



Reinforcement Learning

Lecture 20: Policy Gradients (II)

31 Oct, 2025

Instructor: Raunak P. Bhattacharyya



ScAI

YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

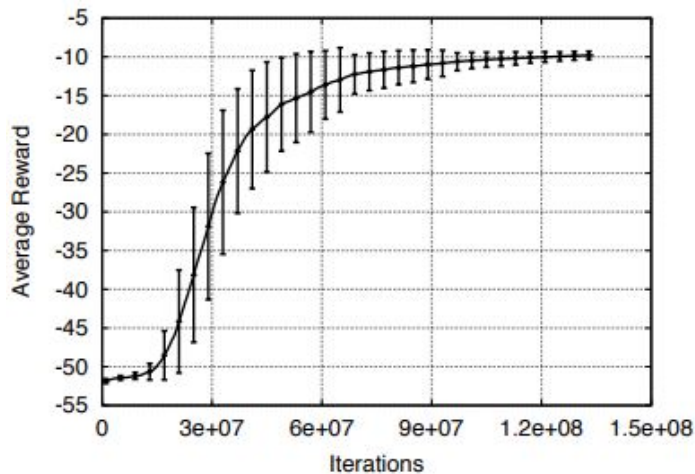
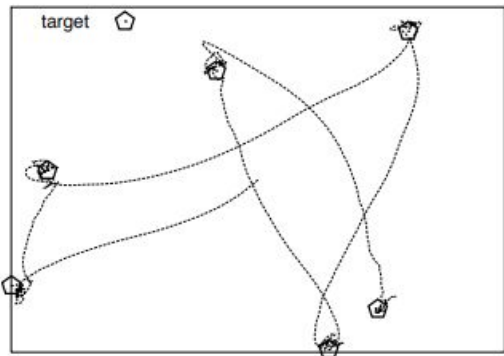
Recap

- Lkhd weighted sampling
- PG for single state single step MDP with Gaussian Policy
- PG intuition for softmax on 2-step 2-state MDP
- PG via autodiff

Plan

- Variance and baselines
- Actor-critic
- Advantage estimation

Example: Puck World



Gradient Estimator

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} \mid s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right)$$

$$\widehat{\nabla_{\theta} J(\theta)} = \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} \mid s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right)$$

A sample mean of the quantity $\nabla_{\theta} \log \pi_{\theta}(\tau_i) r(\tau_i)$

From Sutton and Barto: “...since a Monte-Carlo estimator, will suffer from high variance.”

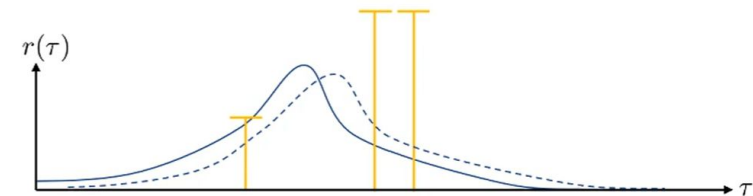
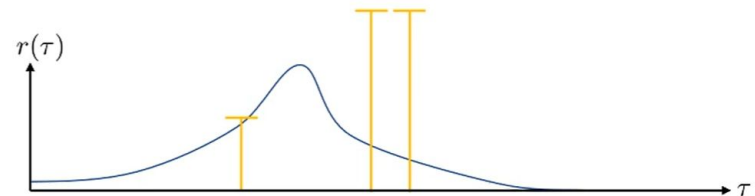
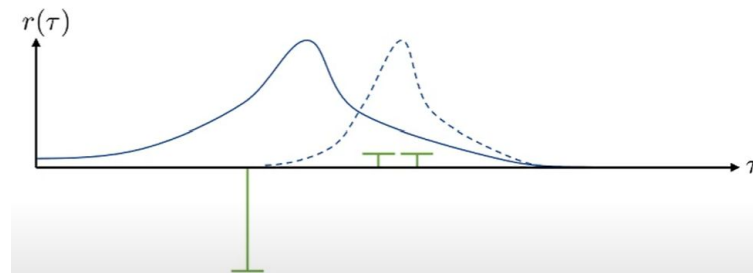
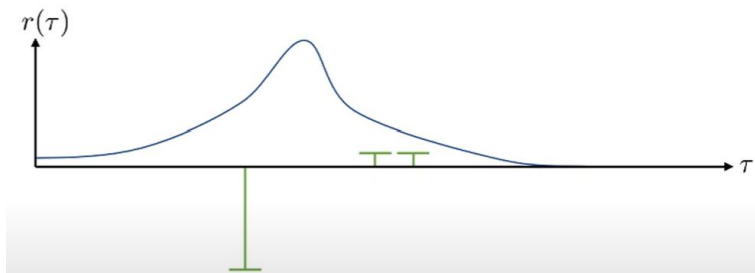
$$\mathbb{E}[f(x)] = \frac{1}{n} \sum_{i=1}^N f(x_i)$$

