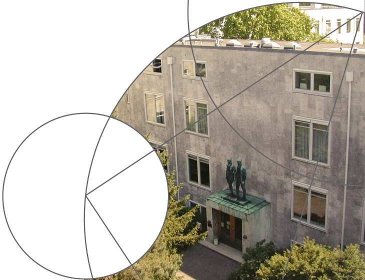
Faculty of Science

Policy Gradient Methods

Reinforcement Learning

Christian Igel
Department of Computer Science



igel@diku.dk @christian_igel

Outline

1. Background
   - Monte Carlo methods
   - Importance weighting
   - Control variates
   - "Log-derivative trick"
2. Policy gradient methods
3. Policy gradient theorem and REINFORCE
   - REINFORCE algorithm
   - Proof undiscounted policy gradient considering start-state
   - Proof discounted policy gradient considering start-state
   - Proof policy gradient average reward*
4. Policy gradients in the wild

Outline

1. Background
   - Monte Carlo methods
   - Importance weighting
   - Control variates
   - "Log-derivative trick"
2. Policy gradient methods
3. Policy gradient theorem and REINFORCE
   - REINFORCE algorithm
   - Proof undiscounted policy gradient considering start-state
   - Proof discounted policy gradient considering start-state
   - Proof policy gradient average reward*
4. Policy gradients in the wild

Warmup: Monte Carlo methods

Crude Monte Carlo methods approximate an integral

$$\mathbb{E}_{p(z)}[f(z)] = \int f(z)p(z)\mathsf{d}z$$

by

$$\frac{1}{n}\sum_{i=1}^{n} f(z_i) \xrightarrow{n\to\infty} \mathbb{E}_{p(z)}[f(z)] \ ,$$

where

$$z_i \sim p(z)$$

for $i = 1, \ldots, n$.

Warmup: Importance sampling

For two distributions p and q with $p(z) > 0 \Rightarrow q(z) > 0$ we have

$$\mathbb{E}_{p(z)}[f(z)] = \int \frac{p(z)}{q(z)} f(z) q(z) \mathsf{d}z = \mathbb{E}_{q(z)}\left[\frac{p(z)}{q(z)} f(z)\right]$$

and thus

$$\frac{1}{n}\sum_{i=1}^{n} \frac{p(z_i)}{q(z_i)} f(z_i) \xrightarrow{n\to\infty} \mathbb{E}_{p(z)}[f(z)] \ ,$$

where $z_i \sim q(z)$ for $i = 1, \dots, n$, that is, we approximate the expectation over p using samples from q.

Warmup: Control variates

Assume we want to estimate $\mathbb{E}[f]$ via Monte Carlo and are worried about the variance of our estimator.

We introduce a helper function (control variate) ϕ correlated with f for which we know $\overline{\phi} = \mathbb{E}[\phi]$ and write

$$\mathbb{E}[f] = \mathbb{E}[f - \phi] + \mathbb{E}[\phi] = \mathbb{E}[f - \phi] + \overline{\phi} = \mathbb{E}[f - \phi + \overline{\phi}] \ .$$

We have

$$\mathrm{Var}[f - \phi + \overline{\phi}] = \mathrm{Var}[f - \phi] = \mathrm{Var}[f] - 2\,\mathrm{Cov}(f, \phi) + \mathrm{Var}[\phi] \ .$$

Thus, the control variate reduces the variance in the sense that
$\mathrm{Var}(f - \phi] < \mathrm{Var}[f]$ if $2\,\mathrm{Cov}(f, \phi) > \mathrm{Var}[\phi]$.
Analogously, $\mathrm{Var}[f + \phi] < \mathrm{Var}[f]$ if $2\,\mathrm{Cov}(f, \phi) < \mathrm{Var}[\phi]$.

Warmup: "Log-derivative trick"

We have

$$\nabla_\theta \log p(z;\theta) = \frac{\nabla_\theta p(z;\theta)}{p(z;\theta)} \Rightarrow \nabla_\theta p(z;\theta) = p(z;\theta)\nabla_\theta \log p(z;\theta)$$

which implies

$$\nabla_\theta \mathbb{E}_{p(z;\theta)}[f(z)] = \int f(z)\nabla_\theta p(z;\theta)\mathsf{d}z = \mathbb{E}_{p(z;\theta)}[f(z)\nabla_\theta \log p(z;\theta)] \ ,$$

where the RHS lends itself to Monte Carlo estimation via

$$\frac{1}{n}\sum_{i=1}^{n} f(z_i)\nabla_\theta \log p(z_i;\theta)$$

with $z_i \sim p(z)$ for $i = 1, \dots, n$. $\nabla_\theta \log p(z;\theta)$ is called *score function*.

