PSET 4

Anusha Murali **CS182** **November 21, 2023**

Problem 1 POMDP Policy Evaluation

(1) POMDP representation of the given problem:

- s: **States** The four rooms $\{S1, S2, S3, S4\}$.
- A: **Actions** Three actions are possible in each state, namely { LEFT (L), RIGHT (R), and OPEN (O) }. Opening the door immediately restarts the whole game, placing the agent in $S4$.
- P: **Transition Probabilities** LEFT (L) and RIGHT (R) actions cause the agent to move in the intended direction with probability 0.8, and in the opposite direction with probability 0.2. The OPEN (O) action opens the door with probability 1.0.
- $R(s)$: **Rewards** Each LEFT (L) or RIGHT (R) action costs -4. The OPEN (O) action generates a reward equal to the number on the door.
- Ω: **Observations**: There aren't any special observation actions; however, observations are available when the agent reaches a room (can observe the color of the room). In Room $S1$ and $S2$, there is ambiguity in that the agent observes Red or Green with equal probability.
- O: **Observation Probabilities** $O(s', a, o)$ (the probability that based on our observation o and the action a that we take, we end up at some state s'). In states $S1$ and $S2$, the probabilities of observing either Red or Green is 0.5.

Belief States of the POMDP: $B = \langle b(S1), b(S2), b(S3), b(S4)\rangle$. The belief state $b(S)$ is the probability that we are at world state S in belief state b, and $0 \leq b(S) \leq 1$. The belief states can be obtained from the color of the rooms. While Room $S3$ and $S4$ are Red and Green respectively, the rooms $S1$ and $S2$ can be either Red or Green with probability = 0.5.

(2) We want to evaluate the policy for $T = 2$ when the action sequence is RIGHT $\rightarrow$ OPEN.

It is given that the initial belief state is $\langle b(S1), b(S2), b(S3), b(S4)\rangle = \langle 0.5, 0.5, 0, 0\rangle$. The agent takes the action RIGHT first followed by OPEN. So, the policy is $p \equiv RIGHT \rightarrow OPEN$. Therefore, we want to evaluate the following:

$$V_p(b) = \sum_S V_p(S)b(S),$$

where $b(S) = \langle b(S1), b(S2), b(S3), b(S4)\rangle = \langle 0.5, 0.5, 0, 0\rangle$ is the initial belief state and $V_p(S)$ is the expected value for an individual state S, where $S \in \{S1, S2, S3, S4\}$.

Since $b(S3) = b(S4) = 0$, our evaluation simplifies into,

$$\begin{aligned} V_p(b) &= \sum_S V_p(S)b(S) = V_p(S1) \times 0.5 + V_p(S2) \times 0.5 + V_p(S3) \times 0 + V_p(S4) \times 0 \\ &= 0.5V_p(S1) + 0.5V_p(S2). \end{aligned}$$

Hence we calculate $V_p(S1)$ and $V_p(S2)$ using the following expression:

$$V_p(S) = R(S, a(p)) + \gamma \sum_{S'} P(S'|S, a(p)) \times \sum_Z P(Z|S', a(p)) \times V(S_Z'^{a(p)}),$$

where

- $R(S, a(p))$: Reward obtained when taking action $a(p)$ from state S,
- $P(S'|S, a(p))$: Probability of arriving in state S' from state S while taking action $a(p)$,
- $P(Z|S', a(p))$: Probability of making observation Z after arriving in state S' by taking action $a(p)$ from state S. The possible observations in our problem are either Green or Red,
- $V(S_Z'^{a(p)})$: Expected value of state S', having taken action $a(p)$ and made the observation Z,
- γ: Discount factor. In our problem $\gamma = 1$.

We can simplify the expression for $V_p(S)$ by dropping $a(p)$ from $P(S'|S, a(p))$. So, we have,

$$V_p(S) = R(S, a(p)) + \gamma \sum_{S'} P(S'|S) \times \sum_Z P(Z|S', a(p)) \times V(S_Z'^{a(p)}),$$

Computation of $V_p(S1)$:

$$\begin{aligned} V_p(S1) &= R(S1, a(p)) + \sum_{S'} P(S'|S1, a(p)) \times \sum_Z P(Z|S', a(p)) \times V(S_Z'^{a(p)}) \\ &= R(S1, \text{RIGHT}) + \sum_{S'} P(S'|S1) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S_Z'^{\text{OPEN}}) \\ &= -4 + P(S1|S1) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S_Z'^{\text{OPEN}}) \\ &\quad + P(S2|S1) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S_Z'^{\text{OPEN}}) \\ &\quad + P(S3|S1) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S_Z'^{\text{OPEN}}) \\ &\quad + P(S4|S1) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S_Z'^{\text{OPEN}}) \end{aligned}$$

