CS 747 (Autumn 2017): End-semester Examination

Instructor: Shivaram Kalyanakrishnan

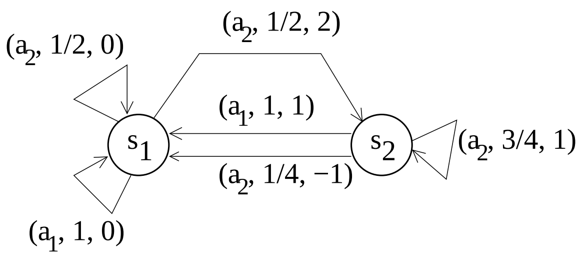
9.30 a.m. – 12.30 p.m., November 22, 2017, 101/103 New CSE Building

Total marks: 25

Note. Provide brief justifications and/or calculations along with each answer to illustrate how you arrived at the answer.

Question 1. Consider an MDP $M = (S, A, T, R, \gamma)$, with a set of states $S = \{s_1, s_2\}$; a set of actions $A = \{a_1, a_2\}$; a transition function T and a reward function R as specified in the table below; and a discount factor $\gamma = \frac{1}{2}$. A state transition diagram corresponding to M is shown below. In the figure, each transition is annotated with (action, transition probability, reward); transitions with zero probabilities are not shown.

Transition probabilities	Rewards
$T(s_1, a_1, s_1) = 1$	$R(s_1, a_1, s_1) = 0$
$T(s_1, a_1, s_2) = 0$	$R(s_1, a_1, s_2) = 0$
$T(s_1, a_2, s_1) = 1/2$	$R(s_1, a_2, s_1) = 0$
$T(s_1, a_2, s_2) = 1/2$	$R(s_1, a_2, s_2) = 2$
$T(s_2, a_1, s_1) = 1$	$R(s_2, a_1, s_1) = 1$
$T(s_2, a_1, s_2) = 0$	$R(s_2, a_1, s_2) = 0$
$T(s_2, a_2, s_1) = 1/4$	$R(s_2, a_2, s_1) = -1$
$T(s_2, a_2, s_2) = 3/4$	$R(s_2, a_2, s_2) = 1$



An RL agent has an initial action value function Q as described in the table below.

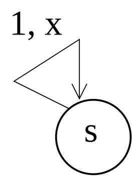
Q	a_1	a_2
s_1	2	0
s_2	1	3

Suppose the agent is in state s_1 to begin, and uses an ϵ-greedy strategy for action selection with $\epsilon = 0.1$. Write down every possible configuration of the Q table after the <u>first</u> learning update is made, under each of the following learning algorithms. Take the learning rate α as 0.1.

1a. Q-learning. [2 marks]

1b. Expected Sarsa. [2 marks]

Question 2. Consider the MDP shown in the figure, with a single state s and a single action a. The reward on taking the action is $x > 0$; the discount factor is γ.



2a. For the unique policy π, which takes a from s, what is $V^{\pi}(s)$? [1 mark]

2b. Consider an agent that learns $V^{\pi}(s)$ using TD(0), starting with an initial estimate $V^0 = 0$. For $t \geq 0$, let V^t denote the estimate after t updates have been made. If the agent uses a <u>constant</u> learning rate $\alpha \in [0, 1]$, what is V^t as a function of x, t, γ, and α? [2 marks]

2c. What value of α minimises the approximation error $|V^{\pi}(s) - V^t|$? Does your answer contradict the standard assumption on the learning rate that we made in class? Explain. [2 marks]

Question 3. In class we primarily considered the *infinite discounted reward* setting; Question 2 above, too, considers the same setting. The present question pertains to the *finite horizon* setting. Consider an MDP with a set of states S, a set of actions A, a transition function T, and a reward function R. For a specified horizon $H > 0$, the value function V^{π} of a policy π is defined as follows, with notations as usual. For $s \in S$:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[r^0 + r^1 + r^2 + \cdots + r^{H-1} | s^0 = s].$$

In other words, $V^{\pi}(s)$ is the expected sum of the first H rewards obtained by starting at s and following π. Assume that the task is continuing (with no terminal states).