Outline

Notation

- Discrete time t, states S, actions A
- State-action-reward sequence $s_0, a_0, r_1, \dots,$ with first time arriving in a terminal state in episodic tasks denoted by T (and for $t > T$: $r_t = 0$ and $s_t = s_T$), start state distribution p_{start}
- $P^a_{ss'} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}$ and $R^a_{ss'} = \mathbb{E}\{r_{t+1} \mid s_t = s, s_{t+1} = s', a_t = a\}$
- $R^a_s = \mathbb{E}\{r_{t+1} \mid s_t = s, a_t = a\} = \sum_{s'} P^a_{ss'} R^a_{ss'}$
- $\pi(s, a; \boldsymbol{\theta}) = \pi_{\boldsymbol{\theta}}(s, a) = \pi(s, a) = \Pr\{a_t = a \mid s_t = s, \boldsymbol{\theta}\}$
- $\Pr\{s \xrightarrow{k} x \mid \pi\}$: probability of going from state s to state x in k steps under policy π ($\Pr\{s \xrightarrow{0} s \mid \pi\} = 1$ and $\Pr\{s \xrightarrow{0} s' \mid \pi\} = 0$ for $s \neq s'$)

Stochastic vs. deterministic policy

- Toy Markov Decision Process (MDP) with $\mathcal{S} = \{s, s_T\}$, s_T is terminal state, s is start state, $\gamma = 1$, $\mathcal{A} = \{\text{left}, \text{right}\}$, all actions lead to s_T, $R_s^{\text{left}} = 1$, $R_s^{\text{right}} = 0$
- Deterministic policy:

$$\pi_\theta(s) = \begin{cases} \text{left} & \text{if } \theta \geq 0 \\ \text{right} & \text{otherwise} \end{cases}$$

- Stochastic policy:

$$\pi'_\theta(s) = \begin{cases} \text{left} & \text{with probability } \sigma(\theta) \\ \text{right} & \text{with probability } 1 - \sigma(\theta) \end{cases} \quad \text{with} \quad \sigma(\theta) = \frac{1}{1 + e^{-\theta}}$$

- $V^\pi(s) = \mathbb{I}[\theta \geq 0], \quad V^{\pi'}(s) = \sigma(\theta)$
- $\frac{\partial}{\partial\theta} V^{\pi'}(s) = (1 - \sigma(\theta))\sigma(\theta)$
  $\frac{\partial}{\partial\theta} V^{\pi}(s) =$ **?**

Introduction: Value function approaches to RL

- "Standard approach" to reinforcement learning (RL) is to
  - estimate a value function (V- or Q-function) and then
  - define a "greedy" policy on top of it.
- One may argue that this approach is
  - somehow "indirect" and
  - oriented towards deterministic policies.
- Problems:
  - "Strong causality" violated (small changes may have drastic effects)

Introduction: Policy gradient approaches to RL

- Model a stochastic policy by a function approximator (the "actor") with own parameters $\boldsymbol{\theta}$, for example for discrete action set

$$\pi(s, a \,|\, \theta) = \frac{e^{h(s,a \,|\, \theta)}}{\sum_{a'} e^{h(s,a' \,|\, \theta)}}$$

  with *preferences* $h(s, a \,|\, \theta)$

- Adapt policy according to

$$\Delta\boldsymbol{\theta} \approx \alpha \nabla_{\boldsymbol{\theta}} J(\pi)$$

  where $J(\pi)$ is a performance measure of the policy π and α a positive step-size/learning-rate

- Subsumes known methods such as actor-critic approaches and the REINFORCE algorithms

Example: Softmax policy

Consider vector of features ($\rightarrow$ neural network) $\boldsymbol{\phi}(s,a), \forall a \in A, s \in S$; policy is a Gibbs distribution in a linear combination of the features

$$\pi(s,a) = \frac{e^{\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\phi}(s,a)}}{\sum_b e^{\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\phi}(s,b)}} \quad .$$

Alternatively, for discrete actions, consider a feature map $\boldsymbol{\phi}$ for the features and

$$\pi(s,a) = \frac{e^{\boldsymbol{\theta}_a^{\mathsf{T}}\boldsymbol{\phi}(s)}}{\sum_b e^{\boldsymbol{\theta}_b^{\mathsf{T}}\boldsymbol{\phi}(s)}}$$

with parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{|A|})$.

