Advanced Policy Gradients

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

Novemer 18, 2021

Overview of this Lecture

1. Introduction
2. Policy Performance Bounds
3. Approximations to Trust Region Formulation
4. Natural Policy Gradient

Introduction

Policy Gradients : The Story So Far

Policy gradient algorithms try to solve the optimization problem

$$\max_{\theta} J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right]$$

by taking stochastic gradient ascent on the policy parameters θ, using the policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\nabla_\theta \log \pi_\theta(a_t|s_t)\Psi_t\right]$$

1. $\Psi_t = \gamma^t Q^{\pi_\theta}(s_t, a_t)$, State action value function
2. $\Psi_t = \gamma^t A^{\pi_\theta}(s_t, a_t) = \gamma^t \left[Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t)\right]$, Advantage function
3. $\Psi_t = \gamma^t \left[r_{t+1} + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)\right]$, TD residual

Limitation 1 : Sample Inefficiency

Gradient of the performance measure is given by

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t)\Psi_t\right]$$

- Policy Gradient seen thus far is **on-policy**
- Gradient update is performed using samples collected from current policy
- Above formulation does not allow to recycle old data
- If we want to use samples from other policies, then the above gradient term needs correction using **importance sampling** weights

Importance Sampling in PG : Possible Remedy ?

What if we do not samples from π_θ instead samples are from π_η(i.e. $\tau \sim \pi_\eta$) ?

We can rewrite the gradient estimate as,

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\eta} \left[\frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\eta)} \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) \ \Psi_t \right]$$

$$\frac{P(\tau|\pi_\theta)}{P(\tau|\pi_\eta)} = \frac{\mu(s_0) \prod_{t=0}^{\infty} P(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t)}{\mu(s_0) \prod_{t=0}^{\infty} P(s_{t+1}|s_t, a_t)\pi_\eta(a_t|s_t)} = \prod_{t=0}^{\infty} \frac{\pi_\theta(a_t|s_t)}{\pi_\eta(a_t|s_t)}$$

Even for policies only slightly different from each other, many small differences multiply to become a big difference and IS weights can **explode** or **vanish**

Problem : How can we efficiently use samples from old policies while avoiding the challenges posed by importance sampling ?

Limitation 2 : Updates in Parameter Space

- Policy gradient takes step in parameter space

$$\theta_{n+1} = \theta_n + \alpha \nabla_{\theta_n} J(\theta_n)$$

- **Distance** in parameter space $\neq$ Distance in policy space
- Hard to get step size right as a result
- Example of a policy space : For finite state and action case, the policy space is given by

$$\Pi = \left\{ \pi : \pi \in \mathbb{R}^{|S| \times |A|}, \sum_a \pi_{sa} = 1, 0 \leq \pi_{sa} \leq 1 \right\}$$

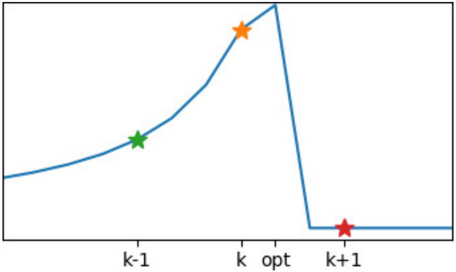
The Step Size Issue

Policy Gradient algorithms perform stochastic gradient ascent

$$\theta_{k+1} = \theta_k + \alpha_k \hat{g}_k$$

with step $\Delta_k = \alpha_k \hat{g}_k$

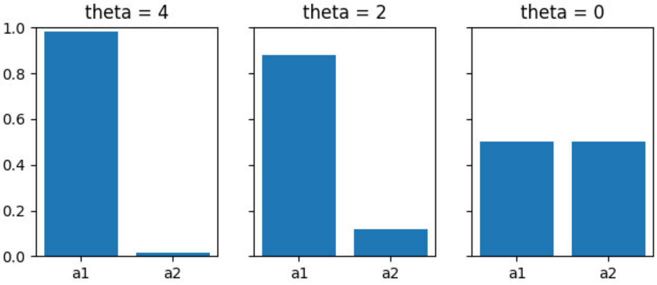
- Step size is too large, performance collapse
  - ★ Consequences in RL setting is more severe than for supervised learning
- Step size is too low, slow progress



