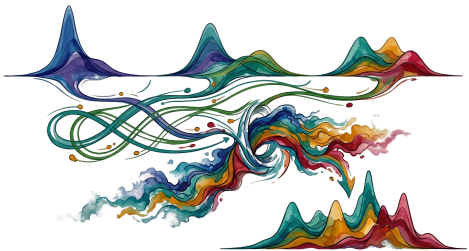
Policy Gradient Methods

Searching the policy directly: scores, plateaus, natural gradients and trust regions

Aurélien Garivier

CR15 Reinforcement Learning — M2, ENS de Lyon

Companion to Chapter 14 of the course notes

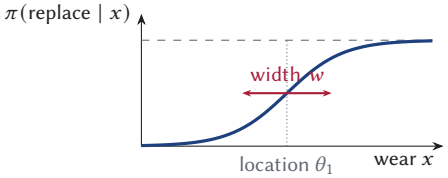


Why search the policy directly?

So far: learn V or Q, then act *greedily*. The other road: **parametrize the policy**, measure its performance, **climb the gradient**. Four reasons:

- **continuous or huge actions**: the $\max_a$ of the Bellman operator is itself a problem;
- the best policy may have to be **stochastic** (games, partial observability);
- the performance is **smooth** in the parameters, whereas greedy policies *jump*;
- a parametrization encodes **prior knowledge** of the shape of good policies.

The price: a non-concave objective (ascent may stop at a bad stationary point), and noisy gradients when estimated from episodes.



two numbers describe a policy
for the machine of Ch. 13:
"replace around θ_1, with hesitation w"

Parametrized policies: four families

A family $(\pi_\theta)_{\theta \in \mathbb{R}^d}$ of randomized policies, $\theta \mapsto \pi_\theta(a \mid s)$ *differentiable and positive*: every action keeps a chance of being tried.

Tabular softmax. One number $\theta(s,a)$ per pair, $\pi_\theta(a \mid s) \propto e^{\theta(s,a)}$. Every randomized policy with positive probabilities is of this form; deterministic policies are the limits $\|\theta\| \to \infty$.

Gaussian, for continuous actions. $\pi_\theta(\cdot \mid s) = \mathcal{N}(m_\theta(s), \sigma_\theta(s)^2)$: a mean action plus an exploration noise. The regulator below: $a = -g\, e + \sigma_e \xi$, one parameter g.

Log-linear. $\pi_\theta(a \mid s) \propto e^{\theta^\top \varphi(s,a)}$ on features φ: d parameters instead of $|\mathcal{S}||\mathcal{A}|$.

Neural network. Session H06: a hidden layer of 128 units maps the four-dimensional cart-pole state to the probabilities of its two actions.

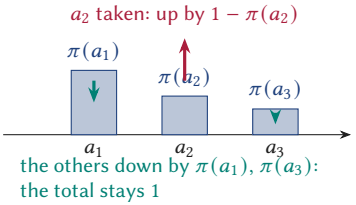
Keep the first family in mind: "deterministic = at infinity" is where the plateaus of this lecture come from.

The score function in one picture

The **score** $\nabla_\theta \ln \pi_\theta(a \mid s)$ is the direction in which to move θ to make the action *a more likely*.

- **Softmax**: $\partial \ln \pi_\theta(a \mid s)/\partial\theta(s, a') = \mathbb{1}\{a' = a\} - \pi_\theta(a' \mid s)$: push up the action taken, push down the others in proportion to their probability.
- **Gaussian**: $\partial_m \ln \pi = (a - m)/\sigma^2$: pull the mean toward the action taken.
- Averaged over $a \sim \pi_\theta$, the score is *zero*: $\sum_a \pi_\theta \nabla \ln \pi_\theta = \nabla \sum_a \pi_\theta = 0$.

Move θ along "score × a number": the action becomes more likely if the number is positive, less likely if negative. The whole lecture is about the number.



The softmax score $e_{a_2} - \pi$ at one state, drawn on the probabilities.

The criterion, and the two regimes of the lecture

The **performance** of θ from an initial law μ is the discounted value, and we climb it:

$$J(\theta) \;=\; V^{\pi_\theta}(\mu) \;=\; \mathbb{E}_\mu^{\pi_\theta}\Big[\sum_{t\geq 1}\gamma^{t-1}R_t\Big].$$

Where π_θ spends its discounted time: the **occupation measure** $d_\mu^{\pi_\theta}(s) = (1-\gamma)\sum_t \gamma^{t-1}\mathbb{P}_\mu^{\pi_\theta}(S_t = s)$.

With the model: planning. d_μ^π and Q^π are *computed*, the gradient is *exact*; gradient ascent is an alternative to value and policy iteration, whose speed can be studied as such (the theorems, the machine, the regulator, the hike, the chain).

From episodes: learning. The *same* gradient is *estimated* from sampled trajectories, and its variance becomes the whole problem: REINFORCE, actor–critic, PPO. Every worked example runs both.

Episodes vs discounting in one line: the theory is discounted, the algorithms run on episodes of length H; the two are related by truncation and the weights γ^{t-1} (Exercise 14.4).

The map of the lecture

Part	The question	The tool	Worked example
1. The theorem	what is the gradient of J, and why is it computable without derivatives of the dynamics?	policy gradient theorem, baselines	one state, three arms
2. From episodes	how to estimate it from trajectories, and how noisy is that?	REINFORCE, actor–critic	the machine (two parameters), the regulator (one gain)
3. Softmax, natural gradient	why is plain ascent slow, and what removes the two damping factors?	multiplicative weights, a dimension-free rate	the windy hike, the two-state chain
4. Trust regions	how far can a step be trusted when it is evaluated with the old policy's data?	surrogate, TRPO, PPO	riverswim
5. Risk	what if the objective is not the mean?	tilted weights, worst fractions	the hike with $\beta = -1, 0.5$

Two words will keep coming back: **plateau** (the gradient vanishes near deterministic policies and at rarely visited states) and **variance** (an estimate from episodes is drowned in noise).

Part 1

The policy gradient theorem

1. The policy gradient theorem
2. Learning from episodes: REINFORCE and actor–critic
3. Softmax policies and the natural gradient
4. Trust regions: TRPO and PPO
5. Risk-sensitive policy gradients

How does J change when θ moves?

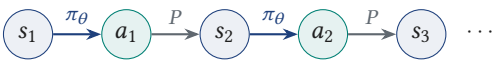
Lemma 14.2

The probability of a trajectory is a *product*, $\mu(s_1)\prod_t \pi_\theta(a_t \mid s_t)P(s_{t+1} \mid s_t, a_t)$, in which only the **policy factors** depend on θ. Differentiating it,

$$\nabla_\theta \ln \mathbb{P}_\theta(\tau) = \sum_{t \le H} \nabla_\theta \ln \pi_\theta(A_t \mid S_t) :$$

the sum of the scores of the actions taken. Hence, for any function f of the trajectory, $\nabla_\theta \mathbb{E}_\theta[f(\tau)] = \mathbb{E}_\theta\big[f(\tau)\sum_t \nabla_\theta \ln \pi_\theta(A_t \mid S_t)\big]$.

We cannot differentiate the dynamics, which we do not know; we do not need to. And f is *any* function: Part 5 will use this for risk criteria.



blue factors: the policy, they depend on θ

gray factors: the dynamics, **unknown**—they drop out

the derivative of the law is the law times the *sum of the scores*: no derivative of P ever appears

The policy gradient theorem

Theorem 14.3 (Sutton et al., 2000)

Theorem 14.3 (Policy gradient theorem). With $\pi = \pi_\theta$,

$$\nabla J(\theta) = \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d_\mu^\pi,\, a\sim\pi(\cdot|s)} \left[\nabla_\theta \ln \pi_\theta(a \mid s)\, Q^\pi(s,a)\right],$$

and Q^π may be replaced by $Q^\pi - b(s)$ for any function b of the state, e.g. by the advantage $A^\pi = Q^\pi - V^\pi$. Finite horizon: the same, summed over t with Q_t^π.

- **In words**: *average, over the states you visit and the actions you take, the score of the action weighted by how good the action was.* Actions better than average get their probability increased; the factor $1/(1-\gamma)$ is the horizon.
- **Why it is remarkable**: no derivative of the dynamics, and *none of the occupation measure* either, although d_μ^π depends on θ—that is the content of the proof.

Score × value, averaged where the policy goes. The rest: how to compute or estimate this average, and how fast ascent along it reaches the optimum.

Proof, 1: differentiate the two Bellman equations

Theorem 14.3

The idea. $V^\pi(s)$ depends on θ through the action chosen *now* at s and through what happens *later*; the two Bellman equations of Chapter 7 separate the two.

1. $V^\pi(s) = \sum_a \pi_\theta(a \mid s)\, Q^\pi(s,a)$, so by the product rule

$$\nabla V^\pi(s) = \underbrace{\sum_a \nabla \pi_\theta(a \mid s)\, Q^\pi(s,a)}_{\coloneqq g(s)\text{: the action now}} + \underbrace{\sum_a \pi_\theta(a \mid s)\, \nabla Q^\pi(s,a)}_{\text{the future}}.$$

Proof, 1: differentiate the two Bellman equations

Theorem 14.3

The idea. $V^\pi(s)$ depends on θ through the action chosen *now* at s and through what happens *later*; the two Bellman equations of Chapter 7 separate the two.