You must provide two temporal difference learning algorithms for an agent to learn V^{π} by interacting with the MDP. The first algorithm must be a Monte Carlo algorithm, with no bootstrapping. The second algorithm must be analogous to TD(0), with full bootstrapping. [3 marks]

Question 4. A human expert who knows the (unique) optimal policy $\pi^{\star}$ for a task $M = (S, A, T, R, \gamma)$ decides to help an agent that is *learning* to perform the same task. The agent learns using a provably convergent RL algorithm (such as Q-learning with suitably annealed exploration and learning rates). In order to help the agent, the expert gives it a reward of 1 whenever it takes the optimal action, and a reward of 0 otherwise. Thus, the agent learns essentially learns on the MDP $M^{\text{human}} = (S, A, T, R^{\text{human}}, \gamma)$, where for $s, s' \in S$ and $a \in A$, $R^{\text{human}}(s, a, s') = 1$ if $a = \pi^{\star}(s)$, and $R^{human}(s, a, s') = 0$ if $a \neq \pi^{\star}(s)$. The agent makes no use of the original rewards (from R).

Will the agent eventually achieve optimal behaviour on M, the original task? Prove that your answer is correct. [4 marks]

Question 5. This question asks you to derive the policy gradient theorem for the simplest of non-trivial MDPs: a 2-armed bandit. Consider a bandit with Bernoulli arms a_1 and a_2, whose mean rewards are p_1 and p_2, respectively, with $0 < p_2 < p_1 < 1$. An agent uses a stochastic policy π to sample the arms; the policy has a single parameter $\theta \in \mathbb{R}$. The specific form of the policy is given by the action-selection probabilities

$$\pi(a_1) = \frac{1}{1 + e^{-\theta}}, \text{ and } \pi(a_2) = \frac{e^{-\theta}}{1 + e^{-\theta}}.$$

The algorithm is initialised with $\theta_0 = 0$, which gives an equal probability to sample each arm. For $t \geq 0$, θ_{t+1} is obtained from θ_t by performing a policy gradient update that depends on the particular reward obtained.

5a. What is the expected reward J (from one pull) of π? [1 mark]

5b. Derive an update rule for θ: that is, write down θ_{t+1} in terms of θ_t, the arm pulled, the reward obtained, and a learning rate. The rule must be consistent with taking a step along the gradient of J. [3 marks]

5c. Describe the expected cumulative regret of this policy gradient algorithm as a function of the number of pulls T. You do not have to provide a mathematical expression: it will suffice to describe the expected cumulative regret as, say, logarithmic, sub-linear, or linear in T. Provide suitable justification. [1 mark]

Question 6. What is the idea behind the Dyna-Q architecture? Briefly describe the main elements of the architecture. You do not have to provide pseudocode. [2 marks]

Question 7. Consider a 2-player general sum matrix game. Player A can take actions a_1 and a_2, and Player B can take actions b_1 and b_2. The table below shows each player's rewards when they play different pairs of actions.

A's action	B's action	A's reward	B's reward
a_1	b_1	2	0
a_1	b_2	1	1
a_2	b_1	-1	2
a_2	b_2	3	0

What strategy must each player follow if the expected sum of their individual rewards is to be maximised? What is the maximal expected sum? Justify your answers. [2 marks]

Solutions

1a. The agent starts from state s_1, and can take either action a_1 or a_2.

- If a_1 is taken, the only possible transition is $s_1, a_1, 0, s_1$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_1) \leftarrow Q(s_1, a_1)(1-\alpha)+\alpha(0+\gamma \max\{Q(s_1, a_1), Q(s_1, a_2)\}) = 2\times 0.9+0.1\times(0.5\times 2) = 1.9.$$

- If a_2 is taken, one possible transition is $s_1, a_2, 0, s_1$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_2) \leftarrow Q(s_1, a_2)(1-\alpha)+\alpha(0+\gamma \max\{Q(s_1, a_1), Q(s_1, a_2)\}) = 0\times 0.9+0.1\times(0.5\times 2) = 0.1.$$

  The other possible transition is $s_1, a_2, 2, s_2$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_2) \leftarrow Q(s_1, a_2)(1-\alpha)+\alpha(2+\gamma \max\{Q(s_2, a_1), Q(s_2, a_2)\}) = 0\times 0.9+0.1\times(2+0.5\times 3) = 0.35.$$

1b. The only difference under Expected Sarsa is that rather than take a maximum over the Q-values of the target state, we use an expectation (according to ϵ-greedy action selection). The working is similar.