Difference in Policy Space : Discrete Case

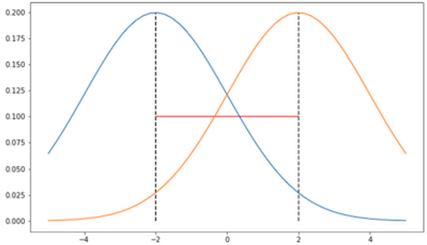
Consider a family of policies with the following parametrization

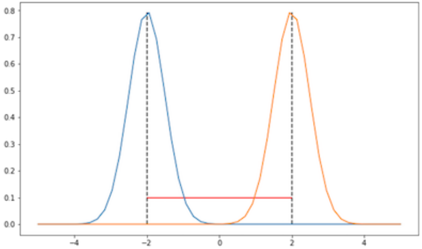
$$\pi_\theta(a) = \begin{cases} \sigma(\theta), & \text{if } a = 1 \\ 1 - \sigma(\theta), & \text{if } a = 2 \end{cases}$$



Small changes in the policy parameters can unexpectedly lead to big changes in the policy

Difference in Policy Space : Continuous Case





Desiderata

- How to make use of data from old policy while avoiding challenges that arise from importance sampling ?
  - ★ At least use roll-outs from **most recent policy** as effectively as possible
- How to design an update rule that doesn't change the policy more than we intend to ?
  - ★ Take steps that respect notion of **distance in policy space** rather than in parameter space

Policy Performance Bounds

Relative Performance of Two Policies

Recall that the performance of a policy π is given by

$$J(\pi) = V^{\pi}(s_0)$$

where $s_0 \in \mu(s)$ is the start state of the trajectory π

Relative Performance Identity of Two Policies

For any two policies π' and π_0 we have,

$$J(\pi') - J(\pi_0) = \mathop{\mathbb{E}}_{\tau \sim \pi'} \left[\gamma^t A^{\pi_0}(s_t, a_t) \right]$$

Proof of Relative Performance Identity of Two Policies

$$
\begin{aligned}
J(\pi') - J(\pi_0)) &\stackrel{?}{=} \mathop{\mathbb{E}}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi_0}(s_t, a_t)\right] \\
&= \mathop{\mathbb{E}}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^t \left[r_{t+1} + \gamma V^{\pi_0}(s_{t+1}) - V^{\pi_0}(s_t)\right]\right] \\
&= J(\pi') + \mathop{\mathbb{E}}_{\tau\sim\pi'}\left[\sum_{t=0}^{\infty}\gamma^{t+1} V^{\pi_0}(s_{t+1}) - \sum_{t=0}^{\infty}\gamma^t V^{\pi_0}(s_t)\right] \\
&= J(\pi') + \mathop{\mathbb{E}}_{\tau\sim\pi'}\left[\sum_{t=1}^{\infty}\gamma^t V^{\pi_0}(s_t) - \sum_{t=0}^{\infty}\gamma^t V^{\pi_0}(s_t)\right] \\
&= J(\pi') - \mathop{\mathbb{E}}_{\tau\sim\pi'}\left[V^{\pi_0}(s_0)\right] \\
&= J(\pi') - J(\pi_0)
\end{aligned}
$$

Policy Improvement Idea

Can we use the identity,

$$J(\pi') - J(\pi_0) = \mathop{\mathbb{E}}_{\tau \sim \pi'} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_0}(s_t, a_t) \right]$$

for policy improvement to go from **old policy** π_0 to **new policy** π' ?

$$\arg\max_{\pi'} J(\pi') = \arg\max_{\pi'} \left[J(\pi') - J(\pi_0) \right] = \arg\max_{\pi'} \left[\mathop{\mathbb{E}}_{\tau \sim \pi'} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_0}(s_t, a_t) \right] \right]$$

- Define performance of π' in terms of advantages from π_0
- Problem : Still requires trajectories from π' !!

