Intro To Artificial Intelligence - Exercise 5

Eran Ston (206704512) and Oded Vaalany (208230474)

July 17, 2024

1 Value Iteration

1.1

Now we want to use MDP with the following transition probabilities:

- S = {s1, s2, s3, s4, s5, s6, s7, s8, s9}
- A = {U, D, L, R}
- $P(s|s', A) = 1$ where we need to do A from s to s'
- $R(s) = -0.05$ for all $s \notin \{s_5, s_7, s_9\}$
- $R(s_5) = -10$
- $R(s_7) = 15$
- $R(s_9) = 30$
- $\gamma = 0.99$

Values of states after each iteration:

step	s1	s2	s3	s4	s5	s6	s7	s8	s9
0	0	0	0	0	0	0	0	0	0
1	-0.05	-0.05	-0.05	-0.05	-10	-0.05	15	-0.05	30
2	-0.0995	-0.0995	-0.0995	14.8	-10	29.65	15	29.65	30
3	14.602	-0.148505	29.3035	14.8	-10	29.65	15	29.65	30
4	14.602	28.960465	29.3035	14.8	-10	29.65	15	29.65	30
5	28.6208	28.96046	29.3035	14.8	-10	29.65	15	29.65	30
6	28.6208	28.96046	29.3035	28.284592	-10	29.65	15	29.65	30
7	28.6208	28.96046	29.3035	28.284592	-10	29.65	15	29.65	30

Optimal actions after each iteration:

step	s1	s2	s3	s4	s5	s6	s7	s8	s9
0	L	L	L	L	L	L	L	L	L
1	R	R	U	U	L	U	L	R	L
2	U	R	U	U	L	U	L	R	L
3	U	R	U	U	L	U	L	R	L
4	U	R	U	U	L	U	L	R	L
5	R	R	U	U	L	U	L	R	L
6	R	R	U	D	L	U	L	R	L
7	R	R	U	D	L	U	L	R	L

The optimal policy is:

←	→	←
↓	←	↑
→	→	↑

1.2

Now we want to use stochastic MDP with the following transition probabilities:

- S = {s1, s2, s3, s4, s5, s6, s7, s8, s9}
- A = {U, D, L, R}
- $P(s|s', A) = 0.9$ where we need to do A from s to s'
- $P(s'|s', A) = 0.9$ where A is not legitimate move for the state
- $P(s|s', A) = \frac{0.1}{\text{number of neighbors -1}}$ where A is not possible from s to s'(neighbors)
- $R(s) = -0.05$ for all $s \notin \{s_5, s_7, s_9\}$
- $R(s_5) = -10$
- $R(s_7) = 15$
- $R(s_9) = 30$
- $\gamma = 0.99$

Values of states after each iteration:

step	s1	s2	s3	s4	s5	s6	s7	s8	s9
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	-0.05000	-0.05000	-0.05000	-0.05000	-10.00000	-0.05000	15.00000	-0.05000	30.00000
2	-0.09950	-0.42785	-0.09950	12.81752	-10.00000	26.18252	15.00000	26.92750	30.00000
3	11.32806	-0.63858	23.23627	12.81507	-10.00000	26.18007	15.00000	26.92750	30.00000
4	11.30501	20.71926	23.21323	13.38074	-10.00000	27.33520	15.00000	26.92750	30.00000
5	19.73555	20.69758	26.35687	13.37960	-10.00000	27.33405	15.00000	26.92750	30.00000
6	19.71613	23.91588	26.35370	17.78188	-10.00000	27.48966	15.00000	26.92750	30.00000
7	23.01945	23.91210	26.81096	17.76457	-10.00000	27.48951	15.00000	26.92750	30.00000

Optimal actions after each iteration:

step	s1	s2	s3	s4	s5	s6	s7	s8	s9
0	L	L	L	L	L	L	L	L	L
1	L	L	L	L	L	L	L	L	L
2	R	D	L	U	L	U	L	R	L
3	U	L	U	U	L	U	L	R	L
4	U	R	U	U	L	U	L	R	L
5	R	R	U	U	L	U	L	R	L
6	R	R	U	D	L	U	L	R	L
7	R	R	U	D	L	U	L	R	L

The optimal policy is:

←	→	←
↓	←	↑
→	→	↑

The values of the optimal policy in the stochastic MDP are lower the values of the optimal policy in the deterministic MDP. This is because the stochastic MDP has a probability of transitioning to a state that is not the desired state, which causes the values to be lower.