- If a_1 is taken, the only possible transition is $s_1, a_1, 0, s_1$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_1) \leftarrow Q(s_1, a_1)(1-\alpha) + \alpha(0 + \gamma((1 - \epsilon + \epsilon/2)Q(s_1, a_1) + (\epsilon/2)Q(s_1, a_2)))$$
  $$= 2 \times 0.9 + 0.1 \times (0.5 \times (0.95 \times 2 + 0.05 \times 0)) = 1.895.$$

- If a_2 is taken, one possible transition is $s_1, a_2, 0, s_1$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_2) \leftarrow Q(s_1, a_2)(1-\alpha) + \alpha(0 + \gamma((1 - \epsilon + \epsilon/2)Q(s_1, a_1) + (\epsilon/2)Q(s_1, a_2)))$$
  $$= 0 \times 0.9 + 0.1 \times (0.5 \times (0.95 \times 2 + 0.05 \times 0)) = 0.095.$$

  The other possible transition is $s_1, a_2, 2, s_2$, which leads to the following change to the Q table (all other entries remain unchanged).

  $$Q(s_1, a_2) \leftarrow Q(s_1, a_2)(1-\alpha) + \alpha(2 + \gamma((1 - \epsilon + \epsilon/2)Q(s_2, a_2) + (\epsilon/2)Q(s_2, a_1)))$$
  $$= 0 \times 0.9 + 0.1 \times (2 + 0.5 \times (0.95 \times 3 + 0.05 \times 1)) = 0.34.$$

2a. $V^\pi(s) = x + \gamma x + \gamma^2 x + \dots$ to $\infty = \frac{x}{1-\gamma}$.

2b. Clearly for $t \geq 1$, $\alpha = 0$, we have $V^t = 0$. For $t \geq 1$, $\alpha > 0$,

$$\begin{aligned}
V^t &= V^{t-1}(1-\alpha) + \alpha(x + \gamma V^{t-1}) \\
&= V^{t-1}(1 - \alpha + \alpha\gamma) + \alpha x \\
&= V^{t-2}(1-\alpha+\alpha\gamma)^2 + \alpha x(1 + (1-\alpha+\alpha\gamma)) \\
&= V^{t-3}(1-\alpha+\alpha\gamma)^3 + \alpha x(1 + (1-\alpha+\alpha\gamma) + (1-\alpha+\alpha\gamma)^2) \\
&= V^0(1-\alpha+\alpha\gamma)^t + \alpha x(1 + (1-\alpha+\alpha\gamma) + (1-\alpha+\alpha\gamma)^2 + \dots + (1-\alpha+\alpha\gamma)^{t-1}) \\
&= 0 + \alpha x \frac{1-(1-\alpha+\alpha\gamma)^t}{1-(1-\alpha+\alpha\gamma)} \\
&= \frac{x}{1-\gamma}(1 - (1-\alpha+\alpha\gamma)^t).
\end{aligned}$$

2c. $|V^\pi(s) - V^t| = \frac{x}{1-\gamma}(1-\alpha(1-\gamma))^t$, which is minimised by setting $\alpha = 1$ (in fact, but for the constraint that $\alpha \in [0,1]$, α can be set even higher). The conditions on the learning rate we considered in class—$\sum_{t=0}^{\infty} \frac{1}{\alpha_t} = \infty$ and $\sum_{t=0}^{\infty} \frac{1}{(\alpha_t)^2} < \infty$, achieved, for example, by setting $\alpha_t = \frac{1}{t}$—are necessary when there is *stochasticity* in the rewards or transitions. The MDP in this question is deterministic.

3. We assume that the agent follows π and goes along a trajectory $s^0, a^0, r^0, s^1, a^1, r^1, s^2, \dots$. For $t \geq 1$, it uses a learning rate $\alpha_t = \frac{1}{t}$.

