# Reinforcement Learning in Finance

Haotian Deng

Shanghai University of Finance and Economics

April 23, 2024

# Contents

# Basic Concepts

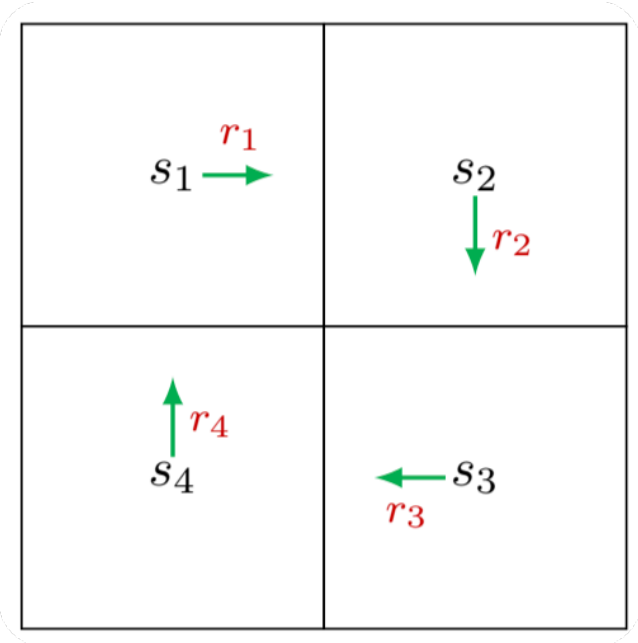


Figure 1: Gird World.

- state-action-reward

$$S_t \xrightarrow{A_t} R_{t+1}, S_{t+1} \xrightarrow{A_{t+1}} \cdots$$

- discounted return

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\ &= R_{t+1} + \gamma \left( R_{t+2} + \gamma R_{t+3} + \dots \right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

- state value function

$$\begin{aligned} v_\pi(s) &= \mathbb{E}\left[ G_t \mid S_t = s \right] \\ &= \mathbb{E}\left[ R_{t+1} + \gamma G_{t+1} \mid S_t = s \right] \\ &= \underbrace{\mathbb{E}\left[ R_{t+1} \mid S_t = s \right]}_{\text{immediate rewards}} \\ &\quad + \underbrace{\gamma \mathbb{E}\left[ G_{t+1} \mid S_t = s \right]}_{\text{discounted future rewards}} \end{aligned}$$

# Bellman Equation

Bootstrapping: the returns rely on each other.

$$
\begin{aligned}
v_1 &= r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = r_1 + \gamma\left(r_2 + \gamma r_3 + \cdots\right) = r_1 + \gamma v_2 \\
v_2 &= r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots = r_2 + \gamma\left(r_3 + \gamma r_4 + \cdots\right) = r_2 + \gamma v_3 \\
v_3 &= r_3 + \gamma r_4 + \gamma^2 r_1 + \cdots = r_3 + \gamma\left(r_4 + \gamma r_1 + \cdots\right) = r_3 + \gamma v_4 \\
v_4 &= r_4 + \gamma r_1 + \gamma^2 r_2 + \cdots = r_4 + \gamma\left(r_1 + \gamma r_2 + \cdots\right) = r_4 + \gamma v_1
\end{aligned}
$$

$$
\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{\mathrm{v}}
\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}
+
\begin{bmatrix} \gamma v_2 \\ \gamma v_3 \\ \gamma v_4 \\ \gamma v_1 \end{bmatrix}
\underbrace{\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}}_{\mathrm{r}}
+\gamma
\underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathrm{P}}
\underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}}_{\mathrm{v}}
$$

Bellman Equation: $\mathrm{v} = \mathrm{r} + \gamma \mathrm{P}\mathrm{v} \;\Rightarrow\; \mathrm{v} = (\mathrm{I} - \gamma \mathrm{P})^{-1}\mathrm{r}$

