function corresponding to the policy?

   (a) no
   (b) yes

   Sol. (a)
   Depending upon the policy, starting from random states alone will not ensure observing returns for each state-action pairs, which is required to fully evaluate q_π.

10. Assuming an MDP where there are n actions $a \in A$ each of which is applicable in each state $s \in S$, if π is an ϵ-soft policy for some $\epsilon > 0$, then

    (a) $\pi(a|s) = \epsilon$, $\forall a, s$
    (b) $\pi(a|s) = \frac{\epsilon}{n}$, $\forall a, s$
    (c) $\pi(a|s) >= \frac{\epsilon}{n}$, $\forall a, s$
    (d) $\pi(a'|s) = 1 - \epsilon + \frac{\epsilon}{n}$, $\pi(a|s) = \frac{\epsilon}{n}$, $\forall a \neq a'$, $\forall s$

    Sol. (c)
    An ϵ-soft policy is one where the probability of selecting any action is at least ϵ/n.

Assignment 6 (Sol.)
Reinforcement Learning
Prof. B. Ravindran

1. In the search procedure listed in the lecture for Monte-Carlo tree search what is/are the uses of the depth parameter?

   (a) allows us to identify leaf states
   (b) allows us to identify terminal states
   (c) can be used to impact the choice of action selection
   (d) allows us to specialise value functions based on the number of steps that have been taken

   Sol. (a), (c), (d)

2. Suppose you are given a finite set of transition data. Assuming that the Markov model that can be formed with the given data is the actual MDP from which the data is generated, will the value functions calculated by the MC and TD methods (in a manner similar to what we saw in the lectures) necessarily agree?

   (a) no
   (b) yes

   Sol. (a)
   In our lecture example, we saw that the value functions calculated by the MC and TD methods did not agree. The same would still hold if the MDP that generated the data was the MDP that we formed from the data in applying the TD method.

3. In the iterative policy evaluation process, we have seen the use of different update equations in DP, MC, and TD methods. With regard to these update equations

   (a) DP and TD make use of estimates but not MC
   (b) TD makes use of estimates but not DP and MC
   (c) MC and TD make use of estimates but not DP
   (d) all three methods make use of estimates

   Sol. (d)
   DP methods update estimates based on other learned estimates, i.e., they bootstrap. MC methods, while they do not bootstrap, make use of estimates because the target in MC methods, i.e., the sample return, is an estimate of the actual expected return; the expected value of course, is not known. TD methods use both the above, i.e., they make use of samples of the expected values as well as bootstrap.

4. Is it necessary for the behaviour policy of an off-policy learning method to have non-zero probability of selecting all actions?

   (a) no
   (b) yes

   Sol. (a)
   If the probability of selecting certain actions in the estimation policy, i.e., the policy which is being evaluated and/or improved, is zero, then the corresponding probability of selecting those same actions in the behaviour policy can also be zero without causing any problem to the off-policy learning procedure.

5. With respect to the Expected SARSA algorithm, is exploration (using for example ϵ-greedy action selection) required as it is in the normal SARSA and Q-learning algorithms?

   (a) no
   (b) yes

   Sol. (b)
   The difference in the update rules that differentiate Expected SARSA from the SARSA algorithm do not obviate the need for exploration in the former, since without exploration the algorithm would, in general, miss out on large parts of the state space, preventing it from correctly converging (in the limit) to an optimal policy.

6. Assume that we have available a simulation model for a particular problem. To learn an optimal policy, instead of following trajectories end-to-end, in each iteration we randomly supply a state and an action to the model and receive the corresponding reward. This information is used for updating the value function. Which method among the following would you expect to work in this scenario?

   (a) SARSA
   (b) Expected SARSA
   (c) Q-learning
   (d) none of the above

   Sol. (d)
   Note that each of the three algorithms listed above require, in addition to the reward, the next state information, which is not provided by the described simulation model.

7. Consider the following transitions observed for an undiscounted MDP with two states P and Q.

   P, +3, P, +2, Q, -4, P, +4, Q, -3

   Q, -2, P, +3, Q, -3

   Estimates the state value function using first-visit Monte-Carlo evaluation.

