Trust Region methods

Konpat Preechakul
Chulalongkorn University
November 2019

Recap policy gradient

- Policy gradient theorem

$$\nabla J(\theta) = \mathbb{E}_{s,a}\left[Q^{\pi}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)\right]$$

  - Variance problem
- Advantage

$$A(s,a) = Q(s,a) - V(s)$$

  - Variance reduction
  - A2C

$$\nabla J(\theta) = \mathbb{E}_{s,a}\left[A(s,a)\nabla\log\pi_{\theta}(a|s)\right]$$

Recap policy gradient

- Off-policy gradient
  - Off-policy critic
  - Off-policy actor

$$\nabla_\theta J(\theta) \approx \sum_s d^b(s) \sum_a Q^\pi(s,a) \nabla_\theta \pi_\theta(a|s)$$

- Deterministic policy gradient
  - DDPG

$$\nabla_\theta J(\theta) = \sum_s d^\pi(s) \nabla_a Q_\phi(s,a)|_{a=\pi(s)} \nabla_\theta \pi_\theta(s)$$

Today’s topic

Today topics

- Why policy gradient fails?
- More robust policy gradient
  - Trust region methods
  - Approximation
- Trust region on critic
- Natural gradients

Why policy gradient fails?

Bad critic

- Critic is not oracle, it has its flaws
- **What are some flaws?**

Bad critic

- Critic is not oracle, it has its flaws
- How to reduce the flaws?
- **Critic is prone to forgetting**
- When does critic forget?

Critic is prone to forgetting

- On-policy training “limits” kind of data the critic sees

- If the data is concentrated in “late game”, the critic forgets “early game” states
  - It is likely to give gibberish Q to the actor

- Underlines using replay, parallel actors

Bad critic

- Critic is not oracle, it has its flaws
- Critic is prone to forgetting
- **How to know when to trust?**

When to trust critic?

- Do critic have “confidence”?

When to trust critic?

- Do critic have “confidence”?
- How to get one?

When to trust critic?

- Do critic have “confidence”?
- How to get one?
- Confidence is challenging:
  - Critic outputs variance?

When to trust critic?

- Do critic have “confidence”?
- How to get one?
- Confidence is challenging:
  - Critic outputs variance?
  - Multiple critics?

When to trust critic?

- Do critic have "confidence"?
- How to get one?
- Confidence is challenging:
  - Critic outputs variance?
  - Multiple critics?
  - Critic dropout?

Gal, Yarin, and Zoubin Ghahramani. 2015. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." *arXiv [stat.ML]*. arXiv. http://arxiv.org/abs/1506.02142.

Bad critic

- Critic is not oracle, it has its flaws
- Critic is prone to forgetting
- How to know when to trust?
- **How to limit the trust?**
- Not updating too much ...

Not updating too much

Not updating too much

- How much is too much?
- The right LR?
- What is the right LR?
- If the critic is “abruptly” large, no small LR is small enough

- We need to be serious about update!

