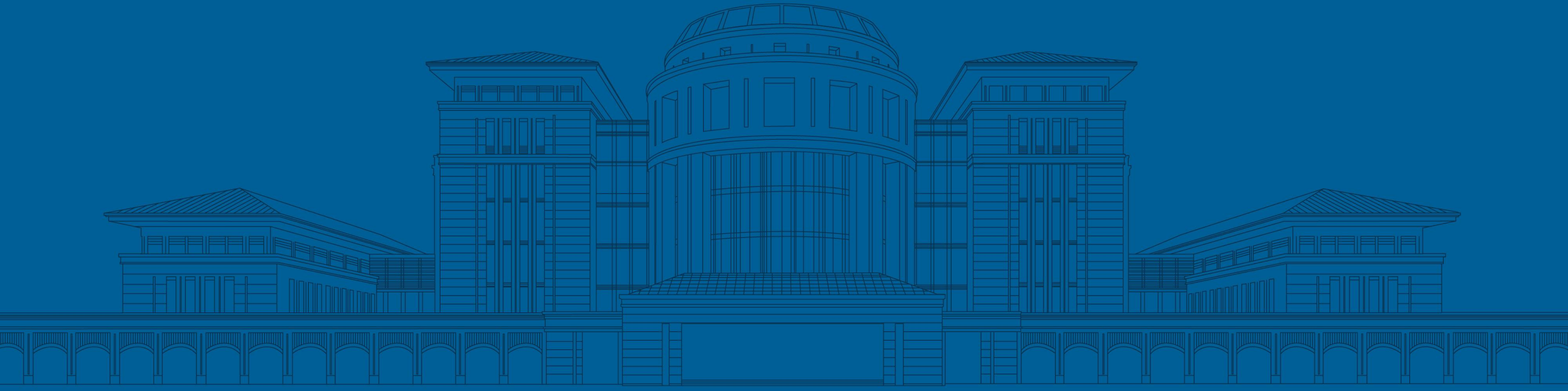
澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Group Report

speaker：Xiongyi Li

23th Nov 2023



Outline

- Deep Reinforcement Learning
  - Part II
    - DQN and Q-learning
    - SARSA
    - Improving

Revision

- Discounted return(折扣回报)

$$U_t \;=\; R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_n.$$

- Action-value function(动作价值函数)

$$Q_\pi(s_t, a_t) \;=\; \mathbb{E}_{S_{t+1},A_{t+1},\cdots,S_n,A_n}\Big[U_t \,\Big|\, S_t = s_t, A_t = a_t\Big].$$

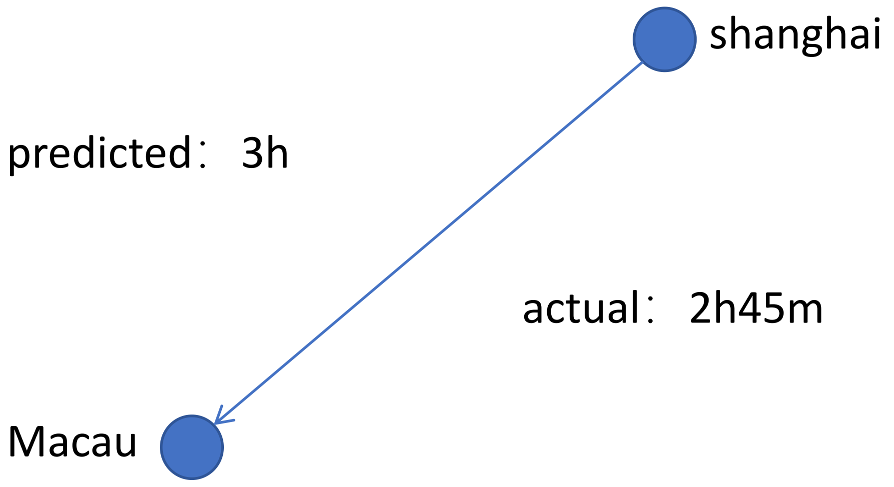
- Optimal action-value function(最优动作价值函数)

$$Q_\star(s_t, a_t) \;=\; \max_\pi \; Q_\pi(s_t, a_t), \qquad \forall\; s_t \in \mathcal{S}, \quad a_t \in \mathcal{A}.$$