We note that $P(S1|S1) = 0.2, P(S2|S1) = 0.8, P(S3|S1) = 0$ and $P(S4|S1) = 0$. Therefore, we find,

$$\begin{aligned}
V_p(S1) &= -4 + 0.2 \times \sum_Z P(Z|S1, \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&\quad + 0.8 \times \sum_Z P(Z|S2, \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&= -4 + 0.2 \times (P(\text{Green}|S1, \text{OPEN}) \times (-40) + P(\text{Red}|S1, \text{OPEN}) \times (-40)) \\
&\quad + 0.8 \times (P(\text{Green}|S2, \text{OPEN}) \times (-100) + P(\text{Red}|S2, \text{OPEN}) \times (-100)) \\
&= -4 + 0.2 \times (0.5 \times -40 + 0.5 \times -40) + 0.8 \times (0.5 \times -100 + 0.5 \times -100) = -92.
\end{aligned}$$

Computation of $V_p(S2)$:

$$\begin{aligned}
V_p(S2) &= R(S2, a(p)) + \sum_{S'} P(S'|S2, a(p)) \times \sum_Z P(Z|S', a(p)) \times V(S'^{a(p)}_Z) \\
&= R(S2, \text{RIGHT}) + \sum_{S'} P(S'|S2) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&= -4 + P(S1|S2) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&\quad + P(S2|S2) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&\quad + P(S3|S2) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&\quad + P(S4|S2) \times \sum_Z P(Z|S', \text{OPEN}) \times V(S'^{\text{OPEN}}_Z)
\end{aligned}$$

We note that $P(S1|S2) = 0.2, P(S2|S2) = 0, P(S3|S2) = 0.8$ and $P(S4|S1) = 0$. Therefore, we find,

$$\begin{aligned}
V_p(S2) &= -4 + 0.2 \times \sum_Z P(Z|S2, \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&\quad + 0.8 \times \sum_Z P(Z|S2, \text{OPEN}) \times V(S'^{\text{OPEN}}_Z) \\
&= -4 + 0.2 \times (P(\text{Green}|S1, \text{OPEN}) \times (-40) + P(\text{Red}|S1, \text{OPEN}) \times (-40)) \\
&\quad + 0.8 \times (P(\text{Green}|S3, \text{OPEN}) \times (100) + P(\text{Red}|S3, \text{OPEN}) \times (100)) \\
&= -4 + 0.2 \times (0.5 \times -40 + 0.5 \times -40) + 0.8 \times (0 \times 100) + 0.8 \times (1 \times 100) = 68.
\end{aligned}$$

Therefore the expected value of the $T = 2$ policy in the given initial belief state is,

$$\begin{aligned}
V_p(b) &= \sum_S V_p(S) b(S) = V_p(S1) \times 0.5 + V_p(S2) \times 0.5 + V_p(S3) \times 0 + V_p(S4) \times 0 \\
&= V_p(S1) \times 0.5 + V_p(S2) \times 0.5 = -92 \times 0.5 + 68 \times 0.5 = \boxed{-12}.
\end{aligned}$$

(3) Since S4 is always Green, we can exclude it from our analysis. The probability that S1 is Red is $\frac{1}{2}$. Similarly the probability that S2 is Red is also $\frac{1}{2}$. But the probability that S3 is Red is 1. Hence the selected room is S3, given that it is Red is,

$$p_{S3|\text{Red}} = \frac{1}{1 + \frac{1}{2} + \frac{1}{2}} = 0.5.$$

The reason is that since the room is known to be Red, it has to be one of S1, S2, or S3. However, rooms S1 and S2 can be Red only with probability 0.5. Since the agent is initialized such that being in any room is equally likely, using Baye's theorem,

$$P(S3|\text{Red}) = \frac{P(S3) \times P(\text{Red}|S3)}{P(\text{Red})} = \frac{(1/4) \times 1}{(1/4)(1/2) + (1/4)(1/2) + (1/4)(1) + 1/4)(0)} = 0.5.$$

PSET 4

Anusha Murali **CS182** **November 21, 2023**

Problem 2 Reinforcement Learning

(1) The value function V and the Q-function are related by the following equations:

$$V(s) = \max_a \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma U(s')\right]$$

$$Q(s,a) = \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma U(s')\right]$$

Therefore,

$$V(s) = \max_a Q(s,a).$$

and

$$Q(s,a) = \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma \max_{a'} Q(s',a')\right].$$

