Reinforcement Learning	May 11, 2019
Recitation 13	
Lecturer:Yishay Mansour	*TA: Lee Cohen*

Exam Solution

Question 1

part (a)

$$
\begin{aligned}
S &= \{1, 2, 3, ..., k\} \\
A &= \{\text{advance}, \text{return}\} \\
S_0 &= 1 \\
R: &R(k, return) = 1 \\
&R(i, return) = 0, R(i, advance) = 0 && \forall i \in \{1..k-1\} \\
P: &P(1|i, return) = 1 && \forall i \in \{1..k\} \\
&P(i+1|i, advance) = 1 && \forall i \in \{1..k-1\}
\end{aligned}
$$

part (b)

The optimal policy is simply advancing until state k and the retrieving the reward 1 by using the *return* action. Formally:

$$
\begin{aligned}
\pi^*(i) &= advance, \forall i \in \{1..k-1\} \\
\pi^*(k) &= return
\end{aligned}
$$

part (c)

We can simply compute V^* using the bellman equation:

$$V^*(s) = r(s, \pi^*(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi^*(s)) V^*(s)$$

By applying the equation for each state in our MDP:

$$
\begin{aligned}
V^*(i) &= r(i, advance) + \gamma V^*(i+1), \forall i \in \{1..k-1\} \\
V^*(k) &= r(i, return) + \gamma V^*(1)
\end{aligned}
$$

It follows that:

$$V^*(i) = \gamma V^*(i+1), \forall i \in \{1..k-1\} \Rightarrow V^*(1) = \gamma^{k-1} V^*(k)$$
$$V^*(k) = 1 + \gamma V^*(1)$$

We can solve for $V^*(1)$ and $V^*(k)$:

$$V^*(1) = \frac{\gamma^{k-1}}{1-\gamma^k}$$
$$V^*(k) = \frac{1}{1-\gamma^k}$$

More generally, we can solve for each state i and get:

$$V^*(i) = \frac{\gamma^{k-i}}{1-\gamma^k}$$

part (d)

First we write the formula for Value Iteration:

$$V_{n+1}(s) = max_{a \in A}\{r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V_n(s')\}$$

In addition, it is given to us that the initial Value Function is $V_0(s) = 0, \forall s \in S$.
For n=1:
We note that all the only state that will update in the first iteration is k, this is because it is the only state with an immediate reward (the *return* action for state 1). We can also verify this by plugging in the values in the formula above. As a result we get:

$$V_1(i) = 0, 1 \leq i \leq k-1$$
$$V_1(k) = 1$$

For n=2:
In this iteration the only states with non-zero values will be $k-1$ and k. For k, we have the explanation above. For $k-1$, we will have non-zero value since $V_1(k)$ isn't zero. As a result we get:

$$V_2(i) = 0, 1 \leq i \leq k-2$$
$$V_2(k-1) = \gamma$$
$$V_2(k) = 1$$

part (e)

In this part we use the equation developed in part (c) and plug in the relevant values($\gamma = 0.5$):

$$V^*(1) = \frac{0.5^2}{1 - 0.5^3} \approx 0.286$$
$$V^*(2) = \frac{0.5}{1 - 0.5^3} \approx 0.571$$
$$V^*(3) = \frac{1}{1 - 0.5^3} \approx 1.143$$

Question 2

part (a)

The final answer is below, the calculations are followed:

Table 1: My caption

	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$
S_0	0	1	1	1	1
S_1	0	0	-2	-2	-1
S_2	0	0	0	1	1

We first write down the formula for $TD(0)$, and note that our parameters are $\alpha = 0.5$ and $\gamma = 0.5$:

$$\begin{aligned} \Delta_t &= r_t + \gamma V(s_{t+1}) - V(s_t) \\ V(s_t) &= V(s_t) + \alpha \Delta_t \end{aligned}$$