DQN

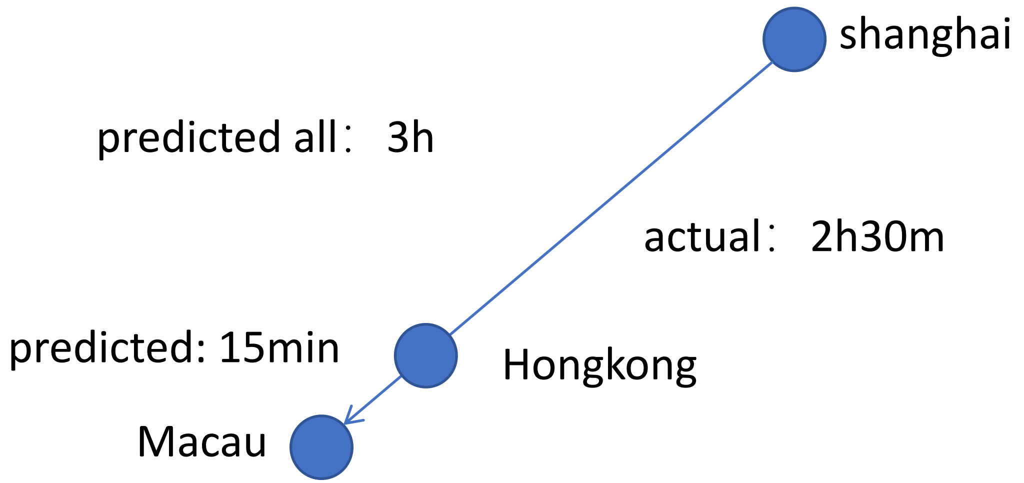
- Deep Q network
- To approximate the optimal action-value function
- Q(s,a;w)

DQN



- Temporal difference(时间差分)
- $Q(s,d;w)$ = 3h
- TD target: $y(t)$ = 2h45m
- Loss: $L = \frac{1}{2}(Q - y(t))^2$
- Gradient: $(Q - y(t))\frac{\partial Q}{\partial w}$
- Gradient descent: $w = w - \alpha(Q - y(t))\frac{\partial Q}{\partial w}$

DQN



- From S to H: 2h30m
- y(t) = 2h30m + 15m = 2h45m
- Loss: $L = \frac{1}{2}[Q - y(t)]^2$
- Gradient: $(Q - y(t))\frac{\partial Q}{\partial w}$
- Gradient descent: $w = w - \alpha(Q - y(t))\frac{\partial Q}{\partial w}$
- $T_{S\text{-}M} \approx T_{S\text{-}H} + T_{H\text{-}M}$
- Predicted date ≈ observed data + predicted data

DQN

- $U_t = R_t + \gamma R_t + \gamma^2 R_t + \gamma^3 R_t + \gamma^4 R_t + \ldots$
- $U_t = R_t + \gamma U_{t+1}$
- $Q(s_t,a_t;w)$ is estimate of expectation$[U_t]$
- $Q(s_{t+1},a_{t+1};w)$ is estimate of expectation$[U_{t+1}]$
- Thus: $Q(s_t,a_t;w) \approx r_t + \gamma Q(s_{t+1},a_{t+1};w)$

DQN

- Prediction: $Q(s_t,a_t;w_t)$
- TD target: $y(t) = r_t + Q(s_{t+1},a_{t+1};w_t)$

定理 4.1. 最优贝尔曼方程

$$\underbrace{Q_\star(s_t,a_t)}_{U_t \text{ 的期望}} = \mathbb{E}_{S_{t+1}\sim p(\cdot|s_t,a_t)}\Big[R_t + \gamma \cdot \underbrace{\max_{A\in\mathcal{A}} Q_\star(S_{t+1},A)}_{U_{t+1} \text{ 的期望}} \Big| S_t = s_t, A_t = a_t\Big].$$

- $Y(t) = r_t + \max Q(s_{t+1},a;w_t)$
- Loss: $L = \frac{1}{2}[Q(s_t,a_t;w) - y(t)]^2$
- Gradient descent: $w_{t+1} = w_t - \alpha(Q - y(t))\frac{\partial L}{\partial w}\Big|_{w = wt}$

SARSA

	第 1 种动作	第 2 种动作	第 3 种动作	第 4 种动作
第 1 种状态	380	-95	20	173
第 2 种状态	-7	64	-195	210
第 3 种状态	152	72	413	-80

- State-action-reward-state-action

$$Q_\pi(s_t, a_t) = \mathbb{E}_{S_{t+1}, A_{t+1}}\left[R_t + \gamma \cdot Q_\pi(S_{t+1}, A_{t+1}) \,\middle|\, S_t = s_t, A_t = a_t\right]$$

$$\approx r_t + \gamma Q_\pi(s_{t+1}, a_{t+1}) \qquad \text{TD target: } y_t$$

- Observe a transition(s_t, a_t, r_t, s_{t+1})
- TD target: $y_t = r_t + \gamma Q_\pi(s_{t+1}, a_{t+1})$
- TD error: $\delta_t = Q_\pi(s_t, a_t) - y_t$
- Update:$Q_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - \alpha\delta_t$

