Reinforcement Learning

CS 59300: RL1

September 23, 2025

Joseph Campbell

Department of Computer Science

PURDUE UNIVERSITY

Today's lecture

1. Actor-critic and advantages

2. Natural policy gradient and trust region policy optimization

Some content inspired by Katerina Fragkiadaki's CMU 10-403

Homework 1 due tomorrow at 11:59 PM

Homework 2 will be issued later this week

Will require you to implement:

- DQN
- Double DQN
- Prioritized experience replay

Project proposals

Discuss your project idea with me!

Due this Friday

Send a confirmation email, even if you have already talked with me

Syllabus updates

Week	Tuesday		Thursday	
1	8/26	Introduction to RL	8/28	Deep Learning Basics and Behavior Cloning
2	9/2	Multi-Armed Bandits & Markov Decision Proc.	9/4	Policy/Value Iteration
3	9/9	Temporal Difference Learning	9/11	Deep Q Learning
4	9/16	Deep Q Learning	9/18	Policy Gradient
5	9/23	Actor-Critic Methods	9/25	Actor-Critic / Model-Based RL
6	9/30	Model-Based RL	10/2	Multi-Agent RL
7	10/7	RL Framework Tutorial	10/9	Monte Carlo Tree Search (Recording)
8	10/14	No Class Fall Break	10/16	Multi-Agent RL
9	10/21	Guest Talk Woojun Kim (CMU)	10/23	Inverse Reinforcement Learning
10	10/28	Offline Reinforcement Learning	10/30	Diffusion Policies and Multimodality
11	11/4	Final Exam	11/6	Intelligent Exploration and Curiosity
12	11/11	Transfer Learning and Lifelong Learning	11/13	Large Language Models
13	11/18	Vision Language Models	11/20	Robotics & Sim-to-Real
14	11/25	Challenges & Open Problems	11/27	No Class Thanksgiving Break
15	12/2	No Class Conference	12/4	No Class Conference
16	12/9	Poster Presentations	12/11	Poster Presentations

Recap: Policy-based reinforcement learning

In previous lectures, we primarily focused on modeling and approximating value functions

- So far, the policy can be obtained from the value function
  - Pure-greedy, ϵ-greedy, …

(Usually) deterministic greedy policies

What if instead we model the policy directly?

Recap: Policy objectives

Policy gradient = gradient of objective with respect to parameters

- So what is a good choice of policy objective?

$$\max_{\theta} J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T} \gamma^t r_{t+1}] = \mathbb{E}_{\pi_\theta}[G_t]$$

In other words, find policy parameters that max the expected return

- Easiest way to do this? Monte Carlo estimation of returns!

Recap: REINFORCE

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Recap: The problem with policy-based RL

Monte Carlo value estimates have a problem: **what is it?**

- Value estimates have a high variance!

In Temporal Difference learning, we got around this by bootstrapping off our own value estimate

- This was used in value-based RL such as Q-learning

Can we combine value-based and policy-based approaches?

