05124265: Reinforcement Learning
Exercise 2

Tal Grossman, 201512282 , Moshe TODO

24/06/2024

1 Theory

1.1 Question 1

TODO
please see attached PDF file **1_Q1.pdf**

1.2 Question 2

1. Formal MDP definition:

   - State space S = $\{(\nu_0, \ldots, \nu_N, k) : \nu_i \in \{0,1\}, k \in \{0,1,\ldots,9\}\}$
     Each state is represented by a binary vector of length N+1 and a number k. The "ON" bits in the vecotrs represent the available digits and the number k represent the last random index position.

   - Action space A = $\{0, 1, \ldots, N\}$
     Possibles actions represents the selection of the index of the next digit. These actions depends on the state space and its current binary vector.
     I.e. the binary vector holds the available positions where the digit can be placed, And so for some vector V, possible actions are: $A(S(V,k)) = \{i : \nu_i = 1\}$
   - Transition probabilities

     $$P(s'|s,a) = P(\nu'_0, \ldots, \nu'_N, k'|\nu_0, \ldots, \nu_N, k, a) = \begin{cases} \frac{1}{10} & a \in A(S(V,k)), \nu'_a = 0, \nu'_i = \nu_i : \forall i \neq a,\ k \in \{0,\ldots,9\} \\ 0 & \text{otherwise} \end{cases}$$

     The transition probability limits the index selection to the available positions in the binary vector. The probability of the next digit is uniform with an equal probability of $\frac{1}{10}$.
   - Reward function $r(s,a)$ = $r((\nu_0, \ldots, \nu_N, k), a)$ = $k \cdot 10^a$
     the reward is determined by the last index position and the value of the digit in that position.
   - initial state $s_0 = (\{0\}^{N+1}, k) : k \in \{0,1,\ldots,9\}\, each\, w.p\, \frac{1}{10}$
     The initial state is a random number k and a binary vector of length N+1 with all zeros.
   - discount factor $\gamma = 1$
     The discount parameter is set to 1, meaning that the agent does not discount future rewards (Finite horizon problem).

2. Optimal policy
   We will prove using induction that the optimal policy depends only on the number of empty slots.
   first, we will proove it holds when there are exactly 2 empty slots, then we will show that if it holds for k empty slots, it holds for k+1 empty slots, thus true for all N+1.
   <u>Base case:</u> $k = 2$: Assuming the empty cells are α and β, where $\alpha > \beta$ and the digit is d,
   i.e. the state is $s = ((0, \ldots, \underbrace{1}_{\alpha}, \ldots, \underbrace{1}_{\beta}, \ldots, 0), d)$. Now we have 2 possible actions, and

$$
\begin{aligned}
V(s) &= \max_{a\in A}\left\{r(s,a)+\sum_{s'}P(s'\mid s,a)V(s')\right\}\\
&= \max\left(d\cdot 10^{\alpha}+\frac{1}{10}\sum_{i=0}^{9}V((0,\dots,\underbrace{1}_{\beta},\dots,0),i),\ d\cdot 10^{\beta}+\frac{1}{10}\sum_{i=0}^{9}V((0,\dots,\underbrace{1}_{\alpha},\dots,0),i)\right)
\end{aligned}
$$

notice that $V((0,\dots,\underbrace{1}_{\beta},\dots,0),i)=i\cdot 10^{\beta}$ and $V((0,\dots,\underbrace{1}_{\alpha},\dots,0),i)=i\cdot 10^{\alpha}$.
now we need to check which is the bigger expression as a function of d, and we get:

$$
\begin{aligned}
d\cdot 10^{\alpha}+\frac{1}{10}\sum_{i=0}^{9}i\cdot 10^{\beta} &> d\cdot 10^{\beta}+\frac{1}{10}\sum_{i=0}^{9}i\cdot 10^{\alpha}\\
\Rightarrow\quad d\cdot 10^{\alpha}+\frac{1}{10}\cdot 10^{\beta}\cdot 45 &> d\cdot 10^{\beta}+\frac{1}{10}\cdot 10^{\alpha}\cdot 45\\
\Rightarrow\quad d\cdot(10^{\alpha}-10^{\beta}) &> 4.5\cdot(10^{\alpha}-10^{\beta})\\
\Rightarrow\quad d &> 4.5
\end{aligned}
$$