1. $V^\pi(s) = \sum_a \pi_\theta(a \mid s)\, Q^\pi(s,a)$, so by the product rule

$$\nabla V^\pi(s) = \underbrace{\sum_a \nabla \pi_\theta(a \mid s)\, Q^\pi(s,a)}_{\coloneqq g(s)\text{: the action now}} + \underbrace{\sum_a \pi_\theta(a \mid s)\, \nabla Q^\pi(s,a)}_{\text{the future}}.$$

2. $Q^\pi(s,a) = r(s,a) + \gamma \sum_{s'} P(s' \mid s,a)\, V^\pi(s')$, and neither r nor P depends on θ, so $\nabla Q^\pi(s,a) = \gamma \sum_{s'} P(s' \mid s,a)\, \nabla V^\pi(s')$: nothing else to differentiate.

Proof, 1: differentiate the two Bellman equations

Theorem 14.3

The idea. $V^\pi(s)$ depends on θ through the action chosen *now* at s and through what happens *later*; the two Bellman equations of Chapter 7 separate the two.

1. $V^\pi(s) = \sum_a \pi_\theta(a \mid s)\, Q^\pi(s,a)$, so by the product rule
$$\nabla V^\pi(s) = \underbrace{\sum_a \nabla \pi_\theta(a \mid s)\, Q^\pi(s,a)}_{\coloneqq g(s)\text{: the action now}} + \underbrace{\sum_a \pi_\theta(a \mid s)\, \nabla Q^\pi(s,a)}_{\text{the future}}.$$
2. $Q^\pi(s,a) = r(s,a) + \gamma \sum_{s'} P(s' \mid s,a)\, V^\pi(s')$, and neither r nor P depends on θ, so $\nabla Q^\pi(s,a) = \gamma \sum_{s'} P(s' \mid s,a)\, \nabla V^\pi(s')$: nothing else to differentiate.
3. Put the second line into the first: with $P^\pi(s,s') = \sum_a \pi_\theta(a \mid s) P(s' \mid s,a)$ the chain of π,
$$\nabla V^\pi(s) \;=\; g(s) + \gamma \sum_{s'} P^\pi(s,s')\, \nabla V^\pi(s').$$

A Bellman *evaluation* equation for the gradient itself: same chain, same discount, and the "reward" is $g(s)$, the immediate effect of moving θ at s.

Proof, 2: unroll, then average

Theorem 14.3

1. **Unroll.** An evaluation equation is solved by summing the discounted "rewards" along the chain (Chapter 7, as for V^π itself):

$$\nabla V^\pi(s) = \sum_{t \geq 1} \gamma^{t-1} \sum_{s'} \mathbb{P}_s^\pi(S_t = s')\, g(s').$$

Proof, 2: unroll, then average

Theorem 14.3

1. **Unroll.** An evaluation equation is solved by summing the discounted "rewards" along the chain (Chapter 7, as for V^π itself):

$$\nabla V^\pi(s) = \sum_{t \geq 1} \gamma^{t-1} \sum_{s'} \mathbb{P}_s^\pi(S_t = s')\, g(s').$$

2. **Average against the initial law μ.** Since $J(\theta) = \sum_s \mu(s) V^\pi(s)$,

$$\nabla J(\theta) = \sum_{s'} \Big[\sum_{t \geq 1} \gamma^{t-1} \mathbb{P}_\mu^\pi(S_t = s') \Big] g(s') = \frac{1}{1-\gamma} \sum_{s'} d_\mu^\pi(s')\, g(s'),$$

the bracket being, by definition, the occupation measure $d_\mu^\pi(s')/(1-\gamma)$.

Proof, 2: unroll, then average

Theorem 14.3

1. **Unroll.** An evaluation equation is solved by summing the discounted "rewards" along the chain (Chapter 7, as for V^π itself):
$$\nabla V^\pi(s) = \sum_{t \geq 1} \gamma^{t-1} \sum_{s'} \mathbb{P}_s^\pi(S_t = s')\, g(s').$$
2. **Average against the initial law μ.** Since $J(\theta) = \sum_s \mu(s) V^\pi(s)$,
$$\nabla J(\theta) = \sum_{s'} \Big[\sum_{t \geq 1} \gamma^{t-1} \mathbb{P}_\mu^\pi(S_t = s') \Big] g(s') = \frac{1}{1-\gamma} \sum_{s'} d_\mu^\pi(s')\, g(s'),$$
the bracket being, by definition, the occupation measure $d_\mu^\pi(s')/(1-\gamma)$.
3. **Score form.** Since $\nabla \pi_\theta = \pi_\theta \nabla \ln \pi_\theta$, $g(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \big[\nabla \ln \pi_\theta(a \mid s)\, Q^\pi(s,a) \big]$: the theorem.

Proof, 2: unroll, then average

Theorem 14.3

1. **Unroll.** An evaluation equation is solved by summing the discounted "rewards" along the chain (Chapter 7, as for V^π itself):
$$\nabla V^\pi(s) = \sum_{t \geq 1} \gamma^{t-1} \sum_{s'} \mathbb{P}_s^\pi(S_t = s')\, g(s').$$
2. **Average against the initial law μ.** Since $J(\theta) = \sum_s \mu(s) V^\pi(s)$,
$$\nabla J(\theta) = \sum_{s'} \Big[\sum_{t \geq 1} \gamma^{t-1} \mathbb{P}_\mu^\pi(S_t = s') \Big] g(s') = \frac{1}{1-\gamma} \sum_{s'} d_\mu^\pi(s')\, g(s'),$$
the bracket being, by definition, the occupation measure $d_\mu^\pi(s')/(1-\gamma)$.
3. **Score form.** Since $\nabla \pi_\theta = \pi_\theta \nabla \ln \pi_\theta$, $g(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \big[\nabla \ln \pi_\theta(a \mid s)\, Q^\pi(s,a) \big]$: the theorem.
4. **The baseline is free.** $\sum_a \nabla \pi_\theta(a \mid s)\, b(s) = b(s)\, \nabla \sum_a \pi_\theta(a \mid s) = b(s)\, \nabla 1 = 0$, so Q^π may be replaced by $Q^\pi - b(s)$.

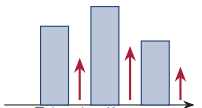
The occupation measure appears by *unrolling*, not by differentiating it: no derivative of d_μ^π, and none of P, is ever needed.

The baseline, in words

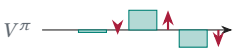
If all values $Q^\pi(s, \cdot)$ are positive (rewards are), *every* action taken is pushed up, the good ones a little more; only the renormalization sorts them out—slowly, and noisily from a single sample.

- **Subtract a baseline** $b(s)$: the expectation does not change, because the average score at a state is zero: $\sum_a \nabla \pi_\theta(a \mid s)\, b(s) = 0$.
- **The variance does change.** With $b = V^\pi$ the weight is the *advantage*: good actions up, bad actions down.
- V^π is close to the optimal baseline (Greensmith et al., 2004) (Exercise 14.3).

Free in expectation, cheap: a running average of returns, or a learned V (the *critic*, Part 2).



$Q^\pi(s, \cdot)$ all positive: every action pushed up



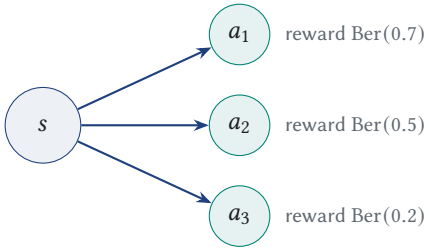
$A^\pi = Q^\pi - V^\pi$: good up, bad down, same mean

Worked example A: one state, three arms

Example 14.4: the MDP

- **State**: one. **Actions**: three arms $a \in \{1, 2, 3\}$.
- **Rewards**: Bernoulli, of means $r = (0.7,\ 0.5,\ 0.2)$.
- **Dynamics**: none—$\gamma = 0$, one decision: a *bandit*.
- **Policy class**: the softmax $\pi_\theta(a) \propto e^{\theta(a)}$, $\theta \in \mathbb{R}^3$.
- **Objective**: $J(\theta) = \sum_a \pi_\theta(a)\, r(a)$, the mean reward of the arm drawn.

Known answer: play arm 1, $J^\star = 0.7$, reached only as $\theta(1) \to \infty$. Two starting points: $\theta = 0$ (uniform) and $\theta = (0, 0, 5)$ (the worst arm has probability 0.987).



one state, one step, three arms:
d_μ^π is trivial and $Q^\pi = r$

A: the gradient by hand

Example 14.4, with the softmax score $\mathbb{1}\{a' = a\} - \pi(a')$

The theorem gives, for each arm, $\dfrac{\partial J}{\partial \theta(a)} = \pi(a)\big(r(a) - J(\theta)\big)$: the probability of an arm moves in proportion to its **advantage** $r(a) - J$ *and* to its **current probability**.

- At $\theta = 0$: $J = 0.467$, gradient $(0.078,\ 0.011,\ -0.089)$.
- At $\theta = (0, 0, 5)$: $J = 0.205$, gradient of norm 0.0065: eighteen times smaller, although the policy is much worse.

A good arm of small probability is promoted slowly; near *every* deterministic policy the gradient vanishes.

	$\theta = 0$	$\theta = (0, 0, 5)$
$\pi(a_1)$	$1/3$	0.0066
$\pi(a_2)$	$1/3$	0.0066
$\pi(a_3)$	$1/3$	0.987
$J(\theta)$	0.467	0.205
$\partial J/\partial\theta(1)$	0.078	0.0033
$\partial J/\partial\theta(2)$	0.011	0.0020
$\partial J/\partial\theta(3)$	-0.089	-0.0053

Arms down the side, starts across. The best arm's push 0.0033 is a large advantage times a tiny probability.

