Aalto University
School of Electrical Engineering

ELEC-E8125 Reinforcement Learning
Policy gradient

Ville Kyrki

6.10.2020

Today

- Direct policy learning via policy gradient.

Learning goals

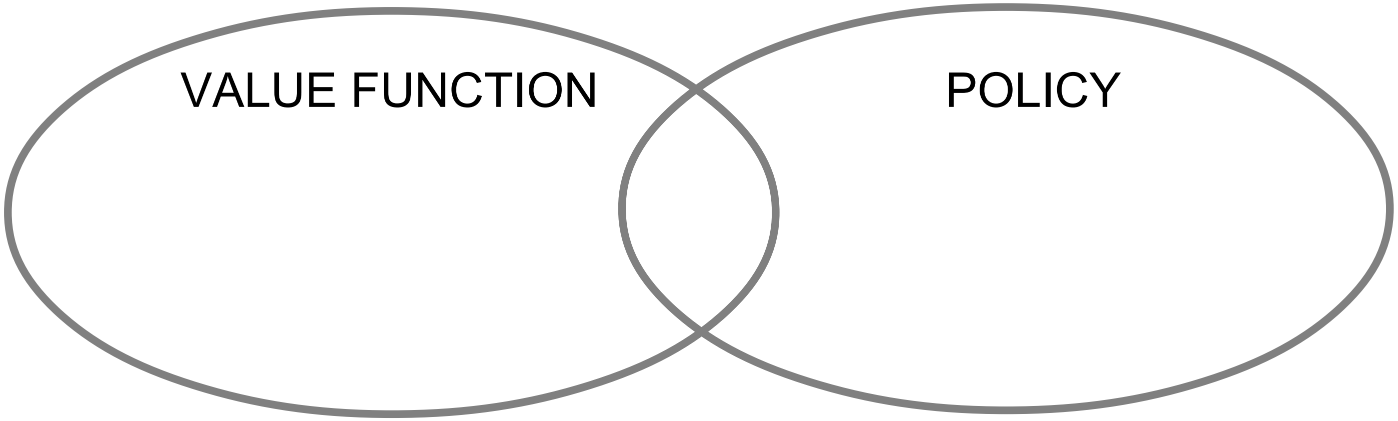
- Understand basis and limitations of policy gradient approaches.

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.
- Non-Markov (partially observable) or adversarial situations might 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

$$\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) = 1/Z\, e^{\theta^T \varphi(\boldsymbol{s}_t, \boldsymbol{a}_t)}$$

Probability portional to expontiated linear combination of features.

Normalization constant

$$Z = \sum_a e^{\theta^T \varphi(\boldsymbol{s}_t, \boldsymbol{a}_t)}$$

- Continuous actions: Gaussian policy

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

Mean is linear combination of features.

Can also be understood as linear policy plus exploration uncertainty

$$\pi_\theta(a_t|\boldsymbol{s}_t) = \theta^T \varphi(\boldsymbol{s}_t) + \epsilon \qquad \epsilon \sim N(0, \sigma^2)$$

Note: Policies include exploration!

But how to fit these?

Note: This is not RL!

Supervised policy learning – behavioral cloning

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

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

Example

Note: This is not RL!

Supervised policy learning – behavioral cloning

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

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

Example

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

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

How to proceed?

Example: Maximum likelihood estimation

- Maximize log-likelihood

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

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

Example: Maximum likelihood estimation

- Maximize log-likelihood

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

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

$$\log P(A|S;\theta)=\sum_t \log \pi_{\boldsymbol{\theta}}(a_t|\boldsymbol{s}_t)$$
$$\nabla \log P(A|S;\theta)=\sum_t \nabla \log \pi_{\boldsymbol{\theta}}(a_t|\boldsymbol{s}_t)$$

But we don't have examples
so we cannot use supervised learning!

What is a good policy?

- How to measure policy quality?

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

- More generally,

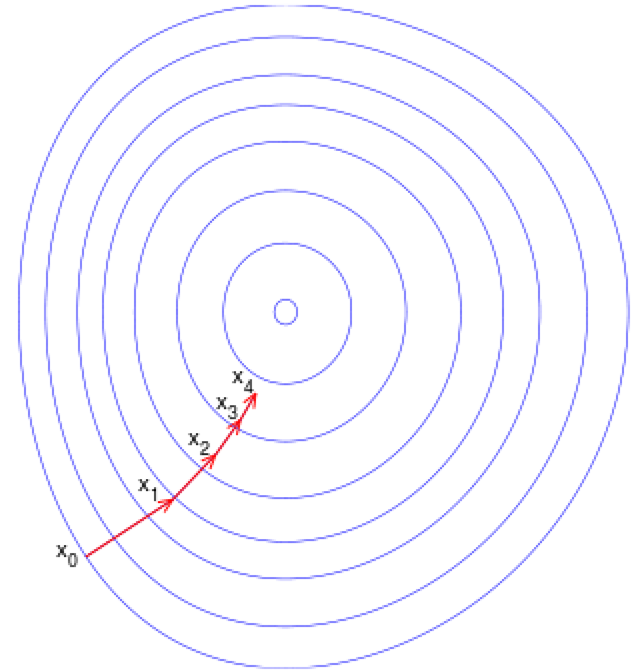
$$R(\boldsymbol{\theta}) = E\left[\sum_{t=0}^{T} c_t r_t\right]$$

Can also represent average reward per time step.