To implement a Monte Carlo algorithm, the agent keeps a value function estimate $V : S \to \mathbb{R}$, which is initialised arbitrarily. At $t \geq H$, the following update is made when state s^t is reached.

$$V(s^{t-H}) \leftarrow V(s^{t-H})(1-\alpha_t) + \alpha_t(r^{t-H} + r^{t-H+1} + \dots + r^{t-1}).$$

It is less straightforward to write down a bootstrapped update. Since the value function is a finite (rather than infinite) sum, it does not give rise to a recursive definition. However, for $h > 1$, the value function V_h^π for a horizon h can be written down in terms of the value function V_{h-1}^π for a horizon $h-1$. For $s \in S$,

$$V_h^\pi(s) = \mathbb{E}_\pi[r^0 + V_{h-1}^\pi(s^1) | s^0 = s], \text{ where}$$

$$V_1^\pi(s) = \mathbb{E}_\pi[r^0 | s^0 = s].$$

A bootstrapping algorithm keeps H value functions $V_1, V_2, \dots, V_H : S \to R$, each meant to predict the sum of a corresponding number of future rewards. At $t \geq 1$, the following $\min\{t, H\}$ updates are made. For $1 < h \leq \min\{t, H\}$,

$$V_h(s^{t-h}) \leftarrow V_h(s^{t-h})(1-\alpha_t) + \alpha_t(r^{t-h} + V_{h-1}(s^{t-h+1})), \text{ and}$$

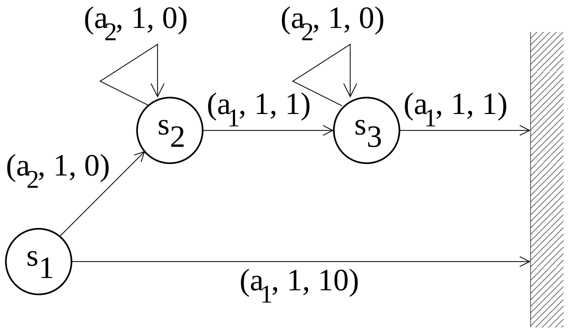
$$V_1(s^{t-1}) \leftarrow V_1(s^{t-1})(1-\alpha_t) + \alpha_t r^{t-1}.$$

Each value function estimate converges in the limit to the corresponding finite sum; in particular V_H converges to V^π.

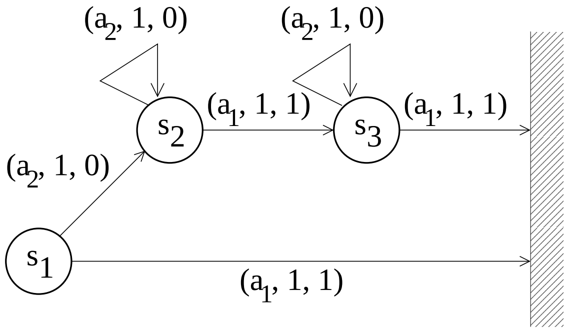
4. The answer depends on whether the task M is continuing or episodic.

Assume that M is continuing (which means M^{human} is also continuing). If $\pi^\star$ is followed in M^{human}, then regardless of the start state, every trajectory will yield only 1's as rewards. Thus, for all $s \in S$, $V^{\pi^\star}_{M^{\text{human}}}(s) = \frac{1}{1-\gamma}$. Take any other policy $\pi \neq \pi^\star$. From at least one state s, the policy will yield 0 as the first reward along trajectories following π. It follows that $Q^{\pi}_{M^{\text{human}}}(s, \pi(s)) < \frac{1}{1-\gamma} = V^{\pi^\star}_{M^{\text{human}}}(s)$. It is also true that Q-values can never exceed $\frac{1}{1-\gamma}$, since 1 is the highest reward available in M^{human}. Hence, by the policy deprovement theorem, π will be strictly dominated by $\pi^\star$ on M^{human}, making $\pi^\star$ the unique optimal policy for M^{human}. The learning agent will converge to $\pi^\star$.

If M is episodic, then the number of steps to termination from a given state can differ based on the policy followed. This property can lead to M and M^{human} having different optimal policies, as illustrated in the example below.



(a) M



(b) M^{human}

Figure 1: Two episodic MDPs with 3 non-terminal states, 2 actions, and no discounting. All transitions are deterministic. The "original" MDP M in (a) has an optimal policy $\pi^\star$ such that $\pi^\star(s_1) = a_1$, $\pi^\star(s_2) = a_1$, and $\pi^\star(s_3) = a_1$. In the MDP M^{human} constructed by the human expert in (b), a_1 is rewarded 1 from all three states, while a_2 is rewarded 0 from all three states. Even so, a_2 becomes the unique optimal action from s_1.

5a. $J = \pi(a_1)p_1 + \pi(a_2)p_2$.

