CS 747 (Autumn 2022)
End-semester Examination

Instructor: Shivaram Kalyanakrishnan

8.30 a.m. – 11.30 a.m., November 17, 2022, LA 001 and LA 002

Note. This exam has **8** questions, given on the pages following this one. Provide justifications/calculations/steps along with each answer to illustrate how you arrived at the answer. You will not receive credit for giving an answer without sufficient explanation.

General instructions.

1. Students whose roll number is odd must sit in LA001; students whose roll number is even must sit in LA002.
2. Barring emergencies, students will be allowed toilet breaks only in these slots.
   - From LA 001: 10.15 a.m. – 10.30 a.m. and 10.45 a.m. – 11.00 a.m.
   - From LA 002: 10.30 a.m. – 10.45 a.m. and 11.00 a.m. – 11.15 a.m.

   At most one student will be allowed at a time; they must enter their name and roll number in the invigilators' register before leaving the room for the break.

Steps for submission.

1. Bring your phone in a pouch or bag, and keep it on the table you are using.
2. Before the exam begins, turn on "flight mode" on the phone, so it cannot communicate. Do not touch the phone while you are writing the exam.
3. When you are finished writing, put your pen away and stand up.
4. Remain standing while retrieving your phone, scanning your paper, turning off "flight mode", then uploading the scanned pdf to Moodle.
5. You will get 15 extra minutes after the test end time for scanning and uploading (you can do it earlier if you have finished). If you are unable to scan and upload your paper, you will be given a slot later to do so.
6. Before leaving, you must turn in your answer paper to the invigilators in the room.
7. We will only evaluate submissions for which the scanned copy matches the physical answer paper that has been turned in.

Question 1. The pseudocode below describes an iterative procedure to update random variable V based on (1) random samples x, y drawn from a finite set X and (2) a real-valued random sample r drawn uniformly from $[0, 1]$.

```
$X \leftarrow \{3, 6, 0, 5\}$.
$V^0 \leftarrow 0$.
For $t = 0, 1, 2, \ldots$:
    $\alpha^t \leftarrow \frac{1}{t+1}$.
    $x^t \leftarrow$ Element of $X$ selected uniformly at random.
    $y^t \leftarrow$ Element of $X$ selected uniformly at random.
    //Note that $x^t$ and $y^t$ need not be distinct.
    $r^t \leftarrow$ Element of $[0, 1]$ drawn uniformly at random.
    $z^t \leftarrow \max\{x^t, y^t\} + r^t$.
    $V^{t+1} \leftarrow V^t(1 - \alpha^t) + \alpha^t z^t$.
```

Describe the limiting behaviour of the sequence $(V^t)_{t=0}^{\infty}$, with an argument for its convergence or non-convergence. If you claim convergence, also provide the limit of the sequence. [4 marks]

Question 2. In an MDP with states $S = \{s_1, s_2, s_3\}$ and actions $A = \{a_1, a_2, \}$, an agent goes along the following state-action-reward trajectory (superscript indicating time step).

$t = 0$			$t = 1$			$t = 2$		
s^0	a^0	r^0	s^1	a^1	r^1	s^2	a^2	r^2
s_1	a_1	2	s_2	a_2	-1	s_1	a_2	4
Q^0			Q^1			Q^2		

The agent keeps a Q-table, with all entries initialised to 0. This table $Q^0 = \mathbf{0}$ gets updated to Q^1 after the first transition, and to Q^2 after the second transition. Updates are made with learning rate $\alpha = \frac{1}{4}$ and no discounting.

2a. Write down Q^1 and Q^2 if the update is made according to Q-learning. [2 marks]

2b. Write down Q^1 and Q^2 if the update is made according to Sarsa. [2 marks]

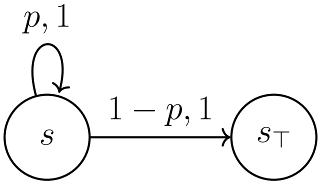
Question 3. An agent interacting with an MDP with non-terminal states $\{s_1, s_2\}$ and terminal state $s_\top$ encounters the following state-reward trajectory while following policy π.

$$s_1, 1, s_1, 2, s_2, 2, s_1, 2, s_2, 1, s_\top.$$

3a. Explain the "Batch TD(0)" algorithm. [1 mark]

3b. What is the estimated value function V^π if Batch TD(0) is run on the episodic data given above? Assume discount factor $\gamma = \frac{3}{4}$. [3 marks]

Question 4. An MDP with a single non-terminal state s and terminal state $s_\top$ is shown in the figure below. With policy π fixed, the agent transitions from s to s with probability $p \in (0, 1)$, and terminates with probability $1 - p$. Each transition yields a reward of 1. The value of s under π is the expected sum of rewards until termination, with no discounting.