Discounted State Distribution

Define discounted future state distribution d^π for a state s as

$$d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s|\pi)$$

Consider the expectation

$$\mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t A^\pi(s_t, a_t)\right]$$

The above expectation can be rewritten using the discounted future state distribution d^π as

$$\frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^\pi \\ a\sim\pi}}\left[A^\pi(s,a)\right]$$

That is,

$$\mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t A^\pi(s_t, a_t)\right] = \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^\pi \\ a\sim\pi}}\left[A^\pi(s,a)\right]$$

Performance Identity and Discounted State Distribution

$$
\begin{aligned}
J(\pi') - J(\pi_0) &= \mathop{\mathbb{E}}_{\tau \sim \pi'} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_0}(s_t, a_t) \right] \\
&= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi'} \\ a \sim \pi'}} \left[A^{\pi_0}(s, a) \right] \\
&= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi'} \\ a \sim \pi_0}} \left[\frac{\pi'(a|s)}{\pi_0(a|s)} A^{\pi_0}(s, a) \right]
\end{aligned}
$$

Only problem is $s \sim d^{\pi'}$!

Approximation of Discounted State Distribution

Under what conditions is it OK to assume

$$\frac{1}{1-\gamma} \underset{\substack{s\sim d^{\pi'} \\ a\sim\pi_0}}{\mathbb{E}} \left[\frac{\pi'(a|s)}{\pi_0(a|s)} A^{\pi_0}(s,a) \right] \overset{?}{\approx} \frac{1}{1-\gamma} \underset{\substack{s\sim d^{\pi_0} \\ a\sim\pi_0}}{\mathbb{E}} \left[\frac{\pi'(a|s)}{\pi_0(a|s)} A^{\pi_0}(s,a) \right]$$

In other words, under what conditions can we have $d^{\pi'} \approx d^{\pi_0}$?

$$\boxed{\text{We can have } d^{\pi'} \approx d^{\pi_0}, \text{ if } \pi' \text{ and } \pi_0, \text{ are close !!}}$$

- Closeness is measured in terms of **KL divergence**
- If the policies are close in **KL divergence**, then the above approximation is good

KL Divergence Between Policies

For any two probability distributions P and Q $D_{KL}(P||Q)$ is defined as

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

KL divergence has the following properties

- $D_{KL}(P||P) = 0$
- $D_{KL}(P||Q) \geq 0$
- $D_{KL}(P||Q) \neq D_{KL}(Q||P)$

KL divergence between policies π' and π_0 is given by

$$D_{KL}(\pi'||\pi_0)[s] = \sum_{a \in \mathcal{A}} \pi'(a|s) \log \frac{\pi'(a|s)}{\pi_0(a|s)}$$

Surrogate Loss Function

If policies π' and π_0 are **'close'** in terms of their KL divergence, then,

$$\begin{aligned} J(\pi') - J(\pi_0) &= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi'} \\ a \sim \pi_0}} \left[\frac{\pi'(a|s)}{\pi_0(a|s)} A^{\pi_0}(s,a) \right] \\ &\approx \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi_0} \\ a \sim \pi_0}} \left[\frac{\pi'(a|s)}{\pi_0(a|s)} A^{\pi_0}(s,a) \right] \\ &= \mathop{\mathbb{E}}_{\tau \sim \pi_0} \left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi'(a_t|s_t)}{\pi_0(a_t|s_t)} A^{\pi_0}(s_t,a_t) \right] \end{aligned}$$

Define the quantity $\mathcal{L}_{\pi_0}(\pi')$ as the **surrogate loss function**, as,

$$\mathcal{L}_{\pi_0}(\pi') = \mathop{\mathbb{E}}_{\tau \sim \pi_0} \left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi'(a_t|s_t)}{\pi_0(a_t|s_t)} A^{\pi_0}(s_t,a_t) \right]$$

- $J(\pi') - J(\pi_0) \approx \mathcal{L}_{\pi_0}(\pi')$

Validity of Approximation

If policies π' and π_0 are 'close' in terms of their KL divergence, then,

$$J(\pi') - J(\pi_0) \approx \; \mathcal{L}_{\pi_0}(\pi')$$

We can have a **relative policy performance bound** using KL divergence as

$$\left[J(\pi') - \left(J(\pi_0) + \mathcal{L}_{\pi_0}(\pi')\right)\right] \leq C\sqrt{\mathop{\mathbb{E}}_{s\sim d^{\pi_0}}\left[D_{KL}(\pi'||\pi_0)[s]\right]}$$

where $C = \frac{4\varepsilon\gamma}{1-\gamma^2}\alpha^2$ with $\alpha = \max_{s\sim d^{\pi_0}}\left[D_{KL}(\pi'||\pi_0)[s]\right]$ and $\varepsilon = \max_{s,a} A^{\pi_0}(s,a)$

