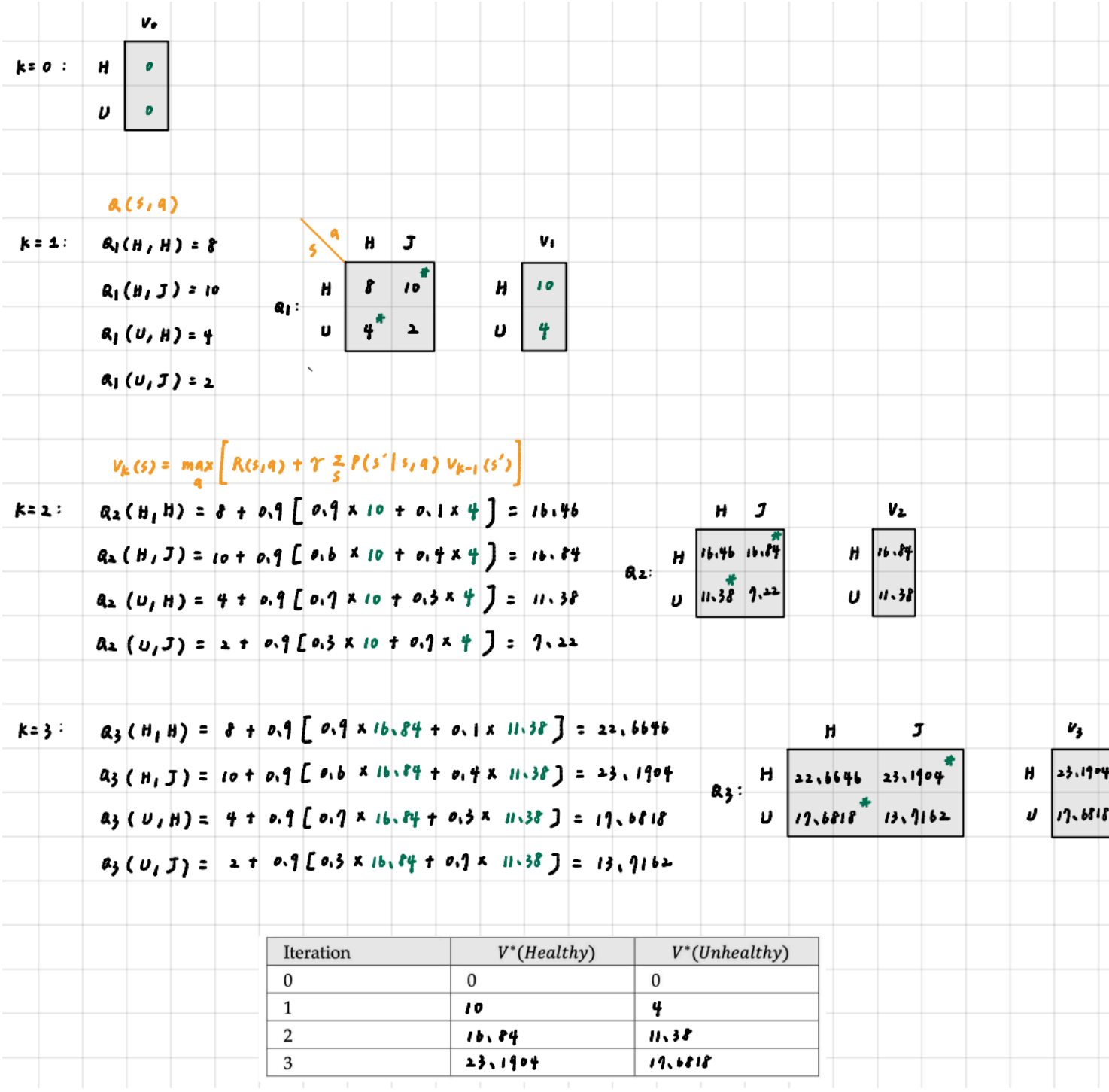
MDP and Reinforcement Learning

AI HW5

數據所 RE6121011 徐仁瓏

1-1 Value Iteration



Iteration	$V^*(Healthy)$	$V^*(Unhealthy)$
0	0	0
1	10	4
2	16.84	11.38
3	23.1904	17.6818