Average reward formulation I

Expected reward per time step

$$J(\pi) = \lim_{t\to\infty} \frac{1}{t}\mathbb{E}\{r_1 + \cdots + r_t \,|\, \pi\} = \sum_s \mu^\pi(s) \sum_a \pi(s,a) R_s^a$$

We assume that the stationary distribution μ^π of states under π exists and is independent of s_0 (e.g., the underlying Markov chain is finite and ergodic, i.e., $\exists k \in \mathbb{N}\, \forall s, s' \in S\, \exists k' \leq k$: $\Pr\{s \xrightarrow{k'} s' \,|\, \pi\} > 0$)

$$\mu^\pi(s) = \lim_{t\to\infty} \Pr\{s_t = s \,|\, s_0; \pi\} = \lim_{t\to\infty} \Pr\{s_t = s \,|\pi\}$$

$\mu^\pi(s)$ is also called the on-policy distribution and corresponds to the fraction of time spent in s under π.

Average reward formulation II

In general it holds

$$\mathsf{Pr}\{s_{t+1} = s' \,|\, \pi\} = \sum_s \mathsf{Pr}\{s_t = s \,|\, \pi\}\mathsf{Pr}\{s_{t+1} = s' \,|\, s_t = s, \pi\}$$

Stationarity

$$\mu^\pi(s') = \lim_{t\to\infty} \mathsf{Pr}\{s_t = s' \,|\, \pi\} = \lim_{t\to\infty} \mathsf{Pr}\{s_{t+1} = s' \,|\, \pi\}$$

implies for $t \to \infty$

$$\mu^\pi(s') = \sum_s \underbrace{\mu^\pi(s)}_{\mathsf{Pr}\{s_t=s\,|\pi\}} \underbrace{\sum_a \pi(s,a) P^a_{ss'}}_{\mathsf{Pr}\{s_{t+1}=s'\,|\,s_t=s,\pi\}}$$

Average reward formulation III

Let's redefine

$$Q^\pi(s,a) = \sum_{t=1}^{\infty} \mathbb{E}\{r_t - J(\pi) \mid s_0 = s, a_0 = a, \pi\}$$

and with

$$V^\pi(s) = \sum_a \pi(s,a) Q^\pi(s,a)$$

we have

$$\begin{aligned} Q^\pi(s,a) &= \sum_{t=1}^{\infty} \mathbb{E}\{r_t - J(\pi) \mid s_0 = s, a_0 = a, \pi\} \\ &= R_s^a - J(\pi) + \sum_{s'} P_{ss'}^a V^\pi(s') \ . \end{aligned}$$

This is actually better referred to as some type of advantage (A, see below) under stationary assumption.

Start-state formulation

Goal is to maximize the expected return

$$J(\pi) = \mathbb{E}\left\{ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \,\middle|\, s_0, \pi \right\}$$

with $\gamma \in [0, 1]$, $\gamma = 1$ only for episodic tasks, and

$$Q^{\pi}(s, a) = \mathbb{E}\left\{ \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \,\middle|\, s_t = s, a_t = a, \pi \right\} \quad .$$

On-policy distribution in episodic tasks

- In episodic tasks, on-policy distribution $\mu^\pi(s)$ depends on initial states, i.e., the start state distribution $p_{\text{start}}(s)$.
- Let $\eta^\pi(s)$ denote the number of time steps spent, on average, in s in a single episode:

$$\eta^\pi(s) = \mathbb{E}_{s_0 \sim p_{\text{start}}} \left[\sum_{k=0}^{\infty} \Pr\{s_0 \xrightarrow{k} s \mid \pi\} \right]$$

- Time is spent in a state s if episodes start in s, or if transitions are made into s from a preceding state s' in which time is spent:

$$\eta^\pi(s) = p_{\text{start}}(s) + \sum_{s'} \eta^\pi(s') \sum_a P^a_{s's} \pi(a, s')$$