Surrogate Loss Function : Properties

$$\mathcal{L}_{\pi_0}(\pi') = \mathop{\mathbb{E}}_{\tau \sim \pi_0}\left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi'(a_t|s_t)}{\pi_0(a_t|s_t)} A^{\pi_0}(s_t, a_t)\right]$$

- $\mathcal{L}_{\pi_0}(\pi')$ is something that we can evaluate using samples from old policy π_0
- Importance sampling is used; but weights depends only on current time-step (not preceding history); hence importance sampling weights don't vanish or explode
- Let π_{θ_k} and π_θ be two parametrized policies. Then, $\nabla_\theta \mathcal{L}_{\pi_{\theta_k}}(\pi_\theta)|_{\theta=\theta_k}$, is equal to the policy gradient

$$\nabla_\theta \mathcal{L}_{\pi_{\theta_k}}(\pi_\theta)|_{\theta=\theta_k} = \mathop{\mathbb{E}}_{\tau \sim \pi_{\theta_k}}\left[\sum_{t=0}^{\infty} \frac{\nabla_\theta \pi_\theta(a_t|s_t)|_{\theta=\theta_k}}{\pi_{\theta_k}(a_t|s_t)} \gamma^t A^{\pi_{\theta_k}}(s_t, a_t)\right]$$

$$= \mathop{\mathbb{E}}_{\tau \sim \pi_{\theta_k}}\left[\sum_{t=0}^{\infty} \nabla_\theta \log(\pi_{\theta_k}(a_t|s_t)|_{\theta=\theta_k} \gamma^t A^{\pi_{\theta_k}}(s_t, a_t)\right]$$

Policy Improvement Guarantee

We can rewrite the relative policy performance bound equation as

$$\left[J(\pi') - J(\pi_0)\right] \geq \mathcal{L}_{\pi_0}(\pi') - C\sqrt{\underset{s\sim d^{\pi_0}}{\mathbb{E}}\left[D_{KL}(\pi'||\pi_0)[s]\right]}$$

Suppose π_{k+1} and π_k are related by

$$\pi_{k+1} = \arg\max_{\pi'}\left[\mathcal{L}_{\pi_k}(\pi') - C\sqrt{\underset{s\sim d^{\pi_k}}{\mathbb{E}}\left[D_{KL}(\pi'||\pi_k)[s]\right]}\right]$$

- Note π_k is a feasible point and the objective at π_k is equal to 0
  - ★ $\mathcal{L}_{\pi_k}(\pi_k) \propto \mathbb{E}[A^{\pi_k}(s_t, a_t)] = 0$
  - ★ $D_{KL}(\pi_k||\pi_k) = 0$
- $\implies$ Optimal value ≥ 0
- By the performance bound inequality, we have $J(\pi_{k+1}) - J(\pi_k) \geq 0$

A First-Cut Algorithm

1: Initialize π_0
2: **for** $k = 0, 1, 2, \cdots$ until convergence **do**
3: Sample a trajectory τ from policy π_k
4: Compute advantage function $A^{\pi_{\theta_k}}(a_t, s_t)$ for all (s_t, a_t) pairs in the trajectory τ
5: Solve the optimization problem

$$\pi_{k+1} = \arg\max_{\pi'} L_{\pi_k}(\pi') - C\sqrt{\mathbb{E}_{s \sim d^{\pi_k}}\left[D_{KL}(\pi' || \pi_k)[s]\right]}$$