   (a) v(P) = 2, v(Q) = -5/2
   (b) v(P) = 2, v(Q) = 0

(c) v(P) = 1, v(Q) = -5/2
(d) v(P) = 1, v(Q) = 0

Sol. (c)
For first-visit MC, we consider only the first occurrence of each state in each transition. Thus, we have

v(P) = (2 + 0)/2 = 1

v(Q) = (-3 - 2)/2 = -5/2

8. Considering the same transition data as above, estimate the state value function using the every-visit Monte-Carlo evaluation.

   (a) v(P) = 2, v(Q) = -5/2
   (b) v(P) = 2, v(Q) = -11/4
   (c) v(P) = 1/2, v(Q) = -11/4
   (d) v(P) = 1/4, v(Q) = -5/2

   Sol. (c)
   In the every-visit case we consider each occurrence of each state in the transitions. Thus, we have

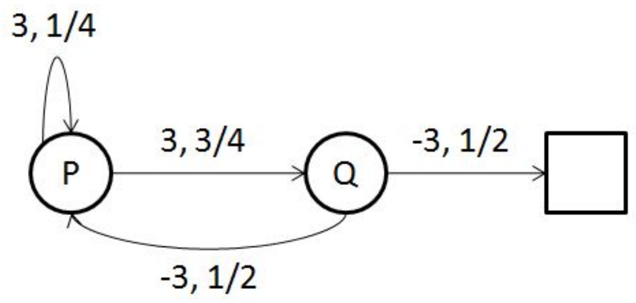
   v(P) = (2 + -1 + 1 + 0)/4 = 1/2

   v(Q) = (-3 - 3 - 2 - 3)/4 = -11/4

9. Construct a Markov model that best explains the observations given in question 7. In this model, what is the probability of transitioning from state P to itself? What is the expected reward received on transitioning from state Q to state P?

   (a) 1/4, -4
   (b) 1/4, -3
   (c) 1/2, -4
   (d) 1/2, -3

   Sol. (b)
   Following is the model that can be constructed from the available data.



10. What would be the value function estimate if batch TD(0) were applied to the above transaction data?

    (a) 1, -1/2
    (b) 1, -2
    (c) 2, -1/2
    (d) 2, -2

    Sol. (d)
    We can solve the Bellman equations directly based on the above model to get

    v(A) = 3 + 1/4 * v(A) + 3/4 * v(B)

    v(B) = -3 + 1/2 * v(A)

    v(A) = 2

    v(B) = -2

Assignment 7 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Consider example 7.1 and figure 7.2 in the text book. Which among the following steps (keeping all other factors unchanged) will result in a decrease in the RMS errors shown in the graphs?

   (a) increasing the number of states of the MDP

   (b) increasing the number of episodes over which error is calculated

   (c) increasing the number of repetitions over which the error is calculated

   (d) none of the above

   Sol. (b)
   Note that the graphs are generated by averaging over the first 10 episodes. If we increase the number of episodes considered, the error shown in the graphs would reduce as evaluation of the policy improves.

2. Considering episodic tasks and for $\lambda \in (0, 1)$, is it true that the one step return always gets assigned the maximum weight in the λ-return?

   (a) no

   (b) yes

   Sol. (a)
   This is not necessarily true and depends on the length of an episode (as well as the value of λ). For example, consider an episode of length 3 and a value of $\lambda = 0.7$.

3. In the TD(λ) algorithm, if $\lambda = 1$ and $\gamma = 1$, then which among the following are true?

   (a) the method behaves like a Monte Carlo method for an undiscounted task

   (b) the eligibility traces do not decay

   (c) the value of all states are updated by the TD error in each episode

   (d) this method is not suitable for continuing tasks

   Sol. (a), (b), (d)
   Note that even if $\lambda = 1$ and the eligibility traces do not decay, states must first be visited before their values can be updated.

4. Assume you have a MDP with $|S|$ states. You decide to use an n-step truncated corrected return for the evaluation problem on this MDP. Do you think that there is any utility in considering values of n which exceed $|S|$ for this problem?