Hence, we can write the temporal update equations in terms of the value function V for the state/action sequence $s - a - s' - a' - s''$ as follows:

$$Q(s,a) = \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma V(s')\right]$$

$$Q(s',a') = \sum_{s''} P(s''|s',a')\left[R(s',a',s'') + \gamma V(s'')\right].$$

Introducing the learning factor α as discussed in the class, we can represent the above two equations as:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[R(s,a,s') + \gamma V(s')\right]$$
$$Q(s',a') \leftarrow (1-\alpha)Q(s',a') + \alpha\left[R(s',a',s'') + \gamma V(s'')\right].$$

(2) We initialize all the entries in the $Q(s,a)$ table to zero:

	$a1$	$a2$
$s1$	0	0
$s2$	0	0
$s3$	0	0
$s4$	0	0
$s5$	0	0

Let us assume that the agent is starting at state $s1$ and taking action $a1$. Hence, the agent could arrive at state $s2$ with probability 0.5 or arrive at state $s3$ with probability 0.5. Therefore, we compute the Q-value for state $s1$ as follows:

$$\begin{aligned} Q(s1,a1) &= 0.5\left[R(s1,a1,s2) + \gamma \times \max_{a'}(Q(s2,a'))\right] \\ &\quad + 0.5\left[R(s1,a1,s3) + \gamma \times \max_{a'}(Q(s3,a'))\right] \\ &= 0.5\left[0 + 1 \times \max(0,0)\right] + 0.5\left[0 + 1 \times \max(0)\right] = 0. \end{aligned}$$

Similarly, if the agent is starting at $s1$ and taking action $a2$, we obtain the following update equation:

$$\begin{aligned} Q(s1, a2) &= 0.25\left[R(s1, a2, s2) + \gamma \max_{a'}(Q(s2, a'))\right] + 0.75\left[R(s1, a2, s3) + \gamma \max_{a'}(Q(s3, a'))\right] \\ &= 0.25[0 + 1 \times \max(0, 0)] + 0.75[0 + \times \max(0)] = 0. \end{aligned}$$

We have completed the first round of actions from $s1$, so we now consider the actions while the agent is in state $s2$.

Taking action $a1$ from state $s2$ lands the agent in $s4$ with probability of 1.0 with a reward of 6. So, we have the following update equation:

$$\begin{aligned} Q(s2, a1) &= R(s2, a1, s4) + \gamma \cdot \max_{a'}(Q(s4, a')) \\ &= 6 + 1 \times 0 = 6. \end{aligned}$$

So, we update the Q-table as follows:

	$a1$	$a2$
$s1$	0	0
$s2$	6	0
$s3$	0	0
$s4$	0	0
$s5$	0	0

Similarly, taking action $a2$ from state $s2$ lands the agent in $s4$ with a probability of 0.5, and in $s5$ with a probability of 0.5. The former gives a reward of 12 and the latter gives a reward of 4. Hence our update equation will be as follows:

$$\begin{aligned} Q(s2, a2) = 0.5 &\left[R(s2, a2, s4) + \gamma \times 0.5 \cdot \max_{a'}(Q(s4, a'))\right] \\ &+ 0.5\left[R(s2, a2, s5) + \gamma \times 0.5 \cdot \max_{a'}(Q(s5, a'))\right] \\ = 0.5\,[12 + 1 \times 0] &+ 0.5\,[4 + 1 \times 0] = 8 \end{aligned}$$

So, we update the Q-table as follows:

	$a1$	$a2$
$s1$	0	0
$s2$	6	8
$s3$	0	0
$s4$	0	0
$s5$	0	0

When the agent is in state $s3$, it takes action $a1$ and lands in $s5$ with a probability of 1. This has a reward of 16. So, our update equation is:

$$\begin{aligned} Q(s3, a1) &= R(s3, a1, s5) + \gamma \cdot \max_{a'}(Q(s5, a')) \\ &= 16 + 1 \times 0 = 16. \end{aligned}$$

So, we update the Q-table as follows:

	$a1$	$a2$
$s1$	0	0
$s2$	6	8
$s3$	16	0
$s4$	0	0
$s5$	0	0

Since we have arrived at a terminal state (i.e: $s5$), we have completed one episode of Q-value iteration.

We again repeat the process using the updated Q table as follows.