A: the plateau

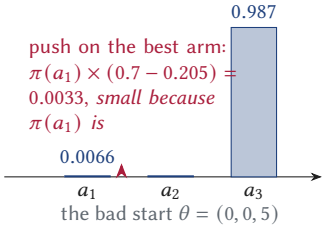
Example 14.4: gradient ascent $\theta_{k+1} = \theta_k + \eta \nabla J(\theta_k)$

Steps until the best arm has probability $\frac{1}{2}$:

start	$\eta = 1$	$\eta = 5$
$\theta = 0$ (uniform)	7	2
$\theta = (0, 0, 5)$ (bad)	136	29

From the bad start with $\eta = 1$, the gap $J^\star - J(\theta_k)$ is still 0.476 after 100 steps. J is nondecreasing along all four runs: nothing is *wrong*, it is *slow*.

A plateau: the iterates sit near a (bad) deterministic policy, where the factor $\pi(a)$ of the good arm is tiny. The ascent does leave it—in 136 steps, for three numbers.



A: the sampled gradient, with and without a baseline

Example 14.4: REINFORCE from one sample, $\hat{g} = \nabla \ln \pi_\theta(A)\,(R - b)$, $A \sim \pi_\theta$, $R \sim \mathrm{Ber}(r(A))$

Recall that $r(a)$ is the *mean* reward of arm a, $r = (0.7, 0.5, 0.2)$, so $J(\theta) = \sum_a \pi_\theta(a) r(a)$ and $\partial J / \partial \theta(a) = \pi_\theta(a)(r(a) - J)$. From episodes we do not see r: we draw an arm $A \sim \pi_\theta$ and observe one Bernoulli reward $R \sim \mathrm{Ber}(r(A))$, of mean $r(A)$. The estimate $\hat{g} \in \mathbb{R}^3$ is unbiased, $\mathbb{E}[\hat{g}] = \nabla J$; "its variance" is $\mathbb{E}\|\hat{g} - \nabla J\|^2$, the trace of its covariance, i.e. the mean squared error of one sample:

	variance, $b = 0$	variance, $b = J$	variance, $b = b^\star$	$\|\nabla J\|^2$ (the signal)
$\theta = 0$	0.297	0.152	0.152 ($b^\star = J$)	0.014
$\theta = (0, 0, 5)$	0.0157	0.0103	0.0063 ($b^\star = 0.596$)	$4.2 \cdot 10^{-5}$

- **At the uniform start**, a signal-to-noise ratio of one needs 11 samples with $b = J$ (also the optimal baseline there: all scores have the same norm).
- **On the plateau**, 150 to 370 samples per unit of signal-to-noise ratio: the gradient is not only small, it is drowned. The optimal baseline 0.596 is far from $J = 0.205$ (rare arms, large scores).

Two lessons for every example below: a baseline halves the variance or better; a plateau is twice a plateau when the gradient is estimated.

A: a preview of the natural gradient

Example 14.4: the update (14.2) of Part 3 on the bandit

Part 3 replaces $\theta \leftarrow \theta + \eta \nabla J$ by a **multiplicative** update of the probabilities,

$$\pi_{k+1}(a) \;\propto\; \pi_k(a)\, e^{\eta\,(r(a)-J_k)},$$

and the tilts compose: $\pi_K(a) \propto \pi_0(a)\, e^{\eta K r(a)}$.

- Probability $\frac{1}{2}$ to the best arm after 3 steps from $\theta = 0$ and 11 from $(0, 0, 5)$ with $\eta = 1$ (1 and 3 with $\eta = 5$): against 7 and 136.
- The gap decays *geometrically*, by $e^{-\eta(0.7-0.5)} = 0.82$ per step: $2.7 \cdot 10^{-2}$ at $K = 10$, $9 \cdot 10^{-6}$ at $K = 50$ from the uniform start; the bound of Thm. 14.13, $\ln 3/(\eta K) + R_{\max}/K$, gives 0.21 and 0.042.

Why so much faster? The factor $\pi(a)$ is gone: the *ratio* of two probabilities moves by the same factor $e^{\eta(r(a)-r(a'))}$ whatever their current size. The plain gradient moves the *probability* itself, by an amount proportional to it.

From the bad start the theorem does not apply (π_0 not uniform), but its proof does with $\mathrm{KL}(\pi^\star \,\|\, \pi_0) = 5.0$ in place of $\ln 3$: bound 0.30 at $K = 20$, where the gap is $6.9 \cdot 10^{-3}$.

The mechanism of the natural gradient is multiplicative; that of the plain gradient is not.

Part 2

Learning from episodes: REINFORCE and actor–critic

1. The policy gradient theorem
2. Learning from episodes: REINFORCE and actor–critic
3. Softmax policies and the natural gradient
4. Trust regions: TRPO and PPO
5. Risk-sensitive policy gradients

REINFORCE, line by line

Algorithm 14.1 (Williams, 1992)

Algorithm 14.1: REINFORCE with a baseline (episodic)

Input: policy π_θ, step sizes (η_k), baseline b (a running average of returns, or a learned value), batch size N

1 **for** $k = 1, 2, \ldots$ **do**

2 run N episodes with π_θ; for each episode i compute the rewards-to-go $W^i_{t:H} = \sum_{k \geq t} R^i_k$ ▷ *collect*

3 $\hat{g} \leftarrow \frac{1}{N} \sum_{i \leq N} \sum_{t \leq H} \nabla_\theta \ln \pi_{\theta,t}(A^i_t \mid S^i_t) \left(W^i_{t:H} - b(S^i_t)\right)$ ▷ *score × (what followed − baseline)*

4 $\theta \leftarrow \theta + \eta_k \hat{g}$; update the baseline ▷ *one ascent step*

- Line 1: *no model*, only trajectories of the current policy. Line 2: the theorem with $Q^\pi_t(S_t, A_t)$ replaced by its one-sample estimate, the reward-to-go. Unbiased.
- **A caution** (Nota and Thomas, 2020): the score at time t should carry the weight γ^{t-1}; dropped in practice, which makes the direction the gradient of no function. Harmless when $H \ll 1/(1-\gamma)$.

Why the reward-to-go, and why a baseline of the state only

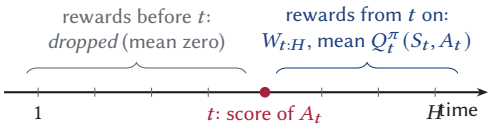
Exercise 14.2

The identity weights the score at time t by the *whole* return W_H. Two pieces can be removed without changing the expectation:

- **the rewards before t**: given the past, the average score of A_t is zero, so "score at t × anything decided before t" averages to zero;
- **a baseline $b_t(S_t)$**: same reason.

What remains, $W_{t:H}$, has conditional mean $Q_t^\pi(S_t, A_t)$: the theorem again.

The action at time t is credited only for what followed it. Less noise, same mean. A baseline depending on A_t would break the argument.



minus $b_t(S_t)$: anything that depends on the state but not on the action, e.g. $V_t^\pi(S_t)$

Actor–critic, line by line

Algorithm 14.2; Konda and Tsitsiklis (2003)

Algorithm 14.2: One-step actor–critic (online)

Input: policy π_θ (the *actor*), value V_w (the *critic*), step sizes $\eta_\theta \ll \eta_w$

for *each episode* **do**
  observe S_1
  for $t = 1, 2, \ldots$ *until the episode ends* **do**
    draw $A_t \sim \pi_\theta(\cdot \mid S_t)$; observe R_t, S_{t+1}
    $\delta_t \leftarrow R_t + \gamma V_w(S_{t+1}) - V_w(S_t)$ ▷ *TD error: an estimate of the advantage*
    $w \leftarrow w + \eta_w \, \delta_t \, \nabla_w V_w(S_t)$ ▷ *critic: TD(0), Algorithm 12.1*
    $\theta \leftarrow \theta + \eta_\theta \, \gamma^{t-1} \, \delta_t \, \nabla_\theta \ln \pi_\theta(A_t \mid S_t)$ ▷ *actor: score × TD error*

- When $V_w \approx V^\pi$, δ_t is an unbiased estimate of $A^\pi(S_t, A_t)$: one reward and a bootstrap replace the reward-to-go. Less variance; a bias when $V_w \neq V^\pi$ (Exercise 14.6). Two time scales: a fast critic, a slow actor; between one step and the full return, generalized advantage estimation (Schulman et al., 2016).

How large is the variance being fought?

For REINFORCE with returns in $[0, HR_{\max}]$ and scores bounded by G, the second moment of $\hat{g}$ is, in the worst case, of order

$$H^2 \cdot (HR_{\max})^2 \cdot G^2 :$$

H scores add up, each multiplied by a return that also grows with H; $(1-\gamma)^{-4}$ when discounted. Most of it sits in the part of the return that the actions *do not control*: that is what baselines and critics remove.

Three remedies, in increasing order of bias: a **baseline** (none), a **critic** (biased if wrong), a **deterministic gradient** (needs a differentiable critic).

Deterministic policy gradient (Silver et al., 2014). For a deterministic policy $a = m_\theta(s)$ on continuous actions, $\nabla J = \frac{1}{1-\gamma}\mathbb{E}_{d_\mu^\pi}[\nabla_\theta m_\theta(s)\, \nabla_a Q^\pi(s,a)|_{a=m_\theta(s)}]$: no sampling over actions, but the *slope* of a critic in a is needed (DDPG, Lillicrap et al., 2016). The regulator below shows it as the limit of the Gaussian policy when its noise vanishes.