Recap: Actor-Critic methods

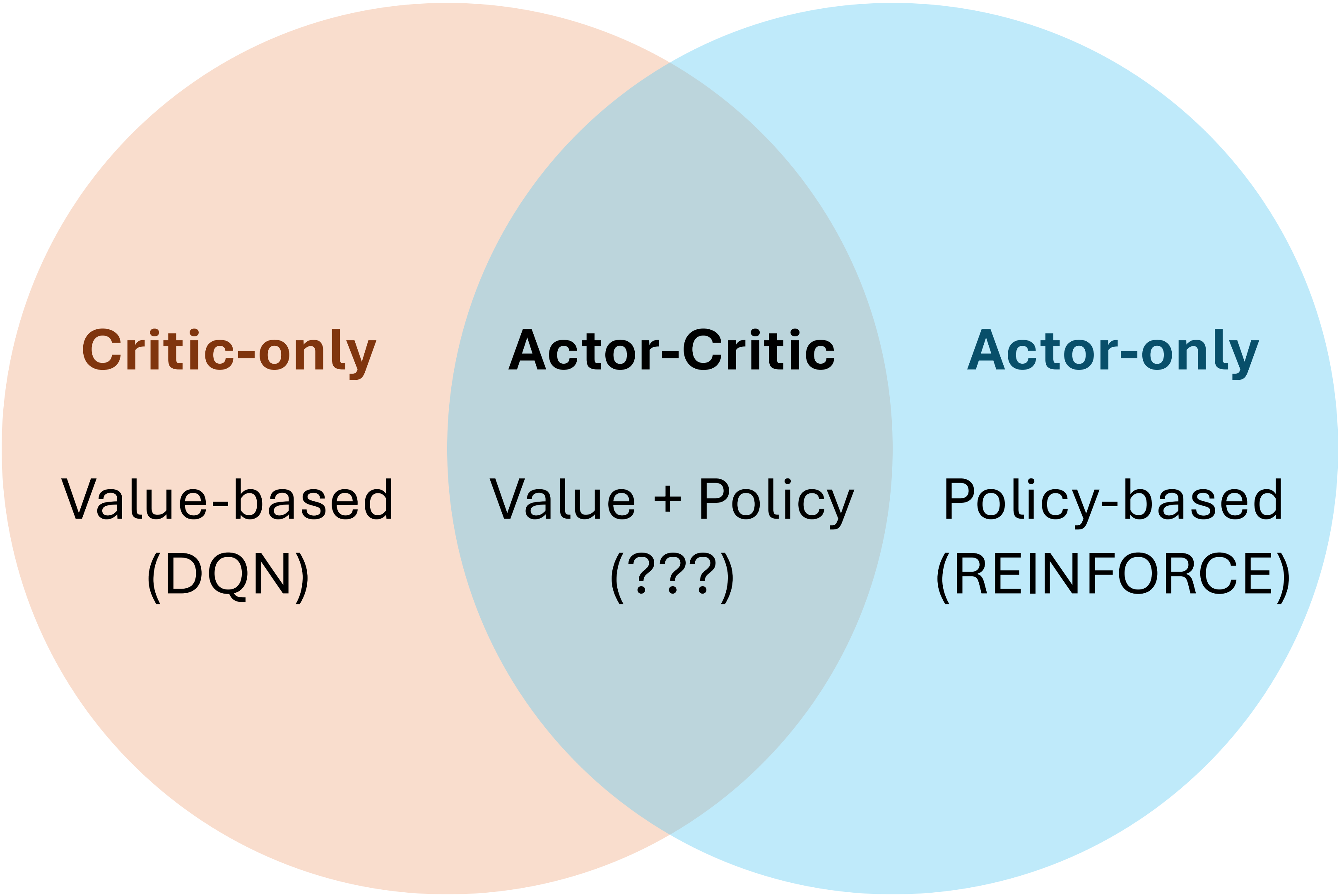
The solution is an actor-critic!

- We approximate both the policy and the value function

We maintain two sets of parameters!

Critic = approximated value function $Q_\phi(s, a) \approx Q_{\pi_\theta}(s, a)$

Actor = approximated policy function $\pi_\theta(a|s)$


Critic-only
Value-based
(DQN)
Actor-Critic
Value + Policy
(???)
Actor-only
Policy-based
(REINFORCE)