General time scaling factor

How to optimize parameters?

Policy gradient



- Use gradient ascent on *R(θ)*.

- Update policy parameters by

$$\boldsymbol{\theta}_{m+1} = \boldsymbol{\theta}_m + \alpha_m \nabla_{\boldsymbol{\theta}} R|_{\boldsymbol{\theta}=\boldsymbol{\theta}_m}$$

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

Guarantees convergence to local minimum.

- How to calculate gradient?

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

Depends on $\boldsymbol{\theta}$.

How to estimate gradient from data (if we have a chance to try different policies)?

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 $R(\theta+\Delta\theta_i) \approx \hat{R}_i = \sum_{t=0}^{H} c_t r_t$
  - Compute gradient $\left[\boldsymbol{g}_{FD}^T, R_{ref}\right]^T = (\Delta\boldsymbol{\Theta}^T \Delta\boldsymbol{\Theta})^{-1} \Delta\boldsymbol{\Theta}^T \hat{\boldsymbol{R}}$
  - Repeat until estimate converged

Not easy to choose.

$$\Delta\boldsymbol{\Theta}^T = \begin{bmatrix} \Delta\theta_1, \ldots, \Delta\theta_I \\ 1, \ldots, 1 \end{bmatrix}$$

$$\hat{\boldsymbol{R}}^T = [\hat{R}_1, \ldots, \hat{R}_I]$$

Where does this come from?

$$\hat{R}_i \approx R_{ref} + \boldsymbol{g}^T \Delta\theta_i$$

Likelihood-ratio approach

- Assume trajectories tau are generated by roll-outs, thus

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

- Expected return can then be written

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

- Gradient is thus

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

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

Likelihood ratio “trick”: Substitute

$$\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau)$$

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

- Why do that?

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = p(\boldsymbol{s}_0) \prod_{t=0}^{H} p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$$

Try substitution for log-gradient! $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = \sum_{t=0}^{H} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$

We know this!

Example differentiable policies

Normalization constant missing.

- Soft-max policy

$$\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) \propto e^{\theta^T \varphi(\boldsymbol{s}_t, \boldsymbol{a}_t)}$$

Probability proportional to exponentiated linear combination of features.

  - Log-policy (*score function*)

$$\nabla_\theta \log \pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) = \varphi(\boldsymbol{s}_t, \boldsymbol{a}_t) - E_{\pi_\theta}[\varphi(\boldsymbol{s}_t, \cdot)]$$

- Gaussian policy

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

Mean is linear combination of features.

  - Log-policy

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

Can also be understood as linear policy plus exploration uncertainty

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

Example differentiable policies

Normalization constant missing.

- Discrete neural net policy

$$\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t) \propto e^{f_\theta(\boldsymbol{s}_t,\, \boldsymbol{a}_t)}$$

Probability proportional to exponentiated neural network output.

- Gaussian neural network policy

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

$$\nabla_\theta \log \pi_\theta(a_t|\boldsymbol{s}_t) = \frac{(u_t - f_\theta(\boldsymbol{s}_t))\, \nabla_\theta f_\theta(\boldsymbol{s}_t)}{\sigma^2}$$

OK, now to applying the policy gradient:

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

MC policy gradient – REINFORCE

- Episodic version shown here.

Reward for trial *i*.

- Approach:
  - Perform episode *J (=1,2,3,...).*
  - Estimate gradient $g_{RE} = E_{\tau}\left[\left(\sum_{t=0}^{H} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t)\right) R(i)\right]$

$$\approx \frac{1}{J}\sum_{i=1}^{J}\left[\left(\sum_{t=0}^{H} \nabla_{\theta} \log \pi_{\theta}(a_t^{[i]}|s_t^{[i]})\right)\left(\sum_t r_{t,i}\right)\right]$$

Use empirical mean.

  - Update policy and repeat with new trial(s) until convergence.
- No need to generate policy variations because of stochastic policy.

Limitations so far

- High variance (uncertainty) in gradient estimate because of stochastic policy.
- 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 gradient estimate.

$$\nabla_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) = E_{\boldsymbol{\tau}}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau})(R(\boldsymbol{\tau}) - b)\right]$$
$$= E_{\boldsymbol{\tau}}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) R(\boldsymbol{\tau})\right]$$

- Does not cause bias because

$$E_{\boldsymbol{\tau}}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) b\right] = \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) b \, d\boldsymbol{\tau} =$$
$$\int \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) b \, d\boldsymbol{\tau} = b \nabla_{\boldsymbol{\theta}} \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) d\boldsymbol{\tau} = b \nabla_{\boldsymbol{\theta}} 1 = 0$$

Intuition:
Modifying rewards by a constant
does not change optimal policy.