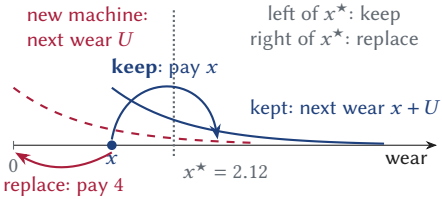
Compatible critic (Exercise 14.5). A critic linear in the score, fitted by least squares where the policy goes, gives the *exact* gradient; its weights are the natural gradient of Part 3.

Worked example D: the machine with a parametrized threshold policy

Example 14.6: the MDP (Examples 8.15 and 13.8)

- **State**: the wear $x \geq 0$, on the grid of step 0.02 on $[0, 60]$ (3 001 states). **Actions**: *keep* or *replace*.
- **Dynamics**: kept, next year's wear is $x + U$; replaced, it is U (a new machine); U exponential of mean 1.
- **Costs**: x per year while keeping, 4 per replacement; $\gamma = 0.9$; *minimize* the discounted cost from a new machine: we descend on $J(\theta) = V^{\pi_\theta}(0)$.

Known answer: replace as soon as the wear exceeds $x^\star = 2.12$ (Thm. 3.22); cost 17.17 from a new machine.



D: a policy with two numbers

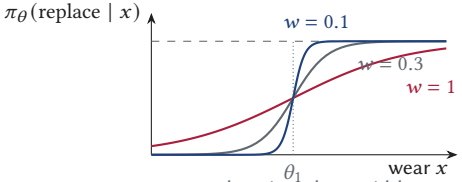
Example 14.6: the class, and the exact gradient

$$\pi_\theta(\text{replace} \mid x) = \sigma\Big(\frac{x - \theta_1}{w}\Big), \qquad w = e^{\theta_2},$$

with σ the logistic function.

- A **location** θ_1 and a **width** w; the threshold rules are the limit $w \to 0$: the class encodes the known shape of the optimal policy.
- **With the model**, everything is exact: $J(\theta)$ is one linear solve on the grid, $d_0^{\pi_\theta}$ one transposed solve, ∇J the theorem's average (checked against finite differences to five decimals).

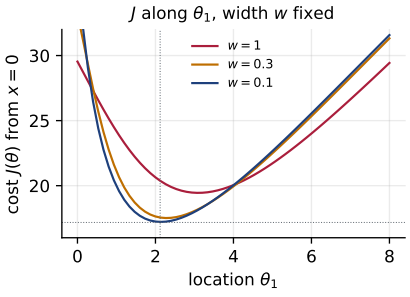
Two parameters, a smooth $J(\theta_1, w)$, an exact gradient: planning in a plane.



same location, three widths;
$w \to 0$: the threshold rule

D: the landscape, and the plateau of this class

Example 14.6, Figure 14.1 (left): J along θ_1 for three widths



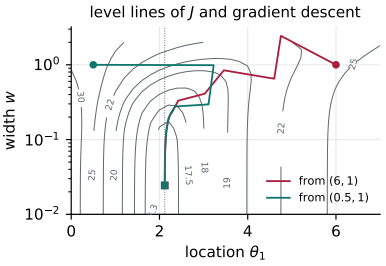
Dotted: $x^\star = 2.12$ and the optimal cost 17.17.

- **Unimodal**, with minima 19.45 at $\theta_1 = 3.10$ ($w = 1$), 17.51 at 2.29 ($w = 0.3$), 17.22 at 2.14 ($w = 0.1$): the best location tends to $x^\star$.
- **The cost of smoothing**, $J - 17.17$: 2.27, 0.34, 0.04—almost an order of magnitude per reduction of the width: a smooth policy is never exactly optimal, the sharper the cheaper.
- **But the gradient shrinks at the same rate**: at $\theta_1 = 2.12$ its norm is 4.4, 0.9, 0.10, 0.008 for $w = 1$, 0.3, 0.1, 0.03—proportional to w^2, like the excess cost.
- **Not where the policy is bad**: $\theta_1 = 20$, $w = 1$ costs 59.6 with a gradient of norm 3.7: the wear reaches every threshold in the end, and the smooth policy keeps the score alive.

The plateau is where the policy becomes *deterministic*: Part 3's phenomenon, with two parameters.

D: descent with the model, and the noise from episodes

Example 14.6, Figure 14.1 (right): step $\eta = 0.5$ on $(\theta_1, \ln w)$



Level lines of J; 200 iterates from $(6, 1)$ and from $(0.5, 1)$.

- **From** $(6, 1)$, replaced too late ($J = 24.01$): the descent first *widens* the policy ($w = 2.44$ at $k = 1$: a wider policy replaces earlier), then moves the location: $(2.42, 0.33)$ at $k = 5$, $(2.12, 0.024)$ at $k = 200$; costs 17.59 and 17.177.
- **From** $(0.5, 1)$ ($J = 26.75$): overshoot to 3.23, then the same path. Within 0.1 of the optimum after 9 and 7 iterations; the remaining 0.003 is the plateau ($w \sim k^{-1/2}$).
- **From episodes** at $(6, 1)$, $N = 100$ episodes of length 50: the REINFORCE estimate of the exact gradient $(2.50, -1.78)$ has standard deviations 1.4 and 3.2; 0.22 and 0.47 with V^{π_θ} as baseline (94% of right signs on θ_1 without it, 100% with it).

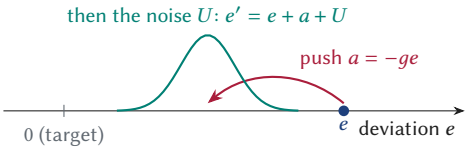
Two parameters instead of 3 001 values, the threshold *imposed*, a smooth objective; the price: never exactly optimal, a plateau at the sharp policies, noise to tame.

Worked example B: the linear-quadratic regulator

Example 14.7: the MDP (Example 13.20)

- **State**: a real number s; what matters is the deviation $e = s - \bar{s}$ from a target. **Action**: a real number a (a push).
- **Dynamics**: $S_{t+1} = S_t + A_t + U_{t+1}$, noise $U \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 1$.
- **Reward**: $-q\,e^2 - a^2$ with $q = 1$—pay for being off target and for pushing; $\gamma = 0.9$; start at $e_0 = 2$; maximize the expected discounted reward.

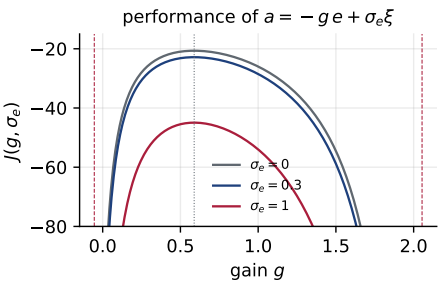
Known answer (Riccati, Ch. 13): the linear feedback $a = -g^\star e$ with $g^\star = \gamma p^\star / (1 + \gamma p^\star) = 0.588$, $p^\star = 1.588$: push back a fixed fraction of the deviation.



cost at each step: $q\,e^2 + a^2$
(off target *and* pushing hard both cost)

B: one gain, one exploration noise

Example 14.7, Figure 14.2 (left): the policy $a = -g\,e + \sigma_e \xi$, $\xi \sim \mathcal{N}(0,1)$

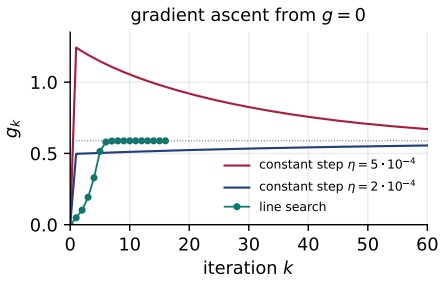


$J(g, \sigma_e)$ on the stable range (dashed boundaries); dotted: $g^\star = 0.588$.

- **One parameter**, the gain g; the noise level σ_e is kept fixed. Score: $\partial_g \ln \pi_g(a \mid e) = -(a + ge)\, e/\sigma_e^2$.
- **Stable iff** $\gamma(1-g)^2 < 1$: $g \in (-0.054,\ 2.054)$; outside, $J = -\infty$.
- **Closed form** inside: $J(g, \sigma_e) = -P(g)D(\sigma_e) - \sigma_e^2/(1-\gamma)$, $P(g)$ the quadratic coefficient of V^{π_g}, D the discounted sum of $\mathbb{E}[E_t^2]$.
- **Stationary at $g^\star = 0.588$ whatever σ_e**: exploration *costs* (the curves drop as σ_e grows) but does not *move* the optimum.
- **Concave on the stable range** (second derivative ≤ -62): the stationary point is the global maximum—the one-dimensional case of Fazel et al. (2018).

B: gradient ascent with the model

Example 14.7, Figure 14.2 (right): from $g = 0$ with $\sigma_e = 0.3$



Two constant steps and a backtracking line search.

The curvature of J ranges from $-9 \cdot 10^4$ at $g = 0$, near the unstable boundary, to -79 at $g^\star$: no single step fits both.

- $\eta = 10^{-3}$ sends $g_1 = 2.49$ out of the stable range, where $J = -\infty$.
- $\eta = 5 \cdot 10^{-4}$ overshoots to $g_1 = 1.24$, then needs 172 iterations to come within 10^{-3} of $g^\star$; $\eta = 2 \cdot 10^{-4}$ needs 276.
- **Planning**: J itself is computable, so a *backtracking line search* costs nothing and reaches $g^\star$ to six digits in 16 iterations.