$$\text{Var}(\mathbb{E}[f(x)]) = \frac{\text{Var}(f(x))}{N}$$

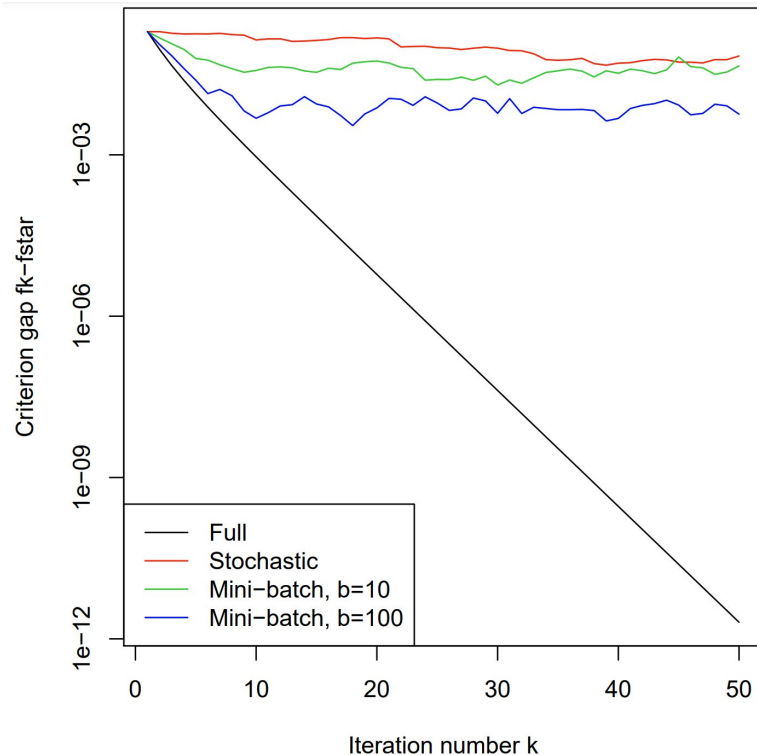
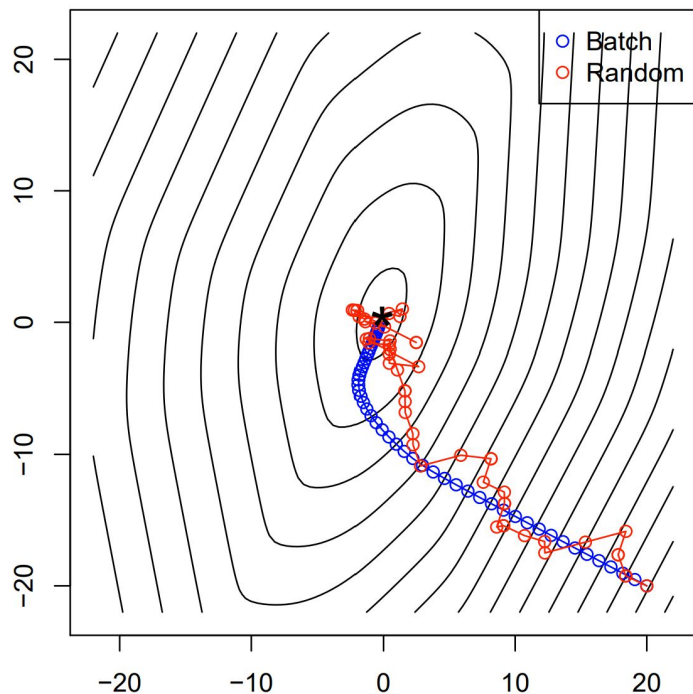
Variance of the estimator is determined by variance of f

Why Does Variance Matter in Policy Gradients

$$\widehat{\nabla_{\theta} J(\theta)} = \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right)$$



Why Does Variance Matter?

