Fisher Information and Policy Gradient Methods

Karl Stratos

Last updated: March, 2026

Contents

1 Fisher Information

Let p_θ denote a distribution over discrete $\mathcal{X}$ parameterized by $\theta \in \mathbb{R}^d$.[1] For $x \in \mathcal{X}$, we write

$$l_x(\theta) := \log p_\theta(x)$$

to view the log probability of x under p_θ as a function of θ.

Lemma 1.1.

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla l_x(\theta)\right] &= 0_d \\ \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla^2 l_x(\theta)\right] &= \mathop{\mathbf{E}}_{x\sim p_\theta}\left[-\nabla l_x(\theta)\nabla l_x(\theta)^\top\right]\end{aligned}$$

Definition 1.1. The **Fisher information matrix** of θ is

$$I(\theta) := \mathop{\mathrm{Cov}}_{x\sim p_\theta}\left(\nabla l_x(\theta), \nabla l_x(\theta)\right) = \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla l_x(\theta)\nabla l_x(\theta)^\top\right] = \mathop{\mathbf{E}}_{x\sim p_\theta}\left[-\nabla^2 l_x(\theta)\right]$$

1.1 Cramér-Rao Lower Bound

Theorem 1.2. Consider any estimator $\hat{\theta} : \mathcal{X}^N \to \mathbb{R}^d$ of θ. Assuming invertible $I(\theta)$,

$$\mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\hat{\theta}(\underline{x}), \hat{\theta}(\underline{x})\right) \succeq \left(\nabla \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat{\theta}(\underline{x})\right]\right)(NI(\theta))^{-1}\left(\nabla \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat{\theta}(\underline{x})\right]\right)^\top$$

An estimator is called **efficient** if the inequality is tight. Note that the lower bound simplifies to $N^{-1}I(\theta)^{-1}$ if $\hat{\theta}$ is unbiased since the Jacobian becomes the identity matrix. Notably, maximum-likelihood estimators (MLEs) are asymptotically efficient (and unbiased), see Appendix B.

1.2 Natural Gradient

For any $\theta^* \in \mathbb{R}^d$, we write $\mathrm{KL}_{\theta^*}(\theta) = D_{\mathrm{KL}}(p_{\theta^*}||p_\theta)$ to denote the KL divergence between p_{θ^*} and p_θ viewed as a function of θ.

Lemma 1.3. For any $\theta^* \in \mathbb{R}^d$,

$$\begin{aligned}\nabla \mathrm{KL}_{\theta^*}(\theta^*) &= 0_d \\ \nabla^2 \mathrm{KL}_{\theta^*}(\theta^*) &= I(\theta^*)\end{aligned}$$

Corollary 1.4. For any $\theta^* \in \mathbb{R}^d$, for all θ infinitesimally close to θ^*

$$\mathrm{KL}_{\theta^*}(\theta) \approx \frac{1}{2}(\theta^* - \theta)^\top I(\theta^*)(\theta^* - \theta) = \frac{1}{2}\left|\left|\theta^* - \theta\right|\right|^2_{I(\theta^*)}$$

Definition 1.2. Let $J : \mathbb{R}^d \to \mathbb{R}$ be a function differentiable at θ with $I(\theta) \succ 0$. The **natural gradient** at θ is

$$\Delta_\theta^{\mathrm{NG}} := I(\theta)^{-1}\nabla J(\theta)$$

[1] A continuous extension is given immediately by replacing sum with integral and assuming mild smoothness conditions that allow interchanging the order of integration and differentiation.

Theorem 1.5. Consider any $\theta \in \mathbb{R}^d$ with $\nabla J(\theta) \neq 0_d$. Then $\Delta_\theta^{\mathrm{NG}}$ can be seen as an approximation of

$$\underset{v:\ \mathrm{KL}_\theta(\theta+v) \leq \frac{1}{2}||\nabla J(\theta)||^2_{I(\theta)^{-1}}}{\arg\max} \quad v^\top \nabla J(\theta) \tag{1}$$

(1) constrains the descent direction v by disallowing $p_{\theta+v}$ from using more than B additional bits to encode the behavior of p_θ, where $B = (1/2)\,||\nabla J(\theta)||^2_{I(\theta)^{-1}}$ can be seen as the amount of information in the gradient of J at θ. In contrast with norm-based objectives, it allows progress by B bits regardless of the coordinate system; recall that even Newton's method is only invariant to affine transformations. While $\Delta_\theta^{\mathrm{NG}}$ is only an approximation to (1), we will see that it has desirable properties when used in the context of reinforcement learning. As an aside, Adam is not natural gradient (Appendix C).

2 Policy Gradient Methods

Definition 2.1. A **Markov decision process (MDP)** is a tuple $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$ where $\mathcal{S}$ and $\mathcal{A}$ are finite sets of states and actions, τ maps any $s, a \in \mathcal{S} \times \mathcal{A}$ to a conditional distribution $\tau(\cdot|s,a)$ over $\mathcal{S}$, r maps any $s, a \in \mathcal{S} \times \mathcal{A}$ to $r(s,a) \in [0,1]$, $\gamma \in [0,1)$ is a discount factor, and μ is a distribution over $\mathcal{S}$.

Definition 2.2. A **policy** π_θ parameterized by $\theta \in \mathbb{R}^d$ maps any $s \in \mathcal{S}$ to a conditional distribution $\pi_\theta(\cdot|s)$ over $\mathcal{A}$. We write $l_{s,a}(\theta) := \log \pi_\theta(a|s)$ and assume that it is smooth.

An MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$ and a policy π_θ together define a random sequence of state-action pairs $(s_t, a_t)_{t=0}^\infty$ as follows:

- Sample an initial state $s_0 \sim \mu$.
- For $t = 0, 1, \ldots, \infty$,
  - Sample an action $a_t \sim \pi_\theta(\cdot|s_t)$ from the policy.
  - Sample a new state $s_{t+1} \sim \tau(\cdot|s_t, a_t)$ from the MDP.