Each episode starts at s and terminates after a random number of steps, according to the transition probabilities. The agent records the trajectories from $N \geq 1$ episodes, based on which it estimates $V^\pi(s)$.

For $i \in \{1, 2, \ldots, N\}$, let M_i denote the number of occurrences of s in episode i. Let $G(i, j)$ denote the long-term reward on episode i starting from the j-th occurrence of s (hence $G(i, j)$ is defined for $j \in \{1, 2, \ldots, M_i\}$). Notice that $G(i, j) = M_i - j + 1$.

4a. The "first-visit" Monte Carlo estimate of $V^\pi(s)$ is given by

$$V_{\text{FV}} = \frac{1}{N} \sum_{i=1}^{N} G(i, 1).$$

What is $\mathbb{E}[V_{\text{FV}}]$? [1 mark]

4b. A "random-visit" Monte Carlo estimate of $V^\pi(s)$ is given by

$$V_{\text{RV}} = \frac{1}{N} \sum_{i=1}^{N} G(i, r_i),$$

where r_i is drawn uniformly at random from $\{1, 2, \ldots, M_i\}$. In other words, the random-visit estimator selects a "G" from each episode by selecting uniformly at random from the ones available, and then computes the average (across episodes) of the selected G's. What is $\mathbb{E}[V_{\text{RV}}]$? [4 marks]

Question 5. A "soft-max" policy for an n-armed bandit, $n \geq 2$, maintains a parameter $w_a \in \mathbb{R}$ for each arm $a \in \{1, 2, \ldots, n\}$, and selects arm a with probability

$$\pi_w(a) = \frac{e^{w_a}}{\sum_{i=1}^{n} e^{w_i}}.$$

Observe that the policy is parameterised by n-dimensional vector $w = (w_1, w_2, \ldots, w_n)$.

An agent interacting with the bandit would like to update its policy based on the REINFORCE algorithm. Suppose the agent's current policy is parameterised by $w \in \mathbb{R}^n$. The agent pulls arm a sampled according to π_w, and receives reward r. Describe how the agent must update to a new parameter vector w' based on this sample, if applying REINFORCE with learning rate $\alpha > 0$. [5 marks]

Question 6. An MDP $M = (S, A, T, R, \gamma)$, with notations as usual, has a finite but large set of states S. Hence, a learning agent resorts to using generalisation to approximate the value function of policy $\pi : S \to A$ that it is following. In particular, the agent uses a linear scheme with the approximation $V : S \to \mathbb{R}$ given by

$$V(s) = w \cdot \phi(s) \text{ for } s \in S,$$

where $w \in \mathbb{R}^d$ for some $d \geq 1$ is the weight vector, and $\phi : S \to \mathbb{R}^d$ gives the d-dimensional feature vector for state $s \in S$.

6a. Recall that we defined the mean-squared value error (MSVE) of V (and hence of w) in the class lecture. Write down the formula for $\text{MSVE}(w)$; recall that Linear TD(1) converges to its minimiser $w^\star = \operatorname{argmin}_{w \in \mathbb{R}^d} \text{MSVE}(w)$. [1 mark]

6b. *Regularisation* is a commonly-used technique in machine learning to contain "overfitting". It is accomplished by constraining w to have "small" coefficients. For regularisation parameter $\beta \geq 0$, the generalised objective function is

$$\text{MSVE}_\beta(w) = \text{MSVE}(w) + \beta \|w\|_2^2,$$

where for $w = (w_1, w_2, \dots, w_d)$, $\|w\|_2^2 = \sum_{i=1}^{d} (w_i)^2$. Notice that our original objective function is MSVE_0. Generalise the update rule for Linear TD(1) if the aim is to minimise the regularised objective function MSVE_β. [1 mark]

6c. Suppose $w_1^\star = \operatorname{argmin}_{w \in \mathbb{R}^d} \text{MSVE}_{\beta_1}(w)$ and $w_2^\star = \operatorname{argmin}_{w \in \mathbb{R}^d} \text{MSVE}_{\beta_2}(w)$, where $\beta_1 > \beta_2 > 0$. Is it guaranteed that $\|w_1^\star\|_2^2 \leq \|w_2^\star\|_2^2$? Justify your answer. [3 marks]