From episodes, neither the line search nor the safety exists: the step is fixed in advance, the gradient is noisy, and an iterate thrown beyond 2.054 by one noisy step diverges before anything signals it.

B: the deterministic gradient as a limit, and the price of sampling the action

Example 14.7

The limit $\sigma_e \to 0$. The score-function gradient of Thm. 14.3 (numerically 45.85 against the closed form 45.87 at $g = 0.3$, $\sigma_e = 0.3$) and the *deterministic* gradient of Silver et al. (2014) have the *same integrand* (Stein's identity: Gaussian policy, quadratic Q); they differ only through the occupation measure, by the factor $D(\sigma_e)/D(0)$: 1.69, 1.06, 1.007 for $\sigma_e = 1, 0.3, 0.1$.

The price of not sampling the action: a critic whose slope $\partial_a Q$ is known. The price of sampling it: on the right.

REINFORCE at $g = 0.3$, one episode of length 100 (from 4 000 episodes); the gradient is near 45:

σ_e	1	0.3	0.1	0.03
std, baseline V^{π_g}	161	167	425	1 372
std, no baseline	397	461	1 163	4 017

- The standard deviation grows like $1/\sigma_e$, as the score does, while the gradient stays put.
- A batch of $N = 100$ episodes at $\sigma_e = 0.3$ estimates 45.9 with a standard deviation of 17; at $\sigma_e = 0.03$, with one of 137: three times the gradient.

Less exploration noise: a better policy, a worse estimate.

Part 3

Softmax policies and the natural gradient

1. The policy gradient theorem
2. Learning from episodes: REINFORCE and actor–critic
3. Softmax policies and the natural gradient
4. Trust regions: TRPO and PPO
5. Risk-sensitive policy gradients

The softmax gradient has two damping factors

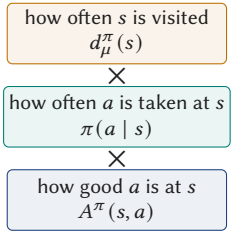
Proposition 14.8

For the tabular softmax, the theorem becomes, entry by entry,

$$\frac{\partial J}{\partial \theta(s,a)} = \frac{1}{1-\gamma}\, d_\mu^\pi(s)\; \pi(a \mid s)\; A^\pi(s,a).$$

- $\pi(a \mid s)$: a good action rarely taken is slowly promoted; near every deterministic policy the gradient vanishes (the bandit's 136 steps, the machine's w^2).
- $d_\mu^\pi(s)$: at states rarely visited, the parameters barely move, *however bad the policy is there*.

At finite θ (full-support μ), $\nabla J = 0$ forces optimality; the danger is convergence towards infinity, to a bad deterministic policy.



damping 1: a state never reached is never corrected

damping 2: a rare action is slowly promoted

the signal: the only factor we actually care about

What is guaranteed: global convergence, in words

Theorem 14.9 (Agarwal et al., 2021; Mei et al., 2020)

Theorem 14.9 (Global convergence of softmax policy gradient, in short). If the initial law μ gives positive mass to *every* state and the step is small enough ($\eta \leq (1-\gamma)^3/(8R_{\max})$), exact gradient ascent converges to the optimal value at every state, with $V^\star(\mu) - V^{\pi_{\theta_k}}(\mu) \leq C/k$.

- **A planning statement**: the gradient is exact, computed from the model.
- **The substance of the proof**: exclude convergence to a suboptimal deterministic policy at infinity; *full-support* μ keeps every state in play.
- **The constant C can be enormous**: on some MDPs the ascent needs a number of iterations exponential in the effective horizon to escape its plateaus (Li et al., 2021).
- **Outside the theorem**—μ concentrated on one start state—the ascent can stall near a suboptimal policy over any budget we tried: the hike, next.

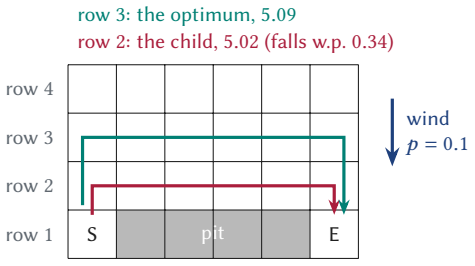
Convergent, yes; fast, no: the two damping factors are not a technicality.

Worked example C: the windy hike

Example 14.10: the MDP (Example 3.6)

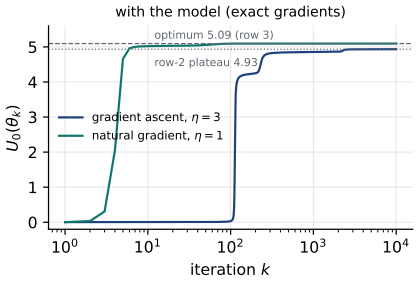
- **States**: the 20 cells of a 4×6 grid outside the pit, plus a *sink*; start at S (bottom left), bar at E (bottom right, absorbing).
- **Actions**: N, S, E, W; with probability $p = 0.1$ the **wind** blows the hiker one cell south instead. Walls block.
- **Pit**: the four bottom cells between S and E; stepping in leads to the sink, reward 0 forever.
- **Horizon** $H = 14$; **reward** 1 per step spent at the bar; maximize $U_0 = \mathbb{E}[W_{14}]$.
- **Policy class**: time-indexed softmax, $14 \times 21 \times 4$ parameters, all 0 at start (uniform).

Known answer: the optimum 5.09 follows row 3; the child's path along row 2 is worth 5.02; the mother's safe path along row 4, 2.43.



C: with the model, the ascent stalls one row too low

Example 14.10, Figure 14.3 (left): exact gradients, $\eta = 3$; natural gradient, $\eta = 1$



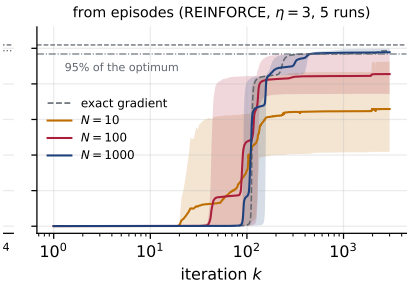
$U_0(\theta_k)$; dashed: the optimum 5.09; dotted: the row-2 plateau 4.93.

- **First plateau**: the uniform policy reaches the bar so rarely that all advantages are tiny; U_0 is still 0.03 at $k = 100$. Then a climb to 4.82 at $k = 300$ (95% of the optimum at $k = 404$).
- **Second plateau**: 4.85 at $k = 1\,000$, 4.93 at $k = 3\,000$ and at $k = 10^4$—a policy along row 2; the child's 5.02 is not even reached.
- **The mechanism is $d_t^\pi(s)$**: once the policy heads east along row 2, a trajectory visits rows 3–4 with probability 0.002; those cells keep a nearly *uniform* rule, so going north looks bad, and the exact gradient there, below $3 \cdot 10^{-6}$, cannot move them.
- **Natural gradient** ($\eta = 1$): 95% at $k = 6$, 5.09 at $k = 85$, 5.0947 from $k = 300$ on, greedy path on row 3.

An observation over a finite budget (10^6 iterations in Ex. 14.21), not a proof: $\mu = \delta_S$ is outside Thm. 14.9.

C: from episodes, the noise freezes the ascent

Example 14.10, Figure 14.3 (right): REINFORCE, N episodes per iteration, $\eta = 3$, 5 runs



Mean and range over 5 runs; each iterate's value is exact.

- $N = 1\,000$: the five runs follow the exact ascent (95% after 365 to 467 iterations), final values 4.86 to 4.99. $N = 100$: one run of five passes 95%; finals 3.72 to 4.86. $N = 10$: none; 2.09 to 4.62.
- **The honest finding**: the noise does *not* lengthen the first plateau (half the optimum at $k = 116$, 95, 90 against 113). It cuts the ascent short: with $\eta = 3$ the parameters grow fast, the final policies are deterministic where they go, and 12 of the 15 runs gain less than 0.03 between $k = 1\,000$ and $3\,000$—frozen at 4.25, 3.72 or 2.09.
- **The noise also helps, in its way**: three runs with $N = 10$ were pushed onto row 3, the right row, with crude rules there; one run with $N = 1\,000$ ends at 4.99.

Premature deterministic policies, reached by chance and never left: the plateaus of Prop. 14.8 from the learning side. A smaller step delays them.

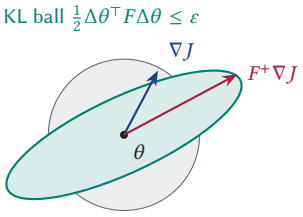
The natural gradient: measure the step in policy space

Amari (1998) and Kakade (2002)

A gradient step moves θ by a small Euclidean distance; but two parameters at the same distance may define *very different* policies, or *almost the same* one.

- Measure distances between *policies*, by the Kullback–Leibler divergence. To second order, $\mathrm{KL}(\pi_\theta(\cdot \mid s) \,\|\, \pi_{\theta'}(\cdot \mid s)) \approx \frac{1}{2}(\theta' - \theta)^\top F_s(\theta)(\theta' - \theta)$, with $F_s = \mathbb{E}_{a \sim \pi_\theta}[\text{score score}^\top]$ the **Fisher information**; averaged over $s \sim d_\mu^\pi$: $F(\theta)$.
- **Natural gradient**: $\widetilde{\nabla} J = F(\theta)^+ \nabla J$, the steepest ascent when the step is measured in policy space.