Definition 2.3. Given an MDP and a policy π_θ, we define for any $s \in \mathcal{S}, a \in \mathcal{A}$ (with the convention $0^0 = 1$)

$$V_s(\theta) = \mathbf{E}\left[\sum_{t=0}^\infty \gamma^t r(s_t, a_t) \middle| s_0 = s\right] \qquad (\textbf{value} \text{ starting from } s)$$

$$Q_{s,a}(\theta) = \mathbf{E}\left[\sum_{t=0}^\infty \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a\right] \qquad (\textbf{action value} \text{ of } a \in \mathcal{A} \text{ starting from } s)$$

$$A_{s,a}(\theta) = Q_{s,a}(\theta) - V_s(\theta) \qquad (\textbf{advantage} \text{ of } a \in \mathcal{A} \text{ starting from } s)$$

$$d_s^\theta(s') = (1-\gamma)\mathbf{E}\left[\sum_{t=0}^\infty \gamma^t \left[[s_t = s']\right] \middle| s_0 = s\right] \qquad (\textbf{occupancy probability} \text{ of } s' \in \mathcal{S} \text{ starting from } s^2)$$

Given an arbitrary distribution ρ over $\mathcal{S}$, we define

$$V_\rho(\theta) = \mathop{\mathbf{E}}_{s\sim\rho}\left[V_s(\theta)\right] \qquad (\textbf{value} \text{ from state distribution } \rho)$$

$$d_\rho^\theta(s') = \mathop{\mathbf{E}}_{s\sim\rho}\left[d_s^\theta(s')\right] \qquad (\textbf{occupancy probability} \text{ of } s' \in \mathcal{S} \text{ from state distribution } \rho)$$

A **policy gradient method** takes local (gradient-based) steps on V_μ to find

$$\theta^* \in \underset{\theta\in\mathbb{R}^d}{\arg\max}\ V_\mu(\theta)$$

This typically assumes stochastic policies so that $V_\mu(\theta)$ is differentiable. In contrast, classical reinforcement learning is concerned with learning a determinstic optimal policy, which reduces to the problem of estimating the value function or the Q function (Appendix D).

Before we proceed, we give some useful identities.

[2] We can easily verify that this is indeed a distribution over $\mathcal{S}$ by the sum of the geometric series $\sum_{t=0}^\infty \gamma^t = 1/(1-\gamma)$.

Lemma 2.1. For any distribution ρ over $\mathcal{S}$, for any $g_{s,a}(\theta)$ that is a function of $s \in \mathcal{S}$, $a \in \mathcal{A}$, $\theta \in \mathbb{R}^d$

$$\mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t g_{s_t,a_t}(\theta)\right] = \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} [g_{s,a}(\theta)]$$

Corollary 2.2. For any distribution ρ over $\mathcal{S}$,

$$V_\rho(\theta) = \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} [r(s,a)]$$

Lemma 2.3 (Performance difference lemma)**.** For any distribution ρ over $\mathcal{S}$,

$$V_\rho(\theta) - V_\rho(\theta') = \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} [A_{s,a}(\theta')]$$

Lemma 2.3, often called the "performance difference lemma" (Kakade *et al.*, 2003), expresses the difference in value between two policies (θ, θ') as the expected advantage under θ' with respect to the actions proposed by θ. Note that the difference is zero for (θ, θ) since the expected advantage is zero.

2.1 Gradient Estimation

Lemma 2.4. For any $f : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$,

$$\nabla V_\mu(\theta) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(Q_{s_t,a_t}(\theta) - f(\theta, s_t)\right) \nabla l_{s_t,a_t}(\theta)\right]$$

The function f is called a "baseline function" or "control variate" and used to reduce the variance. In particular, if we set $f(\theta, s) = V_\theta(s)$ we have

$$\nabla V_\mu(\theta) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t A_{s_t,a_t}(\theta) \nabla l_{s_t,a_t}(\theta)\right] = \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\mu^\theta \\ a \sim \pi_\theta(\cdot|s)}} [A_{s,a}(\theta) \nabla l_{s,a}(\theta)] \tag{2}$$

The utility of Lemma 2.4 is that it gives an expression of the gradient amenable to Monte Carlo estimation, often referred to as REINFORCE (Williams, 1992):

REINFORCE
Input: MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$, differentiable policy π_θ, number of gradient steps R, length of sample sequence T, step size η, baseline function $f : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$
Output: gradient-based estimation of $\theta^* \in \arg\max_{\theta \in \mathbb{R}^d} V_\mu(\theta)$

1. Initialize $\theta^{(0)} \in \mathbb{R}^d$.
2. For $i = 0, 1, \ldots R-1$,
   (a) Sample $s_0 \sim \mu$ and $a_t \sim \pi_{\theta^{(i)}}(\cdot|s_t)$, $s_{t+1} \sim \tau(\cdot|s_t, a_t)$ for $t = 0 \ldots 2T$.
   (b) Calculate

$$\theta^{(i+1)} = \theta^{(i)} + \eta \left(\sum_{t=0}^{T} \gamma^t \left(\sum_{t'=0}^{T} \gamma^{t'} r(s_{t+t'}, a_{t+t'}) - f(\theta^{(i)}, s_t)\right) \nabla l_{s_t,a_t}(\theta^{(i)})\right)$$

3. Return $\theta^{(R)}$.

Note that we sample a trajectory of length $2T$ and at each step t estimate $Q_{s_t,a_t}(\theta)$ by using the next T state-action pairs. It is clear that the gradient estimator is consistent as $T \to \infty$ (using the law of iterated expectations). It is also clear that it has high variance (especially with large T). A natural way to further reduce variance is to take the average of estimates from K independent trajectories. REINFORCE is often used to optimize a degenerate MDP with no state transition (Appendix E).

Aside from high variance, a more fundamental problem with REINFORCE is that the gradient can be "flat" almost all the time. The state/action space is often large and most (s, a) pairs yield zero advantage. On the other hand, the few (s, a) pairs that do have large advantage will never be sampled. This makes the gradient zero and no learning will happen (i.e., we are stuck in a local optimum). To alleviate this one can consider various techniques, such as warm-starting the policy from another task or forcing exploration by regularization (Section 2.3).

