CIS 4190/5190: Applied Machine Learning	Fall 2024
Homework 4	
Handed Out: 11/06/2024	*Due: 11/20/2024*

- **Your Name:** *Yueyang Li*
- **Your PennKey:** *ezel22*
- **Your PennID:** *47700221*

1 Multiple Choice & Written Questions

1. (a) Ans

   The sentence Bob will generate is:
   "Bob loves cookie."

   (b) Ans

   Compute the Log likelihood of "Bob loves cookies":
   Since $\ln(\Pr(loves|Bob)) = -0.693$ and $\ln(\Pr(cookies|Bob, loves)) = -0.916$
   Therefore: $(loglike)(\text{"Bob loves cookie"}) = -1.609$
   Compute the Log likelihood of "Bob loves cookies":
   Since $\ln(\Pr(hatess|Bob)) = -0.916$ and $\ln(\Pr(cookies|Bob, loves)) = -1.609$
   Therefore: $(loglike)(\text{"Bob hates cookie"}) = -1.609 + (-0.916) = -2.525$

   (c) Ans

   No, the greedy sampling strategy will not always give the most probable sentence.
   The greedy strategy selects the most probable word at each step independently but does not consider the joint probability of the entire sequence. This can lead to suboptimal results because:
   Local Optima: Greedy sampling optimizes for the highest probability at each step, which may not lead to the globally most probable sentence.
   Sequence Dependencies: The words in the sentence are dependent on one another, and greedy sampling might miss combinations that have a higher overall probability due to lower initial probabilities.

   Here is a great example, if following the greedy choice, the algorithm will choose "Bob loves cookies" prior to "Bob loves cherry" although the later has high log-likelihood estimation.

(d) Ans

we will only calculate "Bob loves Bob" and "Bob hates cookie"

Compute the Log likelihood of "Bob loves Bob":
Since $\ln(\Pr(loves|Bob)) = -0.693$ and $\ln(\Pr(Bob|Bob, loves)) = -1.386$
Therefore: $(loglike)$("Bob loves cookie") $= -.0693 + (-1.386) = -2.079$
Compute the Log likelihood of "Bob hates cherry":
Since $\ln(\Pr(hatess|Bob)) = -0.916$ and $\ln(\Pr(cherry|Bob, hates)) = -0.357$
Therefore: $(loglike)$("Bob hates cookie") $= -0.916 + (-0.357) = -1.273$
Therefore:
The two highest log-likelihoods are:
1. Bob hates cherry
2. Bob loves cookie

2. (a) Ans

we can calculate E_1, E_2, E_3, E_4 as follows:
$E_1 = (0.5\cdot 0.7)+(0.2\cdot 0.2)+(0.4\cdot 0.3)+(0.1\cdot 0.1) = 0.35+0.04+0.12+0.01 = 0.52$
$E_2 = (0.5\cdot 0.2)+(0.2\cdot 0.7)+(0.4\cdot 0.3)+(0.1\cdot 0.1) = 0.1+0.14+0.12+0.01 = 0.37$
$E_3 = (0.5\cdot 0.0)+(0.2\cdot 0.6)+(0.4\cdot 0.4)+(0.1\cdot 0.3) = 0+0.12+0.16+0.03 = 0.31$
$E_4 = (0.5\cdot 0.1)+(0.2\cdot 0.1)+(0.4\cdot 0.0)+(0.1\cdot 0.9) = 0.05+0.02+0+0.09 = 0.16$
Therefore, we have $E^t = [0.52, 0.37, 0.31, 0.16]$

(b) Ans

calculate the nominators:
$e^{0.52} = 1.682$
$e^{0.37} = 1.447$
$e^{0.31} = 1.363$
$e^{0.16} = 1.174$
$e^{0.52} + e^{0.37} + e^{0.16} + e^{0.16} = 1.682 + 1.447 + 1.363 + 1.174 = 5.666$

Therefore:

$$\alpha^t = \text{softmax}(E^t) = [\frac{1.682}{5.666}, \frac{1.447}{5.666}, \frac{1.363}{5.666}, \frac{1.174}{5.666}] = [0.297, 0.255, 0.241, 0.207]$$