Consider the agent in state $s1$ taking action $a1$ as in the first episode. We obtain the following update equation:

$$\begin{aligned} Q(s1, a1) = 0.5 &\left[R(s1, a1, s2) + \gamma \times \max_{a'}(Q(s2, a'))\right] \\ &+ 0.5 \left[R(s1, a1, s3) + \gamma \times \max_{a'}(Q(s3, a'))\right] \\ &= 0.5\,[0 + 1 \times \max(6, 8)] + 0.5\,[0 + 1 \times \max(16, 0)] = 12. \end{aligned}$$

So, we update the Q-table as follows:

	$a1$	$a2$
$s1$	12	0
$s2$	6	8
$s3$	16	0
$s4$	0	0
$s5$	0	0

Similarly, consider the agent in state $s1$ taking action $a2$. We obtain the following update equation:

$$\begin{aligned} Q(s1, a2) = 0.25 &\left[R(s1, a2, s2) + \gamma \max_{a'}(Q(s2, a'))\right] \\ &+ 0.75 \left[R(s1, a2, s3) + \gamma \max_{a'}(Q(s3, a'))\right] \\ &= 0.25[0 + 1 \times \max(6, 8)] + 0.75[0 + 1 \times \max(16, 0)] \\ &= 14. \end{aligned}$$

So, we update the Q-table as follows:

	$a1$	$a2$
$s1$	12	14
$s2$	6	8
$s3$	16	0
$s4$	0	0
$s5$	0	0

When the agent in $s3$, it takes action $a1$ and lands in $s5$ with probability 1. There will be no further updates to the Q table.

(3) We can obtain the optimal policy from the updated Q table as follows:

Start at state $s1$. We note $Q(s1, a1) = 12$ and $Q(s1, a2) = 14$. Since $Q(s1, a2) > Q(s1, a1)$, take action $a2$ and arrive at state $s3$, which has a Q value of 16. The other option of arriving at $s2$ has a maximum reward of only 8. Then, from $s3$, take action $a1$ to arrive at the terminal state $s5$.

Therefore the optimal policy is $s1 \to a2 \to s3 \to a1 \to s5$.

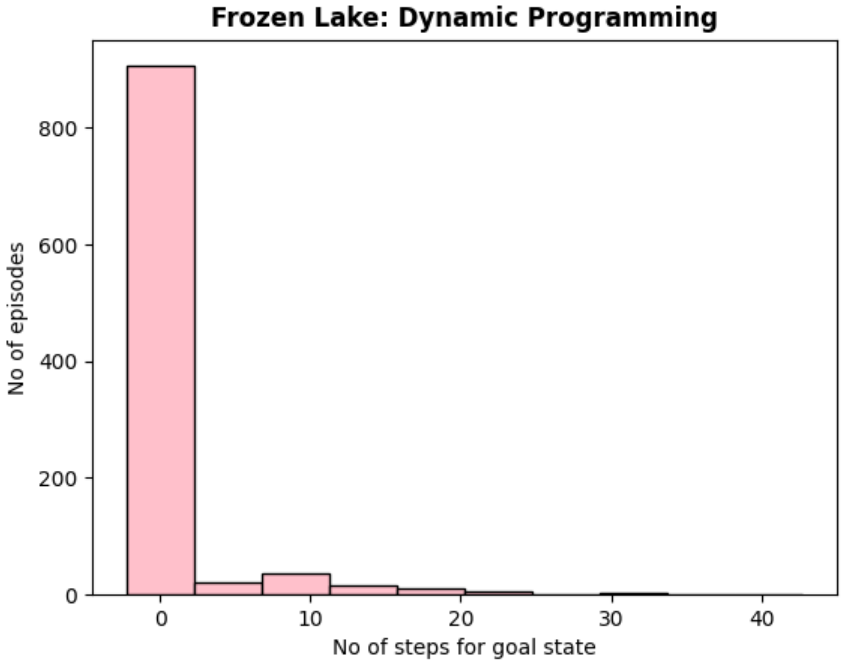
Problem 3 Frozen Lake

(1) I have implemented value iteration in `pset4a.py` within the class `DynamicProgramming`.