   (a) no

   (b) yes

   Sol. (b)
   Note that the number of steps in an n-step truncated corrected return is related to the length of the trajectories which can exceed the number of states in the state space of a problem.

5. Which among the following are reasons to support your answer in the previous question?

   (a) only values of $n \leq |S|$ should be considered as the number of states is only $|S|$

   (b) all implementations with $n > |S|$ will result in the same evaluation at each stage of the iterative process

   (c) the length of each episode may exceed $|S|$, and hence values of $n > |S|$ should be considered

   (d) regardless of the number of states, different values of n will always lead to different evaluations (at each step of the iterative process) and hence cannot be disregarded

   Sol. (c)

6. Consider the text book figure 5.1 describing the first-visit MC method prediction algorithm and figure 7.7 describing the TD(λ) algorithm. Will these two algorithms behave identically for $\lambda = 1$? If so, what kind of eligibility trace will result in equivalence?

   (a) no

   (b) yes, accumulating traces

   (c) yes, replacing traces

   (d) yes, dutch traces, with $\alpha = 0.5$

   Sol. (a)
   The two algorithms are not identical since figure 7.7 describes the online version of the TD(λ) algorithm, whereas in the MC algorithm described in figure 5.1, updates are obviously not made after each individual reward is observed.

7. Given the following sequence of states observed from the beginning of an episode,

$$s_2, s_1, s_3, s_2, s_1, s_2, s_1, s_6$$

   what is the eligibility value, $e_7(s_1)$, of state s_1 at time step 7 given trace decay parameter λ, discount rate γ, and initial value, $e_0(s_1) = 0$, when accumulating traces are used?

   (a) $\gamma^7\lambda^7$

   (b) $(\gamma\lambda)^7 + (\gamma\lambda)^6 + (\gamma\lambda)^3 + \gamma\lambda$

   (c) $\gamma\lambda(1 + \gamma^2\lambda^2 + \gamma^5\lambda^5)$

   (d) $\gamma^7\lambda^7 + \gamma^3\lambda^3 + \gamma\lambda$

Sol. (c)
According to the non-recursive expression for accumulating eligibility trace, we have

$$e_t(s) = \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k}$$

where I_{ss_k} is an indicator function.

Using the above expression along with the given state sequence, we have

$$e_7(s_1) = (\gamma\lambda)^{7-1} + (\gamma\lambda)^{7-4} + (\gamma\lambda)^{7-6} = \gamma\lambda + \gamma^3\lambda^3 + \gamma^6\lambda^6$$

8. For the above question, what is the eligibility value if replacing traces are used?

   (a) $\gamma^7\lambda^7$
   (b) $\gamma\lambda$
   (c) $\gamma\lambda + 1$
   (d) $3\gamma\lambda$

   Sol. (b)
   We know that when using replacing traces, the eligibility trace of a state is set to 1 if that state is visited and decayed by a factor of $\gamma\lambda$ otherwise. Thus, the latest occurrence of state s_1 just before state s_6 would cause $e_6(s_1)$ to be set to 1 and after the occurrence of state s_6, this would decay to $e_7(s_1) = \gamma\lambda$.

9. In solving the control problem, suppose that at the start of an episode the first action that is taken is not an optimal action according to the current policy. Would an update be made corresponding to this action and the subsequent reward received in Watkin's Q(λ) algorithm?

   (a) no
   (b) yes

   Sol. (b)
   This is immediately clear from the Watkin's Q(λ) algorithm described in the text.

10. Suppose that in a particular problem, the agent keeps going back to the same state in a loop. What is the maximum value that can be taken by the eligibility trace of such a state if we consider accumulating traces with $\lambda = 0.25$ and $\gamma = 0.8$?

    (a) 1.25
    (b) 5.0
    (c) ∞
    (d) insufficient data

    Sol. (a)
    For accumulating traces maximum increase in eligibility occurs if the state is selected: $e_t(s) = \gamma\lambda e_{t-1}(s) + 1$. At maximum, $e_t(s) = e_{t-1}(s)$, giving, $e_t(s) = e_{t-1}(s) = \frac{1}{1-\gamma\lambda}$.