# Bellman Equation (element-wise form)

$$
\begin{aligned}
v_\pi(s) &= \mathbb{E}\left[R_{t+1} \mid S_t = s\right] + \gamma \mathbb{E}\left[G_{t+1} \mid S_t = s\right] \\
&= \underbrace{\sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{r \in \mathcal{R}} p(r \mid s, a) r}_{\text{mean of immediate rewards}} \\
&\quad + \underbrace{\sum_{a \in \mathcal{A}} \pi(a \mid s) \gamma \sum_{s' \in \mathcal{S}} p\left(s' \mid s, a\right) v_\pi\left(s'\right)}_{\text{mean of discounted future rewards}} \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) \Big[ \sum_{r \in \mathcal{R}} p(r \mid s, a) r + \gamma \sum_{s' \in \mathcal{S}} p\left(s' \mid s, a\right) v_\pi\left(s'\right) \Big] \\
v_\pi(s) &= r_\pi(s) + \gamma \sum_{s' \in \mathcal{S}} p_\pi\left(s' \mid s\right) v_\pi\left(s'\right)
\end{aligned}
$$

# Bellman Optimality Equation

- state-action value

$$q_\pi(s,a) \doteq \mathbb{E}\Big[G_t \mid S_t = s, A_t = a\Big]$$
$$= \sum_{r\in\mathcal{R}} p(r \mid s,a) r + \gamma \sum_{s'\in\mathcal{S}} p\left(s' \mid s,a\right) v_\pi\left(s'\right)$$

- Bellman optimality equation

$$v(s) = \max_{\pi(s)} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left[\sum_{r\in\mathcal{R}} p(r \mid s,a) r + \gamma \sum_{s'\in\mathcal{S}} p\left(s' \mid s,a\right) v\left(s'\right)\right]$$
$$= \max_{\pi(s)} \sum_{a\in\mathcal{A}} \pi(a \mid s) q(s,a) = \max_{a\in\mathcal{A}} q(s,a)$$

- (deterministic) greedy optimal policy

$$\pi(a \mid s) = \begin{cases} 1 & a = a^* \\ 0 & a \neq a^* \end{cases}, \quad \text{where } a^* = \arg\max_{a\in\mathcal{A}} q(s,a)$$

$$v^* = \max_\pi (r_\pi + \gamma P_\pi v^*) = r_{\pi^*} + \gamma P_{\pi^*} v^* = v_{\pi^*} \geq v_\pi$$

# Optimization Problem for a Social Planner in discrete-time

$$\max_{\{c_t,k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t. } c_t + k_{t+1} = f(k_t) \qquad \Rightarrow -u'[c_t] + \beta u'[c_{t+1}] f'(k_{t+1}) = 0$$

- Traditional method:

$$\begin{aligned} V^*[k_0] &= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u\Big[f(k_t) - k_{t+1}\Big] \\ &= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \Big\{u\Big[f(k_0) - k_1\Big] + \beta\Big[u(f(k_1) - k_2) + \cdots\Big]\Big\} \\ &= \max_{k_1} \Big\{u\Big[f(k_0) - k_1\Big] + \beta V^*[k_1]\Big\} \end{aligned}$$

$$\Downarrow \text{ Bellman Equation}$$

$$V^*[k_t] = \max_{k_{t+1}} \Big\{u\Big[f(k_t) - k_{t+1}\Big] + \beta V^*[k_{t+1}]\Big\}$$

- RL method: $v^*(k_t) = \max_{c(t)} \Big\{u[c(t)] + \beta v^*(k_{t+1})\Big\}$

# Merton's Portfolio Problem in continuous-time

- The state and action at time $t$ is $(t, W_t)$ and $(\pi_t, c_t)$.

$$dW_t = \Big[(r + \pi_t \cdot (\mu - r)) \cdot W_t - c_t\Big] \cdot dt + \pi_t \cdot \sigma \cdot W_t \cdot dz_t$$

- The reward per unit time at time $t$ is

$$\begin{cases} U(c_t) = \frac{c_t^{1-\gamma}}{1-\gamma}, & t < T \\ B(T) \cdot U(W_t) = \varepsilon^\gamma \cdot \frac{W_t^{1-\gamma}}{1-\gamma}, & t = T \end{cases}$$