1.3

Given the following policy: $a_1 = \uparrow \quad a_2 = \rightarrow \quad a_3 = \uparrow \quad a_4 = \uparrow \quad a_5 = * \quad a_6 = \uparrow \quad a_7 = * \quad a_8 = \leftarrow \quad a_9 = *$
So lets define the following Parameters:

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$R = \begin{bmatrix} -0.05 & -0.05 & -0.05 & -0.05 & -10 & -0.05 & 15 & -0.05 & 30 \end{bmatrix}^T$$

$$P = \begin{bmatrix} 0 & 0.1 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & 0 & 0.9 & 0 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 \\ 0.05 & 0 & 0 & 0 & 0.05 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0.05 & 0 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 & 0.05 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Using the following formula: $V = (I - \gamma[P])^{-}1R$ we can calculate the values of the state using the "excat" algorithm.

$$\begin{bmatrix} 14.39238 & 24.09080 & 26.85003 & 13.53242 & -10.00000 & 27.51408 & 15.00000 & 26.92750 & 30.00000 \end{bmatrix}^T$$

The values of the states of the stochastic MDP:

$$\begin{bmatrix} 23.01945 & 23.91210 & 26.81096 & 17.76457 & -10.00000 & 27.48951 & 15.00000 & 26.92750 & 30.00000 \end{bmatrix}^T$$

When comparing the two vectors, we can see how different policies can lead to different state values. In this case, the state values differ because the policies are different. For example, the value of state s_1 in the stochastic MDP is 23.01945, while under the given policy, it is 14.39238. This difference arises because the stochastic MDP finds a better policy for s_1, ultimately leading to a higher score, whereas the given policy is not optimal.

However, when looking at the values of s_2, the non-optimal policy appears to outperform the optimal policy from the stochastic MDP. This highlights the importance of the number of iterations. The stochastic algorithm first searches for a good policy wach iteration and only after achieving the optimal policy do the remaining iterations help it converge.

2 POMDP

2.1

calculate $P(TL)$ for d=1

$$\begin{aligned} P(TL) &= \frac{P(GL|TL)P(TL)}{P(GL|TL)P(TL) + P(GL|TR)P(TR)} \\ &= \frac{0.5P(GL|TL)}{0.5(P(GL|TL) + P(GL|TR))} \\ &= \frac{0.45}{0.45 + 0.05} = 0.9 \end{aligned}$$

calculate $P(TR)$ for d=1

$$\begin{aligned} P(TR) &= \frac{P(GL|TR)P(TR)}{P(GL|TL)P(TR) + P(GL|TR)P(TR)} \\ &= \frac{0.5P(GL|TR)}{0.5(P(GL|TL) + P(GL|TR))} \\ &= \frac{0.1}{0.1 + 0.9} = 0.1 \end{aligned}$$

calculate $P(TL)$ for d=2

$$\begin{aligned} P(TL) &= \frac{P(GL|TL)P(TL)}{P(GL|TL)P(TL) + P(GL|TR)P(TR)} \\ &= \frac{0.5P(GL|TL)}{0.5(P(GL|TL) + P(GL|TR))} \\ &= \frac{0.6}{0.6 + 0.4} = 0.6 \end{aligned}$$

calculate $P(TR)$ for d=2

$$\begin{aligned} P(TR) &= \frac{P(GL|TR)P(TR)}{P(GL|TL)P(TL) + P(GL|TR)P(TR)} \\ &= \frac{0.5P(GL|TR)}{0.5(P(GL|TL) + P(GL|TR))} \\ &= \frac{0.4}{0.6 + 0.4} = 0.4 \end{aligned}$$

