《强化学习与控制》

--

Direct RL with Policy Gradient

Shengbo Eben Li
(李升波)

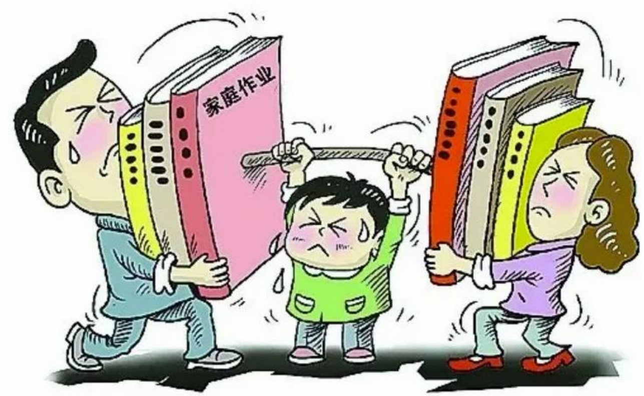
Intelligent Driving Laboratory (iDLab)

Tsinghua University

Never lose a holy curiosity

You cannot teach a man anything;
you can only help him discover it in himself.

-- Galileo Galilei (1564 - 1642)



Outline

1. Indirect RL vs Direct RL
2. Likelihood Ratio Gradient
3. AC from Direct RL
4. Optimization Viewpoint

General Optimal Control Problem

Basis of RL problems

- To find an optimal policy to maximize / minimize a weighted sum of expected return
- Subject to (1) data samples from environment interaction (i.e., model-free) or (2) analytical environment model (i.e., model-based)

$$\max_{\pi}/\min_{\pi} \ \mathbb{E}_{s\sim d(s)}\{v^{\pi}(s)\}$$

Subj. to

$$p(s'|s,a)=\mathcal{P}_{ss'}^{a},$$

or

$$\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \cdots\}$$

?

$\pi^*(a|s)$

Indirect RL

VS

Direct RL

Indirect RL vs Direct RL

(1) Indirect RL

- Sufficient & necessary condition of optimality
  - Hamilton-Jacobi-Bellman equation (continuous-time)
  - Bellman equation (discrete-time)

$$\boldsymbol{\pi}^*(\boldsymbol{a}|\boldsymbol{s}) = \text{Solution of HJB/Bellman equation}$$

- Convergence: Bellman operator is γ-contractive

(2) Direct RL

- Search for a parameterized policy that maximizes the overall objective function

$$\theta^* = \arg\max_{\theta} J(\pi(a|s;\theta))$$

- Search θ^* by using numerical optimization technique
- Convergence: Same as optimization algorithms

Classification of Direct RL

Mainstream direct RL methods

Zero-order optimization

- Evolutionary algorithm (e.g., finite difference)
- Bayesian optimization

First-order optimization

- Likelihood ratio gradient
- Natural policy gradient
- Deterministic policy gradient

Second-order optimization

- Newton method
- Quasi-Newton method

General Optimal Control Problem

Overall RL objective function

$$\max_{\theta} J(\theta) = \mathbb{E}_{s_t \sim d(s_t)}\{v^{\pi_\theta}(s_t)\}$$

$$= \int d(s_t) v^{\pi_\theta}(s_t) \mathrm{d}s_t$$

$$= \mathbb{E}_{s_t, a_t, s_{t+1}, \dots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau \right\}$$

ρ_{π_θ} is joint probability of states and actions in the trajectory

- Revisit value function in terms of trajectory concept

$$v^\pi(s) = \mathbb{E}_{a_t, s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \dots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau \,|\, s_t = s \right\}$$

$$q^\pi(s, a) = \mathbb{E}_{s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \dots \sim \rho_{\pi_\theta}} \left\{ \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau \,|\, s_t = s, a_t = a \right\}$$

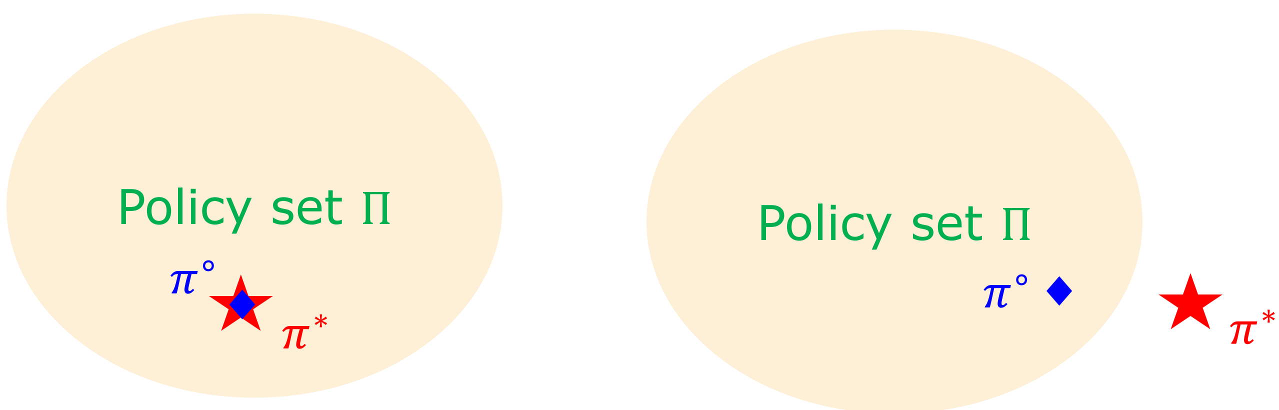
Influence of Initial State Distribution

Define two kinds of "optimal" policies

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^{a} \left(r + \gamma v^*(s')\right), \forall s \in \mathcal{S}$$

$$\pi^{\circ} = \arg\max_{\pi \in \Pi} J\left(\pi(s)\right)$$

- Π is the allowable policy set from designers
- π^* is optimal policy coming from each state element
- π° is optimal policy from overall RL criterion maximization



Case (1): $\pi^* \in \Pi$ Case (2): $\pi^* \notin \Pi$

Influence of Initial State Distribution

- **Case (1): $\pi^*(s)$ is inside allowable policy set Π**
  - 1st step

$$J(\pi^*) \le J(\pi^\circ) = \max_\pi J(\pi)$$

  - 2nd step

$$\begin{aligned} J(\pi^\circ) &= \max_\pi \mathbb{E}_{s\sim d(s)}\{v^\pi(s)\} \\ &\le \max_\pi \mathbb{E}_{s\sim d(s)}\left\{\max_\pi v^\pi(s)\right\} \\ &= \mathbb{E}_{s\sim d(s)}\left\{\max_\pi v^\pi(s)\right\} \\ &= \mathbb{E}_{s\sim d(s)}\{v^*(s)\} \\ &= J(\pi^*) \end{aligned}$$

  - Conclusion

$$J(\pi^*) = J(\pi^\circ)$$

$$\mathbb{E}_{s\sim d(s)}\left\{\max_\pi v^\pi(s)\right\} = \max_\pi \mathbb{E}_{s\sim d(s)}\{v^\pi(s)\}$$

    - Initial state distribution does not affect optimal policy

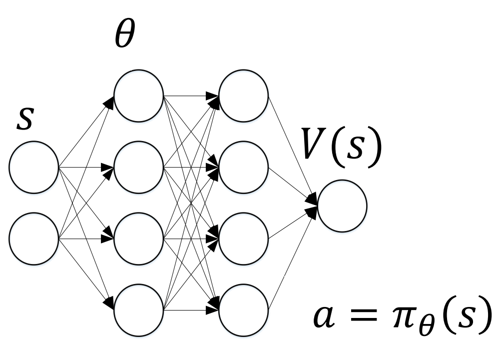
Influence of Initial State Distribution

- **Case (2): $\pi^*(s)$ is NOT inside allowable policy set Π**
  - Only the inequality holds

$$\max_{\pi} \mathbb{E}_{s\sim d(s)}\{v^{\pi}(s)\} \le \mathbb{E}_{s\sim d(s)}\left\{\max_{\pi} v^{\pi}(s)\right\}$$

$$J(\pi^{\circ}) \le J(\pi^*)$$

  - Conclusion
    - Policy $\pi^{\circ}(s) \in \Pi$ gives a less optimal policy than π^*
    - π° becomes dependent of initial state distribution $d(s)$
  - Hint: policy set Π should be as large as possible
    - Neural network is a good choice!