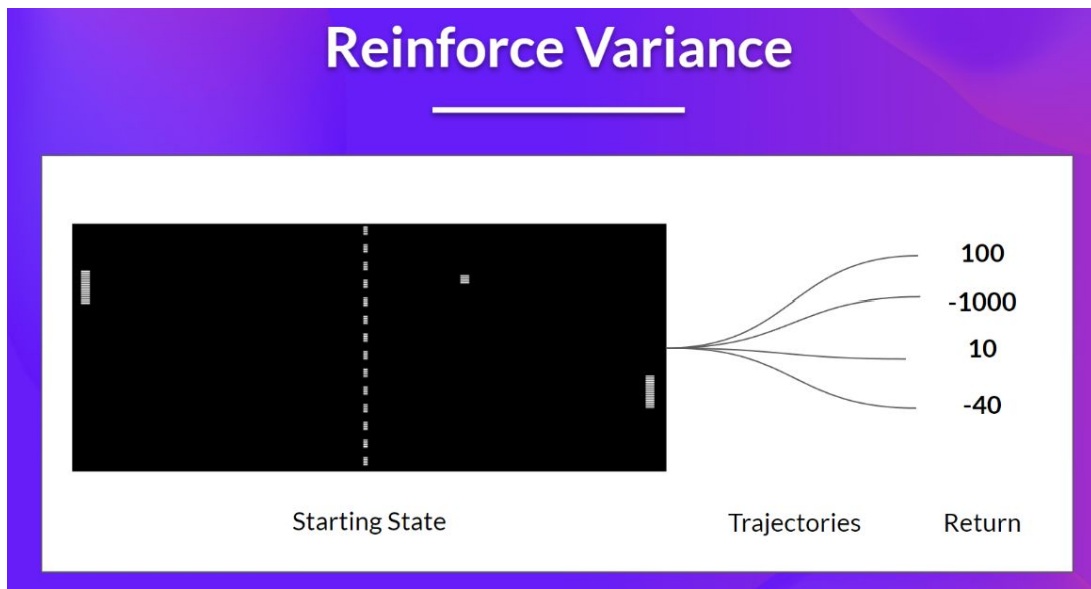


Aside: Control Variates

$$\begin{aligned}\text{Var}(X - Y) &= E[(X - Y)^2] - (E[X - Y])^2 \\&= E[X^2 - 2XY + Y^2] - (E[X] - E[Y])^2 \\&= E[X^2] - 2E[XY] + E[Y^2] - (E[X])^2 - (E[Y])^2 + 2E[X]E[Y] \\&= E[X^2] - (E[X])^2 + E[Y^2] - (E[Y])^2 - (2E[XY] - 2E[X]E[Y]) \\&= \text{Var}(X) + \text{Var}(Y) - 2\text{Cov}(X, Y)\end{aligned}$$

A route to reduce the variance of our gradient estimator

Variance



Let's see a toy example

Baseline

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\tau)} [\nabla_{\theta} \log p_{\theta}(\tau) r(\tau)]$$

$$\text{Let } Y = \nabla_{\theta} \log p_{\theta}(\tau) \cdot (r(\tau) - b)$$

$$\mathbb{E}[Y] = \mathbb{E}[\nabla_{\theta} \log p_{\theta}(\tau) \cdot r(\tau)] - b \cdot \mathbb{E}[\nabla_{\theta} \log p_{\theta}(\tau)]$$

$$E[\nabla_{\theta} \log p_{\theta}(\tau)] = \int p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau) d\tau$$

$$= \int \nabla_{\theta} p_{\theta}(\tau) d\tau$$

Remember this?