- The return at time $t$ is the accumulated discounted reward

$$\int_t^T e^{-\rho(s-t)} \cdot \frac{c_s^{1-\gamma}}{1-\gamma} \cdot ds + \frac{e^{-\rho(T-t)} \cdot \epsilon^\gamma \cdot W_T^{1-\gamma}}{1-\gamma}$$

- Our goal is to determine optimal allocation $\pi(t, W_t)$ and consumption $c(t, W_t)$ at any time $t$ to maximize

$$\mathbb{E}\left[\int_t^T \frac{e^{-\rho(s-t)} \cdot c_s^{1-\gamma}}{1-\gamma} \cdot ds + \frac{e^{-\rho(T-t)} \cdot B(T) \cdot W_T^{1-\gamma}}{1-\gamma} \,\middle|\, W_t\right]$$

# HJB as a Continuous-Time Version of BOE

- BOE in discrete-time:

$$V^*(s) = \max_{a \in \mathcal{A}_t} \left\{ \mathcal{R}(s,a) + \gamma \cdot \sum_{s' \in \mathcal{N}} \mathcal{P}(s,a,s') \cdot V^*(s') \right\}$$

- BOE in continuous-time:

$$V^*(t,s_t) = \max_{a_t \in \mathcal{A}_t} \left\{ \mathcal{R}(t,s_t,a_t) \cdot dt + \mathbb{E}_{(t,s_t,a_t)} \left[ e^{-\rho \cdot dt} \cdot V^*(t+dt, s_{t+dt}) \right] \right\}$$

- Multiplying throughout by $e^{-\rho t}$ and re-arranging, we have

$$\frac{\mathcal{R}(t,s_t,a_t)}{e^{\rho t}} \cdot dt + \mathbb{E}_{(t,s_t,a_t)} \left[ \frac{V^*(t+dt, s_{t+dt})}{e^{\rho(t+dt)}} - \frac{V^*(t,s_t)}{e^{\rho t}} \right]$$

$$\begin{aligned} \frac{V^*(t+dt, s_{t+dt})}{e^{\rho(t+dt)}} - \frac{V^*(t,s_t)}{e^{\rho t}} &= d\left[ e^{-\rho t} \cdot V^*(t,s_t) \right] \\ &= e^{-\rho t} \cdot \left[ dV^*(t,s_t) - \rho \cdot V^*(t,s_t) \cdot dt \right] \end{aligned}$$

# HJB as a Continuous-Time Version of BOE

- Thus we have

$$\max_{a_t \in \mathcal{A}_t} \left\{ \frac{\mathcal{R}(t, s_t, a_t)}{e^{\rho t}} \cdot dt + \mathbb{E}_{(t, s_t, a_t)} \left[ \frac{dV^*(t, s_t) - \rho \cdot V^*(t, s_t) \cdot dt}{e^{\rho t}} \right] \right\} = 0$$

- Multiplying throughout by $e^{\rho t}$ and re-arranging, we have

$$\rho \cdot V^*(t, s_t) \cdot dt = \max_{a_t \in \mathcal{A}_t} \left\{ \mathbb{E}_{(t, s_t, a_t)} \left[ dV^*(t, s_t) \right] + \mathcal{R}(t, s_t, a_t) \cdot dt \right\}$$

- Assume that the transitions for $\boldsymbol{s}$ are given by an Ito process:

$$d\boldsymbol{s}_t = \boldsymbol{\mu}(t, \boldsymbol{s}_t, a_t) \cdot dt + \boldsymbol{\sigma}(t, \boldsymbol{s}_t, a_t) \cdot d\boldsymbol{z}_t$$