2.2 Preconditioning Ascent Directions

It is useful to consider the following generalized form of gradient ascent: for $i = 0, 1, \dots R-1$,

$$\theta^{(i+1)} = \theta^{(i)} + \eta P_\mu(\theta^{(i)}) \nabla V_\mu(\theta^{(i)}) \tag{3}$$

where $\eta > 0$ is a constant step size and $P_\mu(\theta) \in \mathbb{R}^{d \times d}$ is a "preconditioner" for the gradient of the value function $\nabla V_\mu(\theta)$ at $\theta \in \mathbb{R}^d$. We can consider various choices of preconditioner, yielding

$$\begin{aligned}
&P_\mu(\theta) = I_{d\times d} && \text{(gradient ascent)}\\
&P_\mu(\theta) = \left(\nabla^2 V_\mu(\theta)\right)^{-1} && \text{(Newton's method)}\\
&P_\mu(\theta) = I_\mu(\theta)^+ && \text{(natural gradient ascent)}
\end{aligned}$$

where $I_\mu(\theta) \in \mathbb{R}^{d\times d}$ is the Fisher information matrix at θ for the given MDP and policy (assumed nonzero)

$$I_\mu(\theta) = \mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta \\ a\sim\pi_\theta(\cdot|s)}} \left[\nabla l_{s,a}(\theta) \nabla l_{s,a}(\theta)^\top\right]$$

The objective $V_\mu(\theta)$ is generally non-concave and either gradient ascent or Newton's method is susceptible to local optima. Remarkably, natural gradient ascent can give global convergence. Intuitively, $I_\mu(\theta)$ describes a local geometry around θ where the slope represents the policy's predictive bias. By doing a change of coordinates that removes this bias, we enforce exploration.

Compatible function approximation.

Lemma 2.5. Let $\Delta_\theta^{\mathrm{NG}} = I_\mu(\theta)^+ \nabla V_\mu(\theta)$ denote the natural gradient at θ. Then

$$\Delta_\theta^{\mathrm{NG}} \in \arg\min_{w\in\mathbb{R}^d} \mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta \\ a\sim\pi_\theta(\cdot|s)}} \left[\left(\frac{1}{1-\gamma} A_{s,a}(\theta) - w^\top \nabla l_{s,a}(\theta)\right)^2\right]$$

The lemma gives an alternative variational characterization of the natural gradient: it is the best linear predictor of the advantage where the input is the gradient. It also provides a way to compute the natural gradient by minimizing this "compatible function", which avoids computing the Fisher information matrix explicitly. It can be shown that using an approximate minimizer of the compatible function as a substitute for the natural gradient still achieves global convergence for any smooth policy (Corollary 4.23, Agarwal *et al.*, 2019).

2.2.1 Case study with the softmax policy

Definition 2.4. A policy π_θ is called **softmax** if it is parameterized by $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ as

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'\in\mathcal{A}} \exp(\theta_{s,a'})}$$

Lemma 2.6. Let π_θ be a softmax policy. There exists an MDP such that $V_\mu(\theta)$ is not concave in θ.

Lemma 2.7. Let π_θ be a softmax policy. For any distribution ρ over $\mathcal{S}$ and any $s, a \in \mathcal{S} \times \mathcal{A}$, assuming $I_\rho(\theta) \neq 0_{d\times d}$,

$$\frac{\partial}{\partial \theta_{s,a}} V_\rho(\theta) = \frac{1}{1-\gamma} d_\rho^\theta(s) \pi_\theta(a|s) A_{s,a}(\theta)$$
$$\left[I_\rho(\theta)^+ \nabla V_\rho(\theta)\right]_{s,a} = \frac{1}{1-\gamma} A_{s,a}(\theta) + c_s(\theta)$$

where $c_s(\theta)$ is some function of θ and s independent of a.

Note that the gradient of $\theta_{s,a}$ vanishes even if the advantage $A_{s,a}(\theta)$ is large if $\pi_\theta(a|s)$ is close to zero, whereas the natural gradient has a constant advantage term. We have the following global convergence result, with a state-free, sub-linear convergence rate. This is an immediate implication of Theorem A.4 in the appendix.

Corollary 2.8. Let π_θ be a softmax policy. If $\theta^{(R)}$ is the parameter value after taking R natural gradient steps from an arbitrary $\theta^{(0)}$ with some step size $\eta > 0$. Then

$$V_\mu(\theta^{(R)}) \geq V_\mu(\theta^*) - \frac{\log|\mathcal{A}|}{\eta R} - \frac{1}{(1-\gamma)^2 R}$$

In particular, for step size $\eta \geq \log|\mathcal{A}|\,(1-\gamma)^2$, given any $\epsilon > 0$ if

$$R \geq \frac{2}{(1-\eta)^2 \epsilon}$$

we have $V_\mu(\theta^{(R)}) \geq V_\mu(\theta^*) - \epsilon$.

2.3 Regularization

Regularization is a natural way to enforce exploration, thereby addressing the flat gradient problem. In fact, the following theorem states that any approximately stationary point of a KL-regularized value function V_μ is a nearly optimal maximizer of V_μ; this is despite the fact that V_μ is non-concave.

Theorem 2.9 (Theorem 4.13, Agarwal *et al.*, 2019)**.** Given $\lambda \geq 0$, define

$$L_\lambda(\theta) = V_\mu(\theta) - \lambda \mathop{\mathbf{E}}_{s\sim \mathrm{Unif}_\mathcal{S}} \left[D_{\mathrm{KL}}(\mathrm{Unif}_\mathcal{A} || \pi_\theta(\cdot|s))\right]$$

If θ satisfies $||\nabla L_\lambda(\theta)|| \leq \lambda/(2\,|\mathcal{S}|\,|\mathcal{A}|)$, then

$$V_\mu(\theta) \geq V_\mu(\theta^*) - \frac{2\lambda}{1-\gamma}$$

2.3.1 Conservative policy iteration

The performance difference lemma (Lemma 2.3) allows us to express the value of a new parameter ϕ in terms of an old one θ as

$$V_\mu(\phi) = V_\mu(\theta) + \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\phi \\ a\sim\pi_\theta(\cdot|s)}} \left[\frac{\pi_\phi(a|s)}{\pi_\theta(a|s)} A_{s,a}(\theta)\right]$$

where we additionally use importance sampling because we want to sample actions from θ rather than ϕ (i.e., off-policy instead of on-policy learning, which can be extremely inefficient). However, the *state* expectation is still with respect to the new parameter. To better exploit the old parameter, consider replacing d_μ^ϕ with d_μ^θ and define

$$V_{\mu,\theta}(\phi) := V_\mu(\theta) + \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta \\ a\sim\pi_\theta(\cdot|s)}} \left[\frac{\pi_\phi(a|s)}{\pi_\theta(a|s)} A_{s,a}(\theta)\right]$$