Assignment 8 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Is the problem of non-stationary targets an issue when using Monte Carlo returns as targets?

   (a) no

   (b) yes

   Sol. (a)
   When Monte Carlo returns are used as targets, then the targets are stationary as the target values do not change as the parameters evolve.

2. In a parameterised representation of the value function, we use a feature which acts as a counter of some concept in the environment (number of cans the robot has collected, for example). Does such a feature used for representing the state space lead to a violation of the Markov property?

   (a) no

   (b) yes

   Sol. (a)
   As long as states contain adequate information so that the conditional probability distribution of the future states depends only upon the current state of the environment, any kind of feature can be used to describe the state space without violating the Markov property.

3. Which of the following will effect generalisation when using the tile coding method?

   (a) modify the number of tiles in each tiling (assuming the range covered along each dimension by the tilings remains unchanged)

   (b) modify the number of tilings

   (c) modify the size of tiles

   (d) modify the shape of tiles

   Sol. (a), (b), (c), (d)
   Option (a) and (c) are equivalent and effect the range over which values of one state generalise to values of other states. The same effect is also achieved by the number of tilings, since, for example, more tilings (with different overlapping regions) will result in more states being affected by updates to the value of a single state. Finally, the shape of tiles has an effect on generalisation since it allows us to vary between uniform generalisation (for example using square tiles) and non-uniform generalisation (for example using rectangular tiles) where generalisation is more pronounced along some dimensions and not so much along others.

4. For a particular MDP, suppose we use function approximation and using the gradient descent approach, converge to the value function that is the global optimum. Is this value function the same, in general, as the true value function of the MDP?

   (a) no
   (b) yes

   Sol. (a)
   The value function corresponding to the global optimum and the true value function need not be the same considering that, given the representation used, the function approximation may not even be able to express the true value function.

5. Which of the following methods would benefit from normalising the magnitudes of the basis functions?

   (a) on-line gradient descent TD(λ)
   (b) linear gradient descent Sarsa(λ)
   (c) LSPI
   (d) none of the above

   Sol. (a), (b)
   Order of magnitude differences in the values of the features can impact the performance of gradient descent procedures and hence such methods can benefit from normalisation of the inputs. On the other hand LSPI involves solving a system of linear equations, for which procedures exist which are not impacted by scaling issues.

6. Suppose that individual features, $\phi_i(s, a)$, used in the representation of the action value function are non-linear functions of s and a. Is it possible to use the LSTDQ method in such scenarios?

   (a) no
   (b) yes

   Sol. (b)
   When discussing linear methods such as LSTDQ, we are talking about methods which can approximate functions which are linear in the parameter vector, not the feature vector.

7. Which among the following statements about the LSTD and LSTDQ methods is/are correct?

   (a) LSTD learns the state value function
   (b) LSTDQ learns the action value function
   (c) both LSTD and LSTDQ can reuse samples
   (d) both LSTD and LSTDQ can be used along with tabular representations of value functions

   Sol. (a), (b), (d)
   Option (c) is not correct. Given a set of samples collected from the actual process (not from a generative model in which case reusing samples is perhaps not that important) it is useful for these samples to be reused in evaluating different policies. Recall that LSPI is a policy

iteration algorithm where the policy is constantly being improved upon (and hence changing). Such sample reuse is possible in LSTDQ if for any policy π, $\pi(s')$ is available for each s' in the set of samples. This is because for different policies, the same samples can be made use of, as individual policies only determine the $\phi(s', \pi(s'))$ component of $\tilde{A}$. This is not the case with LSTD where freedom in action choices is not available and must be determined from the policy that is being evaluated.

8. Consider the five state random walk task described in the book. There are five states, $\{s_1, s_2, ..., s_5\}$, in a row with two actions each, left and right. There are two terminal states at each end, with a reward of $+1$ for terminating on the right, after s_5 and a reward of 0 for all other transitions, including the one terminating on the left after s_1. In designing a linear function approximator, what is the least number of state features required to represent the value of the equi-probable random policy?