So, if $d>4.5$ the optimal action is to place the digit in the α position, otherwise in the β position.
<u>Inductive step:</u> Assuming the optimal policy depends only on the number of empty slots for k empty slots, we will show that it holds for k+1 empty slots. We'll examin a pair of indices α and β where $\alpha>\beta$ and the digit is d. I.e. the state is $s=((\nu_0,\dots,\underbrace{1}_{\alpha},\dots,\underbrace{1}_{\beta},\dots,\nu_{N+1}),d)$, such that $\sum_{i=0}^{N}\nu_i=k+1$.
We will show that for every specific pair of indices α and β, thus its true in general as well.
<u>for α</u>: $d\cdot 10^{\alpha}+\frac{1}{10}\sum_{i=0}^{9}\underbrace{V((\nu_0,\dots,\underbrace{0}_{\alpha},\dots,\underbrace{1}_{\beta},\dots,\nu_{N+1}),i)}_{\text{k available slots}}$
<u>for β</u>: $d\cdot 10^{\beta}+\frac{1}{10}\sum_{i=0}^{9}\underbrace{V((\nu_0,\dots,\underbrace{1}_{\alpha},\dots,\underbrace{0}_{\beta},\dots,\nu_{N+1}),i)}_{\text{k available slots}}$
by the induction hypothesis, both cases are true for k available digits, and true for every pair of indices.

3. optimal policy for $N=2$:
We will use dynamic programming fir $t=T-1,T-2,\dots,0$ to find the optimal policy:

$$V(s)=\max_{a\in A}r(s,a)+\sum_{s'\in S_{t+1}}P(s'\mid s,a)V_{t+1}(s')$$

$$\pi_t^*(s)=\arg\max_{a\in A}r(s,a)+\sum_{s'\in S_{t+1}}P(s'\mid s,a)V_{t+1}(s')$$

Define $V_3(s)=0$ where

$$V((\nu_0,\nu_1,\nu_2,k),a)=0\quad\forall a\in 0,1,2$$

becuase that $\forall i\in 0,1,2,\nu_i=0$ and no more available slots.
for $t=2$ we have 1 available slot, and so only one possible action where $\forall k\in\{0,\dots,9\}$:

$$
\begin{array}{ccc}
V_2((1,0,0,k),0)=k\cdot 10^{0} & V_2((0,1,0,k),1)=k\cdot 10^{1} & V_2((0,0,1,k),2)=k\cdot 10^{2}\\
\pi_2^*((1,0,0,k))=0 & \pi_2^*((0,1,0,k))=1 & \pi_2^*((0,0,1,k))=2
\end{array}
$$

for $t=1$ we have 2 available slots, and so 2 possible actions.

- case 1 - slots 1, 2 are available, i.e. $s = (0, 1, 1, k)$:

$$\begin{aligned}
\pi_1^*((0,1,1,k)) &= \arg\max_{a\in\{2,1\}}\Big(r((0,1,1,k),2) + \sum_{s'\in S_2} P((0,1,0,k') \mid (0,1,1,k),2)V_2((0,1,0,k')), \\
&\qquad r((0,1,1,k),1) + \sum_{s'\in S_2} P((0,0,1,k') \mid (0,1,1,k),1)V_2((0,0,1,k'))\Big) \\
&= \arg\max_{a\in\{2,1\}}\big(k\cdot 10^2 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^1, k\cdot 10^1 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^2\big) \\
&= \arg\max_{a\in\{2,1\}}\big(100k + 45, 10k + 450\big)
\end{aligned}$$

  we'll choose action 1 for $s = (0, 1, 1, k)$ when

$$100k + 45 \leq 10k + 450 \Rightarrow k \leq 4.5$$

  and action 2 otherwise.

- case 2 - slots 0, 1 are available, i.e. $s = (1, 1, 0, k)$:

$$\begin{aligned}
\pi_1^*((1,1,0,k)) &= \arg\max_{a\in\{1,0\}}\Big(r((1,1,0,k),1) + \sum_{s'\in S_2} P((0,1,0,k') \mid (1,1,0,k),1)V_2((0,1,0,k')), \\
&\qquad r((1,1,0,k),0) + \sum_{s'\in S_2} P((1,0,0,k') \mid (1,1,0,k),0)V_2((1,0,0,k'))\Big) \\
&= \arg\max_{a\in\{1,0\}}\big(k\cdot 10^1 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^0, k\cdot 10^0 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^1\big) \\
&= \arg\max_{a\in\{1,0\}}\big(10k + 4.5, k + 45\big)
\end{aligned}$$

  we'll choose action 0 for $s = (1, 1, 0, k)$ when

$$10k + 4.5 \leq k + 45 \Rightarrow k \leq 4.5$$

  and action 1 otherwise.