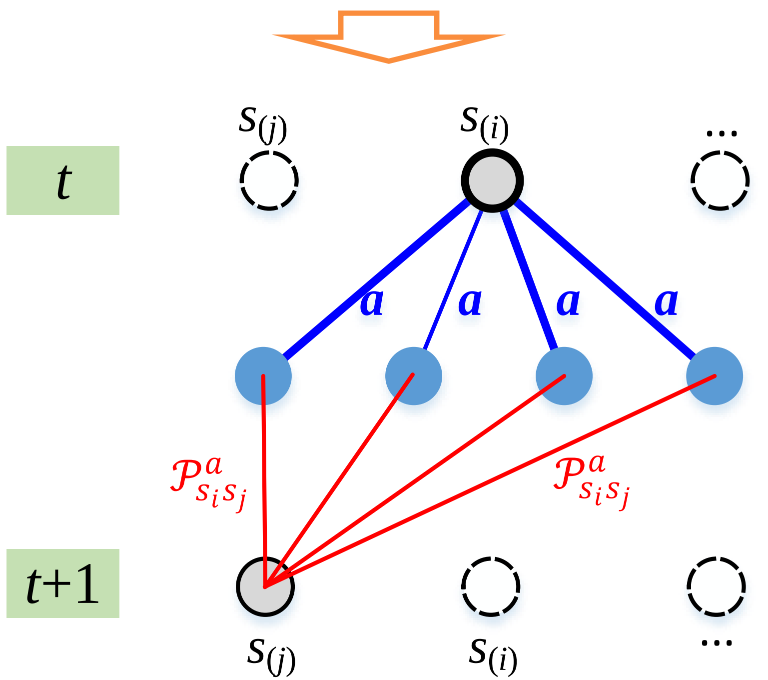
Stationary State Distribution

One-step transition probability

- Given policy $\pi(a|s)$ and environment model $p(s'|s,a)$

$$\mathcal{S} = \{s_{(1)}, s_{(2)}, \cdots, s_{(n)}\}$$

$$\zeta_{i,j} = \sum_{a \in \mathcal{A}} \pi(a|s = s_{(i)}) p(s' = s_{(j)}|s = s_{(i)}, a)$$



Stationary State Distribution

State distribution at time t

- Occurrence frequency of a certain state at time t

$$d_t(s_{(i)}) = \Pr\{s_t = s_{(i)}\}$$

- "Stationary" refers to "stationary in time"

$$\boldsymbol{d}_{t+1} = H_{n\times n}\boldsymbol{d}_t$$

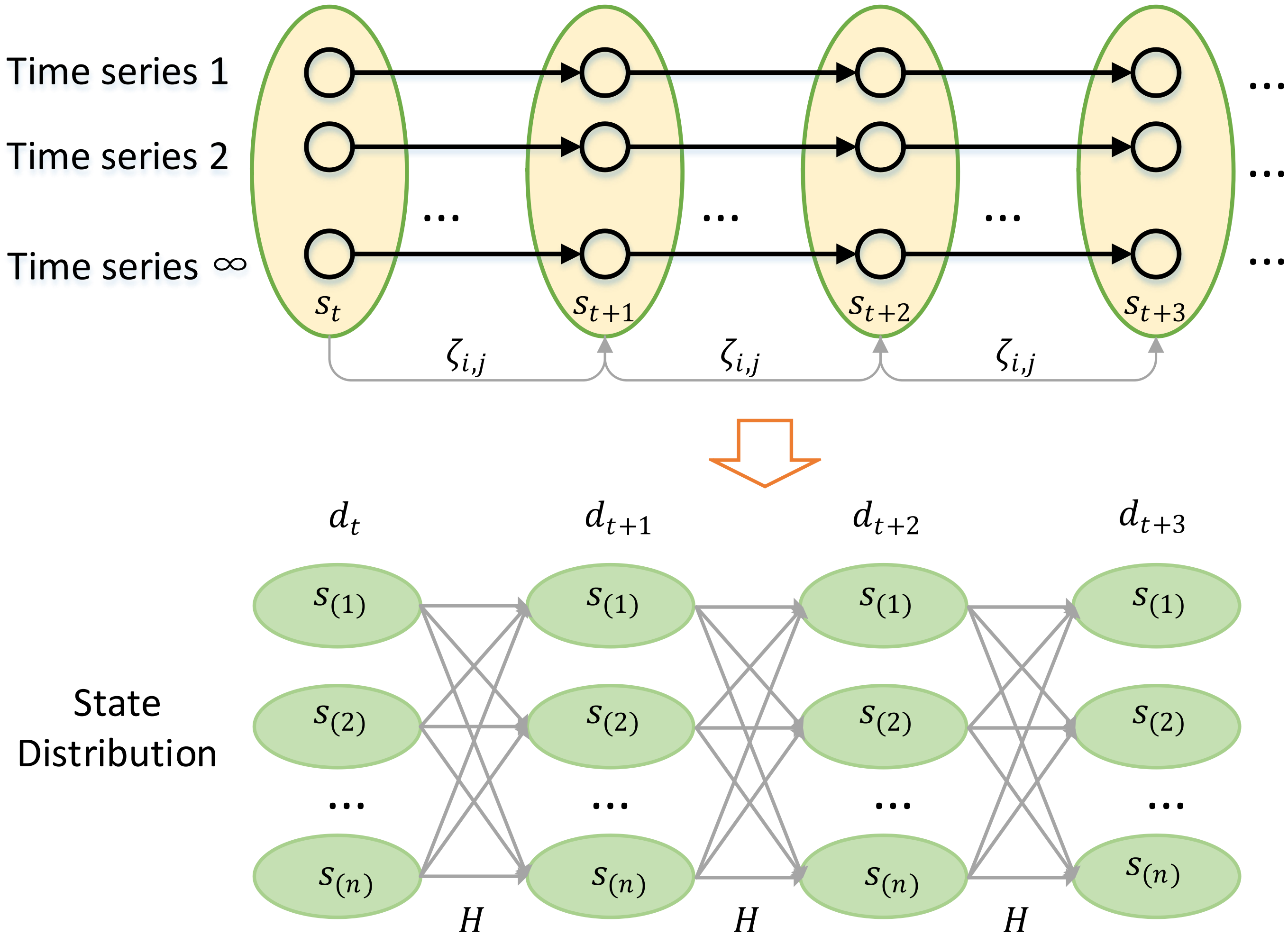
$$H_{n\times n} = \begin{bmatrix} \zeta_{1,1} & \cdots & \zeta_{n,1} \\ \vdots & \ddots & \vdots \\ \zeta_{1,n} & \cdots & \zeta_{n,n} \end{bmatrix} \quad \boldsymbol{d}_t = \begin{bmatrix} d_t(s_{(1)}) & d_t(s_{(2)}) & \cdots & d_t(s_{(n)}) \end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{d}(\boldsymbol{s}) = \mathbf{H}\boldsymbol{d}(\boldsymbol{s})$$

Stationary state distribution (**SSD**)

Stationary State Distribution

Random variable vs State distribution



Stationary State Distribution

Some properties of SSD

- (1) Any finite, irreducible, and ergodic Markov chain has a unique SSD

- (2) For any $i, j \in \mathcal{S}$, the following limit exists, independent of initial state s_0

