CS 234 Session 5
Policy Gradients

Contents

- Motivation
- Deterministic vs. Stochastic Policies
- Policy Gradients Objective
- Finite Difference and Vanilla Policy Gradients
- Variance Reduction
- Off Policy Policy Gradients
- Trust Region Policy Optimization

Motivation

Why use Policy Gradients?

PROS

- Better **convergence** properties (recall Q-learning not guaranteed to converge when using function approx.)
- Effective in **high-dimensional** or **continuous action** spaces
  - Why does vanilla DQN not work on continuous action spaces?
- Can learn **stochastic policies** (see next section for why we may want stochastic policies)

CONS

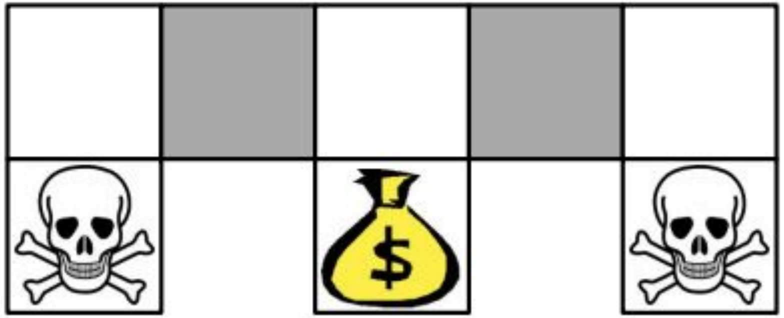
- Typically **data inefficient** and **high variance**

Deterministic vs. Stochastic Policies

Why use stochastic policies?

Deterministic Policy may not be optimal

- What action should we take in the gray state below?



Why use stochastic policies?

Strategic Exploration

- Taking action according to **probability distribution of softmax output** often better exploration strategy than **epsilon-greedy**

Policy Gradients Objective

Episodic Setting / Finite Horizon

- **Probability of a trajectory**

$$\pi_\theta(\tau) = \pi_\theta(s_1, a_1, ..., s_T, a_T) = P(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)P(s_{t+1}|s_t, a_t)$$

- **Objective Function**

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \gamma^t r(s_t, a_t)\right] = \int \pi_\theta(\tau)r(\tau)d\tau \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^t r(s_{i,t}, a_{i,t})$$

- **Optimal Parameters**

$$\theta^* = \arg\max_\theta \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

Exercise 1

- In the lecture notes, for **episodic environments**, the objective function is given below. What **assumption** is made in this objective function?

$$J_1(\theta) = V^{\pi_\theta}(s_1)$$

Answer to Exercise 1

- There is a single start state, s1. In general, there can be a distribution of start states, in which case there should be an **expectation over distribution of start states**, μ.

$$J_1(\theta) = V^{\pi_\theta}(s_1)$$

$$J_1(\theta) = \mathbb{E}_{s\sim\mu}[V^{\pi_\theta}(s)]$$

Continuous Setting / Infinite Horizon

- Define $P_\theta(s,a) = d^{\pi_\theta}(s)\pi_\theta(a|s)$
- Optimal Parameters

$$\begin{aligned}\theta^* &= \arg\max_\theta \sum_{t=1}^{\infty} \mathbb{E}_{(s,a)\sim P_\theta(s,a)}[\gamma^t r(s,a)] \\ &= \arg\max_\theta \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s,a)] \\ &= \arg\max_\theta \mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s,a)]\end{aligned}$$

Exercise 2

- In the lecture notes, for **continuous environments**, two possible objective functions were given. Which of them is the **same** as the **objective** in the previous slide?
- Average Value:

$$J_{avV}(\theta) = \sum_{s} d^{\pi_\theta}(s) V^{\pi_\theta}(s)$$

- Average Reward Per Time Step:

$$J_{avR}(\theta) = \sum_{s} d^{\pi_\theta}(s) \sum_{a} \pi_\theta(s, a) \mathcal{R}_s^a$$

Answer to Exercise 2

- Average Reward per Time Step. In particular, the expectation can be expanded into a summation of states and actions (assuming discrete states and actions; if continuous, use integrals).