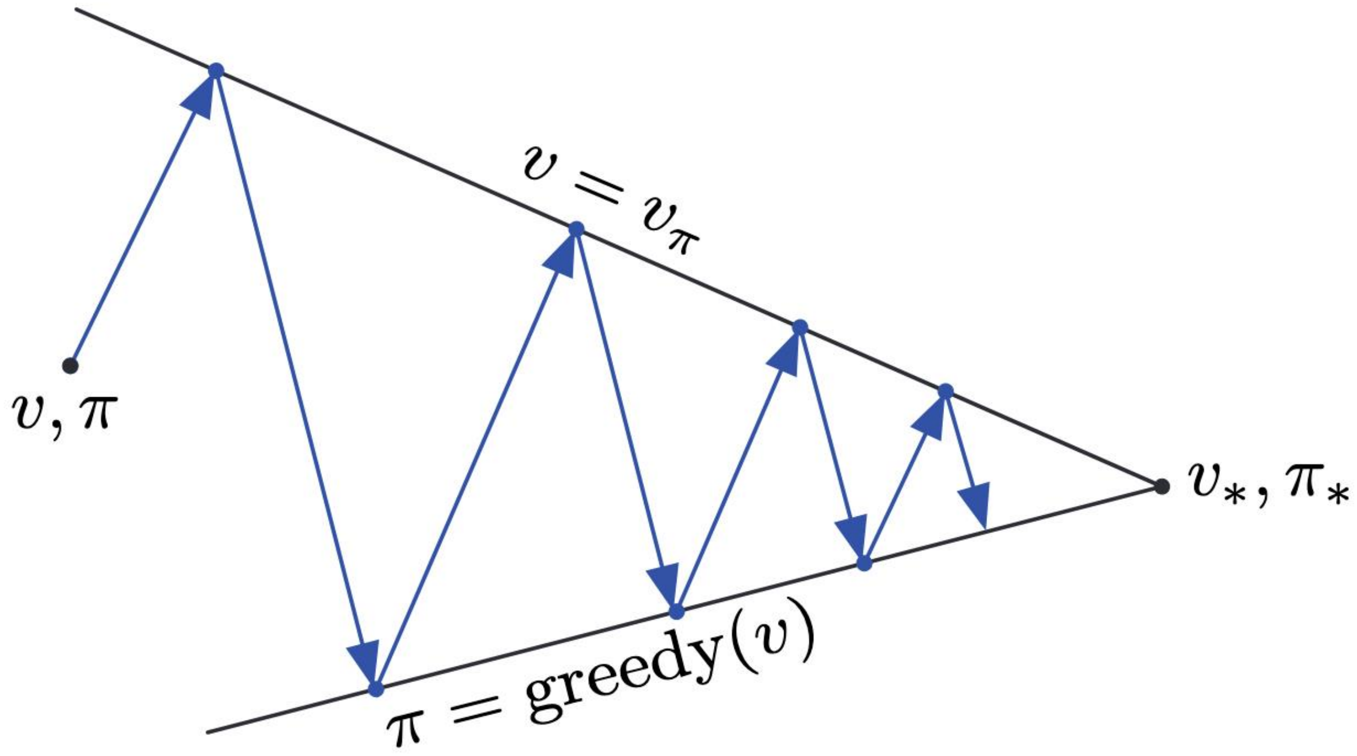
Recap: Generic one-step actor-critic

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
    Initialize S (first state of episode)
    $I \leftarrow 1$
    Loop while S is not terminal (for each time step):
        $A \sim \pi(\cdot|S, \boldsymbol{\theta})$
        Take action A, observe S', R
        $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S,\mathbf{w})$
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S, \boldsymbol{\theta})$
        $I \leftarrow \gamma I$
        $S \leftarrow S'$

Actor-critic and advantages

Actor-critic



Variance of actor-critic

Recall that in standard policy gradient, we have:

$$\theta = \theta + \underbrace{\alpha}_{\text{Learning rate}} \underbrace{\gamma^t}_{\text{Discount factor}} \underbrace{G_t}_{\text{Cumulative Return}} \nabla \ln \pi(a_t | s_t)$$

