



Aalto University
School of Electrical
Engineering

ELEC-E8125 Reinforcement Learning

Policy gradient

Joni Pajarinen

3.10.2023

Today

- Direct policy learning via policy gradient

Learning goals

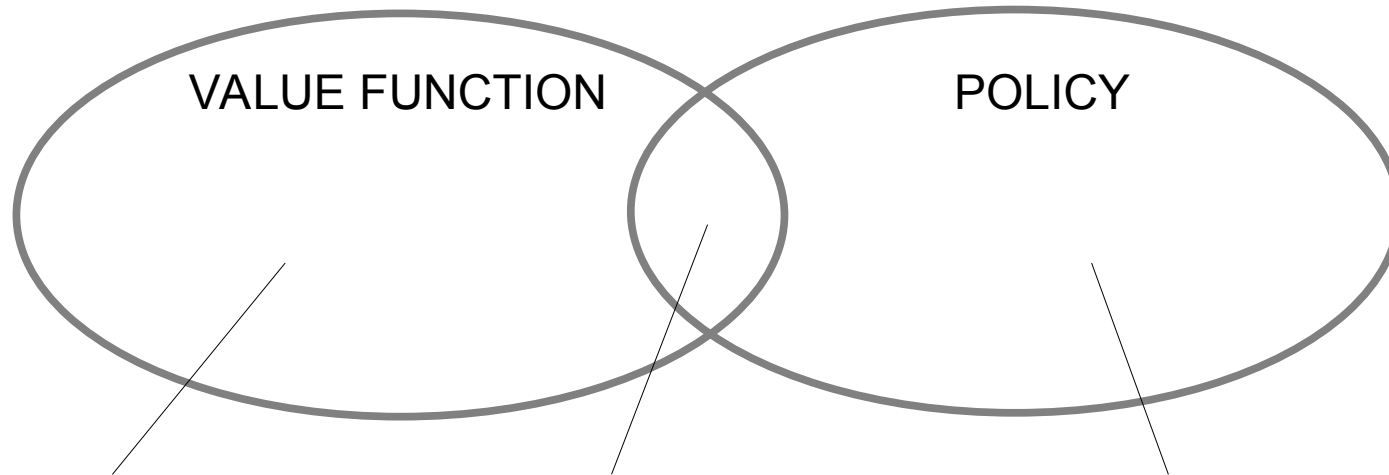
- Understand basis and limitations of policy gradient approaches

Motivation

<https://www.youtube.com/watch?v=xyJAvghtqIM>

- Even with value function approximation, large state spaces can be problematic
- Learning parametric policies $\pi(a|s, \theta)$ directly without learning value functions sometimes easier
 - Actions can be continuous
- Exploration or adversarial situations may benefit from stochastic policies

Value-based vs policy-based RL



Value-based

- Learned value function
- Implicit policy

Actor-critic

- Learned value function
- Learned policy

Policy-based

- No value function
- Learned policy

- Can learn stochastic policies
- Usually locally optimal

Stochastic policies

- Discrete actions: Soft-max policy

$$\pi_{\theta}(a_t | s_t) = 1/Z e^{\theta^T \varphi(s_t, a_t)}$$

Probability proportional to exponential linear combination of features.

Normalization constant

$$Z = \sum_a e^{\theta^T \varphi(s_t, a_t)}$$

- Continuous actions: Gaussian policy

$$\pi_{\theta}(a_t | s_t) \sim N(\theta^T \varphi(s_t), \sigma^2)$$

Mean is linear combination of features.

Can also be understood as linear policy plus exploration uncertainty

$$\pi_{\theta}(a_t | s_t) = \theta^T \varphi(s_t) + \epsilon \quad \epsilon \sim N(0, \sigma^2)$$

Supervised policy learning – behavioral cloning

- Assume examples of policy are given in form of (s,a) pairs
- How to fit a stochastic policy to these?

$$\pi_{\theta}(a_t | s_t) \sim N(\boldsymbol{\theta}^T \boldsymbol{\varphi}(s_t), \sigma^2) \quad \leftarrow \text{Example}$$

Supervised policy learning – behavioral cloning

- Assume examples of policy are given in form of (s,a) pairs. Assume independent examples
- How to fit a stochastic policy to these?

$$\pi_{\theta}(a_t | s_t) \sim N(\boldsymbol{\theta}^T \boldsymbol{\varphi}(s_t), \sigma^2) \quad \leftarrow \text{Example}$$

- Maximum likelihood parameter estimation
 - Here: maximize probability of actions given states and parameters

$$P(A|S; \theta) = \prod_t \pi_{\theta}(a_t | s_t)$$

Example: Maximum likelihood estimation

- Maximize log-likelihood

$$P(A|S; \theta) = \prod_t \pi_{\theta}(a_t | s_t)$$

$$N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(a-\mu)^2}{2\sigma^2}}$$

Example: Maximum likelihood estimation

- Maximize log-likelihood

$$P(A|S; \theta) = \prod_t \pi_\theta(a_t|s_t) \qquad N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(a-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned} \log P(A|S; \theta) &= \sum_t \log \pi_\theta(a_t|s_t) \\ \nabla \log P(A|S; \theta) &= \sum_t \nabla \log \pi_\theta(a_t|s_t) \end{aligned}$$

What is a good policy?