$$= \nabla_{\theta} \int p_{\theta}(\tau) d\tau = \nabla_{\theta} 1$$

$$E[Y] = E[\nabla_{\theta} \log p_{\theta}(\tau) \cdot r(\tau)]$$

Our revised estimator Y does not introduce bias

Optimal Baseline

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\tau)} [\nabla_{\theta} \log p_{\theta}(\tau) r(\tau)]$$

$$Y = \nabla_{\theta} \log p_{\theta}(\tau) \cdot \left(r(\tau) - b \right)$$

$$\text{Var}(\mathbb{E}[f(x)]) = \frac{\text{Var}(f(x))}{N}$$

Thus, sufficient to analyse variance of Y. Then divide by N

$$\text{Var}(Y) = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2$$

Can we ignore the second term?

$$E[Y] = E\left[\nabla_{\theta} \log p_{\theta}(\tau) \cdot r(\tau)\right]$$

Independent of b

Therefore, we are interested in $\mathbb{E}[Y^2]$

Optimal Baseline

$$Y = \nabla_{\theta} \log p_{\theta}(\tau) \cdot (r(\tau) - b)$$

Goal: Find b such that the variance is lowest

Therefore, we are interested in $\mathbb{E}[Y^2]$

$$\mathbb{E} \left[g(\tau)^2 (r(\tau) - b)^2 \right]$$

$$\begin{aligned} \frac{d\text{Var}}{db} &= \frac{d}{db} \mathbb{E} [g(\tau)^2 (r(\tau) - b)^2] \\ &= \frac{d}{db} \mathbb{E} [g(\tau)^2 r(\tau)^2] - 2 \mathbb{E} [g(\tau)^2 r(\tau) b] + b^2 \mathbb{E} [g(\tau)^2] \\ &= -2 \mathbb{E} [g(\tau)^2 r(\tau)] + 2b \mathbb{E} [g(\tau)^2] \\ &= 0 \end{aligned}$$

$$b^{opt} = \frac{\mathbb{E} [g(\tau)^2 r(\tau)]}{\mathbb{E} [g(\tau)^2]}$$

In practice, we use the average reward as the baseline

From Sutton and Barto

REINFORCE with Baseline (episodic), for estimating $\pi_{\theta} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$

Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$

Algorithm parameters: step sizes $\alpha^{\theta} > 0$, $\alpha^{\mathbf{w}} > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):

 Generate an episode $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \theta)$

 Loop for each step of the episode $t = 0, 1, \dots, T - 1$:

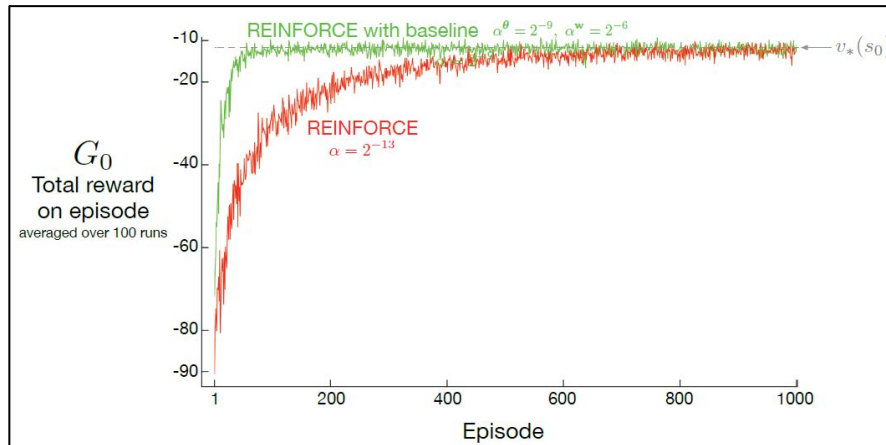
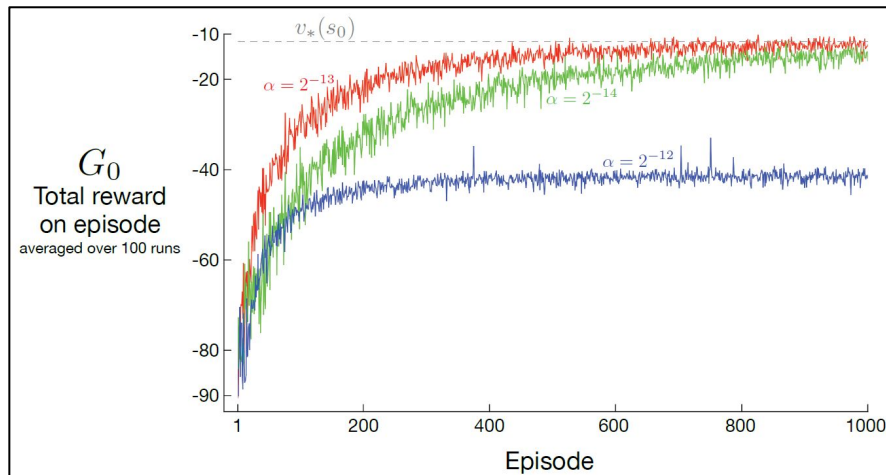
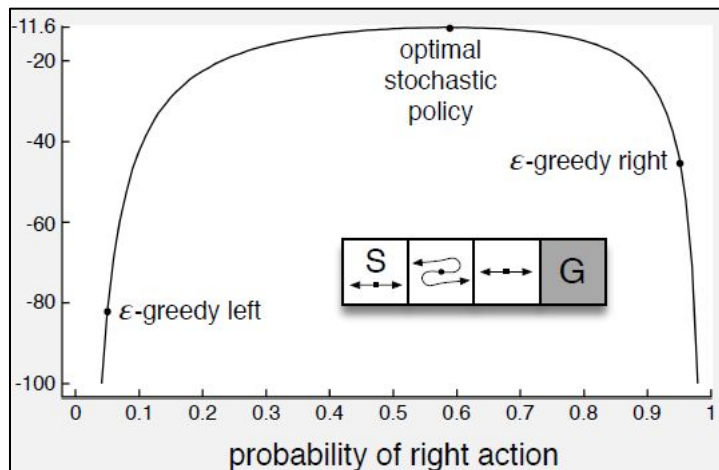
$$G \leftarrow \sum_{k=t+1}^T \gamma^{k-t-1} R_k \tag{G_t}$$

