Reinforcement learning

Part 9

Trust Region Policy Optimization

Let τ be a trajectory $(s_0, a_0, s_1, a_1, ...)$

Recall:

$$J(\pi) = E_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t r(s_t)\right]$$ - **goodness** of the π, π depends on θ.

Policy gradient theorem:

$$\nabla_\theta J(\pi) \approx E_{(s,a)\sim\pi}\left[Q^\pi(s,a)\nabla_\theta log\,\pi(a|s)\right]$$

Let τ be a trajectory $(s_0, a_0, s_1, a_1, ...)$

Recall:

$$J(\pi) = E_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r(s_t)\right]$$

- **goodness** of the π, π depends on θ.

Policy gradient theorem:

$$\nabla_\theta J(\pi) \approx E_{(s,a)\sim\pi}\left[Q^\pi(s,a)\nabla_\theta log\, \pi(a|s)\right]$$

Problem: Value of learning rate doesn't guarantee degree of policy change.

Let's use smarter optimization method.

Optimization

Suppose we want to optimize $\mathrm{F}(\theta)$ by θ in local neighbourhood

We approximate F by $\hat{F}(\theta) \approx F(\theta_0) + \nabla F(\theta)^T(\theta - \theta_0)$

Minimizing $\hat{F}(\theta)$ is the same as minimizing $\nabla F(\theta)^T(d), d = \theta - \theta_0$

We want to find d that

1) minimize $\nabla F(\theta)^T(d)$

2) $\boxed{\mathrm{d}^T d} < \epsilon$

Distance

Using Lagrange multipliers, we find that $\mathrm{d}_{opt} \propto -\nabla F(\theta)$

Optimization

Now, let's measure distance using non-identical matrix K:

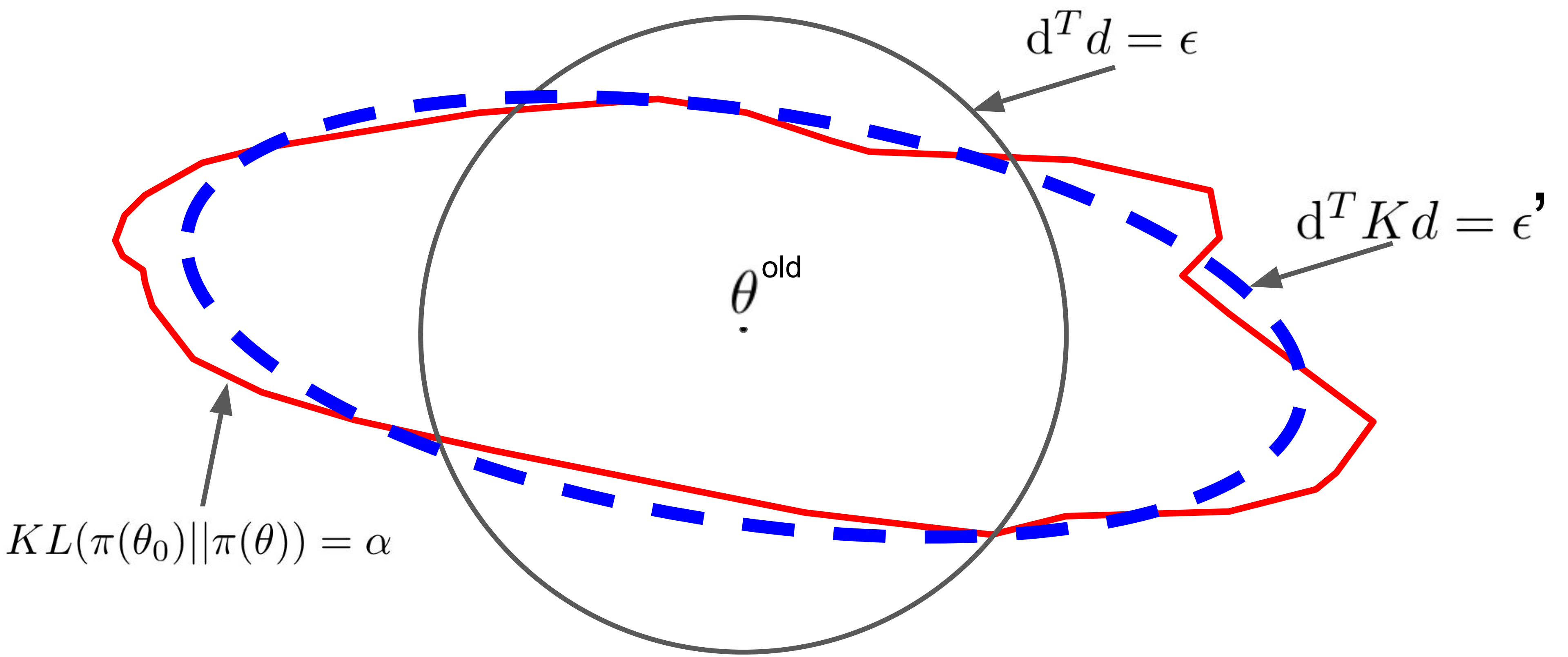
We want to find d that

1) minimize $\nabla F(\theta)^T(d)$

2) $\underbrace{\boxed{\mathrm{d}^T K d}}_{\text{Distance}} < \epsilon$

Using Lagrange multipliers, we find that $\mathrm{d}_{opt} \propto -K^{-1}\nabla F(\theta)$

Space of parameters



Natural Policy Gradient

Suppose:

$$KL(\pi(\theta_0)||\pi(\theta)) \approx 0.5{*}(\theta{-}\theta_0)^T K(\theta{-}\theta_0) = 0.5{*}d^T K d, K = \nabla_\theta^2 KL(\pi(\theta_0)||\pi(\theta))$$

Solve constrained equation: find vector d that

1) Minimize $\nabla J^T d$

2) $d^T K d < \epsilon$

Solution: $\mathrm{d}_{opt} \propto K^{-1} \nabla J(\theta)$

New update rule: $\theta_{t+1} = \theta_t - \alpha K^{-1} \nabla J(\theta)$

Natural Policy Gradient

Suppose:

$$KL(\pi(\theta_0)||\pi(\theta)) \approx 0.5*(\theta-\theta_0)^T K(\theta-\theta_0) = 0.5*d^T Kd, K = \nabla_\theta^2 KL(\pi(\theta_0)||\pi(\theta))$$

Solve constrained equation: find vector d that

1) Minimize $\nabla J^T d$

2) $d^T K d < \epsilon$

Solution: $\mathrm{d}_{opt} \propto K^{-1}\nabla J(\theta)$

New update rule: $\theta_{t+1} = \theta_t - \alpha K^{-1}\nabla J(\theta)$

Problems: Value of learning rate still doesn’t guarantee degree of policy change.
It may be too hard to compute inverse of K.