(2) Using the parameters $\gamma = 0.9$ and $\epsilon = 0.001$, the mean and the variance of the rewards over 1000 episodes are as follows:

- Mean: 0.10
- Variance: 0.09

The plot of histogram showing the number of steps to reach the goal state over 1000 episodes:

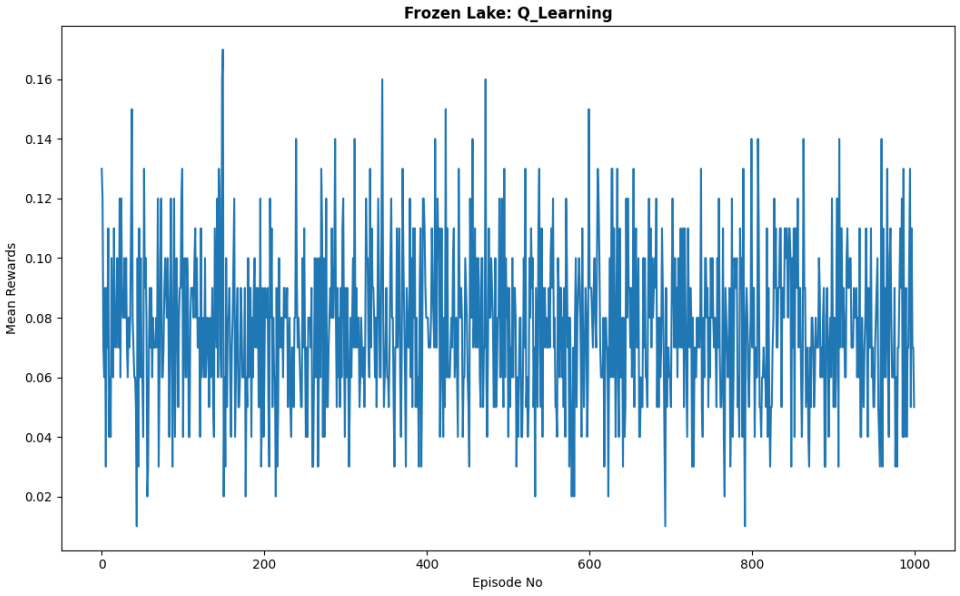


No, the agent does not always reach the goal state. The reason is that when the frozen lake is created with `is_slippery = True`, an action (LEFT, DOWN, RIGHT, or UP) does not necessarily take the agent to the location specified by the action. It performs the intended action with a probability of 1/3, and it moves the agent in the two directions (RIGHT or LEFT) perpendicular to the intended direction with probabilities of 1/3 each. Therefore, the agent can fall into the hole and will never reach the goal state.

The histogram supports this claim as it shows nearly 900 episodes (out of a total of 1000) corresponding to 0 (zero) steps, which means that this many episodes resulted in the agent falling into a hole and never reaching the goal state.

(3) I have implemented the model-free reinforcement learning in the form of Q-learning in `pset4a.py`.

(4) The plot of the mean returns over 100 episodes (based on max-Q values after every 1000 episodes) of the Q-learning agent is shown below.



The average reward obtained using the Q-learning algorithm appears to be similar to that from the value iteration. However, the Q-learning algorithm arrived at the convergence faster.

PSET 4

Anusha Murali **CS182** **November 21, 2023**

Problem 4 Decision Trees

(1) Let $B(p)$ as the entropy of a Boolean random variable that is true with probability p. Hence,

$$B(p) = -\left(p \log_2 p + (1-p) \log_2 (1-p)\right).$$

Let us first consider the attribute A. The entropy of the remainder of attribute A is,

$$\begin{aligned} \textit{Remainder}(A) &= \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right) = \left(\frac{4}{5}\right) B\left(\frac{2}{4}\right) + \left(\frac{1}{5}\right) B\left(\frac{0}{1}\right) \\ &= \left(\frac{4}{5}\right) \left(-0.5 \log_2(0.5) - 0.5 \log_2(0.5)\right) + \left(\frac{1}{5}\right) \left(0 - 1 \log_2 1\right) \\ &= 0.800. \end{aligned}$$

Similarly, the entropy of the remainder of attribute B is,

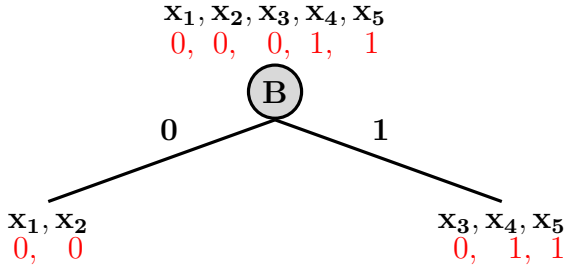
$$\begin{aligned} \textit{Remainder}(B) &= \left(\frac{3}{5}\right) B\left(\frac{2}{3}\right) + \left(\frac{2}{5}\right) B\left(\frac{0}{5}\right) \\ &= \left(\frac{3}{5}\right) \left(-(2/3) \log_2(2/3) - (1/3) \log_2(1/3)\right) + \left(\frac{2}{5}\right) \left(0 - \log_2 1\right) \\ &\approx 0.551. \end{aligned}$$

Finally the entropy of the remainder of attribute C is,

$$\begin{aligned} \textit{Remainder}(C) &= \left(\frac{2}{5}\right) B\left(\frac{1}{2}\right) + \left(\frac{3}{5}\right) B\left(\frac{1}{3}\right) \\ &= 0.4(-0.5 \log_2(0.5) - 0.5 \log_2(0.5)) + 0.6(-(1/3) \log_2(1/3) - (2/3) \log_2(2/3)) = 0.4 \cdot \\ &\approx 0.951. \end{aligned}$$

Since Attribute B has the smallest remainder among the three attributes, it produces the largest information gain. Therefore, we should first use Attribute B to divide the dataset.