$$\delta \leftarrow G - \hat{v}(S_t, \mathbf{w})$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S_t, \mathbf{w})$$

$$\theta \leftarrow \theta + \alpha^{\theta} \gamma^t \delta \nabla \ln \pi(A_t | S_t, \theta)$$

Example



Variance Reduction: Temporal Structure

$$\begin{aligned}\nabla_{\theta} J(\theta) &\approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=1}^T r(s_{i,t'}, a_{i,t'}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=t}^T r(s_{i,t'}, a_{i,t'}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \hat{Q}_{i,t}\end{aligned}$$

Reward-to-go

Reduced variance since we reduced the sum to be a smaller number

Variance Reduction: Temporal Structure

The policy at time t' cannot affect reward at time t when $t < t'$

$$\begin{aligned}\nabla_{\theta} J(\theta) &\approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=1}^T r(s_{i,t'}, a_{i,t'}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=t}^T r(s_{i,t'}, a_{i,t'}) \right) \\&\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \hat{Q}_{i,t}\end{aligned}$$

Reward-to-go

Reduced variance since we reduced the sum to be a smaller number

Reward-To-Go

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \hat{Q}_{i,t}$$

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\tau)} \left[\left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right) \left(\sum_{t=1}^T r(s_t, a_t) \right) \right]$$

The OG

$$\approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(\sum_{t'=t}^T r(s_{i,t'}, a_{i,t'}) \right)$$