   (a) 1

   (b) 2

   (c) 3

   (d) 5

   Sol. (a)
   The value of the states from s_1 to s_5 are 1/6, 2/6, ..., 5/6. Hence a single feature, $\phi(s_i) = i$ is adequate to represent the values.

Assignment 9 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Which among the following is/are the advantages of using the Deep Q-learning method over other learning methods that we have seen?
   (a) a faster implementation of the Q-learning algorithm
   (b) guarantees convergence to the optimal policy
   (c) obviates the need to hand-craft features used in function approximation
   (d) allows the use of off-policy algorithms rather than on-policy learning schemes

   Sol. (c)
   As we have seen with the Atari games example, the input to the network is raw pixel data from which the network manages to learn appropriate features to be used for representing the action value function.

2. In the Deep Q-learning method, is ϵ-greedy (or other equivalent techniques) required to ensure exploration, or is this taken care of by the randomisation provided by experience replay?
   (a) no
   (b) yes

   Sol. (b)
   Some technique to ensure exploration is still required. As with the original Q-learning algorithm, if we only consider transitions generated by the action value function (which is essentially what we'll get with experience replay without any exploration), a large part of the state space will likely remain unexplored.

3. Value function based methods are oriented towards finding deterministic policies whereas policy search methods are geared towards finding stochastic policies. True or false?
   (a) false
   (b) true

   Sol. (b)
   With value function based methods, policies are derived from the value function by considering, for a state, that action which gives maximum value. This leads to a deterministic policy. On the other hand, no such maximisation is at work in policy search methods, where the parameters learned, using the gradient descent method, for example, determine the agent's policy. This is likely to be stochastic if the optimal policy (global or local) is stochastic.

4. Suppose we are using a policy gradient method to solve a reinforcement learning problem. Assuming that the policy returned by the method is not optimal, which among the following are plausible reasons for such an outcome?

   (a) the search procedure converged to a locally optimal policy

   (b) the search procedure was terminated before it could reach the optimal policy

   (c) the sample trajectories arising in the problem were very long

   (d) the optimal policy could not be represented by the parametrisation used to represent the policy

   Sol. (a), (b), (d)
   Option (c) may result in an increase in the time it takes to converge to a policy, but does not necessarily affect the optimality of the policy obtained.

5. In using policy gradient methods, if we make use of the average reward formulation rather than the discounted reward formulation, then is it necessary to consider, for problems that do not have a unique start state, a designated start state, s_0?

   (a) no

   (b) yes

   Sol. (a)
   We used the concept of a designated start state to allow a single value that can be assigned to a policy for the purpose of evaluation. The same result is obtained when using the average reward formulation, i.e, by using the average reward formulation we can compare policies according to their long term expected reward per step, $\rho(\pi)$, where

$$\rho(\pi) = lim_{n\to\infty}\frac{1}{N}E\{r_1 + r_2 + r_3 + ... + r_N|\pi\}$$

6. Using similar parametrisations to represent policies, would you expect, in general, MC policy gradient methods to converge faster or slower than actor-critic methods assuming that the approximation to Q^π used in the actor-critic method satisfies the compatibility criteria?

   (a) slower

   (b) faster

   Sol. (a)
   As we have seen, MC policy gradient algorithms may suffer from large variance due to long episode lengths which can slow down convergence. Actor-critic methods, by relying on value function estimates can lead to reduced variance, and hence, faster convergence.

7. If f_w approximates Q^π and is compatible with the parameterisation used for the policy, then this indicates that we can use f_w in place of Q^π in the expression for calculating the gradient of the policy performance metric with respect to the policy parameter because

   (a) $Q^\pi(s,a) - f_w(s,a) = 0$ in the direction of the gradient of $f_w(s,a)$

   (b) $Q^\pi(s,a) - f_w(s,a) = 0$ in the direction of the gradient of $\pi(s,a)$

(c) the error between Q^π and f_w is orthogonal to the gradient of the policy parameterisation

Sol. (b), (c)
As indicated by options (b) & (c), we can use f_w in place of Q^π if the difference between the two is zero in the direction of the gradient of $\pi(s, a)$.