Finally we can write when d=1: $< 0.9|0.1 >$ and when d=2: $< 0.6|0.4 >$

2.2

Since we konw the probabilities after hearing a roar from the left. We can calculate the probability of the tiger being on the left side. donate the probability to be close to door as p (means d=1)

$$P(TL) = p \cdot P(TL, d=1) + (1-p) \cdot P(TL, d=2) = p \cdot 0.9 + (1-p) \cdot 0.6 = 0.9p - 0.6p + 0.6 = 0.3p + 0.6$$
$$P(TR) = p \cdot P(TR, d=1) + (1-p) \cdot P(TR, d=2) = p \cdot 0.1 + (1-p) \cdot 0.4 = 0.1p - 0.4p + 0.4 = 0.4 - 0.3p$$

2.3

Let's compute the expected return of the following conditional plans, assumie the agent starts in d=2:
$P(GL) = P(GL|TL) \cdot P(TL) + P(GL|TR) \cdot P(TR) = 0.9 * 0.5 + 0.1 * 0.5 = 0.5$
Plan 1:

- Move To d=1: Reward = -2
- Listen: Reward = -1
- If GR then OL :

$$E[OL|GR, d=1] = P(TL|GR, d=1) \cdot -100 + P(TR|GR, d=1) \cdot 10 = 0.1 \cdot (-100) + 0.9 \cdot 10 = -1$$

The expected return of plan 1 is:

$$R(*, move, 1) + R(*, listen, 1) + P(GR) \cdot E[OL|GR, d=1] = -2 - 1 + 0.5 * (-1) = -3.5$$

Plan 2:

- Listen: Reward = -1
- If GL
  - Listen: Reward = -1
- Else GR
  - OL: Reward = $E[OL|GR, d=2] = P(TL|GR, d=2) \cdot -50 + P(TR|GR, d=2) \cdot 10 = 0.4 \cdot (-50) + 0.6 \cdot 10 = -14$

The expected return of plan 2 is:

$$R(*, listen, 2) + P(GL) \cdot R(*, listen, 2) + P(GR) \cdot E[OL|GR, d=2] = -1 + 0.5 \cdot (-1) + 0.5 \cdot (-14) = -8.5$$

2.4

Given that the probability to get GL is 0.5

$$\begin{aligned}
P(GL, GL) &= P(GL|TL, d=1)P(GL|TL, d=1)P(TL) + P(GL|TR, d=1)P(GL|TR, d=1)P(TR) = 0.82 \cdot P(TL)\\
P(GL, GR) &= P(GL|TL, d=1)P(GR|TL, d=1)P(TL) + P(GL|TR, d=1)P(GR|TR, d=1)P(TR) = 0.18 \cdot P(TL)\\
P(GR, GR) &= P(GR|TL, d=1)P(GR|TL, d=1)P(TL) + P(GR|TR, d=1)P(GR|TR, d=1)P(TR) = 0.82 \cdot P(TL)\\
P(GR, GL) &= P(GR|TL, d=1)P(GL|TL, d=1)P(TL) + P(GR|TR, d=1)P(GL|TR, d=1)P(TR) = 0.18 \cdot P(TL)
\end{aligned} \tag{5}$$

Now we can calculate the probabilities:

$$\begin{aligned}
P(TL|GL, GL) &= \frac{P(GL, GL|TL, d=1)P(TL)}{P(GL, GL)} = \frac{P(GL|TL, d=1)P(GL|TL, d=1)P(TL)}{P(GL, GL)} = \frac{0.81}{0.82}\\
P(TL|GL, GR) &= \frac{P(GL, GR|TL, d=1)P(TL)}{P(GL, GR)} = \frac{P(GL|TL, d=1)P(GR|TL, d=1)P(TL)}{P(GL, GR)} = \frac{0.09}{0.18}\\
P(TL|GR, GL) &= \frac{P(GR, GL|TL, d=1)P(TL)}{P(GR, GL)} = \frac{P(GR|TL, d=1)P(GL|TL, d=1)P(TL)}{P(GR, GL)} = \frac{0.09}{0.18}\\
P(TL|GR, GR) &= \frac{P(GR, GR|TL, d=1)P(TL)}{P(GR, GR)} = \frac{P(GR|TL, d=1)P(GR|TL, d=1)P(TL)}{P(GR, GR)} = \frac{0.01}{0.82}
\end{aligned} \tag{6}$$