(c) Ans

We just have to compute the weighted sum:

$$0.297 \cdot [0.7, 0.2, 0.3, 0.1] + 0.255 \cdot [0.2, 0.7, 0.3, 0.1]$$
$$+0.241 \cdot [0.0, 0.6, 0.4, 0.3] + 0.207 \cdot [0.1, 0.1, 0.0, 0.9]$$
$$= [0.280, 0.404, 0.262, 0.314]$$

3. (a) Ans

For this Problem l represents moving left, and r represents moving right
$G_t = R(f, l, e) + 1.0 * R(e, l, d) + ... + 1.0 * R(b, l, a) = 100$
$G_t = R(f, r, g) + 1.0 * 1.0 * R(g, r, h) = 30$
Therefore, the optimal action is moving left to e.

(b) Ans

Similarily, $\gamma = 0.5$:
$G_t = R(f, l, e) + 0.5 * R(e, l, d) + ... + 0.5^5 * R(b, l, a) = 3.125$
$G_t = R(f, r, g) + 1.0 * 0.5 * 0.5 * R(g, r, h) = 7.5$
Therefore, moving to the right to h is the optimal action

(c) Ans

Let's solve γ we have to solve:

$$\gamma^5 \cdot 100 = \gamma^2 \cdot 30$$
$$\gamma^2 \cdot (100\gamma^3 - 30) = 0$$

Therefore, taken $\gamma \in \mathbb{R}$,
$$\gamma = \sqrt[3]{0.3} = 0.67$$

(d) Ans

Initialization:
Initialize $V(s) = 0$ for all states, except $V(\text{terminal}) = 0$.
Compute $V(s)$ for one complete pass through all states.

$$
\begin{aligned}
V(b) &= \max\{100 + 1 \times 0, 0 + 1 \times 0\} && = 100 \\
V(c) &= \max\{0 + 1 \times 100, 0 + 1 \times 0\} && = 100 \\
V(d) &= \max\{0 + 1 \times 100, 0 + 1 \times 0\} && = 100 \\
V(e) &= \max\{0 + 1 \times 100, 0 + 1 \times 0\} && = 100 \\
V(f) &= \max\{0 + 1 \times 100, 0 + 1 \times 0\} && = 100 \\
V(g) &= \max\{0 + 1 \times 100, 30 + 1 \times 0\} && = 100
\end{aligned}
$$

And besides that $V(h) = 0$, and $V(a) = 0$, they are not changed in the loop

(e) Ans

Initialize $V(s) = 0$ for all states. $V(\text{terminal}) = 0$.
Compute $V(s)$ for one complete pass through all states.

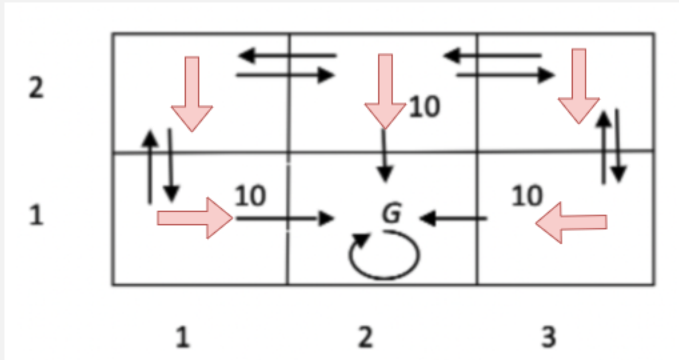
$$
\begin{aligned}
V(g) &= \max\{0 + 1 \times 0, 30 + 1 \times 0\} && = 30 \\
V(f) &= \max\{0 + 1 \times 0, 0 + 1 \times 30\} && = 30 \\
V(e) &= \max\{0 + 1 \times 0, 0 + 1 \times 30\} && = 30 \\
V(d) &= \max\{0 + 1 \times 0, 0 + 1 \times 30\} && = 30 \\
V(c) &= \max\{0 + 1 \times 0, 0 + 1 \times 30\} && = 30 \\
V(b) &= \max\{100 + 1 \times 0, 0 + 1 \times 30\} && = 100
\end{aligned}
$$