- Apply multivariate Ito's Lemma for $V^*$ as a function of $t$ and $s_t$

$$\begin{aligned} dV^*(t, \boldsymbol{s}_t) = & \left( \frac{\partial V^*}{\partial t} + (\nabla_{\boldsymbol{s}} V^*)^T \cdot \boldsymbol{\mu}_t + \frac{1}{2} \operatorname{Tr} \left[ \boldsymbol{\sigma}_t^T \cdot (\Delta_{\boldsymbol{s}} V^*) \cdot \boldsymbol{\sigma}_t \right] \right) \cdot dt \\ & + (\nabla_{\boldsymbol{s}} V^*)^T \cdot \boldsymbol{\sigma}_t \cdot d\boldsymbol{z}_t \end{aligned}$$

# HJB as a Continuous-Time Version of BOE

- Substituting the expression for $dV^*(t, s_t)$, noting that

$$\mathbb{E}_{(t,\boldsymbol{s}_t,a_t)}\left[(\nabla_{\boldsymbol{s}} V^*)^T \cdot \boldsymbol{\sigma}_t \cdot d\boldsymbol{z}_t\right] = 0$$

- Dividing throughout by $dt$, we finally get to the HJB equation

$$\begin{aligned}&\rho \cdot V^*(t, \boldsymbol{s}_t)\\ &= \max_{a_t \in \mathcal{A}_t}\left\{\frac{\partial V^*}{\partial t} + (\nabla_{\boldsymbol{s}} V^*)^T \cdot \boldsymbol{\mu}_t + \frac{1}{2}\operatorname{Tr}\left[\boldsymbol{\sigma}_t^T \cdot (\Delta_{\boldsymbol{s}} V^*) \cdot \boldsymbol{\sigma}_t\right] + \mathcal{R}(t, \boldsymbol{s}_t, a_t)\right\}\end{aligned}$$

- For a finite-horizon problem terminating at time $T$, the above equation is subject to terminal condition:

$$V^*(T, s_T) = \mathcal{T}(s_T)$$

# Merton's Portfolio Problem in continuous-time

- Back to our asset-allocation and consumption problem, the general HJB equation specializes here to the following:

$$\max_{\pi_t, c_t} \left\{ \mathbb{E}_t \left[ dV^* (t, W_t) + \frac{c_t^{1-\gamma}}{1-\gamma} \cdot dt \right] \right\} = \rho \cdot V^* (t, W_t) \cdot dt$$

- Similarly, use Ito's Lemma on $dV^*$, we have

$$\rho \cdot V^* (t, W_t) = \max_{\pi_t, c_t} \{ Q^* (t, W_t, \pi_t, c_t) \}$$

$$Q^* (t, W_t, \pi_t, c_t) = \frac{\partial V^*}{\partial t} + \frac{\partial V^*}{\partial W_t} \cdot \left[ (\pi_t (\mu - r) + r) W_t - c_t \right] + \frac{\partial^2 V^*}{\partial W_t^2} \cdot \frac{\pi_t^2 \cdot \sigma^2 \cdot W_t^2}{2} + \frac{c_t^{1-\gamma}}{1-\gamma}$$

- This HJB equation is subject to the terminal condition:

$$V^* (T, W_T) = \epsilon^\gamma \cdot \frac{W_T^{1-\gamma}}{1-\gamma}$$

# Overview

- How to solve the Bellman optimality equation?

$$v = f(v) = \max_{\pi} (r_\pi + \gamma P_\pi v)$$

- We need algorithms to learn optimal policies!