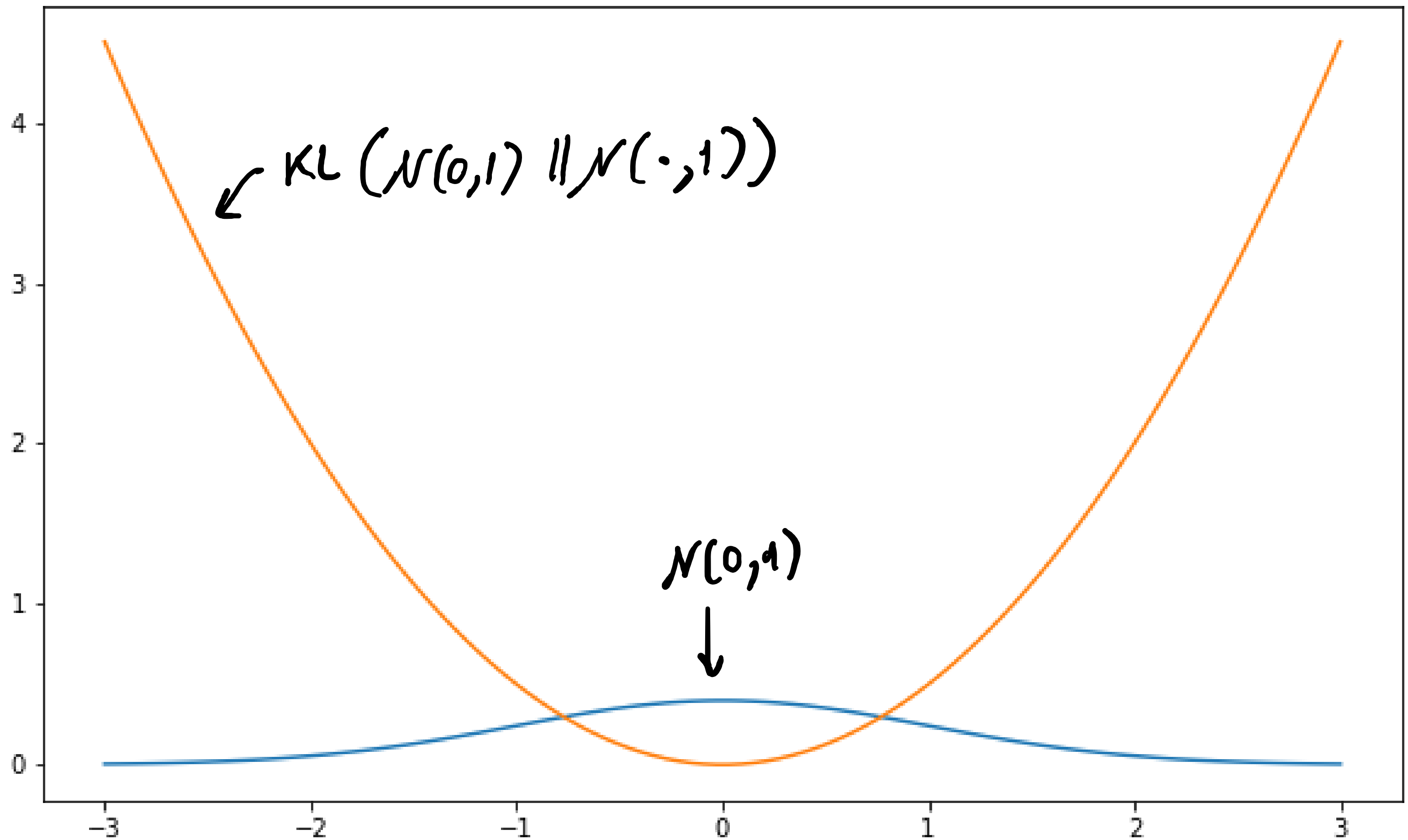
Quantifying "much"

- Policies are probability functions
- We find a "distance" measure on the probabilities
- KL divergence:

$$\mathrm{KL}(P\|Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$$

KL Divergence in picture

$$\mathrm{KL}(P\|Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$$



Quantify "much" in RL

- Context of RL, for a state S:

$$\mathrm{KL}_s(\theta\|\theta') = \sum_a \pi_\theta(a|s)[\log \pi_\theta(a|s) - \log \pi_{\theta'}(a|s)]$$

- For all states:

$$\mathrm{KL}(\theta\|\theta') = \sum_s d^\pi(s)\mathrm{KL}_s(\theta\|\theta')$$

Do policy gradient while making sure not going too far in terms of KL

New objective for policy update

- From the policy gradient objective

$$J(\theta) = \sum_s P_{s_0} V^{\pi}(s)$$

$$\theta \leftarrow \theta + d \qquad d = \alpha \nabla_\theta J(\theta)$$

- A new update direction should be:

don't worry too much

$$d^* = \underset{d}{\operatorname{argmax}}\ J(\theta + d) \quad \text{s.t.}\ \ \mathrm{KL}(\theta \| \theta + d) = c$$

New update direction

$$d^* = \underset{d}{\operatorname{argmax}}\; J(\theta + d) \quad \text{s.t.}\;\; \mathrm{KL}(\theta \| \theta + d) = c$$