8. Suppose we use the actor-critic algorithm described in the lectures where Q^π is approximated and the approximation used is compatible with the parametrisation used for the actor. Assuming the use of differentiable function approximators, we can conclude that the use of such a scheme will result in

   (a) convergence to a globally optimal policy

   (b) convergence to a locally optimal policy

   (c) cannot comment on the convergence of such an algorithm

   Sol. (b)
   The idea behind the two theorems (policy gradient and policy gradient with function approximation) that we saw in the lectures is to show that using such an approach will result in the policy converging. However, we can only prove convergence to a locally optimal policy.

Assignment 10 (Sol.)
Reinforcement Learning
Prof. B. Ravindran

1. Suppose that in solving a problem, we make use of state abstraction in identifying solutions to some of the sub-problems. In this approach, is it possible to obtain a recursively optimal solution to the original problem?

   (a) no
   (b) yes

   Sol. (b)
   A recursively optimal solution to a hierarchical RL problem can be obtained by combining the optimal solutions of all sub-problems. An abstraction is called safe if optimal solutions in the abstract space are also optimal in the original space. Essentially, when performing state abstraction, if we include all relevant features, then the optimal solutions in the abstract space will remain optimal in the original space. Thus, using such abstractions will allow us to obtain solutions to the original problem which are recursively optimal.

2. What kind of solution would you expect to obtain if, in solving a problem, policies for each individual sub-problem are learned in isolation (i.e., without taking into consideration the overall problem)?

   (a) hierarchically optimal solution
   (b) recursively optimal solution
   (c) flat optimal solution

   Sol. (b)
   In isolation, the best you can expect to do is solve each sub-problem optimally. Combining such policies for the sub-problems of a given hierarchical problem will generally result in a recursively optimal solution.

3. Do the policies of individual options need to be defined over the entire state space of the MDP (of the original problem)?

   (a) no
   (b) yes

Sol. (a)
If the termination condition for a particular state is equal to 1, then there is no need for the option policy to be defined over this state. Additionally, by setting the termination condition appropriately, it should be clear that the set of states which the corresponding option can be active in can be controlled and can be designed to be only a subset of the original state space. Hence, the policy of an option does not necessarily need to be defined over the entire state space.

4. Consider the two room example discussed in the lectures. Suppose you define two options, O_1, to take the agent in room 1 (the left room) to room 2, and O_2, to take the agent in room 2 to the goal state. Assuming that you have appropriately specified the initiation sets and termination conditions for both the options and are trying to learn the individual option policies, would you need to use SMDP Q-learning or would conventional Q-learning suffice?

   (a) SMDP Q-learning
   (b) conventional Q-learning

   Sol. (b)
   Since neither option invokes the other in this example, each option's policy comprises of selecting primitive actions only, and hence, SMDP Q-learning is not required.

5. Consider a Markov policy over options $\mu : S \times O \to [0, 1]$, where S is the set of states and O is the set of options. Assume that all options are Markov. While the policy μ selects options, by considering the primitive actions being selected in those options, we can determine another policy, π, which corresponds to μ, but is a conventional policy over actions. In general, will π also be a Markov policy?

   (a) no
   (b) yes

   Sol. (a)
   When considering the conventional policy, π, the action selected in state s_t depends not only on the current state (s_t), but also on the option being followed at that time, which essentially depends on the entire history since the policy μ was initiated.

6. Consider the following problem design. You have a grid world with several rooms, as discussed in the lectures, with the goal state in a corner cell of one of the rooms. You set up an agent with options for exiting each of the rooms into the other. You also allow the agent to pick from the four primitive actions. There is a step reward of -1. The learning algorithm used is SMDP Q-learning, with normal Q-learning updates for the primitive actions. You expect the agent to learn faster due to the presence of the options, but discover that it is not the case. Can you explain what might have caused this?