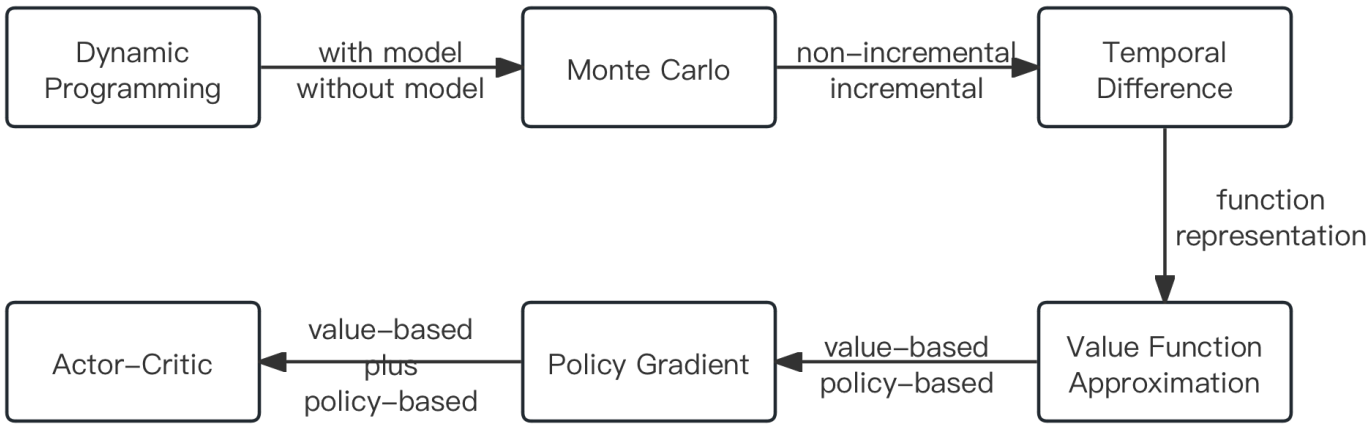


Figure 2: The map of RL algorithms

# Dynamic Programming

- Value Iteration
  1. Policy Update: $\pi_{k+1} = \arg\max_\pi (r_\pi + \gamma P_\pi v_k)$
  2. Value Update: $v_{k+1} = r_{\pi_{k+1}} + \gamma P_{\pi_{k+1}} v_k$
- Policy Iteration
  1. Policy Evaluation: $v_{\pi_k} = r_{\pi_k} + \gamma P_{\pi_k} v_{\pi_k}$
  2. Policy Improvement: $\pi_{k+1} = \arg\max_\pi (r_\pi + \gamma P_\pi v_{\pi_k})$

$$v_k(s) \to q_k(s,a) \xrightarrow{\text{policy update}} \pi_{k+1}(s) \xrightarrow{\text{value update}} \underbrace{v_{k+1}(s) = \max_a q_k(s,a)}_{\text{new value (not state value)}}$$

$$\text{Policy Iteration:} \quad \pi_0 \xrightarrow{\text{PE}} v_{\pi_0} \xrightarrow{\text{PI}} \pi_1 \xrightarrow{\text{PE}} v_{\pi_1} \xrightarrow{\text{PI}} \pi_2 \xrightarrow{\text{PE}} v_{\pi_2} \xrightarrow{\text{PI}} \cdots$$

$$\text{Value Iteration:} \quad v_0 \xrightarrow{\text{PU}} \pi_1' \xrightarrow{\text{VU}} v_1 \xrightarrow{\text{PU}} \pi_2' \xrightarrow{\text{VU}} v_2 \xrightarrow{\text{PU}} \cdots$$

# Monte Carlo

What is the key for the policy improvement step?

$$\pi_{k+1}(s) = \arg\max_{\pi} \left(r_\pi + \gamma P_\pi v_{\pi_k}\right)$$

$$= \arg\max_{\pi} \sum_a \pi(a \mid s) \left[\sum_r p(r \mid s,a) r + \gamma \sum_{s'} p\left(s' \mid s,a\right) v_{\pi_k}\left(s'\right)\right]$$

$$= \arg\max_{\pi} \sum_a \pi(a \mid s) q_{\pi_k}(s,a), \quad s \in \mathcal{S}$$

$$q_{\pi_k}(s,a) = \sum_r p(r \mid s,a) r + \gamma \sum_{s'} p\left(s' \mid s,a\right) v_{\pi_k}\left(s'\right) \leftarrow \text{model-based}$$

$$\Downarrow$$

$$q_{\pi_k}(s,a) = \mathbb{E}\left[G_t \mid S_t = s, A_t = a\right]$$

$$= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s, A_t = a\right]$$

$$q_{\pi_k}(s,a) = \mathbb{E}\left[G_t \mid S_t = s, A_t = a\right] \approx \frac{1}{n} \sum_{i=1}^{n} g_{\pi_k}^{(i)}(s,a) \leftarrow \text{model-free}$$