- A constrained optimization

- We relax them using **Lagrangian:**

$$\mathcal{L}(d, \lambda) = J(\theta + d) + \lambda\left(\mathrm{KL}(\theta \| \theta + d) - c\right)$$

- Optimal d is at the critical point

$$\nabla_{d,\lambda}\mathcal{L} = 0$$

Solving for the update direction

$$\mathcal{L}(d, \lambda) = J(\theta + d) + \lambda \left(\mathrm{KL}(\theta \| \theta + d) - c\right)$$

$$\nabla_{d,\lambda} \mathcal{L} = 0$$

Problematic terms

Cannot calculate easily

$\nabla_d J(\theta + d)$

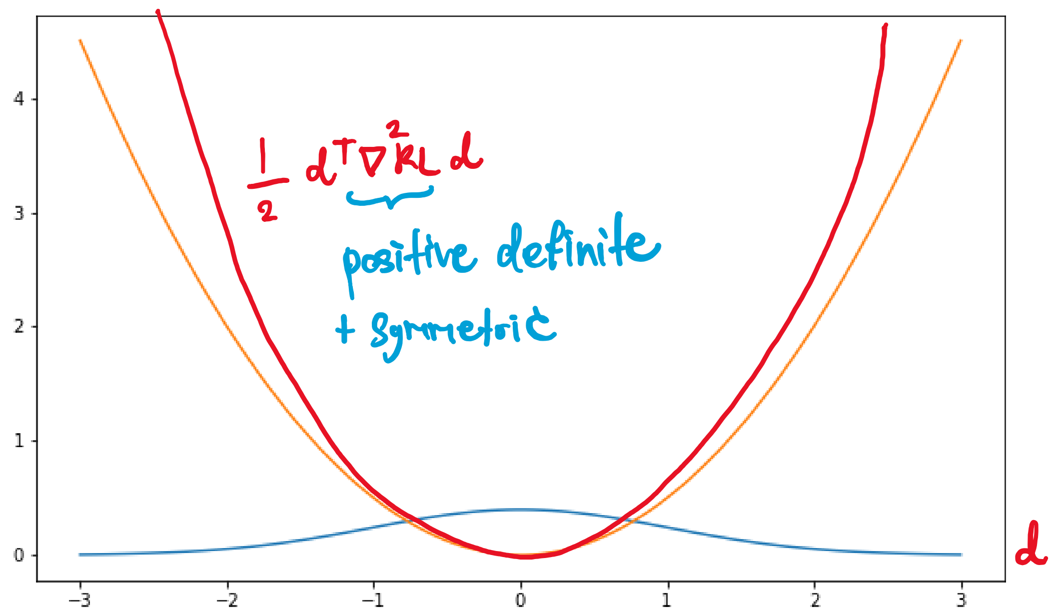
$\nabla_d \mathrm{KL}(\theta \| \theta + d)$

Problematic term

Taylor's to the second order:

$$\mathrm{KL}(\theta \| \theta + d) \approx \overline{\mathrm{KL}}(\theta \| \theta + d)$$

$$= \mathrm{KL}(\theta \| \theta) + \nabla_{\theta'} \mathrm{KL}(\theta \| \theta')|_{\theta'=\theta} d + \frac{1}{2} d^T \nabla^2_{\theta'} \mathrm{KL}(\theta \| \theta')|_{\theta'=\theta} d$$



Solving for the update direction

New update direction

$$d = \frac{s}{\lambda}$$

what does it really mean

$$s = (\nabla_\theta^2 \mathrm{KL})^{-1} \nabla_\theta J(\theta)$$

previously

$\nabla_\theta L(\theta)$

$$\lambda = \sqrt{\frac{s^T \nabla^2 \mathrm{KL} s}{2c}}$$

$$\theta \leftarrow \theta + d$$

Inverse of KL?