$$\lim_{t\to\infty} \Pr\{s_t = s_{(j)} | s_0 = s_{(i)}\}$$

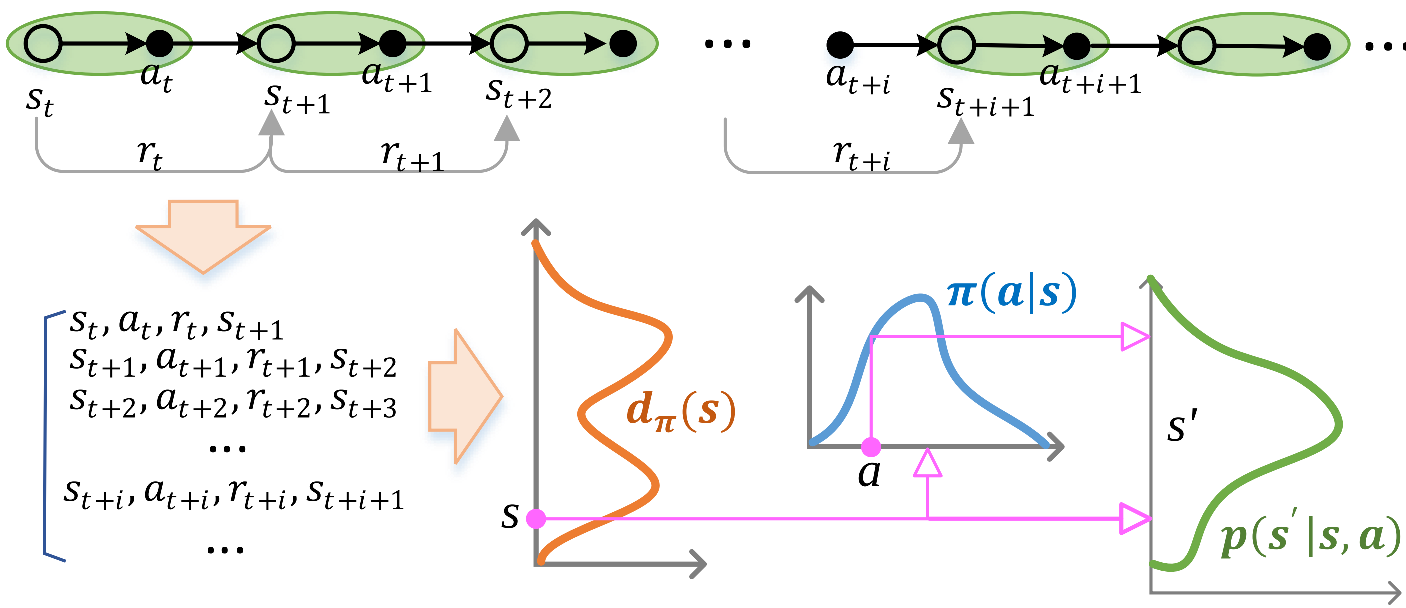
- (3) In an MDP, the SSD under policy π is

$$d_\pi(s_{(j)}) = \lim_{t\to\infty} \Pr\{s_t = s_{(j)} | s_0 = s_{(i)}\}$$

Stationary State Distribution

Graphic understanding

- Limiting distribution that can starts from any initial state distribution
- Temporal order of samples becomes meaningless since each sample could occur randomly with infinite times



Outline

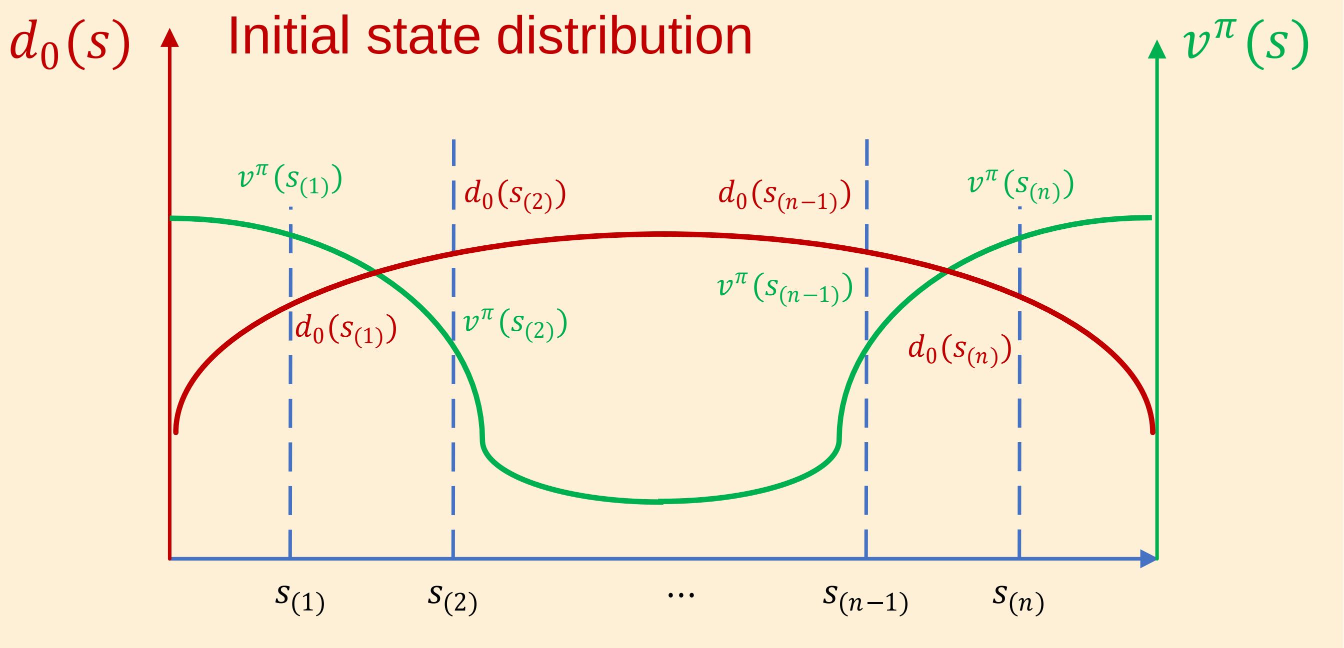
1. Indirect RL vs Direct RL
2. Likelihood Ratio Gradient
3. AC from Direct RL
4. Optimization Viewpoint

Objective function for Direct RL

Overall RL objective function

- Assume that current time $t = 0$
- Finite state space $\mathcal{S} = \{s_{(1)}, s_{(2)}, \cdots, s_{(n)}\}$

$$J(\theta) = \mathbb{E}_{s_0 \sim d_0(s_0)}\{v^\pi(s_0)\} = \sum_{s_0 \in \mathcal{S}} d_0(s_0) v^\pi(s_0)$$



Likelihood Ratio Gradient

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{s_0} d_0(s_0) v^\pi(s_0)$$

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q^\pi(s,a)$$

$$= \nabla_\theta \sum_{s_0} d_0(s_0) \sum_{a_0} \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0)$$

$d_0(s_0)$ is independent of θ

$$= \sum_{s_0} d_0(s_0) \nabla_\theta \sum_{a_0} \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0)$$

Derivation rule

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} [\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) + \pi_\theta(a_0|s_0) \nabla_\theta q^{\pi_\theta}(s_0, a_0)]$$

Relation of q-function and v-function

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} \left[\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) + \pi_\theta(a_0|s_0) \nabla_\theta \left[r_0 + \gamma \sum_{s_1} p(s_1|s_0, a_0) v^{\pi_\theta}(s_1) \right] \right]$$

Likelihood Ratio Gradient

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} \left[\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) \ + \pi_\theta(a_0|s_0) \nabla_\theta \left[r_0 + \gamma \sum_{s_1} p(s_1|s_0, a_0) v^{\pi_\theta}(s_1) \right] \right]$$

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} \left[\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) + \pi_\theta(a_0|s_0) \left[\gamma \sum_{s_1} p(s_1|s_0, a_0) \nabla_\theta v^{\pi_\theta}(s_1) \right] \right]$$

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} \left[\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) + \pi_\theta(a_0|s_0) \left[\gamma \sum_{s_1} p(s_1|s_0, a_0) \nabla_\theta \sum_{a_1} \pi_\theta(a_1|s_1) q^{\pi_\theta}(s_1, a_1) \right] \right]$$

$$= \sum_{s_0} d_0(s_0) \sum_{a_0} \left[\nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0, a_0) \right.$$

Derivation rule

Likelihood Ratio Gradient

Triangular analysis

$$\nabla_\theta J(\theta) = \gamma \sum_{s_0} d_0(s_0) \sum_{a_0} \pi_\theta(a_0|s_0) \sum_{s_1} p(s_1|s_0,a_0) \sum_{a_1} \pi_\theta(a_1|s_1) \nabla_\theta q^{\pi_\theta}(s_1,a_1)$$

$$+\gamma \sum_{s_0} d_0(s_0) \sum_{a_0} \pi_\theta(a_0|s_0) \sum_{s_1} p(s_1|s_0,a_0) \sum_{a_1} \nabla_\theta \pi_\theta(a_1|s_1) q^{\pi_\theta}(s_1,a_1)$$

$$+\sum_{s_0} d_0(s_0) \sum_{a_0} \nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0,a_0)$$