For t=1:

$$\begin{aligned} V_1(0) &= V_0(0) + 0.5[2 + 0.5V_0(1) - V_0(0)] \\ &= 0 + 0.5[2 + 0 - 0] = 1 \end{aligned}$$

For t=2:

$$\begin{aligned} V_2(1) &= V_1(1) + 0.5[-4 + 0.5V_1(2) - V_1(1)] \\ &= 0 + 0.5[-4 + 0 - 0] = -2 \end{aligned}$$

For t=3:

$$\begin{aligned} V_3(2) &= V_2(2) + 0.5[3 + 0.5V_2(1) - V_2(2)] \\ &= 0 + 0.5[3 + 0.5 * (-2) - 0] = 1 \end{aligned}$$

For t=4:

$$\begin{aligned} V_4(1) &= V_3(1) + 0.5[1 + 0.5V_3(1) - V_3(1)] \\ &= -2 + 0.5[1 + 0.5 * (-2) - 1] = -1 \end{aligned}$$

part (b)

The final answer is below, the calculations are followed:

Table 2: My caption

	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$
S_0	0	1	0.1	0.01	0.001
S_1	0	0	1	0.1	1.01
S_2	0	0	0	1	0.1

We first write down the formula for $TD(\lambda)$, and note that our parameters are $\lambda = 0.1$ and $\gamma = 1$. In addition, we have $e_0(s) = 0$:

$$\begin{aligned}\Delta_t &= r_t + \gamma V(s_{t+1}) - V(s_t)\\ e_t(s) &= \gamma\lambda e_{t-1}(s) + I(s_t = s)\\ V(s_t) &= V(s_t) + \alpha e_t \Delta_t\end{aligned}$$

Notes:

1. In $TD(\lambda)$ all the states are updated as opposed to $TD(0)$ where just the state we are in is updated.
2. In the current question we are ONLY asked about the eligibility trace - so handled e_t is enough.

For t=1:

$$e_1 = 0.1 * 1 * \begin{pmatrix}0\\0\\0\end{pmatrix} + \begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\end{pmatrix}$$

For t=2:

$$e_2 = 0.1 * 1 * \begin{pmatrix}1\\0\\0\end{pmatrix} + \begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}0.1\\1\\0\end{pmatrix}$$

For t=3:

$$e_3 = 0.1 * 1 * \begin{pmatrix}0.1\\1\\0\end{pmatrix} + \begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}0.01\\0.1\\1\end{pmatrix}$$

For t=4:

$$e_4 = 0.1 * 1 * \begin{pmatrix} 0.01 \\ 0.1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.001 \\ 1.01 \\ 0.1 \end{pmatrix}$$

Question 3

part (a)

We simply plug in the action value a in the density function of the Pareto probability:

$$\pi(a|\theta, s) = \frac{\alpha(s;\theta)}{a^{\alpha(s;\theta)+1}} \quad \alpha(s;\theta) = (\theta^T \phi(s))^2$$

To calculate the probability of action a being in the range $[1, 10]$, we need to calculate the following integral:

$$\int_1^{10} \frac{\alpha}{x^{\alpha+1}} dx = \alpha \int_1^{10} x^{-\alpha-1} dx = \alpha \Big[-\frac{1}{\alpha x^{\alpha}} \Big]_1^{10} = \Big[-x^{-\alpha} \Big]_1^{10} = 1 - \frac{1}{10^{\alpha}}$$

part (b)

$$\begin{aligned}
\nabla_\theta log(\pi(a|\theta, s)) &= \nabla_\theta log(\frac{\alpha(s;\theta)}{a^{\alpha(s;\theta)+1}}) \\
&= \nabla_\theta [log(\alpha(s;\theta)) - (\alpha(s;\theta) + 1) log(a)] \\
&= [\frac{1}{\alpha(s;\theta)} - log(a)] \nabla_\theta \alpha(s;\theta) \\
&= [\frac{1}{\alpha(s;\theta)} - log(a)] \nabla_\theta (\theta^T \phi(s))^2 \\
&= [\frac{1}{\alpha(s;\theta)} - log(a)] 2 * (\theta^T \phi(s)) * \phi(s)
\end{aligned}$$