And besides that $V(h) = 0$, and $V(a) = 0$, they are not changed in the loop

(f) Ans

1. **Left to Right**: In this case, 1 iteration since after one VI, the array will no longer update its Value functions
2. **Right to Left**: In this case, 1 iteration since after one VI, the array will no longer update its Value functions

4. (a) Ans



(b) Ans

We want to calculate $V^*(1,2)$, There are two options regarding this cell.
Move Right to cell $(2,2)$
Move Down to cell $(1,1)$
Therefore:

$$V^*(1,2) = \max\{0 + 0.8 \cdot V^*(2,2), 0 + 0.8 \cdot V^*(1,1)\}$$

First, we calculate: $V^*(2,2)$

$$V^*(2,2) = \max\{0 + 0.8 \cdot V^*(3,2), 10 + 0.8 \times V^*(G)\}$$

We reached the absorbing state, where its value function can be caculated by:

$$V^*(G) = \frac{R(G,a,G)}{1-\gamma} = 0$$

Now $V^*(2,2) = \max\{0 + 0.8 \cdot V^*(3,2), 10\} = 10$ **second, we calculate** $V^*(1,1)$

$$V^*(1,1) = \max\{10 + 0.8 \cdot V^*(2,1), 0 + 0.8 \cdot V^*(1,2)\} = 10$$

Therefore, finally, we have

$$V^*(1,2) = \max\{8, 8\} = 8$$

(c) Ans

Premise: Q values are intialized to 0 , and $\alpha = 0.1$
From the bottom-left $(1,1)$ to G, we would have the following loop:

$$Q[(1,1), up] = Q[(1,1), up] + 0.1(0 + 0.8 \max_{a'} Q[(1,2), a'] - Q[(1,1), up])$$

Since $Q[(1,2), a'] = 0$, we get:

$$Q[(1,1), \mathrm{up}] = 0$$

For $Q[(1,2), \mathrm{right}]$, we have

$$Q[(1,2), \mathrm{right}] = Q[(1,2), \mathrm{right}] + 0.1 \cdot (0 + 0.8 \max_{a'} Q[(2,2), a'] - Q[(1,2), \mathrm{right}])$$

Since $Q[(2,2), a'] = 0$, we get:

$$Q[(1,2), \mathrm{right}] = 0$$

For $Q[(3,2), \mathrm{down}]$, we have

$$Q[(3,2), \mathrm{down}] = Q[(3,2), \mathrm{down}] + 0.1 \cdot (0 + 0.8 \max_{a'} Q[(3,1), a'] - Q[(3,2),])$$

Since $Q[(2,2), a'] = 0$, we get:

$$Q[(3,2), \mathrm{down}] = 0$$

For $Q[(3,1), \mathrm{left}]$, we have

$$Q[(3,1), \mathrm{left}] = Q[(3,1), \mathrm{left}] + 0.1 \cdot (10 + 0.8 \max_{a'} Q[G, a'] - Q[(3,1), \mathrm{left}])$$

Since we know that $Q[G, a'] = 0$, therefore

$$Q[(3,1), left] = 1$$

This terminates the current episode. So in a nutshell, after the first episode

$$\begin{cases} Q[(1,1), A] & = 0 \\ Q[(1,2), A] & = 0 \\ Q[(2,2), A] & = 0 \\ Q[(3,2), A] & = 0 \\ Q[(3,1), \mathrm{up}] & = 0 \\ Q[(3,1), \mathrm{left}] & = 1 \\ Q[(G), A] & = 0 \end{cases}$$

(d) Ans

Further training won't solve the problem because of the reason as listed:

- The Robot receives +1 rewards regardless of the step taken, there's no penalty for inefficiency, so there's no additional incentive / reward for robot to optimize the path.
- The robot will get stuck in a locally optimal policy that solves the maze but does not explore shorter routes. Training will reinforce the current policy.

Suggest fix to this problem are:

- Have Negative feedback(penalty) for every step it takes, encouraging the agent to find the shortest path.
- include discount factor $\gamma < 1$ os that future rewards are worth less than immediate rewards, this will result in higher total rewards because γ^T decreaeses over the steps taken

5. Programming Assignment

   (a)