  This system of equations can be solved for $\eta(s)$.
- $\mu^\pi(s)$ is then the fraction of time spent in each state:

$$\mu^\pi(s) = \eta^\pi(s) / \sum_{s'} \eta^\pi(s')$$

Outline

Policy gradient theorem, average reward

Theorem

For any MDP, in average-reward formulation

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) \ .$$

- No $\nabla_{\boldsymbol{\theta}} \mu^\pi(s)$ terms
- If s is sampled following π, then

$$\sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)$$

is an unbiased estimate of $\nabla_{\boldsymbol{\theta}} J(\pi)$

Policy gradient theorem, start-state formulation

Theorem

For any MDP, in start-state formulation

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \sum_s \eta_\gamma^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)$$
$$\propto \mathbb{E}_{\substack{\textit{state-action}\\ \textit{sequences}\\ \textit{following } \pi}} \left[\sum_{k=0}^{\infty} \gamma^k \frac{1}{\pi(s_k,a_k)} \nabla_{\boldsymbol{\theta}} \pi(s_k,a_k) Q^\pi(s_k,a_k) \right] \quad .$$

- Here we define

$$\eta_\gamma^\pi(s) = \mathbb{E}_{s_0 \sim p_{\text{start}}} \left[\sum_{k=0}^{\infty} \gamma^k \Pr\{s_0 \xrightarrow{k} s \mid \pi\} \right] \quad .$$

Not normalized ($\sum_s \eta_\gamma^\pi(s) \neq 1$) for $\gamma < 1$, a factor $(1-\gamma)$ is missing.

REINFORCE algorithm

- If s is sampled from distribution following π and $\gamma = 1$, then

$$\sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)$$

is an unbiased estimate of $\nabla_{\boldsymbol{\theta}} J(\pi)$ – note equal weighting of actions (uniform distribution)

- If s and a are sampled from distribution following π, then

$$\frac{1}{\pi(s,a)}$$

re-weights the samples ($\rightarrow$ importance weighting)

- $Q^\pi(s,a)$ is usually unknown; $Q^\pi(s_t,a_t)$ can be estimated using actual returns $R_t = \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}$ or $R_t = \sum_{k=1}^{T-t} r_{t+k}$ (for a finite episode of length T) as estimates

REINFORCE pseudocode

REINFORCE algorithm parameter update:

$$\Delta\boldsymbol{\theta}_t \propto \gamma^t \frac{1}{\pi(s_t, a_t)} \nabla_{\boldsymbol{\theta}} \pi(s_t, a_t) R_t \overset{\text{“log derivative trick”}}{=} \gamma^t \underbrace{\nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t)}_{\text{“score function”}} R_t$$

Algorithm 1: REINFORCE

Input: differential policy π parameterized by $\boldsymbol{\theta}$, learning rate $\alpha > 0$, initial policy parameters $\boldsymbol{\theta}$

repeat

  Generate episode $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$

  foreach $t = 1, \ldots, T-1$ **do**

    $R_t = \sum_{k=1}^{T-t} \gamma^{k-1} r_{t+k}$

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha R_t \gamma^t \nabla_{\boldsymbol{\theta}} \ln \pi(s_t, a_t)$

until *stopping criterion is met*

Adding a baseline

The policy gradient theorem can be generalized, in either average-reward or start-state formulations, to include a baseline, e.g.,

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) \left(Q^\pi(s,a) - b(s)\right) \quad ,$$

where $b(s) : S \to \mathbb{R}$ is an arbitrary baseline function.

$\phi(s) = \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) b(s)$ acts as a control variate. Note $\mathbb{E}[\phi] = 0$.

A possible choice of $b(s)$ would be some estimate of the value function $V^\pi(s)$. We call

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

the *advantage function*.

Proof policy gradient, start-state, undiscounted I

We start by assuming $\gamma = 1$.

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} V^\pi(s) &= \nabla_{\boldsymbol{\theta}} \sum_a \pi(s,a) Q^\pi(s,a) \\
&\overset{\text{product rule}}{=} \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \nabla_{\boldsymbol{\theta}} Q^\pi(s,a)\right] \\
&= \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \nabla_{\boldsymbol{\theta}} \left[R_s^a + \sum_{s'} P_{ss'}^a V^\pi(s')\right]\right] \\
&= \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \sum_{s'} P_{ss'}^a \nabla_{\boldsymbol{\theta}} V^\pi(s')\right]
\end{aligned}
$$