- Naïve inverse is not possible to calculate online

$$\nabla_\theta^2 \mathrm{KL}^{-1} = \left(\mathbb{E}_s\left[\nabla_\theta^2 \mathrm{KL}_s\right]\right)^{-1}$$

$$\nabla^2_{\theta'} \mathrm{KL}\left(\pi_\theta(\cdot|s) \,\|\, \pi_{\theta'}(\cdot|s)\right)\Big|_{\theta'=\theta}$$

- We use:

$$\nabla_\theta^2 \mathrm{KL}^{-1} \approx \mathbb{E}_s\left[\left(\nabla_\theta^2 \mathrm{KL}_s\right)^{-1}\right]$$

Finally, Limited trust PG is

$$\theta \leftarrow \theta + d \qquad d = \frac{s}{\lambda} \qquad \lambda = \sqrt{\frac{s^T \nabla^2 \mathrm{KL} s}{2c}}$$

← Trust region

$$s = (\nabla^2_\theta \mathrm{KL})^{-1} \nabla_\theta J(\theta)$$

$$s \approx \mathbb{E}_s \left[\left(\nabla^2_\theta \mathrm{KL}_s \right)^{-1} \mathbb{E}_a \left[Q^\pi(s,a) \nabla_\theta \log \pi_\theta(a|s) \right] \right]$$

Calculation of inverse is expensive

There are tricks to improve this even further

Schulman, John, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2015. "Trust Region Policy Optimization." *arXiv [cs.LG]*. arXiv. https://doi.org/10.1063/1.4927398.

Bad critic

- Critic is not oracle, it has its flaws
- Critic is prone to forgetting
- How to know when to trust?
- **How to limit the trust?**

Bad critic

- Critic is not oracle, it has its flaws
- Critic is prone to forgetting
- How to know when to trust?
- How to limit the trust?

- **Still doesn’t guarantee policy improvement**

Policy improvement guarantee

Making, Decision, and Control Sergey Levine. n.d. “Advanced Policy Gradients.”

Motivation

- We now have better update
- But, still need to know how large or small the "C" (trust parameter) is
- C could be "varying"
- Too large C could degrade policy
- Too small C is too conservative
- **We want to find C that is just right**

Forming the problem

- We want to "guarantee" policy improvement

$$J(\theta) = \mathbb{E}_{s_0 \sim P(s_0)} \left[V^{\pi}(s_0) \right]$$

$$J(\theta') \geq J(\theta)$$

- Objective becomes:

$$\underset{\theta'}{\operatorname{argmax}} \; J(\theta') - J(\theta)$$

- How to estimate?

The problem of estimation

- We want $J(\theta')$

- We need:
  - Create a new policy
  - Evaluate the policy

- Aim:
  - Estimate the new policy from **what we have**

Write it in another form

Write it in another form

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P'(s_t)} \left[\mathbb{E}_{a_t \sim \pi} \left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)} \gamma^t A^\pi(s_t, a_t) \right] \right]$$

Can we **lower bound** it while using **only what we have?**

$$J(\theta') - J(\theta) \geq \sum_t \mathbb{E}_{s_t \sim P(s_t)} \left[\mathbb{E}_{a_t \sim \pi} \left[\ldots \right] \right]$$

We need to bound: $|P(s_t) - P'(s_t)|$

Bound the state probability

Intuition: $\pi' \approx \pi \rightarrow P' \approx P$

We will show only a "glimpse"

Assume π is deterministic $a_t = \pi(s_t)$

Let:

$$\pi'(\overbrace{a_t \neq \pi(s_t)}^{\text{not taking } a_t}|s_t) \leq \epsilon \leftarrow \text{closeness}$$

Then:

$$P'(s_t) = \underbrace{(1-\epsilon)^t P(s_t)}_{\text{taking action like } \pi} + \underbrace{(1-(1-\epsilon)^t)}_{\text{taking at least 1 wrong action}} P_{\text{oth}}(s_t)$$

Bound the probability

$$|P(s_t) - P'(s_t)|$$

General policy case

- It is also possible to show (not here):

$$|\pi'(a|s) - \pi(a|s)| \leq \epsilon$$

$$|P(s_t) - P'(s_t)| \leq \epsilon t$$

- In optimization, we don't have π'
- We need to estimate it, if so:
- Using KL instead would ease estimation

Using KL

- It is possible to show (not here):

$$|\pi'(a|s) - \pi(a|s)| \leq \sqrt{\frac{1}{2}\mathrm{KL}^{\max}(\pi\|\pi')} = \epsilon'$$

$$|P(s_t) - P'(s_t)| \leq \epsilon' t$$

Schulman, John, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2015. “Trust Region Policy Optimization.” *arXiv [cs.LG]*. arXiv. https://doi.org/10.1063/1.4927398.