Addition is associative

Likelihood Ratio Gradient

Triangular analysis

$$\nabla_\theta J(\theta) = \gamma \sum_{s_0} d_0(s_0) \sum_{a_0} \pi_\theta(a_0|s_0) \sum_{s_1} p(s_1|s_0,a_0) \sum_{a_1} \pi_\theta(a_1|s_1) \nabla_\theta q^{\pi_\theta}(s_1,a_1)$$

$$+ \sum_{s_1} \gamma \sum_{s_0} d_0(s_0) \sum_{a_0} \underline{\pi_\theta(a_0|s_0) p(s_1|s_0,a_0)} \sum_{a_1} \nabla_\theta \pi_\theta(a_1|s_1) q^{\pi_\theta}(s_1,a_1)$$

$$+ \sum_{s_0} d_0(s_0) \sum_{a_0} \nabla_\theta \pi_\theta(a_0|s_0) q^{\pi_\theta}(s_0,a_0)$$

Transition probability for $s_0 \rightarrow s_1$

Roll forward till infinity

$$\nabla_\theta J(\theta) = \sum_{s} \sum_{t=0}^{\infty} \gamma^t p(s_t = s|\pi_\theta) \sum_{a} \nabla_\theta \pi_\theta(a|s) q^{\pi_\theta}(s,a)$$

Likelihood Ratio Gradient

Vanilla Policy Gradient (Sutton et al., 2000)

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \sum_s d_{\pi_\theta}^{\gamma}(s) \sum_a \nabla_\theta \pi_\theta(a|s) q^{\pi_\theta}(s,a)$$

$$d_{\pi_\theta}^{\gamma}(s) \stackrel{\text{def}}{=} (1-\gamma) \sum_{t=0}^{\infty} \gamma^t p(s_t = s | \pi_\theta)$$

Discounted state distribution

When does it become SSD?

$d_0(s)$ is stationary state distribution

$d_0(s)$ is nonstationary but $\gamma \to 1$

Vanilla Policy Gradient

- **Case (1): If** $d_0(s) = d_{\pi_\theta}(s)$**, then** $s_t \sim d_{\pi_\theta}$ **for all** t

$$d_{\pi_\theta}^{\gamma}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p(s_\tau = s|\pi_\theta) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t d_{\pi_\theta}(s) = d_{\pi_\theta}(s)$$

Independent of time

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\sum_{s} d_{\pi_\theta}^{\gamma}(s)\sum_{a}\nabla_\theta \pi_\theta(a|s) q^{\pi_\theta}(s,a)$$

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}\{\nabla_\theta \log \pi_\theta(a|s)\, q^{\pi_\theta}(s,a)\}$$

True action-value function

Vanilla Policy Gradient

- **Case (2): $d_0(s)$ is NOT stationary, but $\gamma \to 1$**
  - Property of normalization (Independent of γ)

$$(1-\gamma)\sum_{s} d_{\pi_\theta}^{\gamma}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \sum_{s} p(s_t = s|\pi_\theta) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t = 1$$

  - Limit of approximation

$$\lim_{\gamma\to 1} d_{\pi_\theta}^{\gamma}(s) = \lim_{\gamma\to 1}\frac{d_{\pi_\theta}^{\gamma}(s)}{\sum_s d_{\pi_\theta}^{\gamma}(s)}$$

$$= \lim_{\gamma\to 1}\lim_{N\to\infty}\frac{\sum_{t=0}^{N}\gamma^t p(s_t = s|\pi_\theta)}{\sum_s \sum_{t=0}^{N}\gamma^t p(s_t = s|\pi_\theta)}$$

$$= \lim_{N\to\infty}\frac{\sum_{t=0}^{N} p(s_t = s|\pi_\theta)}{\sum_{t=0}^{N}\sum_s p(s_t = s|\pi_\theta)}$$

$$= \lim_{N\to\infty}\frac{\sum_{t=0}^{N} p(s_t = s|\pi_\theta)}{N+1}$$

$$= d_{\pi_\theta}(s)$$

$$\sum_{s} p(s_t = s|\pi_\theta) = 1$$

$$d_{\pi_\theta}(s) = \lim_{N\to\infty}\frac{\sum_{t=0}^{N} p(s_t = s|\pi_\theta)}{N+1}$$

Vanilla Policy Gradient

- **Case (2):** $d_0(s)$ **is NOT stationary, but** $\gamma \rightarrow 1$

Vanilla Policy Gradient (Sutton et al., 2000)

$$\lim_{\gamma \rightarrow 1} d^{\gamma}_{\pi_\theta}(s) = d_{\pi_\theta}(s)$$

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \sum_s d^{\gamma}_{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s) q^{\pi_\theta}(s,a)$$

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}\{\nabla_\theta \log \pi_\theta(a|s)\, q^{\pi_\theta}(s,a)\}$$

True action-value function

Special Case: MC Policy Gradient

Policy Gradient with Monte Carlo Estimation

- Monte Carlo estimation of action-value function

$$q^{\pi}(s,a) \approx \mathrm{Avg}\{G_t | s_t = s, a_t = a\}$$

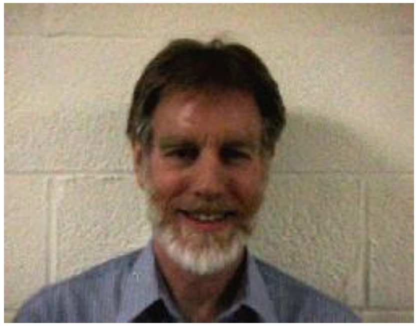
Average returns followed after a particular state-action pair (s, a)

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\{\nabla_\theta \log \pi_\theta(a|s)\, q^{\pi_\theta}(s,a)\}$$

$$q^{\pi_\theta}(s,a) \approx \mathrm{Avg}\{G_t | s_t = s, a_t = a\}$$

$$\theta \leftarrow \theta + \beta \cdot \nabla_\theta \log \pi_\theta(a|s)\, \mathrm{Avg}\{G_t | s_t, a_t\}$$

*REINFORCE (Williams, 1992)



Variance Reduction with Baseline

Baseline technique

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\{q^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|s)\}$$

Baseline

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}\{(q^{\pi_\theta}(s,a) - \zeta(s))\nabla_\theta \log \pi_\theta(a|s)\}$$

- Unbiased estimation only if the baseline is independent of action
  - Proof

$$\begin{aligned}
\mathbb{E}_{\pi_\theta}\{\zeta(s)\nabla_\theta \log \pi_\theta(a|s)\} &= \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \cdot \zeta(s) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \\
&= \sum_s d_{\pi_\theta}(s)\, \zeta(s) \nabla_\theta \sum_a \pi_\theta(a|s) \\
&= \sum_s d_{\pi_\theta}(s)\, \zeta(s) \nabla_\theta 1 \\
&= \sum_s d_{\pi_\theta}(s)\, \zeta(s) \times 0 \\
&= 0
\end{aligned}$$

Variance Reduction with Baseline

What is the optimal baseline?

$$
\begin{aligned}
\Delta\mathbb{D} &= \mathbb{D}\{\nabla_\theta J_{\mathrm{BL}}\} - \mathbb{D}\{\nabla_\theta J\} \\
&= \mathbb{D}_{\pi_\theta}\left\{\left(q^{\pi_\theta}(s,a) - \zeta(s)\right)\nabla_\theta \log \pi_\theta\right\} - \mathbb{D}_{\pi_\theta}\{q^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta\} \\
&= -\mathbb{E}_{\pi_\theta}(\nabla_\theta \log \pi_\theta)^2 \mathbb{E}_{\pi_\theta}\{(2v^{\pi_\theta}(s) - \zeta(s))\zeta(s)\}
\end{aligned}
$$

minimize

$\zeta(s) = v^{\pi_\theta}(s)$

Optimal baseline is state-value function

$$
\Delta\mathbb{D}_{\min} = -\mathbb{E}_{\pi_\theta}(\nabla_\theta \log \pi_\theta)^2 \mathbb{E}_{\pi_\theta}\left\{\left(v^{\pi_\theta}(s)\right)^2\right\} \le 0
$$

Variance Reduction with Baseline

What is the optimal baseline?