$$\mathbb{E}_{(s,a)\sim P_\theta(s,a)}[r(s,a)]$$

$$J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a) \mathcal{R}_s^a$$

Finite Difference and Vanilla Policy Gradients

Finite Difference Methods

- See Lecture Notes for one way to do this
- Another way:
  - Randomly generate K small changes (ΔΘ) to policy and use R rollouts to estimate change in J (ΔJ) for each change in policy parameters. (ΔΘg=ΔJ).

$$\mathbf{g}_{\mathrm{FD}} = \left(\Delta\mathbf{\Theta}^T \Delta\mathbf{\Theta}\right)^{-1} \Delta\mathbf{\Theta}^T \Delta\hat{\mathbf{J}}$$

http://www.scholarpedia.org/article/Policy_gradient_methods

Note on Finite Difference Methods

- Lecture Notes give "**Forward Difference**"

$$\frac{\delta J(\theta)}{\delta \theta_k} \approx \frac{J(\theta + \epsilon u_k - J(\theta)}{\epsilon}$$

- In general, better to use "**Central Difference**"

$$\frac{\delta J(\theta)}{\delta \theta_k} \approx \frac{J(\theta + \epsilon u_k) - J(\theta - \epsilon u_k)}{2\epsilon}$$

https://en.wikipedia.org/wiki/Finite_difference

Exercise 3

- What is the key advantage of using finite difference to estimate policy gradients?

Answer to Exercise 3

- Works for arbitrary policies, even if policy is **not differentiable**

Vanilla Policy Gradients: Log Derivative Trick

- In general, cannot simply move derivative inside expectation. Use **log derivative trick** to do so.

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \int \pi_\theta(\tau) r(\tau) d\tau \\
&= \int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau \\
&= \int \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} r(\tau) d\tau \\
&= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} [\nabla_\theta \log \pi_\theta(\tau) r(\tau)]
\end{aligned}
$$

Exercise 4

- What is the point of the log derivative trick?

Answer to Exercise 4

- By doing so, the gradient estimation will be independent of the dynamics model which, in general, is unknown. See proof on next slide.

Monte-Carlo Estimate of Vanilla Policy Gradients

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) r(\tau)] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta\left[\log P(s_1) + \sum_{t=1}^{T}(\log \pi_\theta(a_t|s_t) + \log P(s_{t+1}|s_t, a_t))\right] r(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta\left[\sum_{t=1}^{T}(\log \pi_\theta(a_t|s_t))\right] r(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=1}^{T}\left(\nabla_\theta\left(\log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=1}^{T}\gamma^t r(s_t, a_t)\right)\right)\right] \\
&\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(\nabla_\theta\left(\log \pi_\theta(a_{i,t}|s_{i,t})\right)\left(\sum_{t=1}^{T}\gamma^t r(s_{i,t}, a_{i,t})\right)\right)
\end{aligned}
$$

Monte-Carlo Vanilla Policy Gradients Algorithm

REINFORCE:

Initialize θ arbitrarily

for each episode $\{s_1, a_1, r_2, \cdots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**

  for $t = 1$ to $T - 1$ **do**

    $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) G_t$

  endfor

endfor

return θ

Variance Reduction

Idea 1: Causality

- **Actions cannot affect past rewards**

$$\nabla_\theta \mathbb{E}[R] = \mathbb{E}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t, s_t, \theta) \sum_{\boxed{t'=t}}^{T-1} r_{t'}\right]$$

- Note: There's something missing in the term above! Can you spot it? (Hint: see the next slide)

Idea 2: Baseline

- Subtract a baseline for every state
- **Baseline compensates for variance introduced by being in different states**

$$\nabla_\theta \mathbb{E}_\tau[R] \approx \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi(a_t|s_t, \theta) \left(\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} \boxed{- b(s_t)} \right) \right]$$

Idea 2: Baseline

- Unbiased if function of state and not action, b(s)

$$
\begin{aligned}
&\mathbb{E}_\tau[\nabla_\theta \log \pi(a_t|s_t,\theta) b(s_t)] \\
&= \mathbb{E}_{s_{0:t},a_{0:(t-1)}}\left[\mathbb{E}_{s_{(t+1):T},a_{t:(T-1)}}[\nabla_\theta \log \pi(a_t|s_t,\theta) b(s_t)]\right] \text{ (break up expectation)} \\
&= \mathbb{E}_{s_{0:t},a_{0:(t-1)}}\left[b(s_t)\mathbb{E}_{s_{(t+1):T},a_{t:(T-1)}}[\nabla_\theta \log \pi(a_t|s_t,\theta)]\right] \text{ (pull baseline term out} \\
&= \mathbb{E}_{s_{0:t},a_{0:(t-1)}}[b(s_t)\mathbb{E}_{a_t}[\nabla_\theta \log \pi(a_t|s_t,\theta)]] \text{ (remove irrelevant variables)} \\
&= \mathbb{E}_{s_{0:t},a_{0:(t-1)}}[b(s_t)\cdot 0]
\end{aligned}
$$