Using Attribute B, the samples x_1 and x_2, which have values 0, are correctly classified as $B = 0$. The resulting decision tree is shown below:



Now, we need to further divide the remaining samples, x_3, x_4, x_4 using either Attribute A or Attribute C.

Let us first consider the attribute A. The entropy of the remainder of attribute A is,

$$\begin{aligned} \textit{Remainder}(A) &= \left(\frac{2}{3}\right) B\left(\frac{2}{2}\right) + \left(\frac{1}{3}\right) B\left(\frac{0}{1}\right) \\ &= \left(\frac{2}{3}\right)(-1\log_2 1 - 0) + \left(\frac{1}{3}\right)(0 - 1\log_2 1) \\ &= 0. \end{aligned}$$

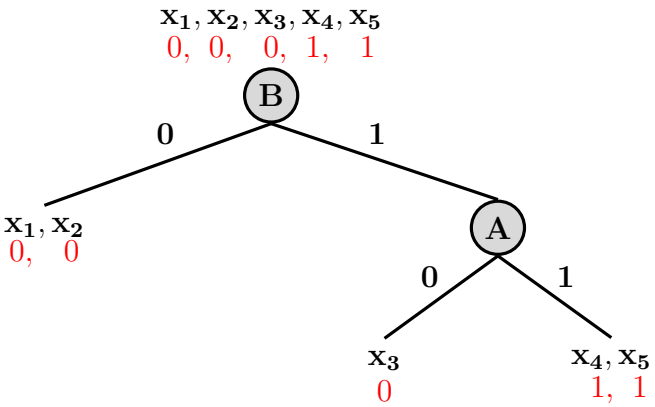
Now, considering Attribute , we find,

$$\begin{aligned} \textit{Remainder}(C) &= \left(\frac{1}{3}\right) B\left(\frac{1}{1}\right) + \left(\frac{2}{3}\right) B\left(\frac{1}{2}\right) \\ &= \left(\frac{1}{3}\right)(-1\log_2 1 - 0) + \left(\frac{2}{3}\right)(-0.5\log_2(0.5) - 0.5\log_2(0.5)) \\ &= 2/3 \approx 0.667. \end{aligned}$$

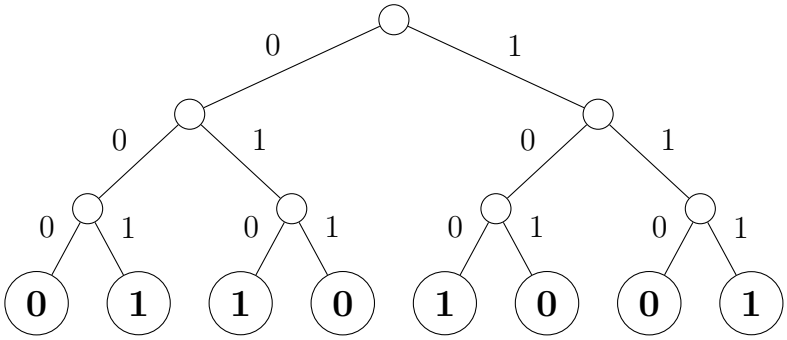
Therefore, Attribute A has the smallest remainder among A and C. Hence, we will divide x_3, x_4, x_5 using Attribute A.

Using Attribute A, the samples, x_3, x_4, x_5, will be divided into a group with only x_3 in it. This has an output value of 0, and corresponds to A's value of 0, and into another group with x_4, and x_5 in it. This latter group has an output value of 1, and corresponds to A's value of 1.