If we want to find $K^{-1}\nabla J(\theta)$ we may solve $Kx = \nabla J(\theta)$

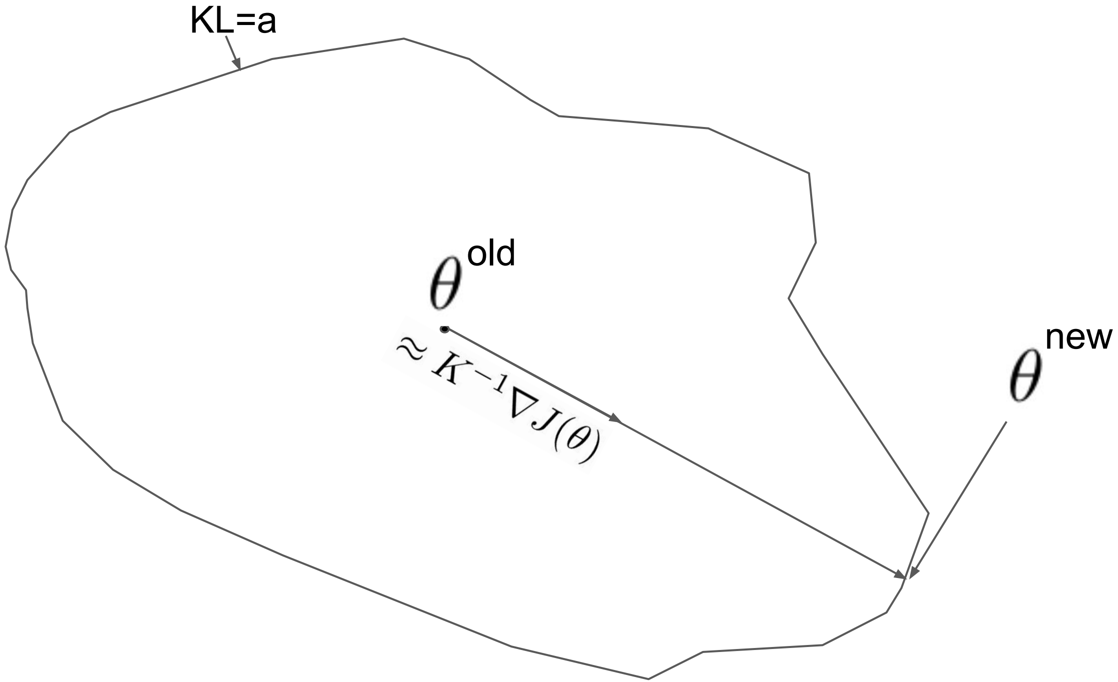
Matrix K is positive-definite so we can use **conjugate gradients**

Number of iterations k allows us to trade-off between precision and time.

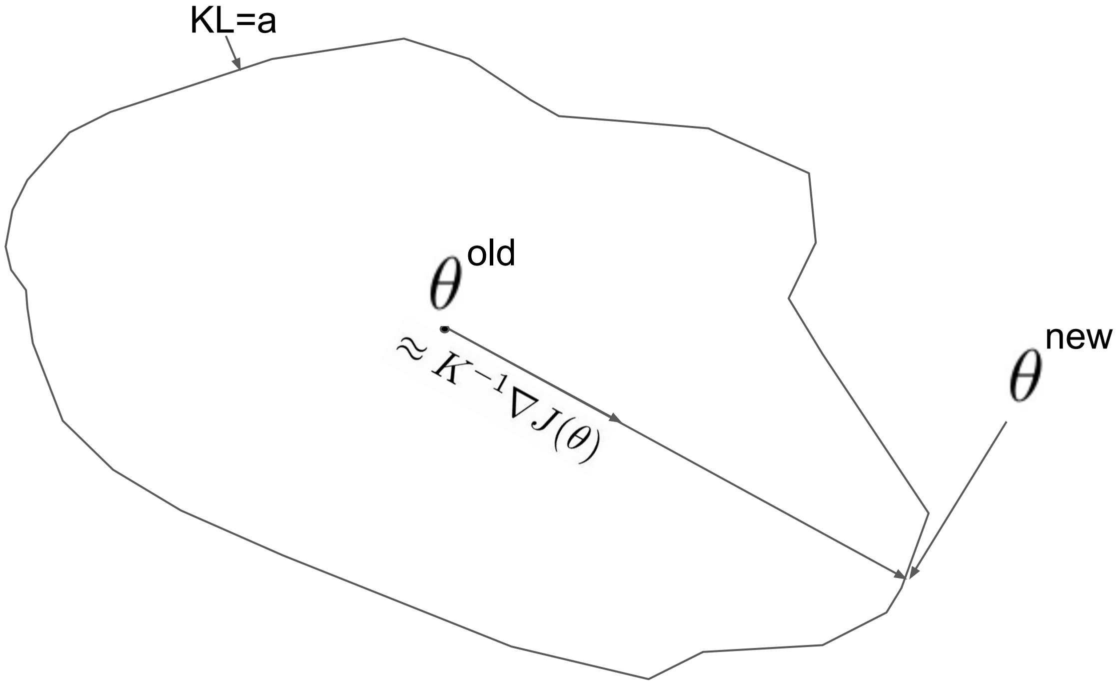
As a result: $\Delta\theta \approx \alpha K^{-1}\nabla J(\theta)$

Last problem: Value of learning rate still doesn't guarantee degree of policy change.