We can learn from this calcuation, that the more we explore the problem the more we can understand where the tiger might be. This helps us to make a better decision , since we open the gap between the probabilities of the tiger being on the left or right side given the observations. We can see that it converge very quickly to a belif, and the inital belif makes a difference.

3

Given a robot that throw a draft in a 1D line. The robot try to throw the draft to location μ^t but its actions are noisy and therefore it have a probability of $y^t = \mu^t + z\ \ ,\ z \sim N(0, \sigma)$

3.1 We would like to calculate the agent's policy $\pi(y|\mu)$ in trems of likelihood

The likelihood of the observation y^t given the state μ^t is $P(y^t|\mu^t) = N(y^t|\mu^t, \sigma)$

$$\begin{aligned} \mathcal{L}(\pi(y|\mu)) = \mathcal{L}(P(y^t|\mu^t, \sigma)) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y^t - \mu^t)^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z^t)^2}{2\sigma^2}} \end{aligned} \tag{7}$$

3.2 Express the update rule

The update rule is the following:

$$\theta^{t+1} = \theta^t + \alpha G^t \nabla \log(\pi(A_t|S_t, \theta_t)) \tag{8}$$

Since in our case we would like to update μ^t so we could write

$$\begin{aligned} \mu_{t+1} &= \mu_t + \alpha G_t \nabla \log(\pi(A_t|S_t, \mu_t)) \\ &= \mu_t + \alpha \cdot G_t \cdot \nabla \left(\log(\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{\|y - \mu\|^2}{2\sigma^2})) \right) \\ &= \mu_t + \alpha \cdot G_t \nabla \left(\log(\frac{1}{\sqrt{2\pi\sigma^2}}) - \frac{\|y - \mu\|^2}{2\sigma^2} \right) \\ &= \mu_t + \alpha \cdot G_t \left(\nabla \log(\frac{1}{\sqrt{2\pi\sigma^2}}) - \nabla \frac{\|y - \mu\|^2}{2\sigma^2} \right) \\ &= \mu_t + \alpha \cdot G_t \left(\frac{y - \mu}{\sigma^2} \right) \end{aligned} \tag{9}$$

Where

$$G_t = \sum_{i=t+1}^{T} r_i$$

3.3

Let's define $r = -(m - y)^2$ as the reward function. we will learn μ using the algorithm

To implement the algorithm for 1 step we can write the following:

Lets calculate G_t

$$G_t = \mathbb{E}[r] = \mathbb{E}[-(m - y)^2] = -\mathbb{E}[(m - \mu - z)^2]$$

$$= -\mathbb{E}[(m - \mu)^2 - 2(m - \mu)z + z^2] = -\mathbb{E}[(m - \mu)^2] - \sigma^2 = -(m - \mu)^2 - \sigma^2$$

Using the update rule from the previous question we can write where μ_t, y_t are scalars:

$$\begin{aligned} \mu_{t+1} &= \mu_t + \alpha \cdot G_t \left(\frac{y_t - \mu_t}{\sigma^2} \right) \\ \mu_{t+1} &= \mu_t + \alpha \cdot \left(\frac{y_t - \mu_t}{\sigma^2} \right) \cdot (-(m - \mu)^2 - \sigma^2) \end{aligned} \tag{10}$$

We whould like to find the μ that maximize the reward, so we can take the derivative of the reward function and set it to 0:

$$\frac{\partial r}{\partial \mu} = 0 \Rightarrow 2(m - \mu) = 0 \Rightarrow \mu = m$$