- How to measure policy quality?

$$R(\boldsymbol{\theta}) = E \left[\sum_{t=0}^T \gamma^t r_t \right]$$

Policy gradient

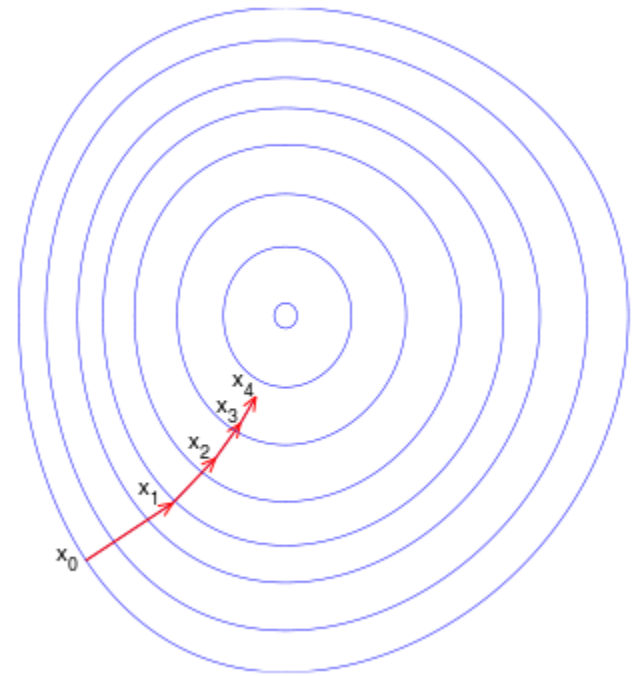
- Use gradient ascent on $R(\theta)$
- Update policy parameters by

$$\theta_{m+1} = \theta_m + \alpha_m \nabla_{\theta} R|_{\theta=\theta_m}$$

- How to calculate gradient?

$$R(\theta) = E \left[\sum_{t=0}^T \gamma^t r_t \right]$$

Depends on θ .



$$\sum_{m=0}^{\infty} \alpha_m = \infty \quad \sum_{m=0}^{\infty} \alpha_m^2 < \infty$$

Required for convergence to a local minimum.

Finite difference gradient estimation

- What is gradient?
 - Vector of partial derivatives
- How to estimate derivative?
 - Finite difference: $f'(x) \approx \frac{f(x+dx) - f(x)}{dx}$
- For policy gradient:
 - Generate variation $\Delta \theta_i$
 - Estimate experimentally $R(\theta + \Delta \theta_i) \approx \hat{R}_i = \sum_{t=0}^H \gamma^t r_t$
 - Compute gradient $\mathbf{g}_{FD} = (\Delta \Theta^T \Delta \Theta)^{-1} \Delta \Theta^T \Delta R$
 - Repeat until estimate converged

Not easy to choose.

$$\Delta R^T = [\hat{R}_1 - R(\theta), \dots, \hat{R}_I - R(\theta)]$$

Where does this come from?

$$\hat{R}_i \approx R(\theta) + \mathbf{g}_{FD}^T \Delta \theta_i$$

Likelihood-ratio approach

- Assume each trajectory τ is generated by a roll-out, thus

$$\tau \sim p_{\theta}(\tau) = p(\tau|\theta) \quad R(\tau) = \sum_{t=0}^H \gamma^t r_t$$

- Expected return can then be written

$$R(\theta) = E_{\tau}[R(\tau)] = \int p_{\theta}(\tau) R(\tau) d\tau$$

- Gradient is thus

$$\nabla_{\theta} R(\theta) = \int \nabla_{\theta} p_{\theta}(\tau) R(\tau) d\tau$$

$$= \int p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau) R(\tau) d\tau \quad \leftarrow \begin{array}{l} \text{Likelihood ratio "trick":} \\ \text{Substitute} \end{array}$$

- Why do that? $= E_{\tau}[\nabla_{\theta} \log p_{\theta}(\tau) R(\tau)] \quad \nabla_{\theta} p_{\theta}(\tau) = p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau)$

$$p_{\theta}(\tau) = p(s_0) \prod_{t=0}^H p(s_{t+1} | s_t, \mathbf{a}_t) \pi_{\theta}(\mathbf{a}_t | s_t)$$

Example differentiable policies

Normalization constant missing.