Exercise 5

- How was the last step performed?

$$= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} \left[b(s_t) \mathbb{E}_{a_t} [\nabla_\theta \log \pi(a_t | s_t, \theta)] \right] \text{ (remove irrelevant variables)}$$

$$= \mathbb{E}_{s_{0:t}, a_{0:(t-1)}} [b(s_t) \cdot 0]$$

Common Baseline Used: V(s)

- Becomes **Advantage Estimator** of the form Return - V(s)

$$A^{\pi,\gamma}(s,a) = Q^{\pi,\gamma}(s,a) - V^{\pi,\gamma}(s)$$

Answer to Exercise 5

$$
\begin{aligned}
\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t|s_t)] &= \int_a \pi_\theta(a_t|s_t)\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}da \\
&= \int_a \nabla_\theta \pi_\theta(a_t|s_t)da \\
&= \nabla_\theta \int_a \pi_\theta(a_t|s_t)da \\
&= \nabla_\theta 1 = 0
\end{aligned}
$$

Idea 3: N-step Estimators

- Instead of using Monte-Carlo estimate of returns can use something similar to **TD Target**
- **Trade-off bias and variance**
- Can still subtract baseline (e.g. V(s))

$$\hat{R}_t^{(1)} = r_t + \gamma V(s_{t+1})$$

$$\hat{R}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2})$$

$$\hat{R}_t^{(\text{inf})} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \cdots$$

Exercise 6

- Which of the following has highest variance?

$$\hat{R}_t^{(1)} = r_t + \gamma V(s_{t+1})$$

$$\hat{R}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2})$$

$$\hat{R}_t^{(\mathrm{inf})} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \cdots$$

Answer to Exercise 6

- Which of the following has highest variance?

$$\hat{R}_t^{(1)} = r_t + \gamma V(s_{t+1})$$

$$\hat{R}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2})$$

$$\boxed{\hat{R}_t^{(\mathrm{inf})} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \cdots}$$

Off Policy Policy Gradients

Why Off Policy Policy Gradients?

- REINFORCE is On Policy. Why?
  - Objective takes expectation over trajectories drawn from $\pi_{\Theta}(\tau)$. Once we change our parameters from Θ to Θ', **old trajectories cannot be reused**.
- **Inefficient use of data**.
- Note: When evaluating algorithms, we care about **performance** (average and asymptotic), **computational complexity** and **sample complexity**.

Importance Sampling

$$
\begin{aligned}
\theta^* &= \arg\max_\theta J(\theta) \\
&= \arg\max_\theta \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[r(\tau)] \\
&= \arg\max_\theta \mathbb{E}_{\tau \sim \bar{\pi}_\theta(\tau)}\left[\frac{\pi_\theta(\tau)}{\bar{\pi}_\theta(\tau)} r(\tau)\right] \\
&= \arg\max_\theta \mathbb{E}_{\tau \sim \bar{\pi}_\theta(\tau)}\left[\frac{P(s_1)\prod_{t=1}^{T} \pi_\theta(a_t|s_t)P(s_{t+1}|s_t,a_t)}{P(s_1)\prod_{t=1}^{T} \bar{\pi}_\theta(a_t|s_t)P(s_{t+1}|s_t,a_t)} r(\tau)\right] \\
&= \arg\max_\theta \mathbb{E}_{\tau \sim \bar{\pi}_\theta(\tau)}\left[\frac{\prod_{t=1}^{T} \pi_\theta(a_t|s_t)}{\prod_{t=1}^{T} \bar{\pi}_\theta(a_t|s_t)} r(\tau)\right]
\end{aligned}
$$

Importance Sampling

- Techniques to reduce variance (causality, baseline, N-step estimators) can still be applied here.