Proof policy gradient, start-state, undiscounted II

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} V^\pi(s) &= \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \sum_{s'} P^a_{ss'} \nabla_{\boldsymbol{\theta}} V^\pi(s') \right] \\
&= \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \nabla_{\boldsymbol{\theta}} V^\pi(s') \\
&= \mathsf{Pr}\{s \xrightarrow{0} s \,|\, \pi\} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s')
\end{aligned}
$$

$$
= \sum_{s'} \left[\mathsf{Pr}\{s \xrightarrow{0} s' \,|\, \pi\} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s',a) Q^\pi(s',a) + \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s') \right]
$$

$$
= \sum_{s'} \sum_{k=0}^{\infty} \mathsf{Pr}\{s \xrightarrow{k} s' \,|\, \pi\} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s',a) Q^\pi(s',a)
$$

Proof policy gradient, start-state: Unrolling

Closer look at last step on previous slide:

$$
\begin{aligned}
&\sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s') \\
&= \sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \left[\sum_a \nabla_{\boldsymbol{\theta}} \pi(s', a) Q^\pi(s', a) + \sum_{s''} \mathsf{Pr}\{s' \xrightarrow{1} s'' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s'') \right] \\
&= \sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \left[\sum_a \nabla_{\boldsymbol{\theta}} \pi(s', a) Q^\pi(s', a) \right] \\
&\quad + \sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \left[\sum_{s''} \mathsf{Pr}\{s' \xrightarrow{1} s'' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s'') \right] \\
&= \sum_{s'} \mathsf{Pr}\{s \xrightarrow{1} s' \,|\, \pi\} \left[\sum_a \nabla_{\boldsymbol{\theta}} \pi(s', a) Q^\pi(s', a) \right] + \sum_{s''} \mathsf{Pr}\{s \xrightarrow{2} s'' \,|\, \pi\} \nabla_{\boldsymbol{\theta}} V^\pi(s'')
\end{aligned}
$$

etc.

Proof policy gradient, start-state, undiscounted III

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \nabla_{\boldsymbol{\theta}} \mathbb{E}\left\{ \sum_{t=1}^{\infty} r_t \,\middle|\, \pi \right\} = \mathbb{E}_{s_0 \sim p_{\text{start}}} \left[\nabla_{\boldsymbol{\theta}} V^{\pi}(s_0) \right]$$

$$\nabla_{\boldsymbol{\theta}} V^{\pi}(s_0) \overset{\text{slide 26}}{=} \sum_s \underbrace{\sum_{k=0}^{\infty} \mathsf{Pr}\{s_0 \xrightarrow{k} s \mid \pi\}}_{\eta^{\pi}(s),\ \text{see slide 18}} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^{\pi}(s,a)$$

Thus we have (note equal weighting of actions in first line):

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \mathbb{E}_{s_0 \sim p_{\text{start}}} \left[\sum_s \sum_{k=0}^{\infty} \mathsf{Pr}\{s_0 \xrightarrow{k} s \mid \pi\} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^{\pi}(s,a) \right]$$

$$\propto \mathbb{E}_{\substack{\text{state-action} \\ \text{sequences} \\ \text{following } \pi}} \left[\sum_{k=0}^{\infty} \frac{1}{\pi(s_k, a_k)} \nabla_{\boldsymbol{\theta}} \pi(s_k, a_k) Q^{\pi}(s_k, a_k) \right]$$

Adding discounting: Time and ensemble average

- Consider an MDP $\mathcal{M}_\gamma$ with discount factor $\gamma \in]0,1[$ leading to the expected return for some policy:

$$J_{\mathcal{M}_\gamma}(\pi) = \mathbb{E}_{\mathcal{M}_\gamma}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t\right]$$

- Let $\mathcal{M}_1$ be the undiscounted version of that MDP.
- Now consider the MDP $\mathcal{M}_{1,\gamma}$, which is the same as $\mathcal{M}_1$ except that after each action (and receiving the reward) the process is terminated with probability of $1-\gamma$ by going to a new terminal state s_{dummy}:

$$J_{\mathcal{M}_{1,\gamma}}(\pi) = \sum_{l=0}^{\infty} (1-\gamma)\gamma^l \mathbb{E}_{\mathcal{M}_1}[r_1 + \dots + r_{l+1}]$$

$(1-\gamma)\gamma^l$ is the probability that the transition to s_{dummy} happens after l steps.