- The best choice of baseline is state-value function

$$\zeta(s) = v^{\pi_\theta}(s)$$

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}\left\{\underbrace{\left(q^{\pi_\theta}(s,a) - v^{\pi_\theta}(s)\right)}_{A(s,a)\text{: advantage function}} \nabla_\theta \log \pi_\theta(a|s)\right\}$$

Two viewpoints

Baseline

Bootstrapping

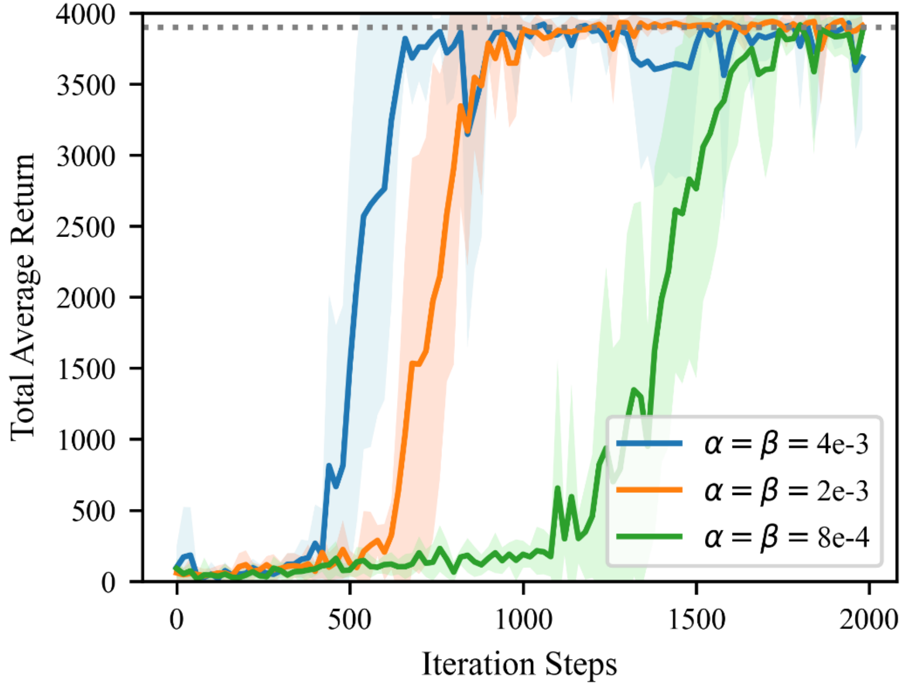
- Replace action-value with state-value

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}\left\{\underbrace{\left(r + \gamma v^{\pi_\theta}(s') - v^{\pi_\theta}(s)\right)}_{\text{One-step TD error}} \nabla \log \pi_\theta(a|s)\right\}$$

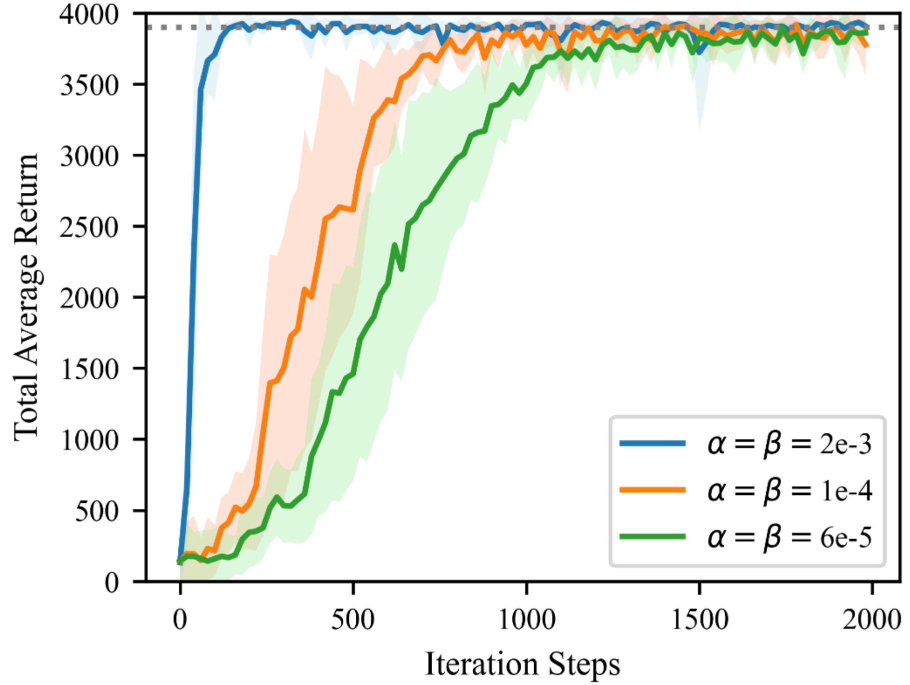
Vanilla policy gradient with state-value function

How Baseline Works?

- **AC (w/o baseline) vs A2C (w/ baseline)**



On-policy AC (w/o baseline)



On-policy A2C (w/ baseline)

On-policy Gradient vs Off-policy Gradient

Off-policy quasi-gradient

- Learn from data generated by old policy and other forms of suboptimal data, including data from expert demonstration

π

$$\max_{\theta} \mathbb{E}_{\mathcal{D}_\pi \sim \pi}\{v^{\pi_\theta}(s)\}$$

Sample-based estimation

$$\nabla J \propto \frac{1}{|\mathcal{D}_\pi|} \sum_{\mathcal{D}_\pi} Q(s,a) \nabla \log \pi_\theta(a|s)$$

On-policy PG

π

On-policy gradient

Off-policy gradient

b

Store IS ratio in buffer

$$\max_{\theta} \mathbb{E}_{\mathcal{D}_b \sim b}\{v^{\pi_\theta}(s)\}$$

Sample-based estimation

$$\nabla J \propto \frac{1}{|\mathcal{D}_b|} \sum_{\mathcal{D}_b} \frac{\pi_\theta(a|s)}{b(a|s)} Q(s,a) \nabla \log \pi_\theta(a|s)$$

Off-policy PG

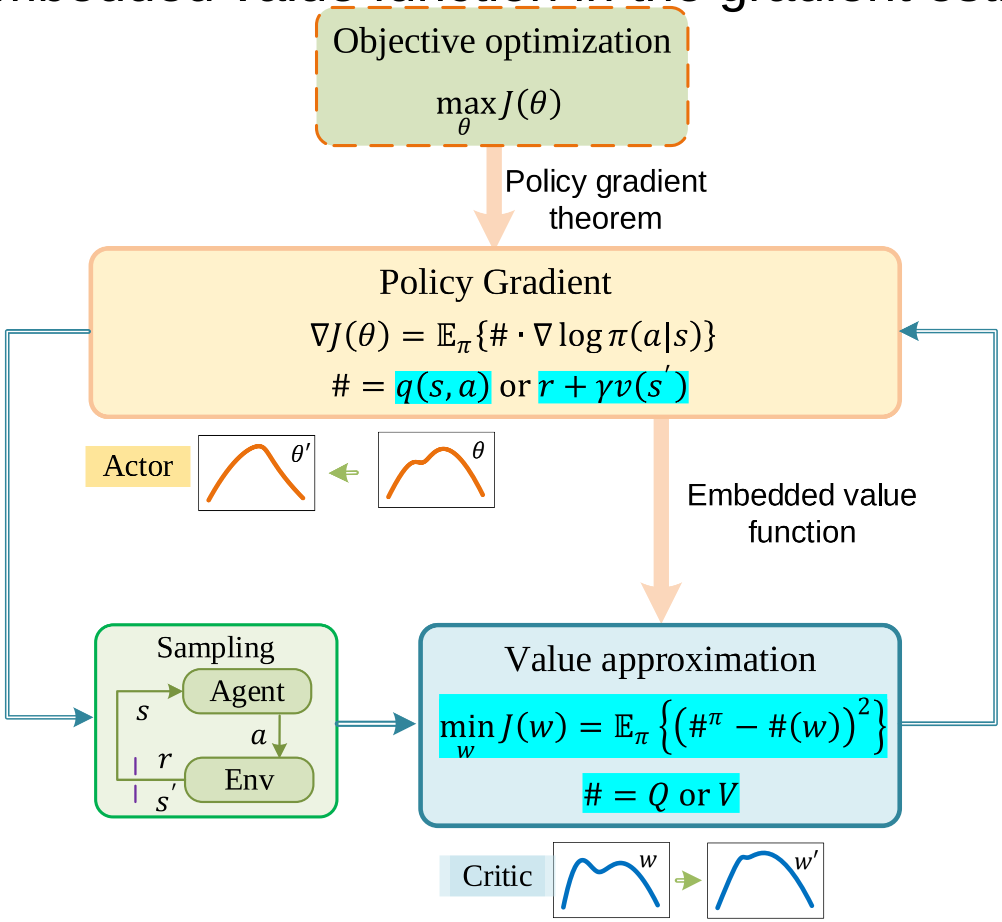
π b

Outline

1. Indirect RL vs Direct RL
2. Likelihood Ratio Gradient
3. AC from Direct RL
4. Optimization Viewpoint

Understand actor-critic with direct RL

- Actor: gradient-based policy updates
- Critic: embedded value function in the gradient estimation