$$\begin{aligned}
\nabla_{\theta'} J(\theta') &= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\frac{\nabla_{\theta'} \pi_{\theta'}(\tau)}{\pi_\theta(\tau)} r(\tau) \right] \\
&= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\frac{\pi_{\theta'}(\tau)}{\pi_\theta(\tau)} \nabla_{\theta'} \log \pi_{\theta'}(\tau) r(\tau) \right] \\
&= \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\left(\prod_{t=1}^{T} \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \right) \left(\sum_{t=1}^{T} \nabla_{\theta'} \left(\log \pi_{\theta'}(a_t|s_t) \right) \right) \left(\sum_{t=1}^{T} \gamma^t r(s_t, a_t) \right) \right]
\end{aligned}$$

Trust Region Policy Optimization

Why TRPO?

- Importance of **step size**.
  - Too small -> Updates are too slow
  - Too large -> Policy may suddenly become bad
- What is wrong with a policy becoming bad? We know for SGD, loss fluctuates anyway.
  - For supervised learning, quickly returns back to good
  - For reinforcement learning, data is collected from policy. **Bad Policy** = **Bad Data**. May never recover.

What do we want to guarantee?

- Monotonic Improvement of Policy!

Exercise 7

- Why can't we perform the following optimization?

$$\max_{\pi'} J(\pi') = \max_{\pi'} J(\pi') - J(\pi)$$
$$= \max_{\pi'} \mathbb{E}_{\tau \sim \pi'} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi}(s_t, a_t) \right]$$

Answer to Exercise 7

- We want to find π'. But, to do that, we need to do rollouts using π'. Unable to do so.
- Importance Sampling to the rescue!

Relative Policy Performance Identity

$$
\begin{aligned}
J(\pi') - J(\pi) &= \mathbb{E}_{\tau \sim \pi'}\left[\sum_{t=0}^{\infty} \gamma^t A^\pi(s_t, a_t)\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi'} \\ a\sim\pi'}}\left[A^\pi(s,a)\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim \boxed{d^{\pi'}} \\ a\sim\pi}}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^\pi(s,a)\right] \\
&\boxed{\approx} \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim \boxed{d^{\pi}} \\ a\sim\pi}}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^\pi(s,a)\right] \\
&= \frac{1}{1-\gamma}L_\pi(\pi')
\end{aligned}
$$

Key Idea in TRPO

- When is the approximation true?
  - If $\pi'=\pi$, then holds with equality. But we want to improve the policy.
- Construct **lower bound of J(π')-J(π) that is tight at π**. When optimizing over lower bound, we are guaranteed improvement!
- Intuitively, the lower bound should depend on how different π and π' are.

Skipping the Proof...

$$L_\pi(\pi') = \mathbb{E}_{\substack{s\sim d^\pi \\ a\sim\pi(\cdot|s)}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi(s,a) \right]$$

$$\epsilon = \max_{s,a} |A^\pi(s,a)|$$

$$\alpha = \max_s D_{TV}(\pi \| \pi')$$

- Lower Bound

$$\frac{1}{1-\gamma} L_\pi(\pi') - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2 \leq V^{\pi'} - V^\pi$$

- Optimizing Lower Bound

$$\max_{\pi'} L_\pi(\pi') - \frac{4\epsilon\gamma}{(1-\gamma)}\alpha^2$$

- Problem: Optimizing Lower Bound results in too small a change in policy (**slow convergence**).

See lecture notes or TRPO paper or CPO paper for detailed proof

Convert to Constraint Optimization

- Constraint Optimization with hyperparameter δ

$$\max_{\pi'} L_\pi(\pi') \quad \text{s.t.} \;\; \alpha^2 \leq \delta$$

- However, α requires taking max over all states. Hard to estimate this. As a heuristic, use expectation so can estimate with samples.

$$\max_{\pi'} L_\pi(\pi')$$

$$\text{s.t.} \;\; \bar{D}_{KL}(\pi, \pi') \leq \delta \qquad \text{where } \bar{D}_{KL}(\pi, \pi') = \mathbb{E}_{s \sim d^\pi}[D_{KL}(\pi \| \pi')[s]]$$

How to solve this optimization problem?

- Natural Policy Gradients
- http://rail.eecs.berkeley.edu/deeprlcourse-fa17/f17docs/lecture_13_advanced_pg.pdf