- case 3 - slots 0, 2 are available, i.e. $s = (1, 0, 1, k)$:

$$\begin{aligned}
\pi_1^*((1,0,1,k)) &= \arg\max_{a\in\{2,0\}}\Big(r((1,0,1,k),2) + \sum_{s'\in S_2} P((0,0,1,k') \mid (1,0,1,k),2)V_2((0,0,1,k')), \\
&\qquad r((1,0,1,k),0) + \sum_{s'\in S_2} P((1,0,0,k') \mid (1,0,1,k),0)V_2((1,0,0,k'))\Big) \\
&= \arg\max_{a\in\{2,0\}}\big(k\cdot 10^2 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^0, k\cdot 10^0 + \frac{1}{10}\sum_{i=0}^{9} i\cdot 10^2\big) \\
&= \arg\max_{a\in\{2,0\}}\big(100k + 4.5, k + 450\big)
\end{aligned}$$

  we'll choose action 0 for $s = (1, 0, 1, k)$ when

$$100k + 4.5 \leq k + 450 \Rightarrow k \leq 4.5$$

  and action 2 otherwise.

for $t = 0$ we have 3 available slots, and so 3 possible actions.

$$\begin{aligned}
\pi_0^*((1,1,1,k)) = \arg\max_{a\in\{0,1,2\}}\Big(& r((1,1,1,k),0) + \sum_{s'\in S_1} P((0,1,1,k') \mid (1,1,1,k),0)V_1((0,1,1,k')), \\
& r((1,1,1,k),1) + \sum_{s'\in S_1} P((1,0,1,k') \mid (1,1,1,k),1)V_1((1,0,1,k')), \\
& r((1,1,1,k),2) + \sum_{s'\in S_1} P((1,1,0,k') \mid (1,1,1,k),2)V_1((1,1,0,k'))\Big) \\
= \arg\max_{a\in\{0,1,2\}}\big(& k + \frac{1}{10}\sum_{i=0}^{4}(10k+450) + \frac{1}{10}\sum_{i=5}^{9}(100k+45), \\
& 10k + \frac{1}{10}\sum_{i=0}^{4}(k+450) + \frac{1}{10}\sum_{i=5}^{9}(100k+45), \\
& 100k + \frac{1}{10}\sum_{i=0}^{4}(k+45) + \frac{1}{10}\sum_{i=5}^{9}(10k+4.5)\big) \\
= \arg\max_{a\in\{0,1,2\}}\big(& k+607.5, 10k+578.25, 100k+60.75\big)
\end{aligned}$$

we'll choose action 0 for $s = (1,1,1,k)$ when

$$\begin{gathered}
k + 607.5 \geq 10k + 578.25 \text{ and } k + 607.5 \geq 100k + 60.75 \\
29.25 \geq 9k \text{ and } 546.75 \geq 99k \\
k \leq 3.25 \text{ and } k \leq 5.5
\end{gathered}$$

we'll choose action 1 for $s = (1,1,1,k)$ when

$$\begin{gathered}
10k + 578.25 \geq k + 607.5 \text{ and } 10k + 578.25 \geq 100k + 60.75 \\
9k \geq 29.25 \text{ and } 517.5 \geq 90k \\
k \geq 3.25 \text{ and } k \leq 5.75
\end{gathered}$$

we'll choose action 2 for $s = (1,1,1,k)$ when

$$\begin{gathered}
100k + 60.75 \geq k + 607.5 \text{ and } 100k + 60.75 \geq 10k + 578.25 \\
99k \geq 546.75 \text{ and } 90k \geq 517.5 \\
k \geq 5.5 \text{ and } k \geq 5.75
\end{gathered}$$

In Total for state $s = (1,1,1,k)$ the optimal policy is:

$$\pi_0^*((1,1,1,k)) = \begin{cases} 0 & 0 \leq k \leq 3 \\ 1 & 4 \leq k \leq 5 \\ 2 & 6 \leq k \leq 9 \end{cases}$$

□