5b.

$$\begin{aligned}
\frac{dJ}{d\theta} &= \frac{d}{d\theta}\left(\sum_{k=1}^{2} \pi(a_k)p_k\right) \\
&= \sum_{k=1}^{2} p_k \frac{d}{d\theta}\left(\pi(a_k)\right) \\
&= \sum_{k=1}^{2} \pi(a_k)\frac{p_k}{\pi(a_k)}\frac{d}{d\theta}\left(\pi(a_k)\right) \\
&= \mathbb{E}_{a_k \sim \pi}\left[\frac{p_k}{\pi(a_k)}\frac{d}{d\theta}\left(\pi(a_k)\right)\right] \\
&= \mathbb{E}_{a_k \sim \pi, r \sim \text{Bernoulli}(p_k)}\left[\frac{r}{\pi(a_k)}\frac{d}{d\theta}\left(\pi(a_k)\right)\right].
\end{aligned}$$

The derivation shows that if arm a_k is drawn with probability $\pi(a_k)$, and $r \in \{0, 1\}$ is drawn as the reward from the arm, then $\frac{r}{\pi(a_k)}\frac{d}{d\theta}\left(\pi(a_k)\right)$ is an unbiased estimate of $\frac{dJ}{d\theta}$. It can be worked out that $\frac{d}{d\theta}\left(\pi(a_1)\right) = \pi(a_1)\pi(a_2)$ and $\frac{d}{d\theta}\left(\pi(a_2)\right) = -\pi(a_1)\pi(a_2)$. As a consequence, we obtain the following set of rules to obtain θ_{t+1} from θ_t, for $t \geq 0$.

- If a_1 was pulled and yielded reward 0, $\theta_{t+1} \leftarrow \theta_t$.
- If a_1 was pulled and yielded reward 1, $\theta_{t+1} \leftarrow \theta_t + \alpha_t \frac{e^{\theta_{t-1}}}{1+e^{\theta_{t-1}}}$.
- If a_2 was pulled and yielded reward 0, $\theta_{t+1} \leftarrow \theta_t$.
- If a_2 was pulled and yielded reward 1, $\theta_{t+1} \leftarrow \theta_t - \alpha_t \frac{1}{1+e^{-\theta_t}}$.

The policy parametrisation is such that larger values of θ will favour a_1. Indeed observe that the policy gradient update increases θ when a_1 gives a 1-reward, and decreases θ when a_2 gives a 1-reward.

5c. In *general*, policy gradient algorithms can get stuck at local minima. If that was to happen in this case, the incurred regret would be linear, since the inferior arm will be played with a non-zero probability for all time.

However, observe that $\frac{dJ}{d\theta}$ becomes zero only at $\theta = \pm\infty$. In other words, there is no local minimum in our example. The actual regret will depend on the rate at which θ_t approaches ∞.[1]

[1] We thank Sabyasachi Ghosh for pointing out an error in an earlier version of the solution.

6. At the heart of Dyna-Q is a model learned from environmental interaction. The agent's experiences are used to refine both the action value function Q and the model M. In addition, M is used to simulate experiences (possibly from a distribution that is different from the on-policy distribution), which are also used to update Q. Updates to M and Q based on experience, as well as updates to Q based on simulated experiences from M, can all be performed asynchronously, making Dyna-Q a very flexible architecture. In particular, simulations can be performed whenever compute time is available; they contribute to speeding up learning.

7. Let Player A play action a_1 with probability x and action a_2 with probability $1-x$. Let Player B play action b_1 with probability y and action b_2 with probability $1-y$. The expected sum of their rewards is

$$S(x,y) = xy(2+0) + x(1-y)(1+1) + (1-x)y(-1+2) + (1-x)(1-y)(3+0).$$

Observe that for any *given* x, $S(x,y)$ is a *linear* function of y: therefore it will be maximised at either $y=0$ or $y=1$. Similarly, for any *given* y, $S(x,y)$ is a *linear* function of x: therefore it will be maximised at either $x=0$ or $x=1$. It follows that $S(x,y)$ must be maximal for one of $(x=0, y=0)$, $(x=0, y=1)$, $(x=1, y=0)$, and $(x=1, y=1)$. From the table we observe that the first configuration is the maximal one. In other words, Player A must play action a_2 with probability 1, and Player B must play action b_2 with probability 1. The resulting sum of their rewards is $3+0=3$.