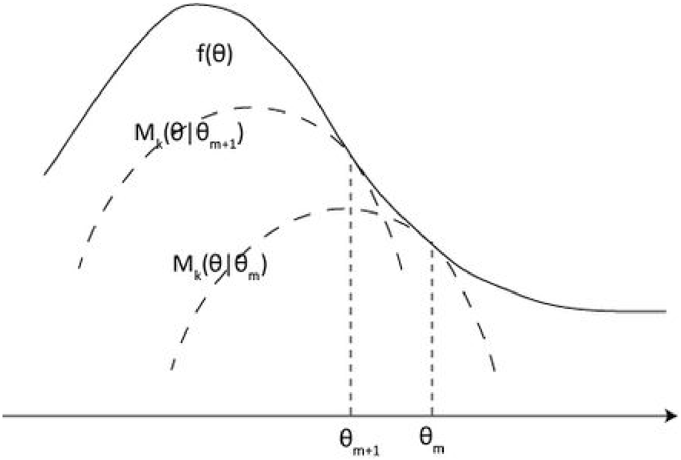
6: **end for**

Issues are :

- C is quite high when γ is close to 1 $\left(C = \frac{4\varepsilon\gamma}{1-\gamma^2}\alpha^2\right)$
- Consequently, step size becomes too small

Majorize Maximize Framework

Majorize-Maximize framework is used to solve the optimization step



Approximate Monotone Improvement

- Instead of KL penalty, use KL constraint
- Can control worst case error through constraint upper limit

$$
\begin{aligned}
\pi_{k+1} = \arg\max_{\pi'} & \left[L_{\pi_k}(\pi')\right] \\
\text{such that } & \mathop{\mathbb{E}}_{s\sim d^{\pi_k}} D_{KL}(\pi' || \pi_k)[s] \leq \delta
\end{aligned}
$$

- From the constraint, **steps respect a notion of distance in policy space**
- Above constrained optimization is basis of many algorithms, Natural Policy Gradient (NPG), truncated NPG, TRPO and PPO
- The objective and the constraint can be estimated from the roll-out of old policies – **sample efficient**
- Update is parametrization invariant

Approximations to Trust Region Formulation

Trust Region Formulation

We have the following optimization problem

$$\pi_{k+1} = \arg\max_{\pi'} \left[\mathcal{L}_{\pi_k}(\pi')\right]$$
$$\text{such that } \bar{D}_{KL}(\pi' || \pi_k) \leq \delta$$

The constraint on the optimization problem is the trust region with size δ and some guarantees on performance improvement are there within the trust region

For parametrized policies the optimization can be written as

$$\pi_{\theta_{k+1}} = \arg\max_{\pi_\theta} \left[\mathcal{L}_{\pi_{\theta_k}}(\pi_\theta)\right]$$
$$\text{such that } \bar{D}_{KL}(\pi_\theta || \pi_{\theta_k}) \leq \delta$$

How do we solve it ?

- Linear approximation for the objective
- Quadratic approximation for the constraint

Approximation of Objective Function

Taylor series expansion for function $f(x)$ around point a is given by

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \cdots$$

- Using Taylor series expansion on objective function $\mathcal{L}_{\pi_{\theta_k}}(\pi_\theta)$ around θ_k (upto first order term) gives us

$$\mathcal{L}_{\pi_{\theta_k}}(\pi_\theta) \approx \cancelto{0}{\mathcal{L}_{\pi_{\theta_k}}(\pi_{\theta_k})} + g^T(\theta - \theta_k) \qquad \text{where } g \doteq \nabla_\theta \mathcal{L}_{\pi_{\theta_k}}(\pi_\theta) \mid_{\theta=\theta_k}$$

- Recall that g is exactly the policy gradient (from previous lecture !)

$$\nabla_\theta \mathcal{L}_{\pi_{\theta_k}}(\pi_\theta)|_{\theta=\theta_k} = \mathop{\mathbb{E}}_{\tau \sim \pi_{\theta_k}} \left[\sum_{t=0}^{\infty} \nabla_\theta \log(\pi_{\theta_k}(a_t|s_t)|_{\theta=\theta_k} \; \gamma^t A^{\pi_{\theta_k}}(s_t, a_t) \right]$$