Variance of actor-critic

Recall that in standard policy gradient, we have:

$$\theta = \theta + \alpha\gamma^{t}G_t \nabla \ln \pi(a_t|s_t)$$

Learning rate

Discount factor

Cumulative Return

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$$

Variance of actor-critic

Recall that in standard policy gradient, we have:

$$\theta = \theta + \alpha\gamma^t G_t \nabla \ln \pi(a_t|s_t)$$

In actor-critic, we can formulate as:

$$\theta = \theta + \alpha\gamma^t Q(s_t, a_t) \nabla \ln \pi(a_t|s_t)$$

Variance of actor-critic

Recall that in standard policy gradient, we have:

$$\theta = \theta + \alpha\gamma^t G_t \nabla \ln \pi(a_t|s_t)$$

In actor-critic, we can formulate as:

$$\theta = \theta + \alpha\gamma^t Q(s_t, a_t) \nabla \ln \pi(a_t|s_t)$$

$$Q(s_t, a_t) = \mathbb{E}[G_t | s_t = s, a_t = a]$$

Reduces variance

Variance of actor-critic

However, we can further reduce the variance of $Q(s_t, a_t)$

As previously discussed, two components to a Q-value:

- How good the state is
- How good the action is relative to the "default"

Variance of actor-critic

However, we can further reduce the variance of $Q(s_t, a_t)$

As previously discussed, two components to a Q-value:

- ~~How good the state is~~ We can cancel this part out!
- How good the action is relative to the "default"

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Variance of actor-critic

Formulate actor-critic with advantages

$$\theta = \theta + \alpha\gamma^t A(s_t, a_t)\nabla\ln \pi(a_t|s_t)$$

Last time we formulated actor-critic with a TD error

$$\delta = r_{t+1} + \gamma V(s_t) - V(s_t)$$

$$\theta = \theta + \alpha\gamma^t \delta\nabla\ln \pi(a_t|s_t)$$

TD error is an estimate of the advantage

$$\theta = \theta + \alpha\gamma^t A(s_t, a_t)\nabla \ln \pi(a_t|s_t)$$

$$\theta = \theta + \alpha\gamma^t Q(s_t, a_t) - V(s_t)\nabla \ln \pi(a_t|s_t)$$

$$\theta = \theta + \alpha\gamma^t r_{t+1} + \gamma V(s_t) - V(s_t)\nabla \ln \pi(a_t|s_t)$$

$$\theta = \theta + \alpha\gamma^t \delta\nabla \ln \pi(a_t|s_t)$$

Controlling the bias/variance trade-off

Consider three cases:

1. 1-step TD. Lowest variance, highest bias.

$$A^{(1)}(s_t, a_t) = r_{t+1} + \gamma V(s_t) - V(s_t)$$

2. N-step TD. Intermediate variance, intermediate bias.

$$A^{(n)}(s_t, a_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^n V(s_{t+n}) - V(s_t)$$

3. Monte Carlo (w/ baseline). Highest variance, lowest bias

$$A^{(\infty)}(\mathrm{s_t}, \mathrm{a_t}) = G_t - V(s_t)$$

Generalized advantage estimation

$$A^{\mathrm{GAE}(\gamma,\lambda)} = (1-\lambda)(A^{(1)} + \lambda A^{(2)} + \lambda^2 A^{(3)} + \cdots)$$
$$= \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}$$

$$\delta_t = r_{t+1} + \gamma V(s_t) - V(s_t)$$

Schulman et al, High-Dimensional Continuous Control Using Generalized Advantage Estimation, 2016.

Generalized advantage estimation

GAE(γ,0) = 1-step TD = high bias, low variance

GAE(γ,1) = Monte Carlo = low bias, high variance

$0 < \lambda < 1$ trades off bias for variance

In practice, we often use $\lambda = [0.9,1.0]$

Note that this does not control how many steps we take, just how much we weigh TD errors

Natural policy gradient and trust region policy optimization

The effect of learning rate

In actor critic, we update model weights with

$$\theta = \theta + \alpha\gamma^t A(s_t, a_t)\nabla \ln \pi(a_t|s_t)$$

The learning rate α dictates our step size

How far should we step?