Same objective, another ruler.



along the long axis of the ellipse the policy barely changes: a long step there, a short one where the policy is sensitive

For the softmax: a multiplicative-weights update

Proposition 14.11, Corollary 14.12

One natural gradient step with the tabular softmax is, at every visited state,

$$\pi_{k+1}(a \mid s) \;=\; \frac{\pi_k(a \mid s)\, \exp\left(\frac{\eta}{1-\gamma} A^{\pi_k}(s,a)\right)}{Z_k(s)} \qquad (14.2)$$

In words: multiply each action's probability by the exponential of its advantage, renormalize.

- **Both damping factors are gone**: $d_\mu^\pi(s)$ (rare states updated as firmly as frequent ones) and $\pi(a \mid s)$ (a good action promoted by a fixed *factor*, whatever its probability).
- The Hedge / multiplicative-weights rule of online learning (Freund and Schapire, 1997), at each state on the advantages (Ch. 17).

A regularized greedy step. The new rule at s maximizes $\langle p, Q^{\pi_k}(s,\cdot)\rangle - \lambda\, \mathrm{KL}(p \,\|\, \pi_k(\cdot \mid s))$ over the simplex, with $\lambda = (1-\gamma)/\eta$: *policy iteration held back by a divergence to the previous policy.* $\eta \to \infty$ gives policy iteration (all the mass on the greedy action).

The tilts compose:
$\pi_K \propto \pi_0 \exp\left(\frac{\eta}{1-\gamma} \sum_{j<K} A^{\pi_j}\right)$, a softmax of the *sum* of past advantages.

One update, five readings

A simplified version of Table 14.1

Reading	What is maximized	What keeps the new policy close to the old
Natural gradient (parameter space)	the linearized J	the KL to second order, $\frac{1}{2}\Delta\theta^\top F \Delta\theta$
Multiplicative weights, Hedge (one state)	$\sum_a p(a) A^{\pi_k}(s,a)$	$\mathrm{KL}(p \,\|\, \pi_k(\cdot \mid s))$
Mirror descent, dual averaging	the same	the Bregman divergence of the entropy, i.e. the KL; the sum of past gradients from a fixed reference
Regularized policy iteration	$\langle p, Q^{\pi_k}(s,\cdot)\rangle$	$\lambda\,\mathrm{KL}(p \,\|\, \pi_k)$, $\lambda = (1-\gamma)/\eta$ (Ch. 15, 16)
Trust region (any parametrization)	the surrogate $L_{\pi_k}(\pi_\theta)$ of Part 4	a KL constraint (TRPO) or clipping (PPO)

- The same step, seen from optimization, online learning, dynamic programming and deep RL.
- The last line is where practice lives: with a network, F cannot be inverted, and the KL constraint is enforced otherwise.

A rate that does not depend on the size of the state space

Theorem 14.13 (Agarwal et al., 2021)

Theorem 14.13 (Convergence of the natural policy gradient, in short). From the uniform policy, with the update (14.2) applied at every state and any step $\eta > 0$, for every K and every μ:

$$V^\star(\mu) - V^{\pi_K}(\mu) \leq \frac{\ln A}{\eta K} + \frac{R_{\max}}{(1-\gamma)^2 K} \quad (A = |\mathcal{A}|).$$

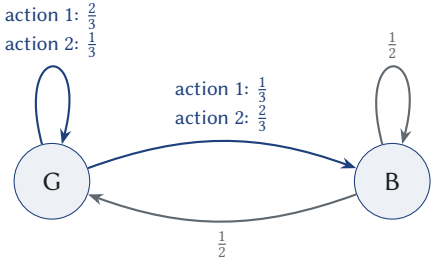
- **What is absent**: the number of states, the initial law, the transition kernel. It is the regret bound of Hedge, paid once per state along the optimal occupation measure, plus a term for the changing advantages.
- **Two ingredients of the proof**: the update improves the value at *every* state (monotone), and the gap to the optimum telescopes through the divergences to $\pi^\star$.
- **Larger steps are better**; with increasing steps, or an entropy regularization, the rate becomes *linear* (Xiao, 2022; Cen et al., 2022).
- **What it hides**: the advantages must be *known*. With estimated advantages, the mismatch between the data and the optimal policy's occupation returns (the coefficient $C^\star$ of Ch. 13).

Worked example F: the two-state chain with a choice

Example 14.14: the MDP (Example 7.12 with a second action)

- **States**: G (good) and B (bad). **Actions**: at G, action 1 moves to (G, B) with probabilities $(\frac{2}{3}, \frac{1}{3})$, action 2 with $(\frac{1}{3}, \frac{2}{3})$; at B a single action, $(\frac{1}{2}, \frac{1}{2})$.
- **Rewards** (shifted by +1 to lie in $[0, 2]$, $R_{\max} = 2$): means $r(\mathrm{G}, 1) = \frac{5}{3}$, $r(\mathrm{G}, 2) = \frac{4}{3}$, $r(\mathrm{B}) = \frac{1}{2}$.
- **Discount** $\gamma = 0.8$; μ uniform; tabular softmax.

Known answer: action 1 at G; $V^\star(\mathrm{G}) = 6.538$, $V^\star(\mathrm{B}) = 5.192$, $V^\star(\mu) = 5.865$; the wrong action loses 0.692 at G. One decision in the whole problem.

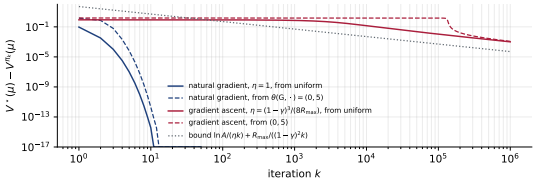


mean rewards: $\frac{5}{3}$ (action 1), $\frac{4}{3}$ (action 2) at G; $\frac{1}{2}$ at B

action 1 keeps the chain in the good state more often

F: linear convergence against a $1/k$ bound

Example 14.14, Figure 14.4



- **Natural gradient,** $\eta = 1$: the log-odds of action 1 grow by $5\Delta_k$ per step, with $\Delta_k \geq 0.53$ the gap between the two action values: the odds are multiplied by at least $e^{2.6} = 14$ per iteration. Gap 0.093, $3.3 \cdot 10^{-3}$, $1.0 \cdot 10^{-4}$, $3.2 \cdot 10^{-6}$ after 1, 2, 3, 4 iterations, machine precision from $k = 12$: linear.
- **The bound** $\ln 2/k + R_{\max}/((1-\gamma)^2 k) = 50.7/k$: 50.7, 25.3, 5.1, 1.0 at $k = 1, 2, 10, 50$. True, dimension-free, worst-case—and useless here: its second term alone exceeds the range $[0, 10]$ of the values up to $k = 5$.
- **Plain gradient** with the guaranteed step $5 \cdot 10^{-4}$: gap 10^{-1} after 13 487 iterations, 10^{-2} after 106 623; $k\times$gap stays near 1 000 up to 10^6: the C/k law, the gradient being $\propto p_k(1-p_k)$. With $\eta = 1$: 7 and 52, but 10^{-4} only after 5 003: the rate, not the constant, is the limitation.
- **Bad start** $\theta(\mathrm{G}, \cdot) = (0, 5)$: the natural gradient is two iterations later (KL distance 5.0 instead of $\ln 2$); gradient ascent needs 146 764 iterations for a gap of 10^{-1}.

Part 4

Trust regions: TRPO and PPO

1. The policy gradient theorem
2. Learning from episodes: REINFORCE and actor–critic
3. Softmax policies and the natural gradient
4. Trust regions: TRPO and PPO
5. Risk-sensitive policy gradients

The surrogate: evaluate the new policy with the old policy's data

Equation (14.5), Theorem 14.16 (Kakade and Langford, 2002; Schulman et al., 2015)

The gain of a new policy π over π_k is the *old* advantages averaged where the *new* policy goes (performance difference lemma). We only have data from the old one, hence the **surrogate**

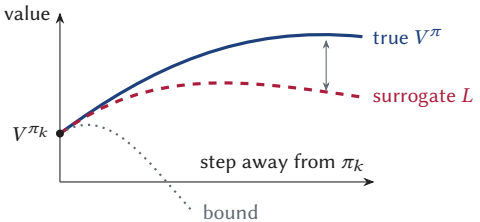
$$L_{\pi_k}(\pi) = V^{\pi_k}(\mu) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_k}}\Big[\sum_a \pi(a \mid s) A^{\pi_k}(s,a)\Big],$$

computable from the old trajectories, equal to $V^\pi(\mu)$ *to first order.*

Theorem 14.16 (Trust-region bound).

$$V^\pi(\mu) \geq L_{\pi_k}(\pi) - \frac{2\gamma\|A^{\pi_k}\|_\infty}{(1-\gamma)^2} \max_s \mathrm{KL}(\pi_k(\cdot \mid s) \,\|\, \pi(\cdot \mid s)).$$

TRPO (Schulman et al., 2015): maximize $L_{\pi_k}(\pi_\theta)$ under $\mathbb{E}_{d_\mu^{\pi_k}} \mathrm{KL}(\pi_k \,\|\, \pi_\theta) \leq \delta_{\mathrm{TR}}$: a natural gradient step, then a line search.