Let’s do linear search!

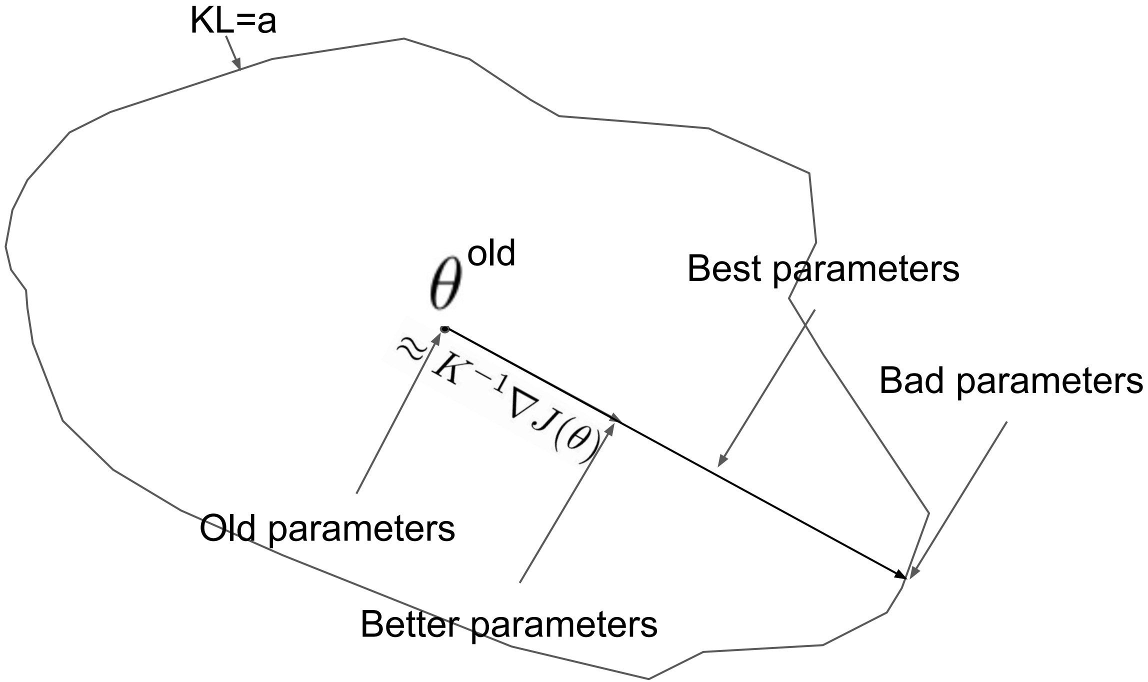


Let’s do linear search!