3.4 We wished to calculate the expected change rate

$$
\begin{aligned}
\mathbb{E}[\mu_{t+1}-\mu_t] &= \mathbb{E}\Bigg[-\alpha\frac{y-\mu}{\sigma^2}\cdot(m-y)^2\Bigg] \\
&= \frac{\alpha}{\sigma^2}\mathbb{E}\Bigg[(z)\cdot(m-(\mu+z))^2\Bigg] \\
&= \frac{\alpha}{\sigma^2}\mathbb{E}\Bigg[(z)\cdot(m-\mu-z)^2\Bigg] \\
&= \frac{\alpha}{\sigma^2}\mathbb{E}\Bigg[(z)\cdot(m-\mu)^2-2(m-\mu)z+z^2\Bigg] \\
&= \frac{\alpha}{\sigma^2}\Bigg(\mathbb{E}\Bigg[(z)\cdot(m-\mu)^2\Bigg]-\mathbb{E}\Bigg[2(m-\mu)z^2\Bigg]+\mathbb{E}\Bigg[z^3\Bigg]\Bigg) \\
&= \frac{\alpha}{\sigma^2}\Bigg(-2(m-\mu)\sigma^2\Bigg) \\
&= -2\alpha(m-\mu)
\end{aligned}
\tag{11}
$$

3.5

Now when the robot learn from μ and σ we would like to find the optimal μ and σ of the learning.

$$
\begin{aligned}
\nabla\log(\pi(y|\mu,\sigma)) &= \nabla\log(\frac{1}{\sqrt{2\pi\sigma^2}}\cdot\exp(-\frac{\|y-\mu\|^2}{2\sigma^2})) \\
&= \nabla\log(\frac{1}{\sqrt{2\pi\sigma^2}})-\nabla\frac{\|y-\mu\|^2}{2\sigma^2} \\
\text{by } \mu &= \frac{y-\mu}{\sigma^2} \\
\text{by } \sigma &= \frac{\frac{-\sqrt{2\pi}}{2\pi\sigma^2}}{\frac{1}{\sqrt{2\pi\sigma^2}}}-\frac{-4\sigma(y-\mu)^2}{4\sigma^4} \\
&= -\frac{1}{\sigma}+\frac{(y-\mu)^2}{\sigma^3}
\end{aligned}
\tag{12}
$$

When we have the gradient we can use the update rule to find the optimal μ and σ:

$$
\begin{aligned}
\mu_{t+1} &= \mu_t+\alpha\cdot G_t\Bigg(\frac{y-\mu}{\sigma^2}\Bigg) \\
\sigma_{t+1} &= \sigma_t+\alpha\cdot G_t\Bigg(-\frac{1}{\sigma}+\frac{(y-\mu)^2}{\sigma^3}\Bigg)
\end{aligned}
\tag{13}
$$

we would like to find the optimal μ and σ that maximize the reward, so we can take the derivative of the reward function and set it to 0: the μ converge is the same as we found in previous sections, now we will look for the converge of σ: we will derive the update rule for σ and set it to 0:

$$
\begin{aligned}
\nabla\sigma &= \nabla\alpha\dot{G}_t\dot{(}-\frac{1}{\sigma}+\frac{(y-\mu)^2}{\sigma^3})=0 \\
&= \nabla\alpha\dot{(}-(m-\mu)^2-\sigma^2)\dot{(}-\frac{1}{\sigma}+\frac{(y-\mu)^2}{\sigma^3})=0 \\
&= \nabla\alpha(\frac{(m-\mu)^2}{\sigma}-\frac{(y-\mu)^2(m-\mu)^2}{\sigma^3}+\sigma-\frac{(y-\mu)^2}{\sigma}) \\
(\mu=m) &= \nabla\alpha(\sigma-\frac{(y-m)^2}{\sigma}) \\
&= \alpha-\alpha\frac{(y-m)^2}{\sigma^2}=0 \\
&\iff \frac{(y-m)^2}{\sigma^2}=1 \\
&\iff \sigma=(y-m)
\end{aligned}
\tag{14}
$$

we get the σ need to be equal to z wich is the noise with variance of σ_{true}.