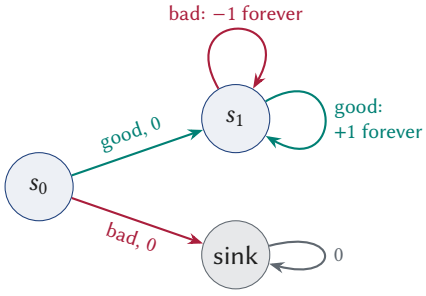
same slope at 0; they part when the new policy visits other states than the old one

Why the *maximum* over states is not conservatism

Example 14.17 (Agarwal et al., 2026, Sect. 12.2.2): the averaged constraint buys nothing

- Old policy π: bad at s_0, good at s_1. It *never* reaches s_1: $d^{\pi}_{s_0}(s_1) = 0$, $V^{\pi}(s_0) = 0$.
- New policy π': good at s_0 with probability α, bad at s_1 *always*.
- **Averaged KL** under $d^{\pi}_{s_0}$: only s_0 counts, $(1-\gamma)\ln\frac{1}{1-\alpha}$, as small as one pleases. **Maximal KL**: infinite at s_1.
- **Surrogate**: $L_{\pi}(\pi') = \alpha\gamma > 0$—reaching s_1 looks good *with π's continuation*. **Truth**: $V^{\pi'}(s_0) = -\alpha\gamma < 0$. Worse, at every radius.

A constraint measured under the old occupation says nothing about states the old policy never visits—exactly those the new one will reach. The design-law lesson of Ch. 13 again.



π takes the crimson arrow at s_0 and never sees s_1; π' sometimes takes the teal one, then plays bad there

PPO: a trust region by clipping

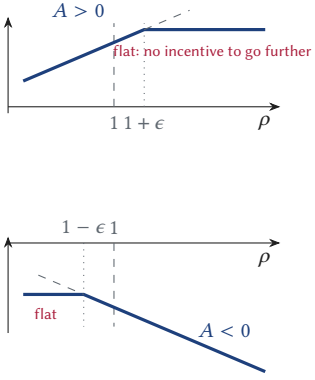
Equation (14.6) (Schulman et al., 2017)

With the ratio $\rho_\theta(s,a) = \pi_\theta(a \mid s)/\pi_k(a \mid s)$, the surrogate is $\mathbb{E}_{s\sim d^{\pi_k}, a\sim\pi_k}[\rho_\theta A^{\pi_k}]$ up to constants. PPO maximizes instead

$$\mathbb{E}\Big[\min\big(\rho_\theta A^{\pi_k},\ \mathrm{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon)A^{\pi_k}\big)\Big].$$

- **Good action** ($A > 0$): no reward for pushing ρ above $1+\epsilon$; **bad action** ($A < 0$): none for pushing it below $1-\epsilon$.
- The min keeps the *pessimistic* branch: moving ρ the *wrong* way is still penalized.

A trust region per state–action pair, enforced by removing the incentive rather than by a constraint; first-order: the workhorse of deep RL.



The clipped objective as a function of the ratio, one pair at a time (dashed: the unclipped ρA).

PPO, line by line

Algorithm 14.3

Algorithm 14.3: Proximal policy optimization (clipped surrogate)

Input: actor π_θ, critic V_w, batch T, epochs E, minibatch B, clipping ϵ_{clip}, weights c_V, c_{H}

1 **for** $k = 1, 2, \ldots$ **do**

2 run π_{θ_k} for T steps; store (S_t, A_t, R_t) and $\pi_{\theta_k}(A_t \mid S_t)$ ▷ *the denominators of the ratios*

3 estimate the advantages $\hat{A}_t$ (generalized advantage estimation with V_w) and the targets $\hat{G}_t = \hat{A}_t + V_w(S_t)$

4 **for** *E epochs, over minibatches of B steps* **do**

5 $\rho_t \leftarrow \pi_\theta(A_t \mid S_t) / \pi_{\theta_k}(A_t \mid S_t)$ ▷ *moves away from 1 as the epochs go*

6 gradient step (Adam) on $\frac{1}{B} \sum_t \left[\min(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1 \pm \epsilon_{\text{clip}}) \hat{A}_t) - c_V (V_w(S_t) - \hat{G}_t)^2 + c_{\text{H}} \mathcal{H}(\pi_\theta(\cdot \mid S_t)) \right]$

7 $\theta_{k+1} \leftarrow \theta$

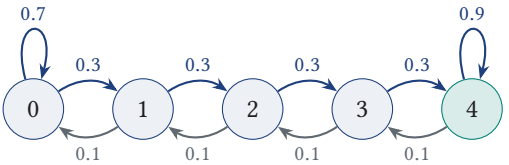
One batch, *several* passes: the data of π_{θ_k} evaluate a policy that is no longer π_{θ_k}—the surrogate in action, hence the clipping. Three terms: actor, critic, entropy bonus against premature determinism.

Worked example E: riverswim

Example 14.18: the MDP (Example 8.16)

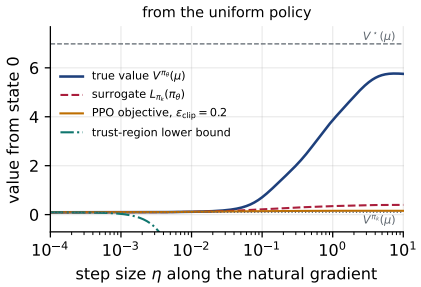
- **States**: 0, 1, 2, 3, 4 along a river; **actions**: L (with the current) and R (against it).
- **Dynamics**: L moves left for sure; R moves right w.p. 0.3, stays w.p. 0.6, is pushed back w.p. 0.1.
- **Rewards**: 1 at every transition into state 4 (self-loop included); 0.005 for L in state 0.
- **Discount** $\gamma = 0.95$ $(1/(1-\gamma) = 20)$; start in state 0; tabular softmax.

Known answer: swim right everywhere, $V^\star(0) = 6.98$. The uniform policy is worth 0.095.



E: from the uniform policy, the surrogate under-estimates

Example 14.18, Figure 14.5 (left): along the natural-gradient direction $\pi_\theta \propto \pi_k e^{\eta A^{\pi_k}/(1-\gamma)}$

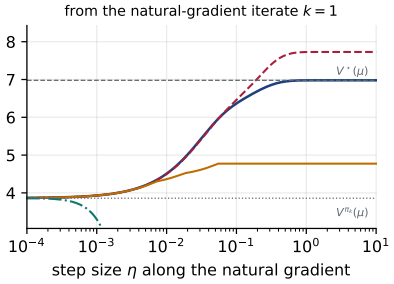


True value, surrogate, PPO objective ($\epsilon_{\text{clip}} = 0.2$), trust-region bound.

- **Where the old policy lives**: $d_\mu^{\pi_k} = (0.76, 0.18, 0.045, 0.011, 0.003)$. Advantage of R: -0.0003 in state 0 (the trickle wins), 0.44 in state 3, 0.76 in state 4.
- **Surrogate vs truth**: same slope at $\eta = 0$, within 1% up to $\eta = 0.006$ only. Then the surrogate saturates at 0.40—it can only reallocate advantages over the states it visits—while the true value climbs to 3.86 at $\eta = 1$ and 5.77 at $\eta = 7.6$: 19 times the announced improvement.
- **The bound** subtracts 579 times the largest KL over states (driven by state 4, occupation 0.003): best at $\eta = 1.2 \cdot 10^{-5}$, guaranteed gain 10^{-5} against 5.67 for the best step, loose by $5 \cdot 10^5$; equal to $-8\,400$ at $\eta = 1$.
- **PPO** is flat from $\eta = 0.47$ (the ratio of R hits 1.2 in states 4, 3, 2, 1 at $\eta = 0.013, 0.023, 0.10, 0.50$): there the true value is 2.66, 45% of the best improvement.

E: from the first iterate, the surrogate over-estimates

Example 14.18, Figure 14.5 (right): from $\pi_1 \propto e^{A^{\pi_0}/(1-\gamma)}$, swimming right w.p. $(0.50, 0.69, 0.98, 1, 1)$



$$V^{\pi_1}(0) = 3.86;$$

$$d_\mu^{\pi_1} = (0.51, 0.13, 0.08, 0.09, 0.18).$$

- **Reliable for larger steps**: within 1% up to $\eta = 0.088$ ($V = 6.29$), 5% up to 0.20; the true value reaches 6.978 at $\eta = 0.95$. **Then the surrogate overshoots** to 7.73, above $V^\star(0) = 6.98$: it credits state 0 (occupation 0.51 under π_1) with the advantage 0.19 of R as if the swimmer still returned there as often (under the new policy, $d_\mu(0) = 0.20$).
- **The bound**: best guaranteed gain 0.002 against 3.12, loose by $1.5 \cdot 10^3$. **PPO** flattens at $\eta = 0.054$ (ratio of R in state 0 at 1.2): true value 5.94, 67% of the available gain, surrogate exact to $3 \cdot 10^{-4}$: it stops *inside* the reliable region; the next iteration collects the rest.

Blind in *either* direction to states the old policy does not visit; the bound loose by 10^3 to 10^5; clipping adequate from π_1 (max KL 0.4), far too cautious from the uniform policy (6.5).

Part 5

Risk-sensitive policy gradients

1. The policy gradient theorem

2. Learning from episodes: REINFORCE and actor–critic

3. Softmax policies and the natural gradient

4. Trust regions: TRPO and PPO

5. Risk-sensitive policy gradients

Any functional of the return has a gradient of the same form

Propositions 14.19, 14.22, 14.23; Theorem 14.20

The score-function identity holds for *any* function of the trajectory. For a criterion ψ of the law of W_H, **weight the scores by the influence of the trajectory on ψ**:
$\nabla J_\psi(\theta) = \mathbb{E}\big[\mathrm{IF}_\psi(W_H)\,\sum_t \nabla \ln \pi_{\theta,t}(A_t \mid S_t)\big]$.