Episodic REINFORCE with optimal baseline

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

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

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

- Approach:
  - Perform trial *J (=1,2,3,...).*
  - For each gradient element *h*
    - Estimate optimal baseline b_h
    - Estimate gradient

Component-wise!

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

  - Repeat until convergence.

Even with optimal baseline, variance can be an issue.

Policy gradient theorem

- Observation: Future actions do not depend on past rewards.

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

"don't take into account past rewards when evaluating the effect of an action" (causality, taking an action can only affect future rewards)

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

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

Note: If only rewards at final time step, this is equivalent to REINFORCE.

Off-policy policy gradient

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

  Optimize $E_{\tau \sim \pi_\theta(\tau)}[R(\tau)]$

  using samples from $\pi'(\tau)$ exploration policy

- Use importance sampling!

Where does this come from?

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

Off-policy policy gradient

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

  Optimize $E_{\tau \sim \pi_\theta(\tau)}[R(\tau)]$

  using samples from $\pi'(\tau)$ ← exploration policy

- Use importance sampling!

$$E_{s \sim p(s)}[f(s)] = \int p(s) f(s)\, ds$$

$$= E_{s \sim q(s)}\left[\frac{p(s)}{q(s)} f(s)\right]$$

Where does this come from?

Weight samples by their relative probability

Aalto University
School of Electrical
Engineering

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

Off-policy policy gradient

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

- We had earlier

$$p_\theta(\boldsymbol{\tau})=p(\boldsymbol{s}_0)\prod_{t=0}^{H}p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$$

- Thus

$$\frac{\pi_\theta(\tau)}{\pi'(\tau)}=\frac{p(\boldsymbol{s}_0)\prod_{t=0}^{H}p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{p(\boldsymbol{s}_0)\prod_{t=0}^{H}p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\pi'(\boldsymbol{a}_t|\boldsymbol{s}_t)}=\frac{\prod_{t=0}^{H}\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\prod_{t=0}^{H}\pi'(\boldsymbol{a}_t|\boldsymbol{s}_t)}$$

Off-policy policy gradient

- Now the gradient

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

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

$$=\boxed{E_{\tau\sim\pi'(\tau)}\left[\left(\prod_t\frac{\pi_\theta(\tau)}{\pi'(\tau)}\right)\left(\sum_t\nabla_\theta\log\pi_\theta(a_t|s_t)\right)\left(\sum_t r_t\right)\right]}$$

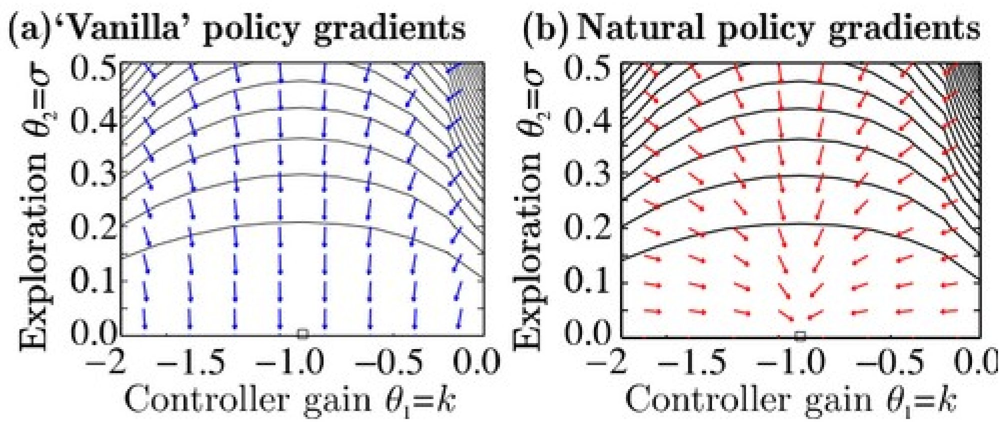
Compare to on-policy (REINFORCE)

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

Will be used later!

Gradient vs natural gradient

- Gradient depends on parametrization.
- Natural gradient parametrization independent.



$$\nabla_\theta^{NG} \pi_\theta(a|\boldsymbol{s}) = \boldsymbol{F}_\theta^{-1} \nabla_\theta \pi_\theta(a|\boldsymbol{s})$$

Intuition: Divide gradient update by second derivative.

Normalizes parameter influence.

- Fisher information matrix

$$\boldsymbol{F}_\theta = E\left[\nabla_\theta \log \pi_\theta(a|\boldsymbol{s}) \nabla_\theta \log \pi_\theta(a|\boldsymbol{s})^T\right]$$

Potentially improves convergence significantly,
in practice sample-based approximation less useful.

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 noisy gradient and small update steps → slow convergence.

Next: Actor-critic approaches

- Can we combine policy learning with value-based methods?

- Readings
  - Sutton&Barto Ch 13.5