1-2 Policy Iteration

Cycle 0: $\begin{cases} \pi_0(H) = J \\ \pi_0(U) = H \end{cases}$

Cycle 1: <1> Eval

$$V(H) = 10 + 0.9\,[\,0.6 \times V(H) + 0.4 \times V(U)\,]$$
$$V(U) = 4 + 0.9\,[\,0.7 \times V(H) + 0.3 \times V(U)\,]$$
$$\Rightarrow \quad V(H) \doteq 80.18, \quad V(U) \doteq 74.68$$

<2> IMPV

$$Q(H,H) = 8 + 0.9\,[\,0.9 \times V(H) + 0.1 \times V(U)\,] = 79.667$$
$$Q(H,J) = 10 + 0.9\,[\,0.6 \times V(H) + 0.4 \times V(U)\,] = 80.182^{*}$$
$$Q(U,H) = 4 + 0.9\,[\,0.7 \times V(H) + 0.3 \times V(U)\,] = 74.699^{*}$$
$$Q(U,J) = 2 + 0.9\,[\,0.3 \times V(H) + 0.7 \times V(U)\,] = 70.699$$
$$\Rightarrow \begin{cases} \pi_1(H) = J \\ \pi_1(U) = H \end{cases}$$

Cycle 2: <1> Eval

$$V(H) = 10 + 0.9\,[\,0.6 \times V(H) + 0.4 \times V(U)\,]$$
$$V(U) = 4 + 0.9\,[\,0.7 \times V(H) + 0.3 \times V(U)\,]$$
$$\Rightarrow \quad V(H) \doteq 80.18, \quad V(U) \doteq 74.68$$

<2> IMPV

$$Q(H,H) = 8 + 0.9\,[\,0.9 \times V(H) + 0.1 \times V(U)\,] = 79.667$$
$$Q(H,J) = 10 + 0.9\,[\,0.6 \times V(H) + 0.4 \times V(U)\,] = 80.182^{*}$$
$$Q(U,H) = 4 + 0.9\,[\,0.7 \times V(H) + 0.3 \times V(U)\,] = 74.699^{*}$$
$$Q(U,J) = 2 + 0.9\,[\,0.3 \times V(H) + 0.7 \times V(U)\,] = 70.699$$
$$\Rightarrow \begin{cases} \pi_1(H) = J \\ \pi_1(U) = H \end{cases}$$

⋮

Cycle K: 已收斂，結果皆相同。

Iteration	Policy($\pi(Healthy)$)	Policy($\pi(Unhealthy)$)
0	eat junk food	eat healthy
1	eat junk food	eat healthy
2	eat junk food	eat healthy
Final	eat junk food	eat healthy

1-3 Q-Learning

Step	State	Action	Next State	Reward
1	Healthy	Eat Junk Food	Healthy	10
2	Healthy	Eat Healthy	Healthy	8
3	Healthy	Eat Junk Food	Unhealthy	10
4	Unhealthy	Eat Healthy	Healthy	4
5	Healthy	Eat Junk Food	Unhealthy	10

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \quad \max_{a'} Q(s',a') = V(s')$$

Q_0:

	H	J
H	0	0
U	0	0

step 1:

$$Q_1(H,J) = Q_0(H,J) + \alpha\left[r + \gamma \times V_0(H) - Q_0(H,J)\right]$$
$$= 0 + 0.5\left[10 + 0.9 \times 0 - 0\right] = 5$$

Q_1:

	H	J
H	0	5*
U	0	0

step 2:

$$Q_2(H,H) = Q_1(H,H) + \alpha\left[r + \gamma \times V_1(H) - Q_1(H,H)\right]$$
$$= 0 + 0.5\left[8 + 0.9 \times 5 - 0\right] = 6.25$$

Q_2:

	H	J
H	6.25*	5
U	0	0

step 3:

$$Q_3(H,J) = Q_2(H,J) + \alpha\left[r + \gamma \times V_2(U) - Q_2(H,J)\right]$$
$$= 5 + 0.5\left[10 + 0.9 \times 0 - 5\right] = 7.5$$

Q_3:

	H	J
H	6.25	7.5*
U	0	0

step 4:

$$Q_4(U,H) = Q_3(U,H) + \alpha\left[r + \gamma \times V_3(H) - Q_3(U,H)\right]$$
$$= 0 + 0.5\left[4 + 0.9 \times 7.5 - 0\right] = 5.375$$