Three readings. Mean: $\mathrm{IF} = W_H - \mathbb{E}[W_H]$, the theorem with the mean as a free baseline. Utility: $u(W_H) - \mathbb{E}[u(W_H)]$. β-mean $\frac{1}{\beta}\ln\mathbb{E}[e^{\beta W_H}]$: proportional to $e^{\beta W_H}$—tilted weights.

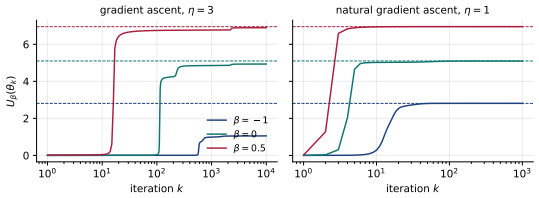
Sampled version for β-means. From N episodes, weights $w_i = e^{\beta W_H^i}/\sum_j e^{\beta W_H^j}$ (Noorani and Baras, 2021). For $\beta < 0$ they concentrate on the *worst* trajectories, few in a batch: the statistical price of risk aversion (Exercise 14.11).

What β-means have in addition. A *Bellman form*: a backward recursion (exponential moments) and a forward one (tilted occupation), assembled like the forward–backward algorithm of hidden Markov models; and a natural-gradient version, a multiplicative update on a *tilted advantage*.

CVaR (Tamar et al., 2015). Keep the $\lceil \delta N \rceil$ worst returns of the batch and apply REINFORCE to them with weights $W_H^i - \hat{q}$ ($\hat{q}$ the empirical quantile): a fraction δ of the data, variance $\propto 1/\delta$.

The hike for $\beta = -1$ and $\beta = 0.5$

Example 14.21, Figure 14.6: exact gradients of the β-means, same class and start as Example 14.10



- **The targets** (Example 5.10): optimal β-values 2.81 (row 4, the prudent mother) for $\beta = -1$ and 6.95 (row 2, the child) for $\beta = 0.5$, against 5.09 (row 3) for the mean.
- **Gradient ascent** ($\eta = 3$): the initial plateau lasts some twenty iterations for $\beta = 0.5$ and close to a thousand for $\beta = -1$; then every run settles on *row* 2: 6.89 for $\beta = 0.5$ (the right row) and 1.05 for $\beta = -1$, far from 2.81. Same mechanism as for the mean (the tilted visitation of rows 3–4 in the role of $d_t^\pi(s)$); still on row 2 after 10^5 iterations (4.94, 6.91, 1.06).
- **Natural gradient** ($\eta = 1$) reaches the optimal β-values in about a hundred iterations: 2.811 and 6.954 (5.095 for the mean), with the greedy paths on rows 4 and 2.
- **Risk aversion is the slowest** for both methods: the informative trajectories are the rare ones.

Summary

The gradient is score × value, where the policy goes. No derivative of the dynamics, none of the occupation measure. A baseline is free; the advantage form is nearly optimal. From episodes: reward-to-go (REINFORCE), or a critic's TD error (actor–critic), at the price of a bias.

Two damping factors make plain ascent slow. $\pi(a \mid s)$: a rare good action is promoted slowly, every deterministic policy is a plateau (bandit: 136 steps; machine: gradient $\propto w^2$). $d_\mu^\pi(s)$: unvisited states are never corrected (hike: row 2 instead of row 3). Global convergence holds with full-support μ, with an enormous constant.

The natural gradient removes both. Measure the step by the KL between policies: for the softmax, a multiplicative update on the advantages, with a rate $\ln A/(\eta K) + R_{\max}/((1-\gamma)^2 K)$ free of S, μ and P—and linear in practice (chain: machine precision in 12 iterations; hike: 5.09 in 85).

Trust the surrogate only near the old policy. It is blind to states the old policy does not visit (riverswim: 19× under, then above $V^\star$); the bound is true and loose by 10^3–10^5; PPO's clipping is a per-pair trust region without guarantee that works. Risk: weight the scores by $e^{\beta W}$ or keep the worst fraction; rare trajectories carry the signal.

To practice

Hands-on

H06 (`cartpole.py`): the actor–critic of Part 2 on the cart-pole (shared layer of 128 units, softmax actor, scalar critic, standardized returns, Adam with step 0.03).

- replace the standardized returns by raw returns, then remove the critic (REINFORCE with and without a running-average baseline): compare the learning curves over several seeds;
- implement the risk-sensitive REINFORCE with $\beta < 0$ on the episode length (weights through a log-sum-exp): a risk-averse agent prefers balancing reliably for a moderate time;
- on the hike, reproduce the exact-gradient runs (`hike_pg.py`, `hike_pg_mean.py`), then replace them by REINFORCE estimates and watch the ascent freeze; try a smaller step.

Exercises (Chapter 14): 14.2 (why the rewards before t and a state baseline may be dropped), 14.3 (the optimal baseline, and why V^π is good but not optimal), 14.7 (the natural gradient step by step; the two-state chain with $\eta \in \{0.3, 3, 10\}$ and 2^k), 14.6 (the bias of a wrong critic, and when it points the wrong way), 14.9 (the coupling argument behind the trust-region bound), 14.13 (the hike: full-support μ, REINFORCE with $N \in \{10, 100, 1000\}$, natural gradient vs its bound), 14.14 (the two-product store under a smoothed can-order rule: four parameters vs 256 values), 14.11 (the effective sample size of tilted weights).

References I

Agarwal, A., S. M. Kakade, J. D. Lee, and G. Mahajan (2021). "On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift". In: *Journal of Machine Learning Research* 22.98, pp. 1–76.

Agarwal, A. et al. (2026). *Reinforcement Learning: Theory and Algorithms*. Working draft of 27 June 2026.

Amari, S.-i. (1998). "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2, pp. 251–276.

Azar, M. G., V. Gómez, and H. J. Kappen (2012). "Dynamic Policy Programming". In: *Journal of Machine Learning Research* 13, pp. 3207–3245.

Borkar, V. S. (2001). "A Sensitivity Formula for Risk-Sensitive Cost and the Actor-Critic Algorithm". In: *Systems & Control Letters* 44.5, pp. 339–346.

Cen, S. et al. (2022). "Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization". In: *Operations Research* 70.4, pp. 2563–2578.

Fazel, M., R. Ge, S. M. Kakade, and M. Mesbahi (2018). "Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Vol. 80. PMLR, pp. 1467–1476.

Freund, Y. and R. E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Greensmith, E., P. L. Bartlett, and J. Baxter (2004). "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning". In: *Journal of Machine Learning Research* 5, pp. 1471–1530.

Kakade, S. (2002). "A Natural Policy Gradient". In: *Advances in Neural Information Processing Systems 14 (NIPS)*.

Kakade, S. and J. Langford (2002). "Approximately Optimal Approximate Reinforcement Learning". In: *Proceedings of the 19th International Conference on Machine Learning (ICML)*.

References II

Konda, V. R. and J. N. Tsitsiklis (2003). "On Actor-Critic Algorithms". In: *SIAM Journal on Control and Optimization* 42.4, pp. 1143–1166.

Li, G. et al. (2021). "Softmax Policy Gradient Methods Can Take Exponential Time to Converge". In: *Proceedings of the 34th Conference on Learning Theory (COLT)*. Vol. 134. PMLR.

Lillicrap, T. P. et al. (2016). "Continuous Control with Deep Reinforcement Learning". In: *International Conference on Learning Representations (ICLR)*.

Mei, J., C. Xiao, C. Szepesvári, and D. Schuurmans (2020). "On the Global Convergence Rates of Softmax Policy Gradient Methods". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 119. PMLR.

Noorani, E. and J. S. Baras (2021). "Risk-Sensitive REINFORCE: A Monte Carlo Policy Gradient Algorithm for Exponential Performance Criteria". In: *Proceedings of the 60th IEEE Conference on Decision and Control (CDC)*.

Nota, C. and P. S. Thomas (2020). "Is the Policy Gradient a Gradient?" In: *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Peters, J. and S. Schaal (2008). "Natural Actor-Critic". In: *Neurocomputing* 71.7–9, pp. 1180–1190.

Schulman, J. et al. (2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Vol. 37. PMLR.

Schulman, J. et al. (2016). "High-Dimensional Continuous Control Using Generalized Advantage Estimation". In: *International Conference on Learning Representations (ICLR)*.

Schulman, J. et al. (2017). "Proximal Policy Optimization Algorithms". arXiv:1707.06347.

Silver, D. et al. (2014). "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. Vol. 32. PMLR.

References III

Singh, S. P., T. Jaakkola, and M. I. Jordan (1994). "Learning Without State-Estimation in Partially Observable Markovian Decision Processes". In: *Proceedings of the Eleventh International Conference on Machine Learning (ICML)*, pp. 284–292.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12 (NIPS)*.

Tamar, A., Y. Glassner, and S. Mannor (2015). "Optimizing the CVaR via Sampling". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Vieillard, N. et al. (2020). "Leverage the Average: An Analysis of KL Regularization in Reinforcement Learning". In: *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

Williams, R. J. (1992). "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Machine Learning* 8.3–4, pp. 229–256.

Xiao, L. (2022). "On the Convergence Rates of Policy Gradient Methods". In: *Journal of Machine Learning Research* 23.282, pp. 1–36.