Actor-Critic RL

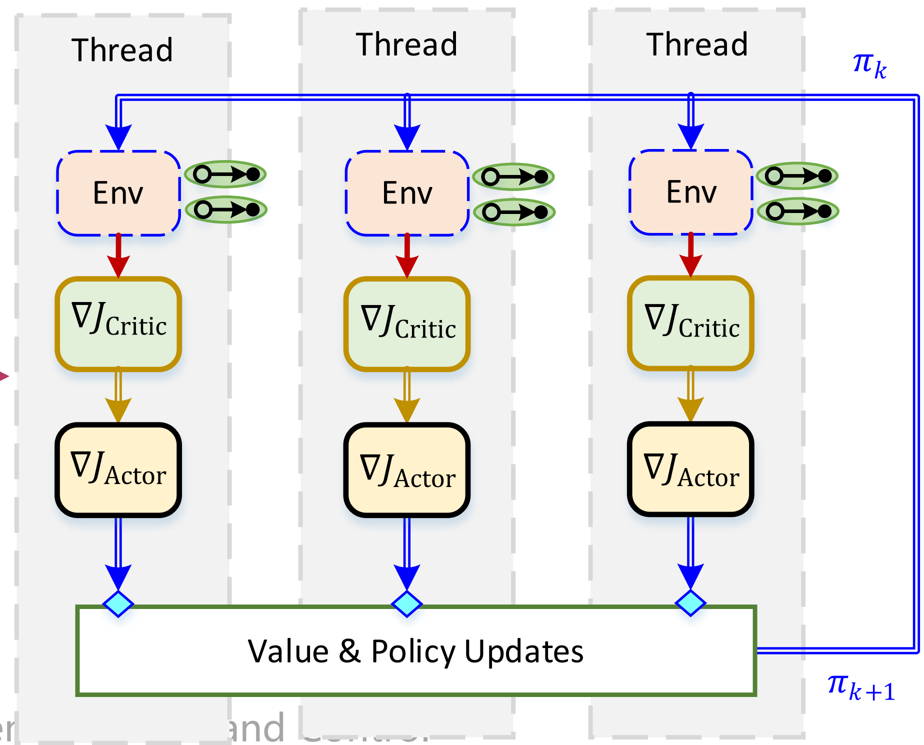
Off-policy AC with Advantage Function (A2C)

$$\nabla_w J_{\text{Critic}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} \rho \cdot \left(r + \gamma V(s';w) - V(s;w)\right)\frac{\partial V(s;w)}{\partial w}$$

$$\nabla_\theta J_{\text{Actor}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} \rho \cdot \nabla_\theta \log \pi(a|s;\theta)\left(r + \gamma V(s';w) - \zeta(s)\right)$$

$\mathcal{B}$: mini-batch

- **A3C** : Asynchronous advantage actor-critic
- **IMPALA**: A3C with importance sampling technique (Google Deepmind)



Actor-Critic RL

Deterministic Policy Gradient (DPG)

$$J_{\text{Actor}}(\theta) = \mathbb{E}_{s\sim d(s)}\{q^{\pi_\theta}(s, \boldsymbol{\pi_\theta(s)})\}$$

DPG

$$\nabla_\theta J_{\text{Actor}}(\theta) \approx \mathbb{E}_{s\sim d_b / s\sim d_\pi}\{\nabla_\theta \pi_\theta(s) \nabla_a q^{\pi_\theta}(s,a)|_{a=\pi_\theta(s)}\}$$

Off-policy Deterministic Actor-Critic

Critic gradient

$$\nabla_w J_{\text{Critic}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} \rho\big(r + \gamma Q(s', a'; w) - Q(s, a; w)\big)\frac{\partial Q(s, a; w)}{\partial w}$$

(1) $\rho = 1$: s' are from behavior policy and $a' \sim \pi(s')$ is from target policy
(2) $\rho = \rho_{t+1}$: s, a, s', a' are from behavior policy

Actor gradient

$$\nabla_\theta J_{\text{Actor}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} \nabla_\theta \pi(s;\theta)\nabla_a Q(s,a;w)$$

$\mathcal{B}$: mini-batch

State-of-the-art of AC Algorithms

Algorithm	Policy	Value	Critic Update	Actor Update	On/Off policy
DDPG	D	Q	TD-based	Vanilla PG	Off
TRPO	S	V	TD-based	Natural PG	On
PPO	S	V	TD-based	Clipped PG	On
TD3	D	Q	Clipped Double Q-learning	Vanilla PG	Off
D4PG	D	Q	Discrete Distributional Q-TD	Vanilla PG	Off
ACKTR	S	V	TD-based	Natural PG	On
A2C/A3C	S	V	TD-based	Vanilla PG	On
Off-PAC	S	V	TD-based	Vanilla PG	Off
ACER	S	Q	TD-based	Vanilla PG	Off
IMPALA	S	V	TD-based	Vanilla PG	Off
Soft Q-learning	S	Q	Soft Q-iteration	Soft PG	Off
SAC	S	Q	Clipped Double-Q	Soft PG	Off
DSAC	S	Q	Continuous Distributional Q-TD	Soft PG	Off

Design using N-step TD

N-step TD error

$$\delta_V^{\mathrm{TD}(n)}(s_t) \stackrel{\text{def}}{=} \underbrace{G_{t:t+n-1} + \gamma^n V^\pi(s_{t+n})}_{n\text{−step TD target}} - V^\pi(s_t)$$

- For off-policy critic update

V	$J_{\mathrm{Critic}} = \mathbb{E}_s\left\{\left(\rho_{t:t+n-1}R^{(n)} - V(s_t;w)\right)^2\right\}, R^{(n)} = G_{t:t+n-1} + \gamma^n V(s_{t+n};w)$
Q	$J_{\mathrm{Critic}} = \mathbb{E}_{s,a}\left\{\left(\rho_{t+1:t+n-1}R^{(n)} - Q(s_t,a_t;w)\right)^2\right\}, R^{(n)} = G_{t:t+n-1} + \gamma^n Q(s_{t+n},a_{t+n};w)$

- For off-policy actor update

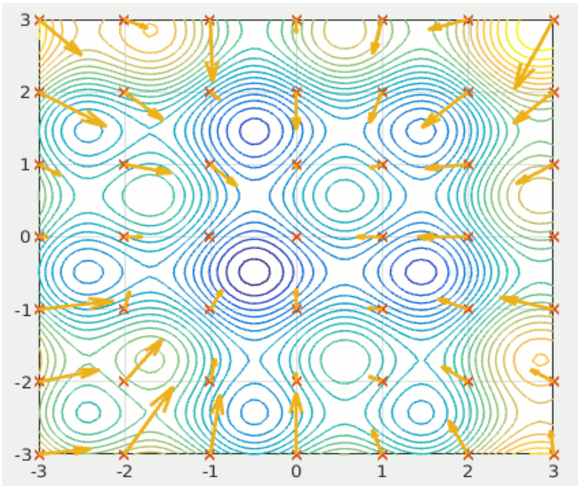
	Stochastic	Deterministic
V	$\nabla_\theta J_{\mathrm{Actor}} = \mathbb{E}_b\left\{\rho_{t:t+n-1}\delta_V^{\mathrm{TD}(n)}\nabla_\theta \log\pi_\theta(a\vert s)\right\}$	
Q	$\nabla_\theta J_{\mathrm{Actor}} = \mathbb{E}_{s\sim d_b,a\sim b}\left\{\frac{\pi_\theta(a\vert s)}{b(a\vert s)}Q(s,a)\nabla_\theta \log\pi_\theta(a\vert s)\right\}$	$\nabla_\theta J_{\mathrm{Actor}} = \mathbb{E}_{s\sim d_b}\{\nabla_\theta\pi_\theta(s)\nabla_a Q(s,a)\}$