- Objective function is simplified to

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

Approximation of Trust Region Constraint

Using Taylor series expansion on the constraint (around θ_k; upto second order) gives us

$$\bar{D}_{KL}(\pi_\theta||\pi_{\theta_k}) \approx \cancelto{0}{\bar{D}_{KL}(\pi_{\theta_k}||\pi_{\theta_k})} + \nabla_\theta \cancelto{0}{\bar{D}_{KL}(\pi_\theta||\pi_{\theta_k})|_{\theta=\theta_k}} + \nabla^2_\theta \bar{D}_{KL}(\pi_\theta||\pi_{\theta_k})\ |_{\theta=\theta_k}$$

The first order term $\nabla_\theta \bar{D}_{KL}(\pi_\theta||\pi_{\theta_k})$ evaluates to zero since the expectation of the score function is zero

$$\nabla_\theta \bar{D}_{KL}(\pi_\theta \,\|\, \pi_{\theta_k}) = \nabla_\theta \mathop{\mathbb{E}}_{\pi_\theta}[\log \pi_\theta] - \nabla_\theta \mathop{\mathbb{E}}_{\pi_\theta}[\log \pi_{\theta_k}] = \mathop{\mathbb{E}}_{\pi_\theta}[\nabla_\theta \log \pi_\theta] = 0$$

Therefore, we are left only with the second order term

$$\bar{D}_{KL}(\pi_\theta||\pi_{\theta_k}) \approx \frac{1}{2}(\theta - \theta_k)^T H\ (\theta - \theta_k) \qquad \text{where } H \doteq \nabla^2_\theta \bar{D}_{KL}(\pi_\theta||\pi_{\theta_k)}\ |_{\theta=\theta_k}$$

Natural Policy Gradient

Natural Policy Gradient

The optimization problem is now simplified as

$$\theta_{k+1} = \arg\max_{\theta} g^T(\theta - \theta_k)$$
$$\text{such that } \frac{1}{2}(\theta - \theta_k)^T H \, (\theta - \theta_k) \leq \delta$$

Linear objective with quadratic constraint

Solution to the approximate problem obtained using Lagrange multiplier method

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

The term $H^{-1}g$ is called the Natural gradient

Idea of the Derivation of Natural Gradient

The optimization problem from previous slide, in general, can be formulated as

$$d^* = \underset{d \text{ s.t. } \mathrm{KL}[\pi_{\theta+d} \| \pi_\theta] = \delta}{\arg\max} J(\theta + d)\,,$$

(Function J here can be thought of as linearized objective function from previous slide)

Recasting the optimization problem using the Lagrangian

$$\begin{aligned} d^* &= \arg\max_d J(\theta + d) + \lambda\left(\mathrm{KL}[\pi_{\theta+d} \| \pi_\theta] - \delta\right) \\ &\approx \arg\max_d J(\theta) + \nabla_\theta J(\theta)^{\mathrm{T}} d + \frac{1}{2}\lambda\, d^{\mathrm{T}} \mathrm{H} d - \lambda\delta\,. \end{aligned}$$

To solve this optimization, we set its derivative wrt. d to zero

$$\begin{aligned} 0 &= \frac{\partial}{\partial d} J(\theta) + \nabla_\theta J(\theta)^{\mathrm{T}} d + \frac{1}{2}\lambda\, d^{\mathrm{T}} \mathrm{H} d - \lambda\delta \\ &= \nabla_\theta J(\theta) + \lambda\, \mathrm{H} d \\ \lambda\, \mathrm{H} d &= -\nabla_\theta J(\theta) \\ d &= -\frac{1}{\lambda} \mathrm{H}^{-1} \nabla_\theta J(\theta) = -\frac{1}{\lambda} \mathrm{H}^{-1} g \end{aligned}$$

Algorithm : Natural Policy Gradient

Algorithm Natural Policy Gradient

1: Initialize π_0
2: **for** $k = 0, 1, 2, \cdots$ **do**
3: Collect trajectories D_k on policy $\pi_k = \pi_{\theta_k}$
4: Estimate all advantages $A^{\pi_{\theta_k}}(s_t, a_t)$
5: Form sample estimates for policy gradients $\hat{g}_k$ (using advantage estimates)
6: Form sample estimates for the Hessian of KL divergence
7: Compute the Natural Policy Gradient update

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g_k^T H_k^{-1} g_k}} H_k^{-1} g_k$$

8: **end for**