This turns out to be a first-order approximation of $V_\mu(\theta)$ *at* $\phi = \theta$ (Kakade and Langford, 2002): check easily that

$$V_{\mu,\theta}(\theta) = V_\mu(\theta)$$
$$\nabla V_{\mu,\theta}(\theta) = \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta \\ a\sim\pi_\theta(\cdot|s)}} \left[A_{s,a}(\theta) \nabla l_{s,a}(\theta)\right] = \nabla V_\mu(\theta)$$

TRPO

Input: MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$, differentiable policy π_θ, number of gradient steps R, length of sample sequence T, baseline function $f : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$ for estimating the advantage term, number of conjugate gradient steps G, trust region size B, line search coefficient $\alpha \in (0, 1)$ and budget L

Update: each gradient update approximately solves (4) by the relaxation (5)

1. Initialize $\theta^{(0)} \in \mathbb{R}^d$.
2. For $i = 0, 1, \dots R-1$,
   (a) Sample $s_0 \sim \mu$ and $a_t \sim \pi_{\theta^{(i)}}(\cdot|s_t)$, $s_{t+1} \sim \tau(\cdot|s_t, a_t)$ for $t = 0 \dots 2T$.
   (b) Estimate the gradient:
   $$\widehat{\nabla} V_\mu(\theta^{(i)}) = \sum_{t=0}^{T} \gamma^t \widehat{A}_{s_t,a_t}(\theta^{(i)}) \nabla l_{s_t,a_t}(\theta^{(i)})$$
   where $\widehat{A}_{s_t,a_t}(\theta^{(i)}) = \sum_{t'=0}^{T} \gamma^{t'} r(s_{t+t'}, a_{t+t'}) - f(\theta^{(i)}, s_t)$ estimates the advantage of a_t at s_t.
   (c) Estimate a symbolic mapping $\xi : u \mapsto \nabla(\nabla \mathrm{KL}_{\theta^{(i)}}(\theta^{(i)})^\top u)$ from samples, for instance in PyTorch style
   $$\xi(u) = \mathrm{grad}\left(\mathrm{grad}\left((1-\gamma)\sum_{t=0}^{2T} \gamma^t \log \frac{\pi_{\theta^{(i)}}(a_t|s_t).\mathrm{detach}()}{\pi_{\theta^{(i)}}(a_t|s_t)}, \theta^{(i)}\right)^\top u, \theta^{(i)}\right)$$
   (d) Run conjugate gradient for G steps: $\widehat{\Delta}^{\mathrm{NG}}_{\theta^{(i)}} = \mathrm{ConjugateGradient}(\xi, \widehat{\nabla} V_\mu(\theta^{(i)}), G)$.
   (e) For $l = 0, 1, \dots L$,
      i. Compute $\theta^{(i+1)} = \theta^{(i)} + \alpha^l \sqrt{2B / \widehat{\nabla} V_\mu(\theta^{(i)})^\top \widehat{\Delta}^{\mathrm{NG}}_{\theta^{(i)}}} \widehat{\Delta}^{\mathrm{NG}}_{\theta^{(i)}}$.
      ii. If
      $$\sum_{t=0}^{T} \gamma^t \frac{\pi_{\theta^{(i+1)}}(a_t|s_t)}{\pi_{\theta^{(i)}}(a_t|s_t)} \widehat{A}_{s_t,a_t}(\theta^{(i)}) \geq 0 \qquad \text{and} \qquad \sum_{t=0}^{2T} \gamma^t \log \frac{\pi_{\theta^{(i)}}(a_t|s_t)}{\pi_{\theta^{(i+1)}}(a_t|s_t)} \leq B$$
      accept the update and continue to the next gradient iteration $i + 1$.
   (f) (If this is reached, no feasible update has been found, so specify some termination protocol.)
3. Return $\theta^{(R)}$.

Figure 1: The TRPO algorithm.

In conservative policy iteration methods, at each update we use an old parameter $\theta \in \mathbb{R}^d$ and find the next parameter by optimizing this first-order approximation with a KL regularization. Specifically, let $\mathrm{KL}_\theta(\phi) := \mathbf{E}_{s \sim d^\theta_\mu}[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_\phi(\cdot|s))]$. We search for a maximizer of $V_{\mu,\theta}(\phi)$ subject to $\mathrm{KL}_\theta(\phi) \leq B$:

$$\theta' \in \underset{\phi \in \mathbb{R}^d:\ \mathrm{KL}_\theta(\phi) \leq B}{\arg\max} \ \underset{\substack{s \sim d^\theta_\mu \\ a \sim \pi_\theta(\cdot|s)}}{\mathbf{E}} \left[\frac{\pi_\phi(a|s)}{\pi_\theta(a|s)} A_{s,a}(\theta)\right] \tag{4}$$

The space of parameters satisfying the KL constraint is called the "trust region" and B is called the trust region size. Natural gradient ascent can be recovered by a first-order approximation of the objective $V_{\mu,\theta}(\phi) \approx \nabla V_\mu(\theta)^\top(\theta - \phi)$ and a second-order approximation of the KL term $\mathrm{KL}_\theta(\phi) \approx \frac{1}{2}(\theta - \phi)^\top I_\mu(\theta)(\theta - \phi)$ (Lemma 1.4) in the constraint (both at $\phi = \theta$):

$$\theta' \in \underset{\phi \in \mathbb{R}^d:\ \frac{1}{2}(\theta-\phi)^\top I_\mu(\theta)(\theta-\phi) \leq B}{\arg\max} \nabla V_\mu(\theta)^\top(\theta - \phi) \tag{5}$$

where the solution is $\theta' = \theta + \eta I_\mu(\theta)^{-1} \nabla V_\mu(\theta)$ with $\eta = \sqrt{2B / \nabla V_\mu(\theta)^\top I_\mu(\theta)^{-1} \nabla V_\mu(\theta)}$.