Outline

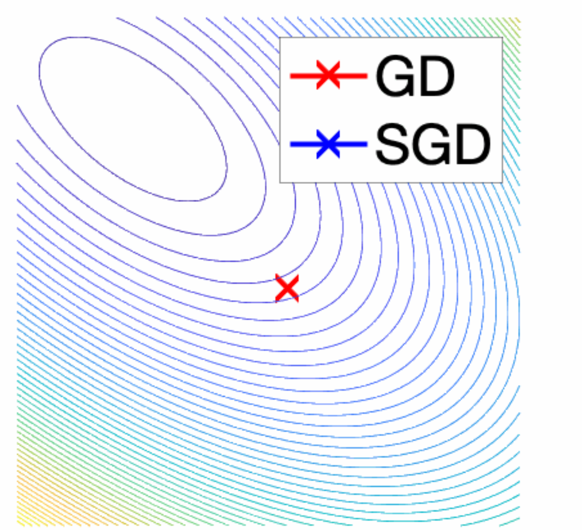
1. Indirect RL vs Direct RL
2. Likelihood Ratio Gradient
3. AC from Direct RL
4. **Optimization Viewpoint**

Types of Stochastic Optimization

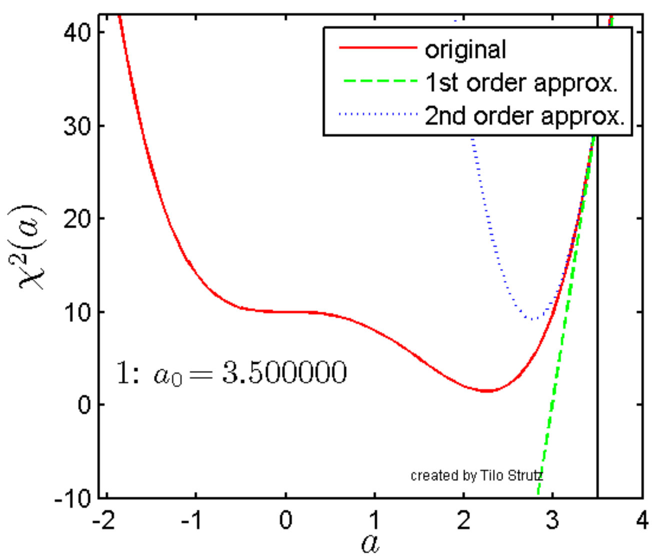
- **Derivative-free optimization**
  - Evolutionary method
  - Bayesian optimization
- **First-order optimization**
  - Stochastic gradient descent
- **Second-order optimization**
  - Newton-Raphson method



Zero-order



First-order



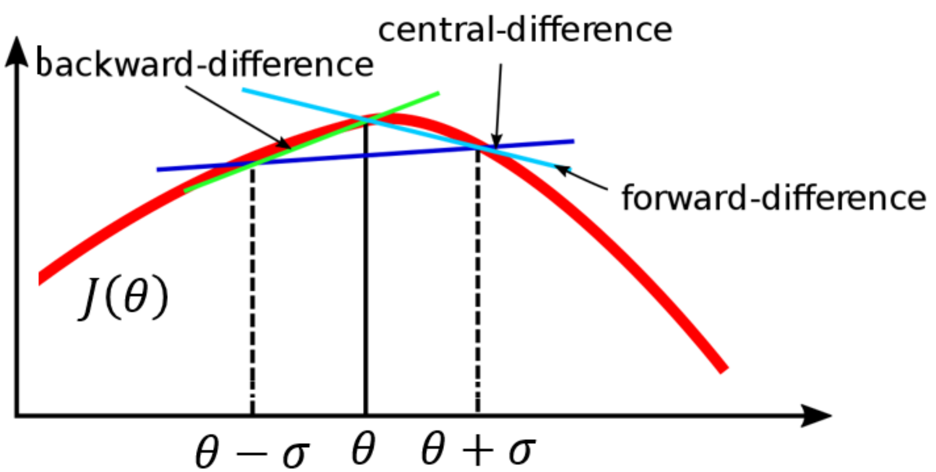
Second-order

Derivative-free Optimization

Derivative-free optimization

- Only zeroth-order information (i.e., function value) is available
- Finite difference method
- Simplest form $\widehat{\nabla J}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{J(\theta + \sigma\epsilon_i) - J(\theta)}{\sigma}\epsilon_i$

Forward difference

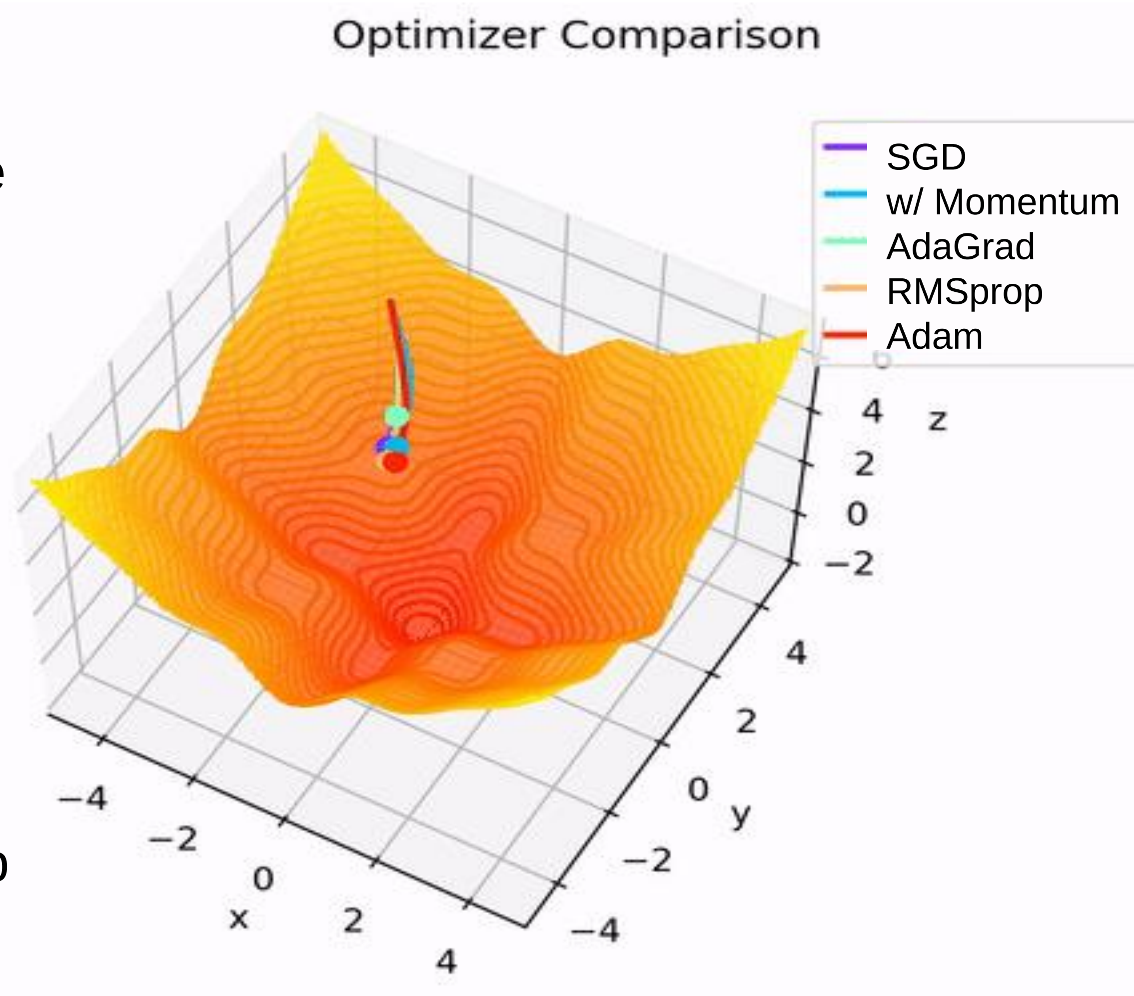


- It is essentially a simple "gradient" estimator
- Scale poorly with the dimension of parameter space

First-order Optimization

Accelerating technique for SGD

- (1) w/ Momentum: accumulate the gradient of past steps to determine the direction to go
- (2) RMSProp: automatically adjust the learning rate and choose a different learning rate for each parameter
- (3) Adam: combination of Momentum and RMSProp



Minorize-maximization optimization

- Primal objective function

$$\max_{\theta} f(\theta)$$

- The lower bound or surrogate function $g(\theta|\theta_k)$ is

$$g(\theta|\theta_k) \le f(\theta), \forall\theta$$
$$g(\theta_k|\theta_k) = f(\theta_k)$$