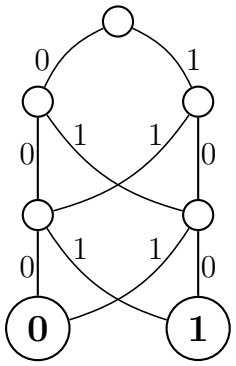
The resulting decision tree is shown below:



(2) (a) A minimal-sized decision tree for the three-input XOR function is shown below:



(b) A minimal-sized decision graph for the three-input XOR function is shown below:



PSET 4

Anusha Murali **CS182** **November 21, 2023**

Problem 5 Drones (Neural Networks)

(1) Let $f(x, z) = w_0 + w_1 x + w_2 z = 5 + x - 5z$. Therefore, we find,

(a) $f(2, 1) = 5 + x - 5z = 5 + 2 - 5 = 2 > 0$. Hence (2,1) is in Class 1.

(b) $f(1, 4) = 5 + x - 5z = 5 + 1 - 20 = -14 < 0$. Hence (1, 4) is in Class 0.

(c) $f(2, 3) = 5 + x - 5z = 5 + 2 - 15 = -8 < 0$. Hence (2, 3) is in Class 0.

(d) $f(5, 4) = 5 + x - 5z = 5 + 5 - 20 = -10 < 0$. Hence (5, 4) is in Class 0.

(e) $f(0, 0) = 5 + x - 5z = 5 > 0$. Hence (0, 0) is in Class 1.

(2) The output of the neural net is calculated as $f(x, z) = \text{Sigmoid}(w_0 + w_1 x + w_2 z)$. Let $w_0 + w_1 x + w_2 z = \alpha$. Therefore, our estimator is $\hat{f}(x, z) = \text{Sigmoid}(\alpha)$. Let us denote $\text{Sigmoid}(x) = \sigma(x)$. Hence,

$$\hat{f}(x, z) = \text{Sigmoid}(w_0 + w_1 x + w_2 z) = \text{Sigmoid}(\alpha) = \frac{1}{1 + e^{-\alpha}} = \sigma(\alpha).$$

Partial differentiating $\sigma(\alpha)$ with respect to w_0, w_1 and w_2, we obtain,

$$\frac{\partial \hat{f}}{\partial w_0} = \sigma(\alpha)(1 - \sigma(\alpha)), \qquad \frac{\partial \hat{f}}{\partial w_1} = \sigma(\alpha)(1 - \sigma(\alpha)) \cdot x, \qquad \frac{\partial \hat{f}}{\partial w_2} = \sigma(\alpha)(1 - \sigma(\alpha)) \cdot z.$$

It is given that the loss function is $L = -(y \cdot \log(f(x, z)) + (1 - y) \cdot \log(1 - f(x, z)))$. Therefore,

$$L = -y \cdot \log\Big(\hat{f}(x, z)\Big) - (1 - y) \cdot \log\Big(1 - \hat{f}(x, z)\Big)$$

$$\frac{\partial L}{\partial \hat{f}} = -\frac{y}{\hat{f}(x, z)} + \frac{1 - y}{1 - \hat{f}(x, z)} = \frac{\hat{f} - y}{\hat{f}(1 - \hat{f})} = \frac{\sigma(\alpha) - y}{\sigma(\alpha)(1 - \sigma(\alpha))}$$

Let the learning rate be η. Then our update equations for the weights are as follows:

$$w_0' = w_0 - \eta \frac{\partial L}{\partial w_0} = w_0 - \eta \frac{\partial L}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial w_0} = w_0 - \eta \left(\sigma(\alpha) - y\right) \tag{1}$$

$$w_1' = w_1 - \eta \frac{\partial L}{\partial w_1} = w_1 - \eta \frac{\partial L}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial w_1} = w_1 - \eta \left(\sigma(\alpha) - y\right) x \tag{2}$$

$$w_2' = w_2 - \eta \frac{\partial L}{\partial w_2} = w_2 - \eta \frac{\partial L}{\partial \hat{f}} \cdot \frac{\partial \hat{f}}{\partial w_2} = w_2 - \eta \left(\sigma(\alpha) - y\right) z \tag{3}$$

(a) **Update using Training Data 1**

For the first update, we use the data, $(x, z) = (2, 1); y = 0$.

Our initial weights are $w_0 = 5, w_1 = 1, w_2 = -5$.