- Soft-max policy

$$\pi_{\theta}(a_t|s_t) \propto e^{\theta^T \varphi(s_t, a_t)}$$

Probability proportional to exponential linear combination of features.

- Log-policy (*score function*)

$$\nabla_{\theta} \log \pi_{\theta}(a_t|s_t) = \varphi(s_t, a_t) - E_{\pi_{\theta}}[\varphi(s_t, \cdot)]$$

- Gaussian policy

$$\pi_{\theta}(a_t|s_t) \sim N(\theta^T \varphi(s_t), \sigma^2)$$

Mean is linear combination of features.

- Log-policy

$$\nabla_{\theta} \log \pi_{\theta}(a_t|s_t) = \frac{(a_t - \theta^T \varphi(s_t)) \varphi(s_t)}{\sigma^2}$$

Can also be understood as linear policy plus exploration uncertainty

$$\pi_{\theta}(a_t|s_t) = \theta^T \varphi(s_t) + \epsilon \quad \epsilon \sim N(0, \sigma^2)$$

Example differentiable policies

Normalization constant missing.

- Discrete neural net policy

$$\pi_{\theta}(a_t | s_t) \propto e^{f_{\theta}(s_t, a_t)}$$

Probability proportional to exponential neural network output.

- Gaussian neural network policy

$$\pi_{\theta}(a_t | s_t) \sim N(f_{\theta}(s_t), \sigma^2)$$

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) = \frac{(a_t - f_{\theta}(s_t)) \nabla_{\theta} f_{\theta}(s_t)}{\sigma^2}$$

OK, now to applying the policy gradient:

$$\nabla_{\theta} R(\theta) = E_{\tau} [\nabla_{\theta} \log p_{\theta}(\tau) R(\tau)]$$

MC policy gradient – REINFORCE

- Episodic version shown here

- Approach:

- Perform J episodes ($i=1,2,3,\dots$)

- Estimate gradient $\mathbf{g}_{RE} = E_{\tau} \left[\left(\sum_{t=0}^H \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) \right) R(i) \right]$ Use empirical mean.
 $\approx \frac{1}{J} \sum_{i=1}^J \left[\left(\sum_{t=0}^H \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t^{[i]} | \mathbf{s}_t^{[i]}) \right) \left(\sum_t \gamma^t r_{t,i} \right) \right]$

Reward for trial i .



- Update policy and repeat with new trial(s) until convergence

- Due to stochastic policy no need to generate additional policy variations

Limitations so far

- High variance (uncertainty) in gradient estimates
- Slow convergence, hard to choose learning rate
 - Parametrization dependent gradient estimate
- On-policy method

Decreasing variance by adding baseline

- Constant baseline can be added to reduce *variance* of the gradient estimate

$$\begin{aligned}\nabla_{\theta} R(\theta) &= E_{\tau}[\nabla_{\theta} \log p_{\theta}(\tau)(R(\tau) - b)] \\ &= E_{\tau}[\nabla_{\theta} \log p_{\theta}(\tau) R(\tau)]\end{aligned}$$

- Does not cause bias because

$$\begin{aligned}E_{\tau}[\nabla_{\theta} \log p_{\theta}(\tau) b] &= \int p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau) b d\tau = \\ \int \nabla_{\theta} p_{\theta}(\tau) b d\tau &= b \nabla_{\theta} \int p_{\theta}(\tau) d\tau = b \nabla_{\theta} 1 = 0\end{aligned}$$

Episodic REINFORCE with optimal baseline

- Optimal baseline for episodic REINFORCE (minimize variance of estimator):

$$b_h = \frac{E_{\tau} \left[\left(\sum_{t=0}^H \nabla_{\theta_h} \log \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) \right)^2 R_{\tau} \right]}{E_{\tau} \left[\left(\sum_{t=0}^H \nabla_{\theta_h} \log \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) \right)^2 \right]}$$

In practice, approximate by empirical mean (average over trials).

- Approach:

- Perform J trials ($i=1,2,3,\dots$)

- For each gradient element h

Component-wise!