Question 7. Several black box optimisation methods involve the step of *selection*. Given $n \geq 2$ candidate solutions $w_1, w_2, \dots, w_n$, the aim is to select some $m \in \{1, 2, \dots, n-1\}$ with the highest fitness values. For candidate solution w, let $f(w)$ denote the fitness. Now, in many cases, simulations are stochastic, hence $f(w)$ is the expected value of real-valued random variable $F(w)$, which has support $[0, f_{\max}]$. One natural approach is to approximate $f(w)$ by $\bar{f}(w)$, the sample average of $N \geq 1$ i.i.d. draws of $F(w)$, obtained by running N simulations using w. Note that this approach would entail Nn total simulations to evaluate the n candidate solutions. Let α denote the probability that the m candidate solutions with the highest $\bar{f}$ values are *not* the m with the highest f values (for convenience we assume no ties in either f or $\bar{f}$). Derive an upper bound on α in terms of f, N, n, m, $w_1, w_2, \dots, w_n$, and $f_{\max}$; the upper bound must go to 0 as $N \to \infty$. [4 marks]

Question 8. This question is based on the paper by Ng et al. (2003), presented in class, on the application of reinforcement learning for helicopter control.

8a. Describe the state and action spaces in the MDP formulation adopted by the authors. Write no more than 8 lines; anything beyond will be ignored. [2marks]

8b. Provide a summary of the methodology used to train a policy for flying the helicopter. Write no more than 12 lines; anything beyond will be ignored. [2 marks]

Solutions

1. By the result of Robbins and Monro, the update must converge to $\mathbb{E}[z]$ (each z^t for $t \geq 0$ is an i.i.d. sample of z). Now, $\mathbb{E}[z] = \mathbb{E}[\max\{x, y\}] + \mathbb{E}[r]$, where x, y, r are again corresponding random variables being repeatedly sampled i.i.d. Since

$$\max\{x, y\} = \begin{cases} 0 & \text{with probability } \frac{1}{16}, \\ 3 & \text{with probability } \frac{3}{16}, \\ 5 & \text{with probability } \frac{5}{16}, \\ 6 & \text{with probability } \frac{7}{16}, \end{cases}$$

we have $\mathbb{E}[\max\{x, y\}] = \frac{0+9+25+42}{16} = \frac{19}{4}$. Also $\mathbb{E}[r] = \frac{1}{2}$. Hence $\mathbb{E}[z] = \frac{21}{4}$.

2a. **Q-learning.** Q^0 is a table with all zeroes. Q^1 remains the same except for

$$Q^1(s_1, a_1) = \frac{1}{2}.$$

Q^2 is the same as Q^1 except for

$$Q^2(s_2, a_2) = \frac{1}{4}\left(-1 + \frac{1}{2}\right) = -\frac{1}{8}.$$

2b. **Sarsa.** Q^0 is a table with all zeroes. Q^1 remains the same except for

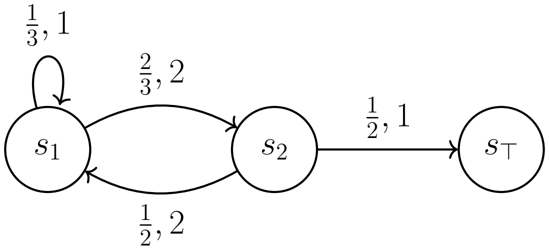
$$Q^1(s_1, a_1) = \frac{1}{2}.$$

Q^2 is the same as Q^1 except for

$$Q^2(s_2, a_2) = \frac{1}{4}\left(-1 + 0\right) = -\frac{1}{4}.$$

3a. Batch TD(0) is a prediction algorithm, which involves performing TD(0) updates to a *batch* of data (that is, to a set of transitions) collected by following some fixed policy π. TD(0) updates are performed in round robin on the set of samples essentially an infinite number of times. The learning rate is annealed appropriately, and hence the method is guaranteed to converge. In the limit, the estimate becomes the value function of π on $\widehat{M}$, which is the MDP with the maximum likelihood of generating the data.

3b. $\widehat{M}$ is obtained by setting transition probabilities based on the empirical fraction observed in the data. In this case, we obtain the following MDP, with transitions annotated with "probability, reward". Transitions with zero probability are not shown.