Using the update equations derived above, we find,

$$\begin{aligned}
w_0' &= w_0 - \eta\,(\sigma(\alpha) - y) \\
&= 5 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right) = 4.911920292202212 \\
w_1' &= w_1 - \eta\,(\sigma(\alpha) - y)\,x \\
&= 1 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)2 = 0.8257530541957325 \\
w_2' &= w_2 - \eta\,(\sigma(\alpha) - y)\,z \\
&= -5 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)1 = -5.08268444695014
\end{aligned}$$

(b) **Update using Training Data 2**

For the next update, we use the data, $(x, z) = (1, 4); y = 1$.
We use the updated weights from above, which are,

$$\begin{aligned}
w_0 &= w_0' = 4.911920292202212 \\
w_1 &= w_1' = 0.8257530541957325 \\
w_2 &= w_2' = -5.08268444695014
\end{aligned}$$

Using the update equations again, we find,

$$\begin{aligned}
w_0' &= w_0 - \eta\,(\sigma(\alpha) - y) \\
&= w_0 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right) = 5.011920246249363 \\
w_1' &= w_1 - \eta\,(\sigma(\alpha) - y)\,x \\
&= w_1 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right)1 = 0.9257530034099853 \\
w_2' &= w_2 - \eta\,(\sigma(\alpha) - y)\,z \\
&= w_2 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right)4 = -4.6826846714578405
\end{aligned}$$

(c) **Update using Training Data 3**

For the next update, we use the data, $(x, z) = (2, 3); y = 1$.
We use the updated weights from above, which are,

$$\begin{aligned}
w_0 &= w_0' = 5.011920246249363 \\
w_1 &= w_1' = 0.9257530034099853 \\
w_2 &= w_2' = -4.6826846714578405
\end{aligned}$$

Using the update equations again, we find,

$$\begin{aligned}
w_0' &= w_0 - \eta\left(\sigma(\alpha) - y\right) \\
&= w_0 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right) = 5.111844488568275 \\
w_1' &= w_1 - \eta\left(\sigma(\alpha) - y\right)x \\
&= w_1 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right)2 = 1.1255855790521632 \\
w_2' &= w_2 - \eta\left(\sigma(\alpha) - y\right)z \\
&= w_2 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 1\right)3 = -4.383059043736215
\end{aligned}$$

(d) **Update using Training Data 4**

For the next update, we use the data, $(x, z) = (5, 4); y = 0$.
We use the updated weights from above, which are,

$$\begin{aligned}
w_0 &= w_0' = 5.111844488568275 \\
w_1 &= w_1' = 1.1255855790521632 \\
w_2 &= w_2' = -4.383059043736215
\end{aligned}$$

Using the update equations again, we find,

$$\begin{aligned}
w_0' &= w_0 - \eta\left(\sigma(\alpha) - y\right) \\
&= w_0 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right) = 5.111732394310733 \\
w_1' &= w_1 - \eta\left(\sigma(\alpha) - y\right)x \\
&= w_1 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)5 = 1.1250251705161354 \\
w_2' &= w_2 - \eta\left(\sigma(\alpha) - y\right)z \\
&= w_2 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)4 = -4.383506117494596
\end{aligned}$$

(e) **Update using Training Data 5**

For the next update, we use the data, $(x, z) = (0, 0); y = 0$.
We use the updated weights from above, which are,

$$\begin{aligned}
w_0 &= w_0' = 5.111732394310733 \\
w_1 &= w_1' = 1.1250251705161354 \\
w_2 &= w_2' = -4.383506117494596
\end{aligned}$$

Using the update equations again, we find,

$$\begin{aligned}
w_0' &= w_0 - \eta\,(\sigma(\alpha) - y) \\
&= w_0 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right) = 5.012331348739938 \\
w_1' &= w_1 - \eta\,(\sigma(\alpha) - y)\,x \\
&= w_1 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)0 = 1.1250251705161354 \\
w_2' &= w_2 - \eta\,(\sigma(\alpha) - y)\,z \\
&= w_2 - 0.1\left(\frac{1}{1+e^{-(w_0+w_1x+w_2z)}} - 0\right)0 = -4.383506117494596
\end{aligned}$$

Therefore, the final weights after the above 5 updates are,

$$\begin{aligned}
w_0 &= 5.012331348739938, \\
w_1 &= 1.1250251705161354, \\
w_2 &= -4.383506117494596.
\end{aligned}$$

Comment: Using a small Python script, I executed the above steps implementing the Gradient Descent algorithm until the weights have reasonably converged. The converged values are as follows:

$$\begin{aligned}
w_0 &= -5.418224568735031, \\
w_1 &= -5.841145131697083, \\
w_2 &= 7.297350955235475.
\end{aligned}$$

I further confirmed that using the above weights, the class values output by the neural net are the same as the ideal ones given in the table in the problem.

(3) I have implemented the `train` and `predict` methods for the neural net in `pset4b.py`.

PSET 4

Anusha Murali **CS182** **November 21, 2023**

Problem 6

(1) I worked on this by myself. I did not use any other resources besides the lecture slides and the textbook.

(2) I spent 35 hours (15 on the coding part, and 20 on the theory part) on this assignment.