- Optimize surrogate function $g(\theta|\theta_k)$ at the k-th step

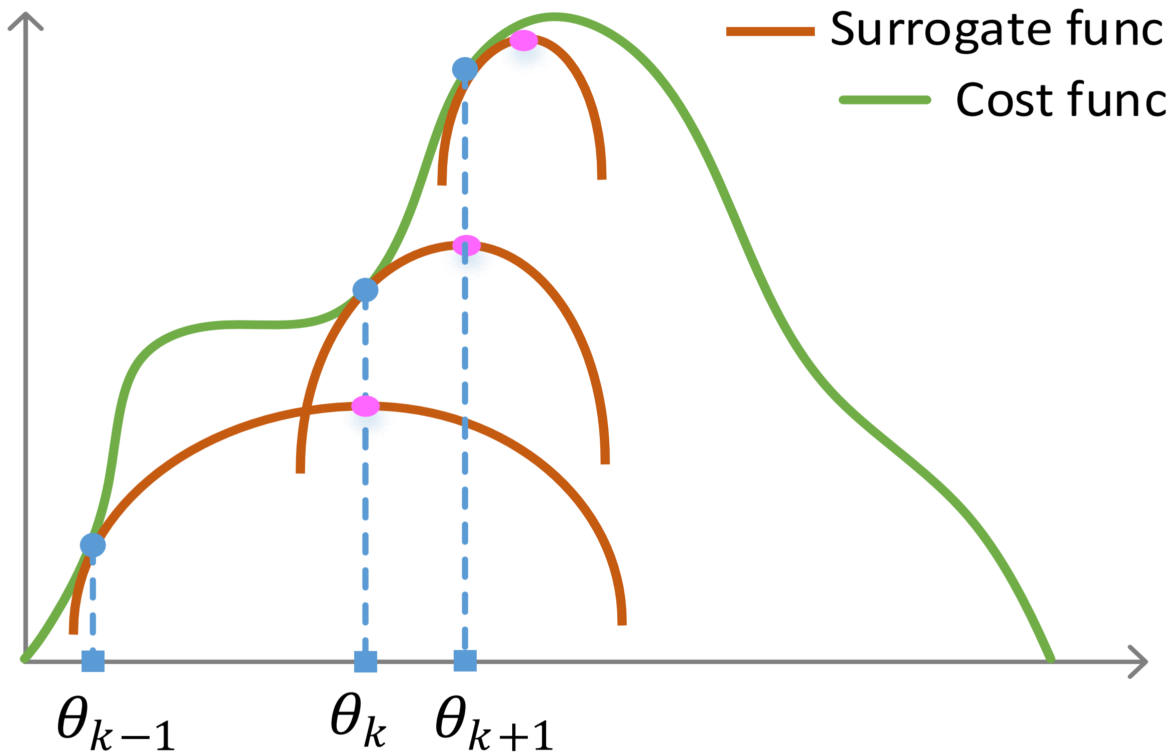
$$\theta_{k+1} = \arg\max_{\theta} g(\theta|\theta_k)$$

This iteration will guarantee convergence to the optimum

$$f(\theta_{k+1}) \ge g(\theta_{k+1}|\theta_k) \ge g(\theta_k|\theta_k) = f(\theta_k)$$

First-order Optimization

Minorize-maximization optimization



Surrogate function = The lower bound of primal objective function

First-order Optimization

Natural Policy Gradient

- Surrogate function for RL objective function

$$J(\pi) \geq L_{\pi_{\text{old}}}(\pi) - C \cdot D_{\text{KL}}^{\max}(\pi_{\text{old}}, \pi)$$

  - C-penalty coefficient, $L_{\pi_{\text{old}}}(\pi)$ - local approximate function

$$L_{\pi_{\text{old}}}(\pi) = J(\pi_{\text{old}}) + \sum_s d_{\pi_{\text{old}}}^{\gamma}(s) \sum_a \pi(a|s) A^{\pi_{\text{old}}}(s, a)$$

MM Optimization

$$\max_{\pi} \left\{ L_{\pi_{\text{old}}}(\pi) - C \cdot D_{\text{KL}}^{\max}(\pi_{\text{old}}, \pi) \right\}$$

First-order Optimization

Natural Policy Gradient

- Consider penalty coefficient C as a Lagrange multiplier

$$\max_{\theta} L_{\pi_{\text{old}}}(\pi_\theta)$$

Subj. to

$$D_{\text{KL}}^{\max}(\pi_{\text{old}}, \pi_\theta) \leq \delta$$

- Replace max operator with average operator

$$\max_{\theta} L_{\pi_{\text{old}}}(\pi_\theta) = \max_{\theta} \mathbb{E}_{\pi_{\text{old}}} \left\{ \frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} A^{\pi_{\text{old}}}(s,a) \right\}$$

$$D_{\text{KL}}^{\max}(\pi_{\text{old}}, \pi_\theta) \approx \bar{D}_{\text{KL}}(\pi_{\text{old}}, \pi_\theta) = \mathbb{E}_{s \sim d_{\pi_{\text{old}}}} \{ D_{\text{KL}}(\pi_{\text{old}}(\cdot\,|s), \pi_\theta(\cdot\,|s)) \}$$

- Trust Region Policy Optimization (TRPO)

$$\max_{\theta} \mathbb{E}_{\pi_{\text{old}}} \left\{ \frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)} A^{\pi_{\text{old}}}(s,a) \right\}$$

Subject to

$$\bar{D}_{\text{KL}}(\pi_{\text{old}}, \pi_\theta) \leq \delta$$

⇨ Natural policy gradient

Second-order Optimization

Newton Method

- Approximating RL objective function by second-order Taylor's expansion

$$\max_{\Delta\theta} g^{\mathrm{T}}\Delta\theta + \frac{1}{2}\Delta\theta^{\mathrm{T}}F\Delta\theta$$

$g = \nabla_\theta J(\theta)$ first-order derivative

$F = \nabla_\theta^2 J(\theta)$ second-order derivative (i.e., Hessian matrix)

- Analytical solution

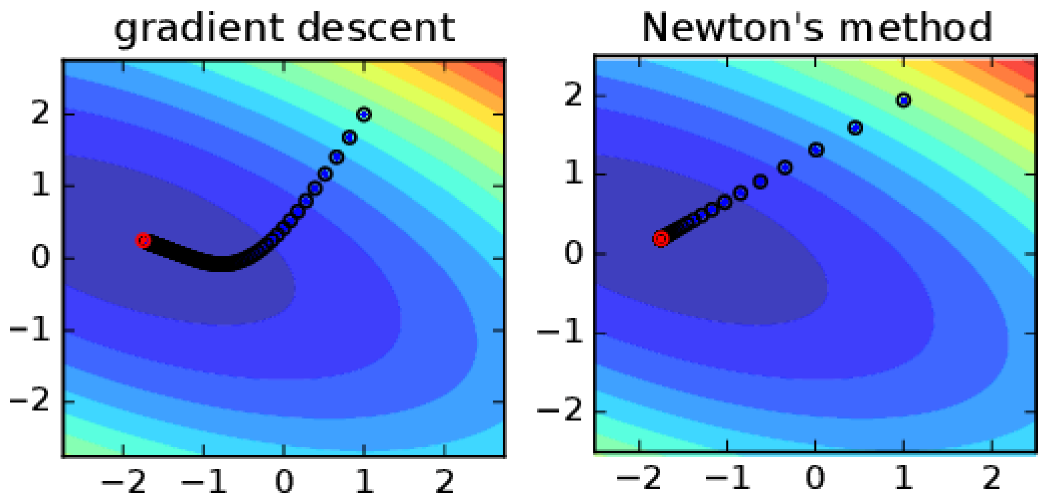
$$\Delta\theta^* = F^{-1}g = \left[\nabla_\theta^2 J(\theta)\right]^{-1}\nabla_\theta J(\theta)$$

- Updating rule $\theta \leftarrow \theta + \Delta\theta^*$

The key is how to
efficiently and accurately compute Hessian and its inverse matrix

Second-order Optimization

Convergence: super-linear rate



Disadvantage

- High cost of computing inverse Hessian matrix
  - Quasi-Newton method: BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm
- Poor performance in non-convex optimization
  - Decreasing step size: $\theta \leftarrow \theta + \alpha_n \Delta\theta^*$

The End!