Time and ensemble average II

$$
\begin{aligned}
J_{\mathcal{M}_{1,\gamma}}(\pi) &= \sum_{l=0}^{\infty}(1-\gamma)\gamma^l \mathbb{E}_{\mathcal{M}_1}[r_1 + \cdots + r_{l+1}] \\
&= \mathbb{E}_{\mathcal{M}_1}\left[\sum_{l=0}^{\infty}(1-\gamma)\gamma^l[r_1 + \cdots + r_{l+1}]\right] \\
&= \mathbb{E}_{\mathcal{M}_1}\left[\sum_{t=1}^{\infty}(1-\gamma)\sum_{i=t}^{\infty}\gamma^{i-1} r_t\right]
\end{aligned}
$$

First sum picks the time step t, second sum accumulates r_t terms:

$$
\begin{aligned}
&(1-\gamma)\gamma^0[r_1] \\
&(1-\gamma)\gamma^1[r_1 + r_2] \\
&(1-\gamma)\gamma^2[r_1 + r_2 + r_3] \\
&\qquad\vdots
\end{aligned}
$$

Time and ensemble average III

We have ($\rightarrow$ geometric series):

$$(1-\gamma)\sum_{t=1}^{\infty}\sum_{i=t}^{\infty}\gamma^{i-1}r_t = \sum_{t=1}^{\infty}\gamma^{t-1}r_t(1-\gamma)\sum_{i=0}^{\infty}\gamma^{i} = \sum_{t=1}^{\infty}\gamma^{t-1}r_t$$

and thus:

$$J_{\mathcal{M}_{1,\gamma}}(\pi) = J_{\mathcal{M}_{\gamma}}(\pi)$$

As this holds for all start-state distributions, this implies

$$V^{\pi}_{\mathcal{M}_{1,\gamma}}(s) = V^{\pi}_{\mathcal{M}_{\gamma}}(s)$$

for all states s. As $\mathcal{M}_{1,\gamma}$ differs from $\mathcal{M}_1$ only *after* taking each action and receiving the corresponding reward, this implies $Q^{\pi}_{\mathcal{M}_{1,\gamma}}(s,a) = Q^{\pi}_{\mathcal{M}_{\gamma}}(s,a)$ for all state-action pairs.

Policy gradient theorem discounted case I

We derive the policy gradient theorem for $\mathcal{M}_\gamma$ using $\mathcal{M}_{1,\gamma}$ and $\mathcal{M}_1$ and apply the undiscounted start-state version to $\mathcal{M}_{1,\gamma}$.

If $\Pr\{s_0 \xrightarrow{k} s \,|\, \pi\}$ is the probability of going from s_0 to s in k steps in $\mathcal{M}_1$, then the corresponding probability in $\mathcal{M}_{1,\gamma}$ is $\gamma^k \Pr\{s_0 \xrightarrow{k} s \,|\, \pi\}$.

Thus last step of the proof on slide 28 for $\mathcal{M}_{1,\gamma}$ becomes:

$$\mathbb{E}_{s_0\sim p_{\text{start}}}\left[\nabla_{\boldsymbol{\theta}} V^\pi(s_0)\right] =$$

$$\mathbb{E}_{s_0\sim p_{\text{start}}}\left[\sum_s \sum_{k=0}^\infty \underbrace{\gamma^k \underbrace{\Pr\{s_0 \xrightarrow{k} s \,|\, \pi\}}_{\text{probability under } \mathcal{M}_1}}_{\text{probability under } \mathcal{M}_{1,\gamma}} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)\right] =$$

$$\mathbb{E}_{s_0\sim p_{\text{start}}}\left[\sum_s \eta_\gamma^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)\right]$$

Policy gradient theorem discounted case II

And in the same way:

$$\mathbb{E}_{s_0 \sim p_{\text{start}}}\left[\nabla_{\boldsymbol{\theta}} V^\pi(s_0)\right] =$$

$$\mathbb{E}_{s_0 \sim p_{\text{start}}}\Bigg[\sum_s \sum_{k=0}^{\infty} \underbrace{\gamma^k \underbrace{\Pr\{s_0 \xrightarrow{k} s \,|\, \pi\}}_{\text{probability under } \mathcal{M}_1}}_{\text{probability under } \mathcal{M}_{1,\gamma}} \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)\Bigg] \propto$$

$$\mathbb{E}_{\substack{\text{state-action}\\ \text{sequences}\\ \text{following } \pi \text{ on } \mathcal{M}_\gamma}}\left[\sum_{k=0}^{\infty} \gamma^k \frac{1}{\pi(s_k,a_k)} \nabla_{\boldsymbol{\theta}} \pi(s_k,a_k) Q^\pi(s_k,a_k)\right]$$