TRPO. TRPO (Schulman *et al.*, 2015) proposes to optimize (4), but in the end it uses the approximation (5) and is back to "just" natural gradient ascent with practical tweaks. First, note that naively computing the ascent

PPOAdaptiveKL
Input: MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$, differentiable policy π_θ, number of gradient steps R, length of sample sequence T, advantage estimator $\widehat{A}$, number of inner gradient steps E, initial KL penalty weight β_0 (e.g., 1), trust region size B
Update: each gradient update approximately solves (4) with a soft constraint

1. Initialize $\theta^{(0)} \in \mathbb{R}^d$.
2. For $i = 0, 1, \ldots R-1$,
   (a) Run $\pi_{\theta^{(i)}}$ to collect state-action pairs $(s_0, a_0) \ldots (s_T, a_T)$ and estimate advantages $\widehat{A}_{s_t,a_t}(\theta^{(i)})$.
   (b) Define

$$J(\phi) = \sum_{t=0}^{T} \gamma^t \frac{\pi_\phi(a_t|s_t)}{\pi_{\theta^{(i)}}(a_t|s_t)} \widehat{A}_{s_t,a_t}(\theta^{(i)}) + \beta_i \underbrace{(1-\gamma) \sum_{t=0}^{T} \gamma^t \log \frac{\pi_{\theta^{(i)}}(a_t|s_t)}{\pi_\phi(a_t|s_t)}}_{\widehat{\mathrm{KL}}_{\theta^{(i)}}(\phi)}$$

   (c) Take E gradient steps on $J(\phi)$ from $\phi^{(0)} = \theta^{(i)}$. Set $\theta^{(i+1)} = \phi^{(E)}$.
   (d) (Adaptive step) If $\widehat{\mathrm{KL}}_{\theta^{(i)}}(\theta^{(i+1)}) \geq 1.5B$, set $\beta_{i+1} = 2\beta_i$; if $\widehat{\mathrm{KL}}_{\theta^{(i)}}(\theta^{(i+1)}) < B/1.5$, set $\beta_{i+1} = \beta_i/2$.

Figure 2: The PPO algorithm (KL).

direction $\Delta_\theta^{\mathrm{NG}} = I_\mu(\theta)^{-1} \nabla V_\mu(\theta)$ would require $O(d^3)$ time/space and cannot scale to large models. It uses fixed-iteration conjugate gradient to approximate $\Delta_\theta^{\mathrm{NG}}$ in $O(d)$ time/space. Conjugate gradient can be viewed abstractly as a method for estimating $x = A^{-1}y$ given only a mapping $z \mapsto Az$ which dominates its runtime. In this case the mapping is particularly cheap to compute. For any function $J : \mathbb{R}^d \to \mathbb{R}$ of θ, the product of the Hessian $\nabla^2 J(\theta)$ and a vector $u \in \mathbb{R}^d$ can be written as

$$\nabla^2 J(\theta) u = \nabla \left(\nabla J(\theta)^\top u \right)$$

where the latter expression can be computed in $O(d)$ time/space.[3] Since $I_\mu(\theta) = \nabla^2 \mathrm{KL}_\theta(\theta)$ (Lemma 1.3), we can simply estimate $\mathrm{KL}_\theta(\theta)$ and provide the mapping $u \mapsto \nabla(\nabla \mathrm{KL}_\theta(\theta)^\top u)$ to conjugate gradient to estimate $\Delta_\theta^{\mathrm{NG}}$. Second, $\Delta_\theta^{\mathrm{NG}}$ is only an approximation to (4), so there is no guarantee that it will improve the objective or respect the constraint. We can enforce this by explicitly choosing a step size based on line search, that is choose the largest scalar multiplier that ensures (1) $V_{\mu,\theta}(\theta') \geq V_{\mu,\theta}(\theta)$ and (2) $\mathrm{KL}_\theta(\theta) \leq B$. The TRPO algorithm is summarized in Figure 1.

PPO. TRPO optimizes a specific approximation of the original objective and explicitly estimates the ascent direction. A much more straightforward approach is to take gradient steps directly on (4) while "softly" enforcing the constraint. This is PPO (Schulman *et al.*, 2017). The first variant of PPO includes a KL penalty in the objective and adaptively adjusts its weight based on the provided trust region size (Figure 2). The second variant of PPO is based on clipping and even simpler to implement (Figure 3). Let $\mathrm{clip}_{1-\epsilon}^{1+\epsilon} : \mathbb{R} \to [1-\epsilon, 1+\epsilon]$ denote the clipping operation for a given value of ϵ, that is $\mathrm{clip}_{1-\epsilon}^{1+\epsilon}(x) = \min\{\max\{x, 1+\epsilon\}, 1-\epsilon\}$. Instead of having an explicit KL constraint, we now optimize

$$J(\phi) = \sum_{t=0}^{T} \gamma^t \min \left\{ \frac{\pi_\phi(a_t|s_t)}{\pi_\theta(a_t|s_t)} \widehat{A}_{s_t,a_t}(\theta), \mathrm{clip}_{1-\epsilon}^{1+\epsilon} \left(\frac{\pi_\phi(a_t|s_t)}{\pi_\theta(a_t|s_t)} \right) \widehat{A}_{s_t,a_t}(\theta) \right\} \tag{6}$$

where θ is the reference parameter. Note that the objective is always a lower bound on the original objective (4). From the perspective of gradient updates, this should be understood as step-wise term dropping. At each time step t, the second argument is selected iff (i) the advantage is positive and $\pi_\phi(a_t|s_t) > (1+\epsilon)\pi_\theta(a_t|s_t)$, or (ii) the advantage is negative and $\pi_\phi(a_t|s_t) < (1-\epsilon)\pi_\theta(a_t|s_t)$. Since clipping results in a flat gradient, no update will happen when ϕ tries to excessively amplify the advantage or reduce the disadvantage, disincentivizing exploitation. As with REINFORCE, we can derive batched versions of TRPO/PPO by sampling K independent trajectories (an embarrassingly parallel extension) and taking the average.

[3] In PyTorch, it would be `grad(torch.dot(grad(J, theta, retain_graph=True)[0], u), theta)`.

PPOClipping
Input: MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$, differentiable policy π_θ, number of gradient steps R, length of sample sequence T, advantage estimator $\widehat{A}$, trust region size B, number of inner gradient steps E, clipping parameter ϵ (e.g., 0.2)
Update: each gradient update approximately solves (4) with a soft constraint

1. Initialize $\theta^{(0)} \in \mathbb{R}^d$.
2. For $i = 0, 1, \ldots R-1$,
   (a) Run $\pi_{\theta^{(i)}}$ to collect state-action pairs $(s_0, a_0) \ldots (s_T, a_T)$ and estimate advantages $\widehat{A}_{s_t,a_t}(\theta^{(i)})$.
   (b) Define

$$J(\phi) = \sum_{t=0}^{T} \gamma^t \min \left\{ \frac{\pi_\phi(a_t|s_t)}{\pi_{\theta^{(i)}}(a_t|s_t)} \widehat{A}_{s_t,a_t}(\theta^{(i)}), \text{clip}_{1-\epsilon}^{1+\epsilon} \left(\frac{\pi_\phi(a_t|s_t)}{\pi_{\theta^{(i)}}(a_t|s_t)} \right) \widehat{A}_{s_t,a_t}(\theta^{(i)}) \right\}$$

   (c) Take E gradient steps on $J(\phi)$ from $\phi^{(0)} = \theta^{(i)}$. Set $\theta^{(i+1)} = \phi^{(E)}$.