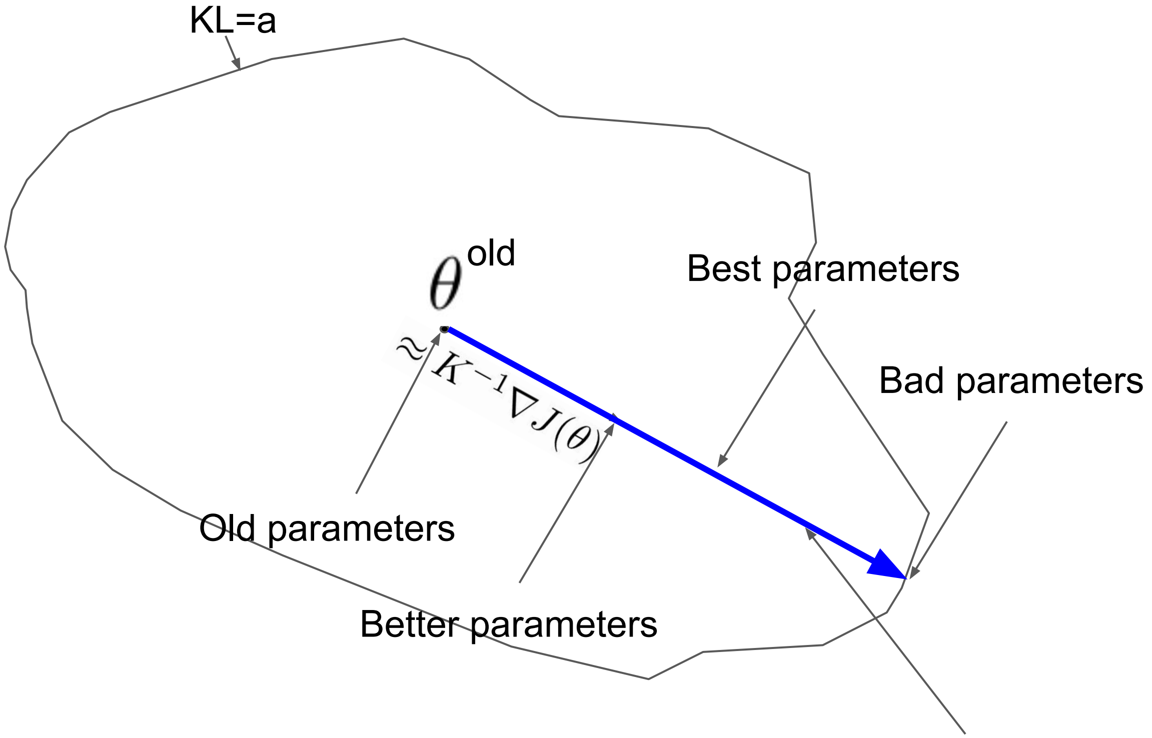


Problem?

Imagine this situation:



Imagine this situation:



We want to compute loss function here!

Let's define ρ_π as

$$\rho_\pi(s) = p(s_0 = s) + \gamma p(s_1 = s|\pi) + \gamma^2 p(s_2 = s|\pi) + \ldots$$

Suppose we have these uniformly distributed trajectories that may be generated by policy π

$$s^0 \rightarrow s^1 \rightarrow s^2$$

$$s^1 \rightarrow s^2$$

$$s^0 \rightarrow s^2$$

$$s^2$$

Suppose $\gamma = 0.8$

So $\rho_\pi(s^2) = 1/4 + \gamma 1/2 + \gamma^2 1/4 = 0.81$

Recall: $J(\pi) = E_{\tau \sim \pi}\Big[\sum_{t=0}^{T} \gamma^t r(s_t)\Big]$

It can be proven that $J(\tilde{\pi}) = J(\pi) + E_{\tau \sim \tilde{\pi}}\Big[\sum_{t=0}^{T} \gamma^t A_\pi(s_t, a_t)\Big]$

Let's rewrite it this way $J(\tilde{\pi}) = J(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$

Trust Region trick:

If $E_s\Big[KL(\pi \,||\, \tilde{\pi})\Big]$ is small,

$$J(\tilde{\pi}) \approx J(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

Then:

$$J(\tilde{\pi}) \approx J(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a) =$$

$$= J(\pi) + \sum_s \rho_\pi(s) \sum_a \pi(a|s) * \frac{\tilde{\pi}(a|s)}{\pi(a|s)} * A_\pi(s,a) =$$

$$= J(\pi) + \boxed{E_{(s,a)\sim\pi}[\frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s,a)]}$$

Can be computed at every point!

If $\pi = \tilde{\pi}$ (Let $\tilde{\pi}$ be a function of θ')

$$\nabla_{\theta'} E_{(s,a)\sim\pi}[\frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s,a)] = E_{(s,a)\sim\pi}[\nabla_{\theta'} log\, \tilde{\pi}(a|s) A_\pi(s,a)]$$

Trust Region Policy Optimization

1) Sample state-action pairs from on-policy distribution

2) Compute $g = \nabla_{\theta'} \hat{J}(\tilde{\pi})$ = $\nabla_{\theta'} \frac{1}{N} \sum_{i=0}^{N} \frac{\tilde{\pi}(s_i, a_i)}{\pi(s_i, a_i)} A_\pi(s_i, a_i)$

$$K = \nabla^2_{\theta'} \frac{1}{N} \sum_{i=0}^{N} KL(\pi(s_i) \mid\mid \tilde{\pi}(s_i))$$