Reinforcement learning vs imitation learning

In on-policy reinforcement learning, our objective is

$$\max_{\theta} \mathbb{E}_{s,a\sim\pi_\theta}\left[\ln \pi_\theta(a|s) A(s,a)\right]$$

In behavior cloning, our objective is

$$\max_{\theta} \mathbb{E}_{(s,a)\sim\pi^*}\left[\ln \pi_\theta(a|s)\right]$$

Reinforcement learning vs imitation learning

In supervised learning, we tune step size to fit "labels" on a dataset

In (on-policy) RL, the step size *changes the dataset we see next*

- Step size is too big: bad policy update which means we collect bad data for next gradient update
- Step size is too small: we don't update policy enough, which means we collect very similar data each time

Step size depends on parameters

To make matters worse, the step size induces different changes to action probabilities depending on our policy parameters

- Does not account for difference between π_{old} and π_{new}

Example: assume Bernoulli policy

$$\pi_\theta(a) = \begin{cases} \sigma(\theta) & a = 0 \\ 1 - \sigma(\theta) & a = 1 \end{cases}$$

Step size depends on parameters

Let $p = \pi_\theta(a = 1)$

$$\Delta p = p(1 - p)\Delta\theta$$

For a fixed step $\Delta\theta$, the sensitivity of Δp depends on $p(1 - p)$

If $\Delta\theta = 0.2$

- If $p = 0.5$ then $\Delta p = 0.05$
- If $p = 0.99$ then $\Delta p = 0.00198$

Step size depends on parameters

Let $p = \pi_\theta(a = 1)$

For a fix

The same parameter step affects the policy differently depending on where we are in the parameter space.

)

If $\Delta\theta = 0.2$

- If $p = 0.5$ then $\Delta p = 0.05$
- If $p = 0.99$ then $\Delta p = 0.00198$

Intuition

Remember that in PG our gradients are approximate!

- What if there is error? Or we take too big of a step?

Some images borrowed from [Jonathan Hui's nice write-up]

Intuition

Instead, we define a “trust region” and only take steps within it

Line search
(like gradient ascent)

Trust region

Intuition

Do this enough times, and we safely follow our gradient

Gradient descent in terms of policies

Consider a parameterized distribution π_θ and objective $J(\theta)$

$$\theta_{\text{new}} = \theta_{\text{old}} + \Delta\theta$$

Gradient descent in terms of policies

Consider a parameterized distribution π_θ and objective $J(\theta)$

$$\theta_{\text{new}} = \theta_{\text{old}} + \Delta\theta^*$$

Gradient descent: step in parameter space is dictated by Euclidean distance of parameter vectors before/after update

$\Delta\theta^* = \arg\max J(\theta + \Delta\theta)$ where $\left|\left|\Delta\theta\right|\right| \leq \epsilon$ *(how far can we step)*

In other words, gradient descent is the steepest descent under the Euclidean norm.

Gradient descent in terms of policies

Consider a parameterized distribution π_θ and objective $J(\theta)$

$$\theta_{\text{new}} = \theta_{\text{old}} + \Delta\theta^*$$

Natural gradient descent: step in parameter space is dictated by KL divergence in the distributions before/after update

$\Delta\theta^* = \arg\max J(\theta + \Delta\theta)$ where $D_{\text{KL}}(\pi_\theta || \pi_{\theta+\Delta\theta}) \leq \epsilon$

In other words, natural gradient descent is the steepest direction under KL divergence (local approximation of Fisher information)

Refresher: Kullback-Leibler divergence

Measure of how much probability distribution Q differs from true distribution P

$$D_{\mathrm{KL}}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

Expected extra "surprise" if using data drawn from Q instead of P when the actual distribution is P

- $D_{\mathrm{KL}}(P||Q) = 0$ if $P = Q$

Goal: KL-constrained optimization

We want to improve our expected policy returns while subject to the constraint that we don't change the policy *too much* at once

- Constraint dictated by KL divergence

$$\max_{\Delta\theta} g^T \Delta\theta \quad \text{s.t.} \quad D_{\mathrm{KL}}\left(\pi_\theta || \pi_{\theta+\Delta\theta}\right) \leq \epsilon \quad \text{where } g = \nabla\theta J(\theta)$$

Maximize gains while ensuring KL $\leq \epsilon$

KL-constrained optimization

Small parameter steps can be approximated as quadratic

$$D(\theta, \theta + \Delta\theta) = \frac{1}{2} \Delta\theta^T H(\theta) \Delta\theta + \mathrm{h.o.t.}$$

Tells us curvature

due to Taylor expansion.