Figure 3: The PPO algorithm (clipping).

2.4 Advantage Estimation

In many reinforcement learning algorithms, we need to estimate the advantage term. Recall the definition: the advantage of action a at state s under policy π_θ is

$$A_{s,a}(\theta) = Q_{s,a}(\theta) - V_s(\theta) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a \right] - \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s \right] \tag{7}$$

where the expectations are with respect to states and actions drawn from the underlying MDP and π_θ (e.g., the old policy in conservative policy iteration).

2.4.1 Actor-critic

One approach is to introduce an additional parameter θ_v which predicts an estimate $V_{\theta_v}(s) \approx V_s(\theta)$ for any state s, and then use this to estimate the advantage. For historical reasons, the policy whose value is being predicted is called an "actor" and the value estimator is called a "critic". If $(s_0, a_0) \ldots (s_T, a_T)$ are state-action pairs drawn using policy π_θ producing a final observed state s_{T+1}, the advantage at time step $t = 0, 1, \ldots T$ can be estimated as

$$\widehat{A}_{s_t,a_t}(\theta) = \sum_{t'=0}^{T-t} \gamma^{t'} r(s_{t+t'}, a_{t+t'}) + \gamma^{T-t+1} V_{\theta_v}(s_{T+1}) - V_{\theta_v}(s_t)$$

Note the careful indexing: at every step we use only the given trajectory to estimate the corresponding advantage, while appling the critic on s_{T+1} to "bootstrap" the future (it is zero if s_{T+1} corresponds to a null terminal state, but in general may not be, e.g., due to truncation). The critic is trained along with the actor. A simplest way is to minimize a regression loss term

$$J_{\text{critic}}(\theta_v) = \sum_{t=0}^{T} \left(V_{\theta_v}(s_t) - \left(\sum_{t'=0}^{T-t} \gamma^{t'} r(s_{t+t'}, a_{t+t'}) + \gamma^{T-t+1} V_{\theta_v}(s_{T+1}) \right) \right)^2$$

2.4.2 Group relative policy optimization

While training a separate critic allows for a faithful estimation of (7), it can introduce a substantial compute overhead. But when viewing PPO (6) merely as an objective to maximize (repeated here with no discount factor)

$$\max_\phi \sum_{t=0}^{T} \min \left\{ \frac{\pi_\phi(a_t|s_t)}{\pi_\theta(a_t|s_t)} \widehat{A}_{s_t,a_t}(\theta), \text{clip}_{1-\epsilon}^{1+\epsilon} \left(\frac{\pi_\phi(a_t|s_t)}{\pi_\theta(a_t|s_t)} \right) \widehat{A}_{s_t,a_t}(\theta) \right\} \tag{8}$$

we are motivated to drop the exact definition (7) and treat $\widehat{A}_{s,a}(\theta) \in \mathbb{R}$ as a constant weight that captures the "reward profit" of performing action a in state s. Shao *et al.* (2024) make additional observations that in practice:

- We sample $N \geq 2$ iid trajectories $\{(s_0, a_0^{(j)}) \dots (s_{T_j}^{(j)}, a_{T_j}^{(j)})\}_{j=1}^N \sim \pi_\theta$ to compute a batched estimate of (8) *anyway* before making one update to ϕ.
- We typically only get a sequence-level reward.

Importantly, the trajectories are all sampled from the same initial state s_0 (i.e., prompt), making them a "group". In (outcome-supervision) *group relative* policy optimization (GRPO), the advantage estimator is

$$\widehat{A}_{s_t^{(j)}, a_t^{(j)}}(\theta) = \frac{r_j - \text{mean}_{l=1}^N(r_l)}{\text{stdev}_{l=1}^N(r_l)} \qquad r_j = \text{sequence-level reward for trajectory } j$$

for each step t in trajectory j. Note that all actions in one trajectory receive the same advantage. This estimator is practical since it does not require a critic, can be computed instantly on the fly, and yields values with zero mean and unit variance within the group.

References

Agarwal, A., Jiang, N., and Kakade, S. M. (2019). Reinforcement learning: Theory and algorithms. Technical report, Technical Report, CS Department, UW Seattle.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274.

Kakade, S. M. *et al.* (2003). *On the sample complexity of reinforcement learning*. Ph.D. thesis, University of London London, England.

Kim, Y., Rush, A. M., Yu, L., Kuncoro, A., Dyer, C., and Melis, G. (2019). Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1105–1117.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kunstner, F., Hennig, P., and Balles, L. (2019). Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, pages 4158–4169.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., *et al.* (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Tripathi, G. (1999). A matrix extension of the cauchy-schwarz inequality. *Economics Letters*, **63**(1), 1–3.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, **8**(3-4), 229–256.

A Missing Proofs and Lemmas

Proof of Lemma 1.1. The expected gradient can be written as

$$\mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla l_x(\theta)\right] = \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\frac{\nabla p_\theta(x)}{p_\theta(x)}\right] = \sum_{x\in\mathcal{X}} \nabla p_\theta(x) = \nabla \sum_{x\in\mathcal{X}} p_\theta(x) = \nabla 1 = 0_d$$

The expected Hessian can be written as

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla^2 l_x(\theta)\right] &= \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\frac{p_\theta(x)\nabla^2 p_\theta(x) - \nabla p_\theta(x)\nabla p_\theta(x)^\top}{p_\theta(x)^2}\right] \\
&= \sum_{x\in\mathcal{X}} \nabla^2 p_\theta(x) - \sum_{x\in\mathcal{X}} \frac{\nabla p_\theta(x)\nabla p_\theta(x)^\top}{p_\theta(x)} \\
&= \nabla^2 \sum_{x\in\mathcal{X}} p_\theta(x) - \sum_{x\in\mathcal{X}} \frac{p_\theta(x)^2 \nabla l_x(\theta)\nabla l_x(\theta)^\top}{p_\theta(x)} \\
&= \mathop{\mathbf{E}}_{x\sim p_\theta}\left[-\nabla l_x(\theta)\nabla l_x(\theta)^\top\right]
\end{aligned}$$