# Temporal-Difference: Robbins-Monro Algorithm

- Bellman expectation equation: state value function

$$\begin{aligned} v_\pi(s) &= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\ &= \mathbb{E}\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = s\right] \\ v_\pi(s_t) &= \mathbb{E}\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = s_t\right] \end{aligned}$$

- RM algorithm to solve bellman expectation equation

$$\begin{aligned} g\left(v_\pi\left(s_t\right)\right) \doteq & v_\pi\left(s_t\right) - \mathbb{E}\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = s_t\right] = 0 \\ \tilde{g}\left(v_\pi\left(s_t\right)\right) = & v_\pi\left(s_t\right) - \left[r_{t+1} + \gamma v_\pi\left(s_{t+1}\right)\right] \\ = & \underbrace{\left(v_\pi\left(s_t\right) - \mathbb{E}\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = s_t\right]\right)}_{g(v_\pi(s_t))} \\ & + \underbrace{\left(\mathbb{E}\left[R_{t+1} + \gamma v_\pi\left(S_{t+1}\right) \mid S_t = s_t\right] - \left[r_{t+1} + \gamma v_\pi\left(s_{t+1}\right)\right]\right)}_{\eta} \end{aligned}$$

$$\begin{aligned} v_{t+1}\left(s_t\right) = & v_t\left(s_t\right) - \alpha_t\left(s_t\right) \tilde{g}\left(v_t\left(s_t\right)\right) \\ = & v_t\left(s_t\right) - \alpha_t\left(s_t\right)\left(v_t\left(s_t\right) - \left[r_{t+1} + \gamma v_\pi\left(s_{t+1}\right)\right]\right) \end{aligned}$$

# Temporal-Difference: TD Target and TD Error

$$\underbrace{v_{t+1}(s_t)}_{\text{new estimate}} = \underbrace{v_t(s_t)}_{\text{current estimate}} - \alpha_t(s_t)\left\{\overbrace{v_t(s_t) - \underbrace{[r_{t+1} + \gamma v_t(s_{t+1})]}_{\text{TD target } \bar{v}_t}}^{\text{TD error}\delta_t}\right\}$$

- TD target: $v(s_t) \to \bar{v}_t$

$$\begin{aligned}
& v_{t+1}(s_t) = v_t(s_t) - \alpha_t(s_t)[v_t(s_t) - \bar{v}_t] \\
\Longrightarrow \quad & v_{t+1}(s_t) - \bar{v}_t = v_t(s_t) - \bar{v}_t - \alpha_t(s_t)[v_t(s_t) - \bar{v}_t] \\
\Longrightarrow \quad & v_{t+1}(s_t) - \bar{v}_t = [1 - \alpha_t(s_t)][v_t(s_t) - \bar{v}_t] \\
\Longrightarrow \quad & |v_{t+1}(s_t) - \bar{v}_t| = |1 - \alpha_t(s_t)|\,|v_t(s_t) - \bar{v}_t| \\
\Longrightarrow \quad & |v_{t+1}(s_t) - \bar{v}_t| \leq |v_t(s_t) - \bar{v}_t|
\end{aligned}$$

- TD error: $\delta_t = v_t(s_t) - \underbrace{[r_{t+1} + \gamma v_t(s_{t+1})]}_{\text{from innovation } (s_t, r_{t+1}, s_{t+1})}$

$$\begin{aligned}
\mathbb{E}[\delta_t \mid S_t = s_t] &= \mathbb{E}[v_\pi(S_t) - (R_{t+1} + \gamma v_\pi(S_{t+1})) \mid S_t = s_t] \\
&= v_\pi(s_t) - \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s_t] = 0
\end{aligned}$$

# Temporal-Difference: Sarsa

Sarsa

- Bellman expectation equation for state-action value

$$q_\pi(s,a) = \mathbb{E}\Big[R + \gamma q_\pi\left(S', A'\right) \mid s, a\Big], \quad \text{for all } (s,a)$$