Bound some function

$$|P(s_t) - P'(s_t)| \leq \epsilon t$$

$$\mathbb{E}_{s_t \sim P'(s_t)} [f(s_t)]$$

Returning to our objective

$$\mathcal{A}_\pi(\pi') = \mathbb{E}_{a_t \sim \pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)}\gamma^t A^\pi(s_t, a_t)\right]$$

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P'(s_t)}\left[\mathcal{A}_\pi(\pi')\right]$$

Lower bound:

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P(s_t)}\left[\mathcal{A}_\pi(\pi')\right] - \overbrace{\sum_t \epsilon t \mathcal{O}\left(\frac{r_{\max}}{1-\gamma}\right)}^{\text{make } \epsilon \text{ small, ignore this!}}$$

$$\text{s.t. } \sqrt{\frac{1}{2}\mathrm{KL}^{\max}(\pi \| \pi')} = \epsilon$$

New objective

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P(s_t)} \left[\mathcal{A}_\pi(\pi') \right] - \sum_t \epsilon t \mathcal{O}\left(\frac{r_{\max}}{1-\gamma} \right)$$

$$\text{s.t.} \ \sqrt{\frac{1}{2} \mathrm{KL}^{\max}(\pi \| \pi')} = \epsilon$$

- Bound is very loose
- We should interpret as:

Keep KL small, policy improves!

Approaching optimization

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P(s_t)} \left[\mathcal{A}_\pi(\pi')\right] - \sum_t \epsilon t \mathcal{O}\left(\frac{r_{\max}}{1-\gamma}\right)$$

$$\text{s.t. } \sqrt{\frac{1}{2}\mathrm{KL}^{\max}(\pi \| \pi')} = \epsilon$$

Becomes:

$$\underset{\theta'}{\operatorname{argmax}}\ \mathcal{J}(\theta') \quad \text{s.t. } \mathrm{KL}^{\max}(\pi \| \pi_{\theta'}) = \epsilon$$

Approaching optimization

$$\underset{\theta'}{\text{argmax}}\ \mathcal{J}(\theta') \quad \text{s.t. } \mathrm{KL}^{\max}(\pi \| \pi_{\theta'}) = \epsilon$$

$\mathrm{KL}^{\max}(\pi \| \pi_{\theta'})$ is impractical to estimate

We need to go for all S to get the max

Approximate as:

$$\underset{\theta'}{\text{argmax}}\ \mathcal{J}(\theta') \quad \text{s.t. } \mathbb{E}_s\left[\mathrm{KL}_{\mathrm{s}}(\pi \| \pi_{\theta'})\right] = \epsilon$$

Seem familiar?

Policy improvement guarantee:

$$\underset{\theta'}{\operatorname{argmax}} \; \mathcal{J}(\theta') \quad \text{s.t.} \;\; \mathbb{E}_s\left[\mathrm{KL}_\mathrm{s}(\pi \| \pi_{\theta'})\right] = \epsilon$$

How to calculate $\nabla_{\theta'} J(\theta')$

Limited trust PG:

$$\underset{d}{\operatorname{argmax}} \; J(\theta + d) \quad \text{s.t.} \;\; \mathrm{KL}(\theta \| \theta + d) = c$$

Estimating the gradient

$$\mathcal{J}(\theta') = \sum_t \mathbb{E}_{s_t, a_t \sim \pi}\left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi(a_t|s_t)} \gamma^t A^\pi(s_t, a_t)\right]$$

Same old Taylor trick:

$$\nabla_{\theta'} \mathcal{J}(\theta') \approx \nabla_{\theta'} \mathcal{J}(\theta')|_{\theta'=\theta}$$