   (a) the options are not useful for solving the problem, hence the slowdown
   (b) initially, the agent will focus more on using primitive actions, causing slowdown
   (c) due to the presence of options, we can no longer achieve the optimal solution, hence it takes longer
   (d) it takes longer because we are using SMDP Q-learning compared to conventional Q-learning which is faster

Sol. (b)
Consider the contrast between selecting a primitive action and an option during the initial phase of learning in this problem. On selecting a primitive action from states near the start state or a doorway state, you do not reach the goal state and get a unit of negative reward. On selecting an option, however, you are transported to some doorway state (which is not the goal state) and get a negative reward of larger magnitude (recall SMDP rewards). This suggests that initially, the primitive options will appear more promising and will be explored more whereas the benefit of the options will not initially be realised or exploited. Thus, the expected speedup would in general not be observed.

7. Using intra-option learning techniques, we can learn about options even without ever executing them. True or false?

   (a) false

   (b) true

   Sol. (b)
   As we have seen, in intra-option learning, we can update estimates of multiple options based on observing the rewards received on executing a single option. Many of the options whose estimates are updated need not even have been executed.

8. Suppose that you have identified a set of sub-tasks for solving a large problem using the hierarchical learning approach. To solve each sub-task efficiently, you want to constrain the primitive actions that can be executed within each sub-task. Specifying such constraints is possible in

   (a) options

   (b) HAMs

   (c) both

   Sol. (c)
   Specifying the option policies allows us to constrain the primitive actions that are executed in each option/sub-task. Similar effect can be obtained in HAMs by limiting the choice states in each machine.

Assignment 11 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Using the MAXQ approach leads to solutions which are

   (a) hierarchically optimal

   (b) recursively optimal

   (c) flat optimal

   Sol. (b)
   Since the MAXQ policy of the core MDP is the set of policies of individual sub-tasks, with individual sub-task policies aiming to solve the sub-tasks optimally, you can expect to obtain recursively optimal solutions using the MAXQ approach.

2. We saw that each sub-task has an associated pseudo-reward function. Are the rewards of the core MDP available to the agent while it is learning policies of individual sub-tasks or is the agent restricted to the corresponding sub-task's pseudo rewards?

   (a) only pseudo rewards are available

   (b) both pseudo rewards and core MDP rewards are available

   Sol. (b)
   As we observed in the example taxi problem, rewards of the core MDP are available while learning the policies of the sub-tasks.

3. In the MAXQ framework, is termination in a sub-task deterministic or stochastic as in the options framework?

   (a) deterministic

   (b) stochastic

   Sol. (a)
   We saw in the sub-task definition that for each sub-task, all states of the core MDP are partitioned into a set of active states and a set of terminal states, where sub-task termination is immediate (and deterministic) whenever a terminal state is entered.

4. Each sub-task M_i is an SMDP because

   (a) the state space of the sub-task is a subset of the state space of the core MDP

   (b) each sub-task has its own policy

(c) actions in a sub-task can be temporally extended

(d) the rewards received in sub-tasks depend not only on the state but also on the sub-task in which an action was executed

Sol. (c)
In the definition, we saw that the actions in a sub-task comprise both, primitive actions as well as other sub-tasks. The invocation of a sub-task results in a sequence of actions being executed (similar to an option). Thus, each sub-task is an SMDP.

5. The expected reward function $\bar{R}(s, a)$ of the SMDP corresponding to sub-task M_i is equivalent to the projected value function $V^{\pi_i}(a, s)$. True or false?

   (a) false

   (b) true

   Sol. (a)
   Recall that $\bar{R}(s, a) = V^{\pi}(a, s)$ not $V^{\pi_i}(a, s)$.

6. In the MAXQ approach to solving a problem, suppose that sub-task M_i invokes sub-task M_j. Do the pseudo rewards of M_j have any effect on sub-task M_i?

   (a) no

   (b) yes

   Sol. (b)
   The pseudo rewards of one sub-task are not directly considered when solving a different sub-task regardless of their connectivity. However, since sub-task M_i invokes sub-task M_j, and hence depends upon the policy of M_j, the rewards of M_j do effect sub-task M_i, as the pseudo rewards of sub-task M_j would be a factor determining the policy of M_j.