- some experience (online): $\{(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})\}_t$

$$q_{t+1}\left(s_t, a_t\right) = q_t\left(s_t, a_t\right) - \alpha_t\left(s_t, a_t\right)\Big\{q_t\left(s_t, a_t\right) - [r_{t+1} + \gamma q_t\left(s_{t+1}, a_{t+1}\right)]\Big\}$$

n-step Sarsa

$$\{s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}\} \to \{s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, \ldots, r_{t+n}, s_{t+n}, a_{t+n}\}$$

$$\begin{aligned}
\text{Sarsa} \longleftarrow G_t^{(1)} &= R_{t+1} + \gamma q_\pi\left(S_{t+1}, A_{t+1}\right) \\
&\vdots \\
\text{n-step Sarsa} \longleftarrow G_t^{(n)} &= R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n q_\pi\left(S_{t+n}, A_{t+n}\right) \\
&\vdots \\
\text{MC} \longleftarrow G_t^{(\infty)} &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} \cdots
\end{aligned}$$

# Temporal-Difference: Q-learning

- Bellman optimality equation: directly find the maximize state-action value $q(s,a)$

$$q(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_a q\left(S_{t+1}, a\right) \mid S_t = s, A_t = a\right], \quad \forall s, a$$

- Q-learning method

$$q_{t+1}\left(s_t, a_t\right) = q_t\left(s_t, a_t\right) - \alpha_t\left(s_t, a_t\right)\left\{q_t\left(s_t, a_t\right) - \left[r_{t+1} + \gamma \max_{a \in \mathcal{A}(s_{t+1})} q_t\left(s_{t+1}, a\right)\right]\right\}$$

- On-policy and off-policy: behavior policy = target policy ?
  - behavior policy: generate experience samples
  - target policy: constantly updated toward an optimal policy
- What exactly is the TD algorithm? A stochastic approximation algorithm for solving BE or BOE $q(s,a) = \mathbb{E}\left[\bar{q}_t \mid s, a\right]$.

$$q_{t+1}\left(s_t, a_t\right) = q_t\left(s_t, a_t\right) - \alpha_t\left(s_t, a_t\right)\left[q_t\left(s_t, a_t\right) - \underbrace{\bar{q}_t}_{\text{TD target}}\right]$$

# Value Function Approximation

- Our goal is to find the best $w$ that can minimize $J(w)$.

$$\min_w J(w) = \mathbb{E}\left[\left(v_\pi(S) - \hat{v}(S, w)\right)^2\right]$$

- To minimize the objective function $J(w)$, we can use gradient-descent algorithm: $w_{k+1} = w_k - \alpha_k \nabla_w J(w_k)$

$$\begin{aligned}\nabla_w J(w_k) &= \nabla_w \mathbb{E}\left[\left(v_\pi(S) - \hat{v}(S, w_k)\right)^2\right] = \mathbb{E}\left[\nabla_w \left(v_\pi(S) - \hat{v}(S, w_k)\right)^2\right] \\ &= 2\mathbb{E}\left[\left(v_\pi(S) - \hat{v}(S, w_k)\right)\left(-\nabla_w \hat{v}(S, w_k)\right)\right] \\ &= -2\mathbb{E}\left[\left(v_\pi(S) - \hat{v}(S, w_k)\right)\nabla_w \hat{v}(S, w_k)\right]\end{aligned}$$

- Use the stochastic gradient to replace the true gradient:

$$\text{SGD: } w_{t+1} = w_t + \alpha_t[v_\pi(s_t) - \hat{v}(s_t, w_t)]\nabla_w \hat{v}(s_t, w_t)$$

- By the spirit of TD learning, $r_{t+1} + \gamma \hat{v}(s_{t+1}, w_t)$ can be viewed as an approximation of $v_\pi(s_t)$. Then, the algorithm becomes

$$\text{TD: } w_{t+1} = w_t + \alpha_t \left[r_{t+1} + \gamma \hat{v}(s_{t+1}, w_t) - \hat{v}(s_t, w_t)\right] \nabla_w \hat{v}(s_t, w_t)$$