■

Proof of Theorem 1.2. Define $l_{\underline{x}}(\theta) := \log p_\theta^N(\underline{x})$. Note that

$$\mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\nabla l_{\underline{x}}(\theta)\right] = \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\nabla \sum_{i=1}^N l_{x_i}(\theta)\right] = \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\sum_{i=1}^N \nabla l_{x_i}(\theta)\right] = N \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla l_x(\theta)\right] = 0_d$$

Thus

$$\mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\hat\theta(\underline{x}), \nabla l_{\underline{x}}(\theta)\right) = \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta(\underline{x})\nabla l_{\underline{x}}(\theta)^\top\right] = \nabla \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta(\underline{x})\right]$$

The last equality can be directly checked as follows:

$$\mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta_i(\underline{x})\frac{\partial l_{\underline{x}}(\theta)}{\partial\theta_j}\right] = \sum_{\underline{x}\in\mathcal{X}^N} \frac{\partial p_\theta^N(\underline{x})}{\partial\theta_j}\hat\theta_i(\underline{x}) = \frac{\partial}{\partial\theta_j}\sum_{\underline{x}\in\mathcal{X}^N} p_\theta^N(\underline{x})\hat\theta_i(\underline{x}) = \frac{\partial}{\partial\theta_j}\mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta_i(\underline{x})\right]$$

Furthermore, using the linearity of covariance and the fact that $\underline{x}$ has iid entries,

$$\mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\nabla l_{\underline{x}}(\theta), \nabla l_{\underline{x}}(\theta)\right) = N \mathop{\mathrm{Cov}}_{x\sim p_\theta}\left(\nabla l_x(\theta), \nabla l_x(\theta)\right) = NI(\theta)$$

which is invertible since $I(\theta)$ is by premise. Thus by Cauchy-Schwarz (Corollary G.2),

$$\begin{aligned}
\mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\hat\theta(\underline{x}), \hat\theta(\underline{x})\right) &\succeq \mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\hat\theta(\underline{x}), \nabla l_{\underline{x}}(\theta)\right) \mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\nabla l_{\underline{x}}(\theta), \nabla l_{\underline{x}}(\theta)\right)^{-1} \mathop{\mathrm{Cov}}_{\underline{x}\sim p_\theta^N}\left(\nabla l_{\underline{x}}(\theta), \hat\theta(\underline{x})\right) \\
&= \left(\nabla \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta(\underline{x})\right]\right)(NI(\theta))^{-1}\left(\nabla \mathop{\mathbf{E}}_{\underline{x}\sim p_\theta^N}\left[\hat\theta(\underline{x})\right]\right)^\top
\end{aligned}$$

■

Proof of Lemma 1.3.

$$\begin{aligned}
\mathrm{KL}_{\theta^*}(\theta) &= \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[l_x(\theta^*)\right] - \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[l_x(\theta)\right] \\
\nabla\mathrm{KL}_{\theta^*}(\theta) &= \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[-\nabla l_x(\theta)\right] \\
\nabla^2\mathrm{KL}_{\theta^*}(\theta) &= \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[-\nabla^2 l_x(\theta)\right]
\end{aligned}$$

Thus at $\theta = \theta^*$, by Lemma 1.1 and Definition 1.1:

$$\begin{aligned}
\nabla\mathrm{KL}_{\theta^*}(\theta^*) &= \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[-\nabla l_x(\theta^*)\right] = 0_d \\
\nabla^2\mathrm{KL}_{\theta^*}(\theta^*) &= \mathop{\mathbf{E}}_{x\sim p_{\theta^*}}\left[-\nabla^2 l_x(\theta^*)\right] = I(\theta^*)
\end{aligned}$$

■

Proof of Theorem 1.5. Plugging in the local KL approximation in Lemma 1.4, we have the relaxed optimization problem

$$\widehat{\Delta} = \underset{v:\ ||v||_{I(\theta)} \leq ||\nabla J(\theta)||_{I(\theta)^{-1}}}{\arg\max} v^\top \nabla J(\theta)$$

This is just steepest descent with the $I(\theta)$-norm constraint, and the solution is $\widehat{\Delta} = I(\theta)^{-1}\nabla J(\theta)$.

■

Proof of Lemma 2.1.

$$\begin{aligned}
\underset{\substack{s\sim d_\rho^\theta \\ a\sim\pi_\theta(\cdot|s)}}{\mathbf{E}}[g_{s,a}(\theta)] &= \sum_{s\in\mathcal{S}} d_\rho^\theta(s) \sum_{a\in\mathcal{A}} \pi_\theta(a|s) g_{s,a}(\theta) \\
&= \sum_{s\in\mathcal{S}} (1-\gamma) \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t [[s_t = s]]\right] \sum_{a\in\mathcal{A}} \pi_\theta(a|s) g_{s,a}(\theta) \\
&= (1-\gamma) \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \sum_{s\in\mathcal{S}} [[s_t = s]] \sum_{a\in\mathcal{A}} \pi_\theta(a|s) g_{s,a}(\theta)\right] \\
&= (1-\gamma) \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \sum_{a\in\mathcal{A}} \pi_\theta(a|s_t) g_{s_t,a}(\theta)\right] \\
&= (1-\gamma) \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t g_{s_t,a_t}(\theta)\right]
\end{aligned}$$

■

Proof of Lemma 2.3.