On $\widehat{M}$, we obtain the following Bellman equations.

$$\begin{aligned} V^{\pi}_{\widehat{M}}(s_1) &= \frac{1}{3}(1 + \gamma V^{\pi}_{\widehat{M}}(s_1)) + \frac{2}{3}(2 + \gamma V^{\pi}_{\widehat{M}}(s_2)), \\ V^{\pi}_{\widehat{M}}(s_2) &= \frac{1}{2}(2 + \gamma V^{\pi}_{\widehat{M}}(s_1)) + \frac{1}{2}(1 + \gamma V^{\pi}_{\widehat{M}}(s_\top)). \end{aligned}$$

Substituting values, we observe:

$$\begin{aligned} V^{\pi}_{\widehat{M}}(s_1) &= \frac{1}{3}(1 + \frac{3}{4} V^{\pi}_{\widehat{M}}(s_1)) + \frac{2}{3}(2 + \frac{3}{4} V^{\pi}_{\widehat{M}}(s_2)), \\ V^{\pi}_{\widehat{M}}(s_2) &= \frac{1}{2}(2 + \frac{3}{4} V^{\pi}_{\widehat{M}}(s_1)) + \frac{1}{2}. \end{aligned}$$

Simplifying:

$$\begin{aligned} 9V^{\pi}_{\widehat{M}}(s_1) &= 20 + 6V^{\pi}_{\widehat{M}}(s_2), \\ 8V^{\pi}_{\widehat{M}}(s_2) &= 12 + 3V^{\pi}_{\widehat{M}}(s_1). \end{aligned}$$

Solving these linear equations yields

$$V^{\pi}_{\widehat{M}}(s_1) = \frac{116}{27}, V^{\pi}_{\widehat{M}}(s_2) = \frac{28}{9}.$$

4a.

$$\begin{aligned}
\mathbb{E}[V_{\text{FV}}] &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[G(i,1)] \\
&= \frac{1}{N}\sum_{i=1}^{N}((1-p)1 + p(1-p)2 + p^2(1-p)3 + \dots) \\
&= \frac{1}{1-p}.
\end{aligned}$$

4b.

$$\begin{aligned}
\mathbb{E}[V_{\text{RV}}] &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[G(i,r_i)] \\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{M_i=1}^{\infty}\mathbb{P}\{\text{Episode } i \text{ lasts } M_i \text{ steps}\}\frac{1}{M_i}\sum_{r=1}^{M_i}(M_i - r + 1) \\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{M_i=1}^{\infty} p^{M_i-1}(1-p)\cdot\frac{M_i+1}{2} \\
&= \sum_{k=1}^{\infty} p^{k-1}(1-p)\cdot\frac{k+1}{2} \\
&= 1 + \frac{p}{2(1-p)}.
\end{aligned}$$

5. Since the task in question is a bandit (with no states and inter-state transitions), the episodic reward simply amounts to $J(w) = \mathbb{E}[r]$, where r is the reward obtained by pulling arm $a \sim \pi_w$. Hence the policy update must be

$$w' \leftarrow w + \alpha\{\nabla_w \ln(\pi_w(a))\}r.$$

Now,

$$\begin{aligned}
\frac{\partial}{\partial w_i}\ln(\pi_w(a)) &= \frac{\partial}{\partial w_i}\ln(\frac{e^{w_a}}{\sum_{j=1}^{n} e^{w_j}}) \\
&= \frac{\partial}{\partial w_i}(w_a) - \frac{\sum_{j=1}^{n} e^{w_j}}{(\sum_{j=1}^{n} e^{w_j})^2}\frac{\partial}{\partial w_i}(\sum_{j=1}^{n} e^{w_j}) \\
&= \begin{cases} 1-\pi_w(a) & i = a, \\ -\pi_w(a) & i \neq a. \end{cases}
\end{aligned}$$

Hence $\nabla_w \ln(\pi_w(a))$ is a vector whose element of a is $1 - \pi_w(a)$, and whose elements for $i \neq a$ are all $-\pi_w(a)$.

6a.

$$\mathrm{MSVE}(w) \overset{\mathrm{def}}{=} \sum_{s \in S} \mu^\pi(s)(V^\pi(s) - w \cdot \phi(s))^2,$$

where $\mu^\pi : S \to [0, 1]$ is the stationary distribution of π.

6b. Notice that $\nabla_w \mathrm{MSVE}_\beta(w) = \nabla_w \mathrm{MSVE}(w) + 2\beta w$. For $t = 0, 1, \ldots,$ we get the update

$$w_{t+1} \leftarrow w_t - \alpha_t \left((G_{t:\infty} - w_t \cdot \phi(s^t))\phi(s^t) + 2\beta w_t \right)$$

where (1) s^t is the state encountered at time step t, (2) $G^{t:\infty}$ is the long-term reward from t onwards, and (3) α_t is the learning rate at time step t.