Note: $\Pr\{s_0 \xrightarrow{k} s \,|\, \pi\}$ is the same for $\mathcal{M}_\gamma$ and $\mathcal{M}_1$ and Q^π is the same for $\mathcal{M}_\gamma$ and $\mathcal{M}_1$ as can be shown using the "time and ensemble average" approach as before

Proof policy gradient, average reward I

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} V^\pi(s) &= \nabla_{\boldsymbol{\theta}} \sum_a \pi(s,a) Q^\pi(s,a) \\
&\overset{\text{product rule}}{=} \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \nabla_{\boldsymbol{\theta}} Q^\pi(s,a) \right] \\
&= \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \nabla_{\boldsymbol{\theta}} \left[R_s^a - J(\pi) + \sum_{s'} P_{ss'}^a V^\pi(s') \right] \right] \\
&= \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \left[-\nabla_{\boldsymbol{\theta}} J(\pi) + \sum_{s'} P_{ss'}^a \nabla_{\boldsymbol{\theta}} V^\pi(s') \right] \right]
\end{aligned}
$$

Proof policy gradient, average reward II

$$\nabla_{\boldsymbol{\theta}} V^\pi(s) = \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \left[-\nabla_{\boldsymbol{\theta}} J(\pi) + \sum_{s'} P^a_{ss'} \nabla_{\boldsymbol{\theta}} V^\pi(s') \right] \right] \Rightarrow$$

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \sum_a \left[\nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \pi(s,a) \sum_{s'} P^a_{ss'} \nabla_{\boldsymbol{\theta}} V^\pi(s') \right] - \nabla_{\boldsymbol{\theta}} V^\pi(s) \Rightarrow$$

$$\sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} J(\pi) = \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) + \sum_s \mu^\pi(s) \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \nabla_{\boldsymbol{\theta}} V^\pi(s') - \sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} V^\pi(s)$$

Proof policy gradient, average reward III

$$\begin{aligned}
\sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} J(\pi) &= \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) \\
&\quad + \underbrace{\sum_s \mu^\pi(s) \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}}_{\mu^\pi(s),\ \text{see slide 15}} \nabla_{\boldsymbol{\theta}} V^\pi(s') - \sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} V^\pi(s) \\
&= \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a) \\
&\quad + \sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} V^\pi(s) - \sum_s \mu^\pi(s) \nabla_{\boldsymbol{\theta}} V^\pi(s)
\end{aligned}$$

$$\nabla_{\boldsymbol{\theta}} J(\pi) = \sum_s \mu^\pi(s) \sum_a \nabla_{\boldsymbol{\theta}} \pi(s,a) Q^\pi(s,a)$$

Outline

1. Background
   - Monte Carlo methods
   - Importance weighting
   - Control variates
   - "Log-derivative trick"
2. Policy gradient methods
3. Policy gradient theorem and REINFORCE
   - REINFORCE algorithm
   - Proof undiscounted policy gradient considering start-state
   - Proof discounted policy gradient considering start-state
   - Proof policy gradient average reward*
4. Policy gradients in the wild

Policy gradients in the wild

In the wild, you will find several expressions for the policy gradient, which have the form

$$\boldsymbol{g} = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(s_t, a_t)\right] \quad , \tag{1}$$

where Ψ_t may be one of the following (Schulman et al., 2016):

1. $\sum_{t=0}^{\infty} r_t$: total reward of the trajectory
2. $\sum_{t'=t}^{\infty} r_{t'}$: reward following action a_t
3. $\sum_{t'=t}^{\infty} r_{t'} - b(s_t)$: baselined version of previous formula
4. $Q^\pi(s_t, a_t)$: state-action value function
5. $A^\pi(s_t, a_t)$: advantage function
6. $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$: temporal difference (TD) residual

References

R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning 8:229-256, 1992

R. S. Sutton, A. G. Barto. Reinforcement Learning: An Introduction. 2nd edition, MIT Press, 2018 [Chapter 5]

C. Nota and P. S. Thomas. Is the Policy Gradient a Gradient? Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), pp. 939–947, 2020

J. Schulman, P. Moritz, S. Levine, M. I. Jordan, P. Abbeel. High-dimensional continuous control using generalized advantage estimation, International Conference on Learning Representations (ICLR), 2016