Explanation:

1. First we plug in π.
2. The we apply log rules.
3. We apply the chain rule: derive by α and multiply by derivative of α according to θ.
4. Plug in α.
5. Derive α according to θ.

part (c)

By definition of gradient policy:

$$\theta = \theta + \alpha * G * [\frac{1}{\alpha(s;\theta)} - log(a)] 2 * (\theta^T \phi(s)) * \phi(s)$$

Where G is the return from step t.

Question 4

part (a)

Definition:
Operator H is γ-**contracting** if there exists $\gamma \in (0,1)$ such that

$$\|H(Q_1) - H(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

for every $Q_1, Q_2 \in \mathbb{R}^{|S|\times|A|}$.

part (b)

Proof:
First, we write the expectation explicitly:

$$H(Q(s,a)) = r(s,a) + \gamma max_{a\in A}\{\sum_{s'\in S} p(s'|s,b)Q(s',b)\}$$

Note: here we already see that the operator isn't a policy, since the action we use to move forward(b) isn't the same as the one we use for the immediate reward(a).

To handle the max operator in our proof we define b_s^i to be:

$$b_s^i \in argmax_{b\in A}\{\sum_{s'\in S} p(s'|s,b)Q_i(s',b)\}$$

We also separate the proof in to two: first we bound for $H(Q_1) - H(Q_2)$ and then $H(Q_2) - H(Q_1)$. It follows:

$$\begin{aligned}
H(Q_1(s,a)) - H(Q_2(s,a)) &= r(s,a) + \gamma \sum_{s'\in S} p(s'|s,b_s^1)Q_1(s',b_s^1) + \\
&\quad - r(s,a) - \gamma \sum_{s'\in S} p(s'|s,b_s^2)Q_2(s',b_s^2) \\
&\leq \gamma \sum_{s'\in S} p(s'|s,b_s^1)[Q_1(s',b_s^1) - Q_2(s',b_s^1)] \\
&\leq \gamma \sum_{s'\in S} p(s'|s,b_s^1) max_{s''\in S}[Q_1(s'',b_s^1) - Q_2(s'',b_s^1)] \\
&\leq \gamma max_{s''\in S}[Q_1(s'',b_s^1) - Q_2(s'',b_s^1)] \\
&\leq \gamma \|Q_1 - Q_2\|_\infty
\end{aligned}$$

The first inequality, follow the fact that b_s^1 isn't the $argmax$ of Q_2. The second inequality, follows the the fact that we can bound by choosing the state that results the max difference. The third inequality, is because the probabilities sum to 1. Finally, we further bound the difference since we can also choose to bound using the biggest difference amongst all states and actions (∞-norm).

part (c)

We can simply compute $H(Q)$ by activating the operator over the initial $Q = (0, 0, 0, 0, 0)$. It follows that:

$$\begin{aligned} H(Q) &= (1, 0, 0, 0, 1) \\ H(H(Q)) &= (1 + \gamma, \gamma, 0, 0, 1) \\ H(H(H(Q))) &= (1 + \gamma, \gamma, 0, 0, 1) \end{aligned}$$

Since $H(H(Q)) = H(H(H(Q)))$ we can derive that the fixed-point of the operator is $Q^c = (1 + \gamma, \gamma, 0, 0, 1)$.

part (d)

The answer is NO.

As noted, the operator doesn't match the way policies are activated since the operator receives reward for action a while the next step is decided by another action - b.

We can use the example in part (c) as a counter example. We received that the maximal reward in Q^c is $1+\gamma$ while the maximal reward possible for an optimal policy is 1. Therefore, there doesn't exists a policy achieves such reward.