3) Find $\hat{d} = -1 * ConjGrad(Kx = g)$

4) Do linear search in direction of $\hat{d}$, constraint $\frac{1}{N} \sum_{i=0}^{N} KL(\pi(s_i) \mid\mid \tilde{\pi}(s_i)) < \alpha$

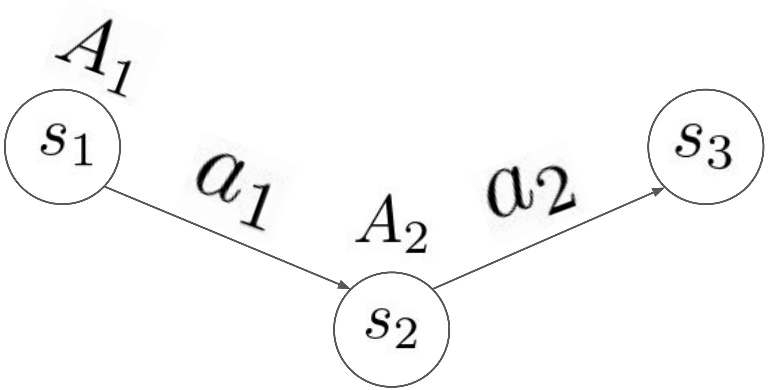
and simultaneously check value of $\frac{1}{N} \sum_{i=0}^{N} \frac{\tilde{\pi}(s_i, a_i)}{\pi(s_i, a_i)} A_\pi(s_i, a_i)$

Sampling

Single path (naive approach)

Sample **(state, action, return)** from on-policy distribution

$$\hat{J}(\tilde{\pi}) = \frac{1}{N}\sum_{i=0}^{N}\frac{\tilde{\pi}(s_i,a_i)}{\pi(s_i,a_i)}A_\pi(s_i,a_i)$$

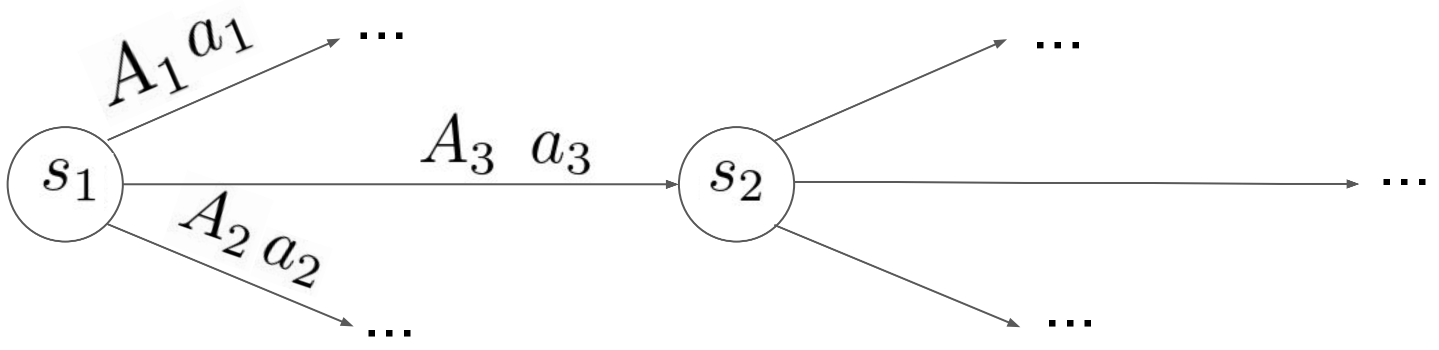


Sampling

Vine (works only if we may use checkpoints)

Sample **(state, returns for all a)** from on-policy distribution

$$\hat{J}(\tilde{\pi}) = \frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{N_a}\frac{\tilde{\pi}(s_i,a_j)}{\pi(s_i,a_j)}A_\pi(s_i,a_j)$$



TRPO

Advantages

- Very stable training
- Good result

Disadvantages

- Cheap sampling is necessary
- Not easy to implement

Thank you for your attention!

Questions?