Assignment 12 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. In the partial observability example that we saw, there were some observations which uniquely identify the state of the environment. Suppose in a problem, no such observations exist, i.e., there are no observations which allow us to exactly determine the state of the environment. In such problems, is it ever possible to be sure which state you are in given only the sequence of observations?

   (a) no

   (b) yes

   Sol. (b)
   It is not necessary that there exist observations that uniquely identify the state of the system. In many problems, the sequence of observations suffice to determine with certainty, the state of the system. For example, in a problem, a particular observation may correspond to multiple states. However, it is possible that the sequence of observations that precede such an observation allows us to distinguish between the multiple states which that observation corresponds to. In addition, the agent has knowledge of its actions which can further help make such distinctions.

2. Suppose that some of the actions that are available to an agent in a partially observable environment have non-deterministic outcomes. How would this impact the agent's ability to determine the state it is in?

   (a) it would make determining the state easier

   (b) it would make determining the state harder

   (c) it would have no effect in determining the state

   Sol. (b)
   Non-deterministic action outcomes introduce another layer of uncertainty, since the agent can now not be sure what the effect of taking an action in a state is (as the observations do not allow the agent to deterministically evaluate the state to which the agent transitions to after taking an action). This would generally make the problem of state determination harder.

3. Referring to the partial observability example considered in the lectures and assuming that there is no noise in the sensor outputs, is it true that all observation sequences of length 3 or more result in the elimination of uncertainty regarding the position of the agent in the environment?

(a) no

(b) yes

Sol. (a)
Consider the plausible sequence of observations 1010, 1010, 1010, 1010.

4. To solve a POMDP problem, suppose you decide to maintain belief states. Would the estimation of the belief states benefit from access to the history of observations and/or actions?

   (a) access to past observations, but not actions, would improve belief state estimation

   (b) access to past actions, but not observations, would improve belief state estimation

   (c) access to past observations and actions would improve belief state estimation

   (d) access to past observations or actions will not improve belief state estimation

   Sol. (d)
   An agent needs to update its belief upon taking an action and observing the corresponding observation. Since the state is Markovian, maintaining a belief over the states solely requires knowledge of the previous belief state, the action taken, and the current observation. The information from past actions and observations is already encapsulated in the belief state.

5. Suppose that you are given a problem in which the agent is able to determine the state it transitions to after each action taken by the agent. However, the state space is described in a manner which requires the agent to consider information in the current, as well as the latest past state in the sequence, to make action decisions. Is the problem so defined an MDP, a POMDP, or neither? Can this problem be solved using RL techniques that we have studied (perhaps with possible modifications to the problem definition)?

   (a) MDP, yes

   (b) POMDP, yes

   (c) neither, no

   (d) neither, yes

   Sol. (d)
   The problem so defined is neither an MDP (the Markov property does not hold) nor a POMDP (partial observability is not an issue here). We can reformulate the problem by considering a state in the new formulation to be a subset of the set of all pairs of states in the original formulation. With appropriate modifications to the transition probability and reward function, we can obtain a MDP which can be solved using the various RL techniques that we have seen. Another approach would be to consider this problem as a POMDP where each 'observation' gives only partial information regarding the 'state' of the system and use history based methods.

6. Recall the example partially observable environment encountered in the lectures. Suppose the goal in this problem is to reach the bottom-right state, with the usual -1 rewards for each transition. Assume that the agent starts two cells below the top left state and the first action selected by the QMDP procedure is to move down. Assuming deterministic action outcomes with no noise in the observations, as well as the use of the QMDP procedure where the underlying MDP has been correctly solved, what can you say about the optimality of the agent's policy in this scenario?

(a) the agent will follow an optimal policy to reach the goal

(b) the agent's policy to reach the goal will not be optimal

(c) the optimality of the agent's policy towards reaching the goal in this scenario cannot be determined with certainty

Sol. (a)
Note that on taking its first action, the agent reaches a state which is uniquely identifiable by the the observation corresponding to that state. Thus, the first action of the agent dispels all state uncertainty and this allows the QMDP procedure to follow the optimal policy to reach the goal.