$$\hat{Q}_{i,t} = \sum_{t'=t}^T r(s_{i,t'}, a_{i,t'})$$

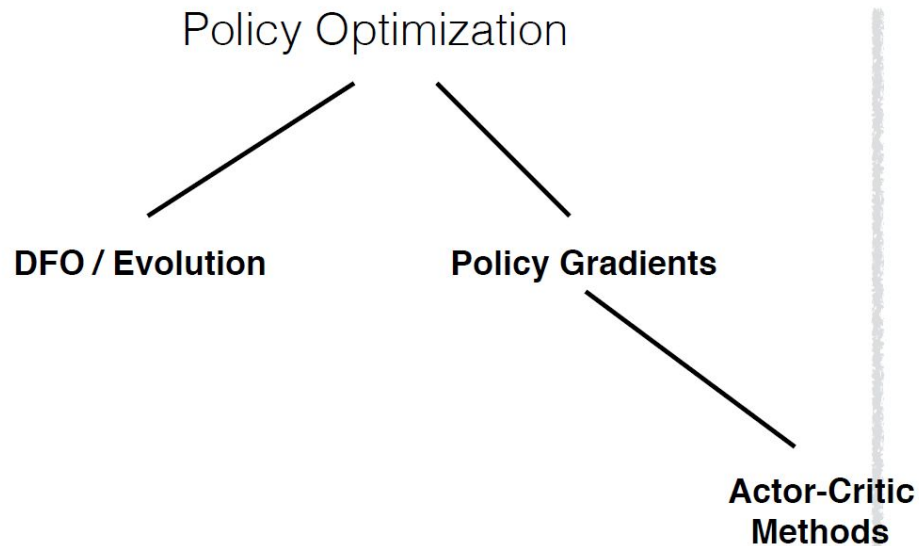
Single sample estimate of
expected reward

If we land at same state and action, we could experience a very different trajectory

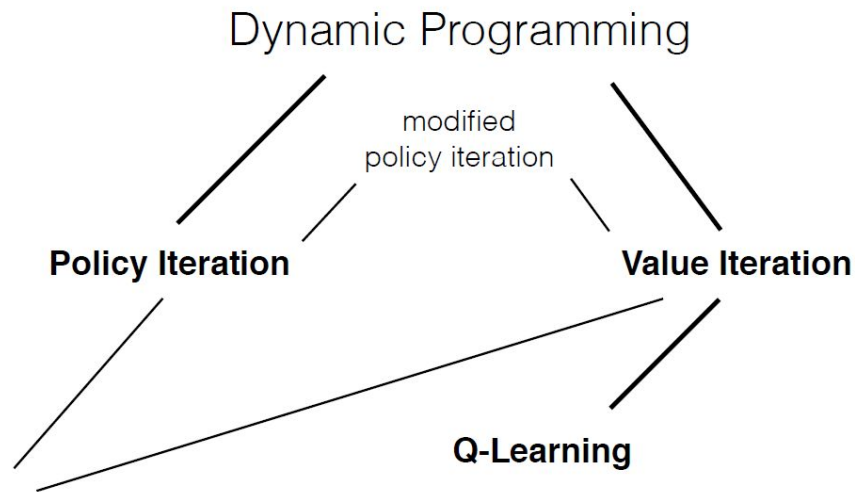
Less samples means higher variance of expected value

Approaches to RL

Direct search in policy space



If we know the policy's corresponding Q-function, we can improve the policy



True Expected Reward-To-Go

$$\hat{Q}_{i,t} = \sum_{t'=t}^T r(s_{i,t'}, a_{i,t'})$$

$$Q(s_t, a_t) = \sum_{t'=t}^T \mathbb{E} \left[r(s_{t'}, a_{t'}) \mid s_t, a_t \right]$$

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) Q(s_{i,t}, a_{i,t})$$

Lower variance gradient estimator

We have already seen that baselines reduce variance. Let's do that here as well

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(Q(s_{i,t}, a_{i,t}) - b \right)$$

Baseline

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \left(Q(s_{i,t}, a_{i,t}) - b \right)$$

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_{i,t} | s_{i,t}) \right) \left(\sum_{t=1}^T r(s_{i,t}, a_{i,t}) - b \right)$$

Single sample

Avg reward

$$b = \frac{1}{N} \sum_{i=1}^N r(\tau_i)$$

Let's work with the true reward-to-go

$$b_t = \frac{1}{n} \sum_{i=1}^N Q(s_{i,t}, a_{i,t})$$

State-Dependent Baseline

$$b_t = \frac{1}{N} \sum_{i=1}^N Q(s_{i,t}, a_{i,t})$$

Avg reward-to-go over all possible trajectories that start at that timestep

But, we want average reward-to-go over all possible trajectories that start at that state.

$$V(s_t) = \mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)} \left[Q(s_t, a_t) \right]$$

How much better is a compared to average action in that state

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t}) \left(Q(s_{i,t}, a_{i,t}) - V(s_{i,t}) \right)$$

Framework for actor-critic algorithm