Policy improvement guarantee

$$s \approx \mathbb{E}_s \left[\left(\nabla_\theta^2 \mathrm{KL}_s \right)^{-1} \mathbb{E}_a \left[Q^\pi(s,a) \nabla_\theta \log \pi_\theta(a|s) \right] \right]$$

- Turns out to be the same as “Limited trust PG”
- Limit trust => Policy improvement with high chance
- We still cannot “guarantee” we don’t know epsilon

- There is another interpretation of $\nabla_\theta^2 \mathrm{KL}_s$

Natural gradients

Motivation

- Gradient descent is not always “steepest”
- Under a more realistic assumption, a better gradient could be derived
- **Leading to faster convergence**
- But higher computation

What is steepest descent?

- Given a fixed budget, what is the update that "reduces" the loss the most
- Euclidean space:

$$\underset{d}{\operatorname{argmin}}\; L(\theta + d) \quad \text{s.t. } \|d\| = c$$

- If the update "d" is of norm "c", what should be its direction?

Gradient descent is steepest in Euclidean

- d is steepest if it reduces the loss fastest

$$d^* = \underset{d}{\operatorname{argmin}}\ L(\theta + d) \quad \text{s.t. } \|d\| = c$$

- Lagrangian

$$\mathcal{L}(d, \lambda) = L(\theta + d) + \lambda(d^T d - c^2)$$

- Solve for critical point:

$$\nabla_d \mathcal{L} = \nabla_d L(\theta + d) + \lambda d = 0$$

$$\nabla_\lambda \mathcal{L} = d^T d - c^2 = 0$$

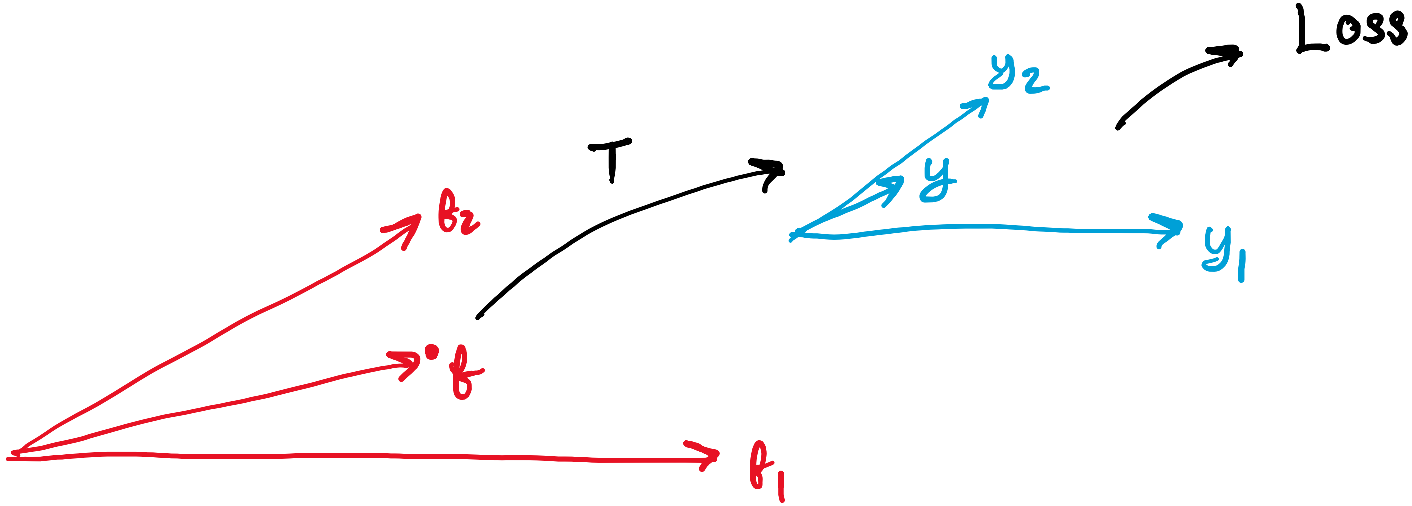
Gradient descent is steepest in Euclidean

Solve for critical point:

$$\nabla_d \mathcal{L} = \nabla_d L(\theta + d) + \lambda d = 0$$

$$\nabla_\lambda \mathcal{L} = d^T d - c^2 = 0$$

If we have another space above



- What is now the steepest descent wrt. the space above?