Q_4:

	H	J
H	6.25	7.5*
U	5.375*	0

step 5:

$$Q_5(H,J) = Q_4(H,J) + \alpha\left[r + \gamma \times V_4(U) - Q_4(H,J)\right]$$
$$= 7.5 + 0.5\left[10 + 0.9 \times 5.375 - 7.5\right] = 11.16875$$

Q_5:

	H	J
H	6.25	11.16875*
U	5.375*	0

Step	0	1	2	3	4	5
Q(Healthy, Eat Healthy)	0	0	6.25	6.25	6.25	6.25
Q(Healthy, Eat Junk Food)	0	5	5	7.5	7.5	11.16875*
Q(Unhealthy, Eat Healthy)	0	0	0	0	5.375	5.375*
Q(Unhealthy, Eat Junk Food)	0	0	0	0	0	0

1-4 Compare Policies

三種方法（Value Iteration、Policy Iteration、Q-Learning）在該問題中得出的最佳策略是一樣的，即「健康時選擇吃垃圾食物；不健康時選擇吃健康食物」。這是因為三種方法基於相同的 MDP 定義，該問題的最佳策略唯一，所有算法均能正確收斂到相同的結果。

2-1 Formulate this problem as an MDP.

- States: 城市集合，例如$\{C_1, C_2, \ldots, C_n\}$
- Actions: 對每個城市 i，動作為嘗試移動到可到達的鄰近城市 j，即$\{R(i, j)\}$
- Rewards: 成功移動的獎勵為 $-v_{i,j}$，失敗的獎勵為 $-u_{i,j}$
- Transitions: 成功移動到鄰近城市 j 的機率為 $1 - \alpha_i$，失敗並留在城市 i 的機率為 α_i

2-2 Construct a deterministic state space model

- States: 保持與 MDP 相同的狀態集合，即城市集合$\{C_1, C_2, \ldots, C_n\}$
- Actions: 在每個城市 C_i，嘗試移動到鄰近城市 C_j。

在確定性模型中，城市之間的邊的權重表示從城市 C_i 到城市 C_j 的期望通勤時間。計算公式：

$$w_{i,j} = (1 - \alpha_i) \cdot v_{i,j} + \alpha_i \cdot (u_{i,j} + \text{期望時間}(C_i))$$

其中 $v_{i,j}$ 為成功通勤的時間，$u_{i,j}$ 為失敗停留的時間，α_i 失敗的機率。每次失敗後，依然需要重新嘗試通勤（即返回到起始城市的狀態），因此失敗的成本不僅是$u_{i,j}$，還包括未來所有可能的嘗試所花費的期望時間，這樣的設計可以反映「長期平均通勤時間」。MDP 的最佳策略與確定性模型的最短路徑是一致的，因為兩者的計算邏輯都基於期望效用最小化。

3-1 $\gamma = 0.0$

$\gamma = 0.0$ 意味著 agent 完全忽略未來的回報，專注於當前步驟的即時回報。長期回報策略無法被學習，因此代理可能學不到真正的最佳策略。若回報具有延遲性（例如需要多步行為才能獲得高回報），代理的表現會很差。

3-2 $\gamma = 0.9$

$\gamma = 0.9$ 代表 agent 會考慮未來回報，並在即時回報與長期回報之間尋求平衡。這樣的策略學習的穩定性增加，能適應延遲回報的問題。長期行為的策略變得更合理，回報最大化的表現更好。學習過程稍慢於 $\gamma = 0.0$ 的情況，但表現結果較佳。

3-3 $\gamma = 1.0$

$\gamma = 1.0$ 表示完全專注於未來回報。策略傾向於採取「看似高風險但長期有潛在高回報」的動作。在不穩定的回報環境中，可能會導致學習過程不穩定。

3-4 $\epsilon = 0.0$

$\epsilon = 0.0$ 意味著 agent 只選擇當前認為最優的動作，完全不進行探索。缺乏探索可能導致 agent 陷入局部最優解，而非全局最優解。如果初始策略接近最佳策略，學習會非常快速且穩定。如果初始策略不佳，agent 無法修正策略，表現可能非常糟糕。