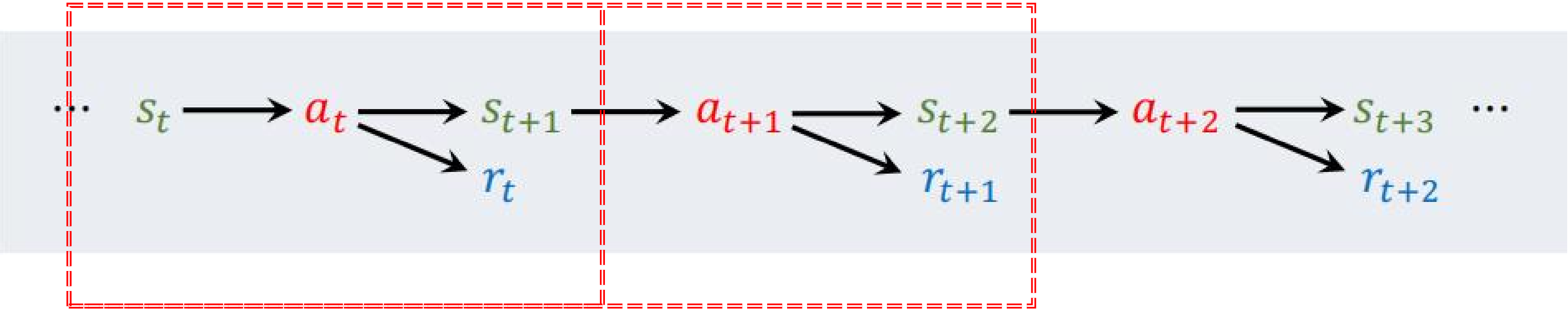
Q-learning

	第 1 种动作	第 2 种动作	第 3 种动作	第 4 种动作
第 1 种状态	380	-95	20	173
第 2 种状态	-7	64	-195	210
第 3 种状态	152	72	413	-80

$$Q_\pi(s_t, a_t) = \mathbb{E}_{S_{t+1},A_{t+1}}\left[R_t + \gamma \cdot Q_\pi(S_{t+1}, A_{t+1}) \,\middle|\, S_t = s_t, A_t = a_t\right]$$

- $Q^*(s_t,a_t;w) \approx \text{expectation}[R_t + Q^*(S_{t+1},A_{t+1})]$
- $A_{t+1} = \text{argmax}Q^*(S_{t+1},a)$
- $Q_\pi(s_t,a_t) \approx \underline{r_t + \max Q^*(S_{t+1},a)}$ TD target
- TD target: $y_t = r_t + \gamma \max Q^*(s_{t+1},a)$
- TD error: $\delta_t = Q^*(s_t,a_t) - y_t$
- Update: $Q^*(s_t,a_t) = Q^*(s_t,a_t) - \alpha\delta_t$

Multi-step TD target



$$U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^{n-t} R_n.$$

$$U_t = \left(\sum_{i=0}^{m-1} \gamma^i R_{t+i} \right) + \gamma^m U_{t+m}.$$

m-step TD target for **Sarsa**:

$$y_t = \sum_{i=0}^{m-1} \gamma^i \cdot r_{t+i} + \gamma^m \cdot Q_\pi(s_{t+m}, a_{t+m}).$$

One-step TD target for **Sarsa**:

$$y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1}).$$

Experience replay

- Transition(s_t,a_t,r_t,s_{t+1})
- Experience: all transitions
- Compared to TD learning:
  - use experience
  - eliminate correlation

Overestimation

- Bootstrapping(自举): TD target: $y(t) = r_t + Q(s_{t+1},a_{t+1};w_t)$
- Maximization: TD target: $y_t = r_t + \gamma \max Q^*(s_{t+1},a)$
- Solutions:
  - use an another target network to compute TD targets
  - use double DQN

Dueling network

- State-value function(状态价值函数):

$$V_\pi(s) = \mathbb{E}_{A\sim\pi}\Big[Q_\pi(s, A)\Big].$$

- Optimal state-value function(最优状态价值函数):

$$V^\star(s) = \max_\pi V_\pi(s).$$

- optimal advantage function(最优优势函数):

$$A^\star(s,a) = Q^\star(s,a) - V^\star(s).$$

Dueling network

Theorem 1: $V^\star(s) = \max_{a} Q^\star(s,a)$.

- $\max_a A^*(s,a) = \max_a Q^*(s,a) - V^*(s) = 0$

Theorem 2: $Q^\star(s,a) = V^\star(s) + A^\star(s,a) - \max_{a} A^\star(s,a)$.

- to avoid non-identifiability(不唯一性)

Dueling network

- Neural network $A(s,a;w^A)$ approximates $A^*(s,a)$
- Neural network $V(s;w^V)$ approximates $V^*(s)$
- $Q(s,a;w^A,w^V) = V(s;w^V) + A(s,a;w^A) - \max A(s,a;w^A)$

Thank You!

Avenida da Universidade, Taipa, Macau, China
Email： mc35289@um.edu.mo Website： www.um.edu.mo