6c. Yes. Since $w_1^\star$ minimises $\mathrm{MSVE}_{\beta_1}(\cdot)$, we have

$$\mathrm{MSVE}_{\beta_1}(w_1^\star) \leq \mathrm{MSVE}_{\beta_1}(w_2^\star).$$

Since $w_2^\star$ minimises $\mathrm{MSVE}_{\beta_2}(\cdot)$, we have

$$\mathrm{MSVE}_{\beta_2}(w_2^\star) \leq \mathrm{MSVE}_{\beta_2}(w_1^\star).$$

Adding the inequalities above yields

$$\begin{gathered}
\mathrm{MSVE}_{\beta_1}(w_1^\star) + \mathrm{MSVE}_{\beta_2}(w_2^\star) \leq \mathrm{MSVE}_{\beta_1}(w_2^\star) + \mathrm{MSVE}_{\beta_2}(w_1^\star) \\
\Longrightarrow \\
\mathrm{MSVE}(w_1^\star) + \beta_1\|w_1^\star\|_2^2 + \mathrm{MSVE}(w_2^\star) + \beta_2\|w_2^\star\|_2^2 \leq \mathrm{MSVE}(w_2^\star) + \beta_1\|w_2^\star\|_2^2 + \mathrm{MSVE}(w_1^\star) + \beta_2\|w_1^\star\|_2^2 \\
\Longrightarrow \\
(\beta_1 - \beta_2)(\|w_1^\star\|_2^2 - \|w_2^\star\|_2^2) \leq 0 \\
\Longrightarrow \\
\|w_1^\star\|_2^2 \leq \|w_2^\star\|_2^2.
\end{gathered}$$

7. Without loss of generality, assume

$$f(w_1) > f(w_2) > \cdots > f(w_n).$$

The idea is that if N is large enough, each $\bar{f}(w)$ will be "close" to the corresponding $f(w)$, and hence the top m w's according to $\bar{f}$ will be $w_1, w_2, \ldots, w_m$. Let c be an arbitrary number in $(f(w_{m+1}), f(w_m))$. We can be sure that the selection is accurate if for $i \in \{1, 2, \ldots, m\}$, $\bar{f}(w_i) > c$, and for $i \in \{m+1, m+2, \ldots, n\}$, $\bar{f}(w_i) \leq c$.. By separately upper-bounding the probability of deviation for each candidate solution (using Hoeffding's inequality), and then combining them, we get

$$\begin{aligned}\alpha &\leq \sum_{i=1}^{m} \mathbb{P}\{\bar{f}(w_i) > c\} + \sum_{i=m+1}^{n} \mathbb{P}\{\bar{f}(w_i) \leq c\} \\ &\leq \sum_{i=1}^{n} e^{-2N\frac{(f(w_i)-c)^2}{(f_{\max})^2}}.\end{aligned}$$

Since the bound holds for arbitrary $c \in (f(w_{m+1}), f(w_m))$, we have

$$\alpha \leq \min_{c \in (f(w_{m+1}), f(w_m))} \sum_{i=1}^{n} e^{-2N\frac{(f(w_i)-c)^2}{(f_{\max})^2}}.$$

Alternatively, we may upper-bound α by upper-bounding the probabilities for each pair $i \in \{1, 2, \ldots, m\}$ and $j \in \{m+1, m+2, \ldots, n\}$ that $\bar{f}(w_i) > \bar{f}(w_j)$—in turn by enforcing that each $\bar{f}$ fall on the appropriate side of a constant $c_{ij} \in (f_{w_j}, f_{w_1})$. This reasoning would yield the following upper bound:

$$\alpha \leq \sum_{i=1}^{m} \sum_{i=m+1}^{n} \min_{c_{ij} \in (f(w_j), f(w_i))} \left(e^{-2N\frac{(f(w_i)-c_{ij})^2}{(f_{\max})^2}} + e^{-2N\frac{(f(w_j)-c_{ij})^2}{(f_{\max})^2}} \right).$$

8a. A total of 12 state features are used, including position (three coordinates), orientation (three coordinates), and the rates of change of the position and orientation (another three plus three coordinates). The action is four-dimensional. For each of the two rotors, there is one control corresponding to speed and one corresponding to tilt.

8b. A human pilot flies the helicopter, and the trajectory is logged. Thereafter a model is trained using supervised learning. The predictor used is locally-weighted linear regression. On this model, policy search (through hill climbing) is performed to obtain a successful control policy. The policy is represented a a neural network, whose weights are the parameters being optimised. Domain knowlegde is incorporated into the policy by setting or removing network connections; reward functions are carefully designed to guide training towards effective policies. Learned policies for hovering and trajectory-following are evaluated on the real helicopter.