Norm of the new space

$\|\theta\| = \langle \theta, \theta \rangle = \theta^T \theta$

$T : \theta \to y \qquad y = T\theta \qquad$ Linear map

$\|y\| = \langle y, y \rangle = y^T y$

$= (T\theta)^T (T\theta)$

$= \theta^T T^T T \theta \neq \theta^T \theta$

Norm of the new space

Steepest is "subjective" because "norm" is subjective

$$d^* = \underset{d}{\operatorname{argmin}}\ L(\theta + d) \quad \text{s.t. } ||d|| = c$$

different meaning

$d \neq -\frac{\nabla L}{\lambda}$

Is there a more “natural” space than parameter space?

- Policy is a probability function
- It is more natural to think in **“space of probability functions”**
- What is the steepest descent in the probability function space?

Steepest in prob. fn. space

- Define the “distance” in the function space
- KL Divergence comes into mind:

$$\mathrm{KL}(P\|Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$$

- Steepest descent can get from solving:

$$d^* = \underset{d}{\mathrm{argmax}}\ J(\theta + d) \quad \text{s.t. } \mathrm{KL}(\theta\|\theta + d) = c$$

Connection

- Limited trust policy gradient
- Policy improvement guarantee
- Steepest descent on probability function space (Natural gradient)

They are doing the same thing

Related works

- Natural policy gradient
- Trust region policy optimization (TRPO)
  - We present here a “mini” version
- Proximal policy optimization (PPO)
  - An approximation of TRPO
  - Works well and easy to implement

More on policy gradient

Action dependent baseline

- **PG** has high variance because it uses “indirect gradient”

$$\nabla J(\theta) = \mathbb{E}_{s,a}\left[Q^{\pi}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)\right]$$

- **DPG** has lower variance because it can “backprop”

$$\nabla J(\theta) = \sum_{s} d^{\pi}(s)\nabla_{a}Q_{\phi}(s,a)|_{a=\pi(s)}\nabla_{\theta}\pi_{\theta}(s)$$

Action dependent baseline

- **Can we combine the two?**
- Q-Prop

$$\nabla J(\theta) = \mathbb{E}_{s,a}\left[(Q^{\pi}(s,a) - \overline{Q}_{\theta}(s,a))\nabla_{\theta}\log\pi_{\theta}(a|s)\right]$$
$$+ \mathbb{E}_{s}\left[\nabla_{a}Q_{\phi}(s,a)|_{a=u_{\theta}(s)}\nabla_{\theta}u_{\theta}(s)\right]$$

$$u_{\theta}(s) = \sum_{a}\pi_{\theta}(a|s)a$$

- Taylor expansion (first order)

$$\overline{Q}_{\phi}(s,a) = Q_{\phi}(s,u_{\theta}(s)) + \nabla_{a}Q_{\phi}(s,a)|_{a=u_{\theta}(s)}(a - u_{\theta}(s))$$

Gu, Shixiang, Timothy Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. 2016. "Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1611.02247.

Policy gradient from minimizing KL

- If we look at Q as "unnormalized" policy
  - A little bit sharper of Q is exp(Q)
  - This is our target policy
- We could use a KL:

$$\pi = \underset{\pi \in \Pi}{\operatorname{argmin}} \ \mathrm{D}_{KL} \left(\pi(\cdot|s) \middle\| \frac{\exp(Q(\cdot, s))}{Z} \right)$$

- Minimizing KL is an optimization task

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1801.01290.

Policy gradient from minimizing KL

- KL policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_s \left[\nabla_\theta \mathrm{D}_{KL} \left(\pi(\cdot|s) \middle\| \frac{\exp(Q(\cdot, s))}{Z} \right) \right]$$

- Z is a constant, ignored
- Policy improve to Q
- Policy eval: Q gets even sharper
- Repeat

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1801.01290.