- Estimate optimal baseline b_h

- Estimate gradient

- Repeat until convergence

$$g_h = \frac{1}{J} \sum_{i=1}^J \left[\left(\sum_{t=0}^H \nabla_{\theta_h} \log \pi_{\theta}(\mathbf{a}_t^{[i]} | \mathbf{s}_t^{[i]}) \right) (R(i) - b_h^{[i]}) \right]$$

Policy gradient theorem (PGT)

- Observation: Future actions do not depend on past rewards. “don't take into account past rewards when evaluating the effect of an action” (causality, taking an action can only affect future rewards)

$$E\left[\nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) r_k\right] = 0 \quad \forall t > k$$

- PGT:
 - Reduces variance of estimate → Fewer samples needed on average

$$\mathbf{g}_{PGT} = E_{\tau} \left[\sum_{k=0}^H \left(\sum_{t=0}^k \nabla_{\theta_h} \log \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) \right) (\gamma^k r_k - b_k^h) \right]$$

Off-policy policy gradient

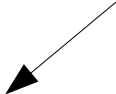
- What if we have samples from another policy, for example, from earlier time steps?

Optimize $E_{\tau \sim \pi_{\theta}(\tau)}[R(\tau)]$
using samples from $\pi'(\tau)$

- Use importance sampling!

$$\begin{aligned} E_{s \sim p(s)}[f(s)] &= \int p(s) f(s) ds \\ &= E_{s \sim q(s)} \left[\frac{p(s)}{q(s)} f(s) \right] \end{aligned}$$

Where does this
come from?



Off-policy policy gradient

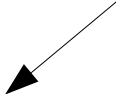
- What if we have samples from another policy (e.g. earlier time steps?)

Optimize $E_{\tau \sim \pi_{\theta}(\tau)} [R(\tau)]$
using samples from $\pi'(\tau)$

- Use importance sampling!

$$\begin{aligned} E_{s \sim p(s)} [f(s)] &= \int p(s) f(s) ds \\ &= E_{s \sim q(s)} \left[\frac{p(s)}{q(s)} f(s) \right] \end{aligned}$$

Where does this
come from?



$$E_{\tau \sim \pi'(\tau)} \left[\frac{\pi_{\theta}(\tau)}{\pi'(\tau)} R(\tau) \right]$$

Off-policy policy gradient

$$E_{\tau \sim \pi'(\tau)} \left[\frac{\pi_{\theta}(\tau)}{\pi'(\tau)} R(\tau) \right]$$

- We had earlier

$$p_{\theta}(\tau) = p(s_0) \prod_{t=0}^H p(s_{t+1} | s_t, a_t) \pi_{\theta}(a_t | s_t)$$

- Thus

$$\frac{\pi_{\theta}(\tau)}{\pi'(\tau)} = \frac{p(s_0) \prod_{t=0}^H p(s_{t+1} | s_t, a_t) \pi_{\theta}(a_t | s_t)}{p(s_0) \prod_{t=0}^H p(s_{t+1} | s_t, a_t) \pi'(a_t | s_t)} = \frac{\prod_{t=0}^H \pi_{\theta}(a_t | s_t)}{\prod_{t=0}^H \pi'(a_t | s_t)}$$

Off-policy policy gradient

- Now the gradient

$$\begin{aligned}\nabla_{\theta} E_{\tau \sim \pi'(\tau)} \left[\frac{\pi_{\theta}(\tau)}{\pi'(\tau)} R(\tau) \right] &= E_{\tau \sim \pi'(\tau)} \left[\frac{\nabla_{\theta} \pi_{\theta}(\tau)}{\pi'(\tau)} R(\tau) \right] \\ &= E_{\tau \sim \pi'(\tau)} \left[\frac{\pi_{\theta}(\tau)}{\pi'(\tau)} \nabla_{\theta} \log \pi_{\theta}(\tau) R(\tau) \right] \\ &= E_{\tau \sim \pi'(\tau)} \left[\left(\prod_t \frac{\pi_{\theta}(a_t | s_t)}{\pi'(a_t | s_t)} \right) \left(\sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right) \left(\sum_t \gamma^t r_t \right) \right]\end{aligned}$$

Compare to on-policy (REINFORCE)

$$\nabla_{\theta} E_{\tau \sim \pi_{\theta}(\tau)} [R(\tau)] = E_{\tau \sim \pi_{\theta}(\tau)} \left[\left(\sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \right) \left(\sum_t \gamma^t r_t \right) \right]$$

Summary

- Policy gradient methods can be used for stochastic policies and continuous action spaces
- Finite-difference approaches approximate gradient by policy adjustments
- Likelihood ratio-approaches calculate gradient through known policy
- Policy gradient often requires very many updates because of a noisy gradient and small update steps resulting in slow convergence

Next: Actor-critic approaches

- Can we combine policy learning with value-based methods?
- Readings
 - Sutton & Barto Ch 13.5