# Value Function Approximation: Deep Q-learning

- Deep Q-learning aims to minimize the Bellman optimality error

$$\min J = \mathbb{E}\left[\left(R + \gamma \max_{a \in \mathcal{A}(S')} \hat{q}\left(S', a, w\right) - \hat{q}(S, A, w)\right)^2\right]$$

- The parameter $w$ appears in two $\hat{q}\left(S', a, w\right)$.

$$J = \mathbb{E}\left[\left(R + \gamma \max_{a \in \mathcal{A}(S')} \hat{q}\left(S', a, w\right) - \hat{q}(S, A, w)\right)^2\right]$$

- Assume that $w$ in $y(w) = R + \gamma \max_{a \in \mathcal{A}(S')} \hat{q}\left(S', a, w\right)$ is fixed (at least for a while) when we calculate the gradient.

$$\nabla_w J = \mathbb{E}\left[\left(R + \gamma \max_{a \in \mathcal{A}(S')} \underbrace{\hat{q}\left(S', a, w_T\right)}_{\text{target network}} - \underbrace{\hat{q}(S, A, w)}_{\text{main network}}\right) \nabla_w \hat{q}(S, A, w)\right]$$

# Policy Gradient

- Policies can be represented by functions: $\pi(a|s,\theta)$
- Use gradient-based optimization algorithms to search for optimal policies:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta_t)$$

$$J(\theta) = \lim_{n\to\infty} \mathbb{E}\left[\sum_{t=0}^{n} \gamma^t R_{t+1}\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1}\right]$$

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_{s\in\mathcal{S}} \eta(s) \sum_{a\in\mathcal{A}} \nabla_\theta \pi(a \mid s,\theta) q_\pi(s,a) \\ &= \mathbb{E}_{S\sim\eta, A\sim\pi(S,\theta)} \left[\nabla_\theta \ln \pi(A \mid S,\theta) q_\pi(S,A)\right]\end{aligned}$$

# Monte Carlo Policy Gradient (REINFORCE)

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta_t)$$
$$= \theta_t + \alpha \mathbb{E}\left[\nabla_\theta \ln \pi(A \mid S, \theta_t)\, q_\pi(S, A)\right]$$
$$\Downarrow \text{ replace the true gradient with a stochastic gradient}$$
$$\theta_{t+1} = \theta_t + \alpha \underbrace{\nabla_\theta \ln \pi(a_t \mid s_t, \theta_t)}_{\frac{\nabla_\theta \pi(a_t|s_t,\theta_t)}{\pi(a_t|s_t,\theta_t)}} \underbrace{q_t(s_t, a_t)}_{\text{MC estimation}}$$
$$\Downarrow$$
$$\theta_{t+1} = \theta_t + \alpha \underbrace{\left(\frac{q_t(s_t, a_t)}{\pi(a_t \mid s_t, \theta_t)}\right)}_{\beta_t} \nabla_\theta \pi(a_t \mid s_t, \theta_t)$$
$$\Downarrow$$
$$\theta_{t+1} = \theta_t + \alpha \beta_t \nabla_\theta \pi(a_t \mid s_t, \theta_t)$$

# Actor-Critic

- Actor: Policy Update Algorithm

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta \ln \pi \left( a_t \mid s_t, \theta_t \right) q_t \left( s_t, a_t \right)$$

- Critic: Sarsa + Value Function Approximation

$$\begin{aligned} w_{t+1} = & w_t \\ & + \alpha_w \left[ r_{t+1} + \gamma q \left( s_{t+1}, a_{t+1}, w_t \right) - q \left( s_t, a_t, w_t \right) \right] \nabla_w q \left( s_t, a_t, w_t \right) \end{aligned}$$

- A framework which can be extended to many other algorithms
  - Deterministic AC (DPG)
  - DDPG
  - A2C, A3C
  - SAC
  - PPO
  - TD3

# *Thanks!*