Locally, small steps look like the squared distance defined by the Hessian $H(\theta)$

KL-constrained optimization

Intuition: when we take a ”small step” the quadratic term tells us how fast the function curves away from our starting point.

When $D(\theta, \theta + \Delta\theta)$ is the KL divergence, the Hessian is known as the Fisher information matrix

$$F(\theta) = \mathbb{E}_{x\sim\theta+\Delta\theta}[\nabla\theta \log p_\theta(x) \nabla\theta \log p_\theta^T]$$

Measures how sensitive the log-likelihood is to parameter changes

Natural Policy Gradient

Putting the pieces together

$$D_{\mathrm{KL}}(\pi_\theta || \pi_{\theta+\Delta\theta}) \approx \frac{1}{2} \Delta\theta^T F(\theta) \Delta\theta$$

So natural gradient descent is then given by

$$\max_{\Delta\theta} g^T \Delta\theta \quad \mathrm{s.t.}\, \Delta\theta^T F(\theta) \Delta\theta \ \leq 2\epsilon$$

$$\Delta\theta^* = \alpha F^{-1} \nabla\theta J(\theta)$$

Remember that this is
$\nabla\theta \ln \pi(a|s) A(a,s)$

Kakade, A Natural Policy Gradient, 2001.

Natural Policy Gradient

$$\theta_{\text{new}} = \theta_{\text{old}} + \Delta\theta^*$$

Recalling back to our Taylor expansion:

$$D_{\text{KL}}(\pi_\theta || \pi_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^T F(\theta)\Delta\theta = \frac{1}{2}(\alpha\Delta\theta)^T F(\alpha\Delta\theta)$$

$$\theta_{\text{new}} = \theta_{\text{old}} + \sqrt{\frac{2\epsilon}{\Delta\theta F^{-1}\Delta\theta}} F^{-1}(\theta)\nabla\theta J(\theta)$$

Quadratic approximations are *local*

Since the quadratic is only local, in practice the actual empirical KL update may be larger than ϵ

Simple solution: perform a backtracking line search

1. Compute progressively smaller proposed steps
2. If step yields positive advantage and KL $\leq \epsilon$ then repeat

Trust region policy optimization

Algorithm 2 Line Search for TRPO

Compute proposed policy step $\Delta_k = \sqrt{\frac{2\epsilon}{\hat{g}_k^T \hat{F}_k^{-1} \hat{g}_k}} \hat{F}_k^{-1} \hat{g}_k$
for $j = 0, 1, 2, ..., L$ **do**
  Compute proposed update $\theta = \theta_k + \alpha^j \Delta_k$
  if $\bar{\mathbb{A}}_{\pi_{old}}(\pi) \geq 0$ and $\bar{D}_{KL}(\theta || \theta_k) \leq \delta$ **then**
    accept the update and set $\theta_{k+1} = \theta_k + \alpha^j \Delta_k$
    break
  end if
end for

Useful link:
https://spinningup.openai.com/en/latest/algorithms/trpo.html

Schulman et al, Trust Region Policy Optimization, 2015.

Trust region policy optimization

Algorithm 2 Line Search for TRPO

end for

The "trust region" is the area within the parameter space in which our quadratic model is accurate.

Useful link:
https://spinningup.openai.com/en/latest/algorithms/trpo.html

Schulman et al, Trust Region Policy Optimization, 2015.

Deep RL Foundations in 6 Lectures

Lecture 4: TRPO, PPO

Pieter Abbeel

https://youtu.be/KjWF8VIMGiY?si=OHdqO8uZQ_njEhKx&t=741

Take-aways

Policy gradient leverages advantages to reduce variance.

- Generalized advantage estimation can control bias/variance

Natural policy gradient controls step sizes in distribution space

TRPO adds a backtracking line search to natural policy gradient