$$\begin{aligned}
V_\rho(\theta) - V_\rho(\theta') &= \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)\right] - V_\rho(\theta') \\
&= \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \left(r(s_t,a_t) + V_{s_t}(\theta') - V_{s_t}(\theta')\right)\right] - V_\rho(\theta') \\
&\overset{*}{=} \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \left(r(s_t,a_t) + \gamma V_{s_{t+1}}(\theta') - V_{s_t}(\theta')\right)\right] \\
&= \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \left(r(s_t,a_t) + \gamma \mathbf{E}\left[V_{s_{t+1}}(\theta')|s_t,a_t\right] - V_{s_t}(\theta')\right)\right] \\
&= \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t \left(Q_{s_t,a_t}(\theta') - V_{s_t}(\theta')\right)\right] \\
&= \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t A_{s_t,a_t}(\theta')\right] \\
&= \frac{1}{1-\gamma} \underset{\substack{s\sim d_\rho^\theta \\ a\sim\pi_\theta(\cdot|s)}}{\mathbf{E}} [A_{s,a}(\theta')]
\end{aligned}$$

where the fourth equality is the tower rule and the final equality is by Lemma 2.1. To verify the equality with *, pull $V_{s_t}(\theta')$ in the expected sum outside and observe

$$\begin{aligned}
\underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^t V_{s_t}(\theta')\right] &= \underset{s_0\sim\rho}{\mathbf{E}} \left[V_{s_0}(\theta') + \sum_{t=1}^\infty \gamma^t V_{s_t}(\theta')\right] \\
&= V_\rho(\theta') + \underset{s_0\sim\rho}{\mathbf{E}} \left[\sum_{t=0}^\infty \gamma^{t+1} V_{s_{t+1}}(\theta')\right]
\end{aligned}$$

■

Proof of Lemma 2.4. Pick any $s \in \mathcal{S}$. Note that

$$V_s(\theta) = \sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q_{s,a}(\theta)$$

By the chain rule,

$$\nabla V_s(\theta) = \sum_{a \in \mathcal{A}} Q_{s,a}(\theta) \nabla \pi_\theta(a|s) + \sum_{a \in \mathcal{A}} \pi_\theta(a|s) \nabla Q_{s,a}(\theta)$$

The first term can be written as

$$\sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q_{s,a}(\theta) \nabla l_{s,a}(\theta) = \mathop{\mathbf{E}}_{a \sim \pi_\theta(\cdot|s)} [Q_{s,a}(\theta) \nabla l_{s,a}(\theta)]$$

For the second term, note that

$$\nabla Q_{s,a}(\theta) = \nabla \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a\right] = \nabla \mathop{\mathbf{E}}_{s' \sim \tau(\cdot|s,a)}\left[\sum_{t=0}^{\infty} \gamma^{t+1} r(s_t, a_t) \middle| s_0 = s'\right] = \gamma \mathop{\mathbf{E}}_{s' \sim \tau(\cdot|s,a)} [\nabla V_{s'}(\theta)]$$

Thus we can write

$$\nabla V_s(\theta) = \mathop{\mathbf{E}}_{a \sim \pi_\theta(\cdot|s)}\left[Q_{s,a}(\theta) \nabla l_{s,a}(\theta) + \gamma \mathop{\mathbf{E}}_{s' \sim \tau(\cdot|s,a)} [\nabla V_{s'}(\theta)]\right] = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t Q_{s_t,a_t}(\theta) \nabla l_{s_t,a_t}(\theta) \middle| s_0 = s\right]$$

where we get the second equality by unwinding $\nabla V_{s'}(\theta)$. Taking an expectation over $s \sim \mu$ gives

$$\nabla V_\mu(\theta) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t Q_{s_t,a_t}(\theta) \nabla l_{s_t,a_t}(\theta)\right]$$

Now, let $f : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$ be any function. Using its independence from actions,

$$\mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t f(\theta, s_t) \nabla l_{s_t,a_t}(\theta)\right] = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t f(\theta, s_t) \mathop{\mathbf{E}}_{a \sim \pi_\theta(\cdot|s_t)} [\nabla l_{s_t,a}(\theta)]\right] = 0$$

Thus

$$\nabla V_\mu(\theta) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(Q_{s_t,a_t}(\theta) - f(\theta, s_t)\right) \nabla l_{s_t,a_t}(\theta)\right]$$

■

Proof of Lemma 2.6. We construct an MDP $(\mathcal{S}, \mathcal{A}, \tau, r, \gamma, \mu)$ as follows. Let $\mathcal{S} = \{s\}$, $\mathcal{A} = \{1, 2, 3\}$, $\tau(s|s, a) = 1$ for all $a \in \mathcal{A}$, $\gamma = 0$, and $\mu(s) = 1$ with rewards

$$r(s, 1) = 1$$
$$r(s, 2) = 0$$
$$r(s, 3) = 1$$

so that

$$V_\mu(\theta) = \mathop{\mathbf{E}}_{s_0 \sim \mu}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] = \pi_\theta(1|s) + \pi_\theta(3|s)$$

Define $\theta_1 = (0, 0, -\infty)$ and $\theta_2 = (-\infty, 0, 0)$ where ∞ is an extremely large constant. Then $\bar{\theta} = (\theta_1 + \theta_2)/2 = (-\infty, 0, -\infty)$ and the corresponding policies are $\pi_{\theta_1}(\cdot|s) = (0.5, 0.5, 0)$, $\pi_{\theta_2}(\cdot|s) = (0, 0.5, 0.5)$, and $\pi_{\bar{\theta}}(\cdot|s) = (0, 0.5, 0)$ with values $V_\mu(\theta_1) = 0.5$, $V_\mu(\theta_2) = 0.5$, and $V_\mu(\bar{\theta}) = 0$. Thus we have shown that

$$\frac{V_\mu(\theta_1) + V_\mu(\theta_2)}{2} > V_\mu\left(\frac{\theta_1 + \theta_2}{2}\right)$$

■

Proof of Lemma 2.7. For any fixed $s', a' \in \mathcal{S} \times \mathcal{A}$, we have

$$\frac{\partial}{\partial \theta_{s,a}} l_{s',a'}(\theta) = [[s' = s \wedge a' = a]] - [[s' = s]]\, \pi_\theta(a|s) \tag{9}$$

for all $s, a \in \mathcal{S} \times \mathcal{A}$. By Lemma 2.4,

$$\begin{aligned}
\frac{\partial}{\partial \theta_{s,a}} V_\rho(\theta) &= \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t A_{s_t,a_t}(\theta) \frac{\partial}{\partial \theta_{s,a}} l_{s_t,a_t}(\theta)\right] \\
&= \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s \wedge a_t = a]\right] A_{s_t,a_t}(\theta)\right] - \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right] \pi_\theta(a|s) A_{s_t,a_t}(\theta)\right]
\end{aligned}$$

The first term is

$$\begin{aligned}
\mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s \wedge a_t = a]\right] A_{s_t,a_t}(\theta)\right] &= \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s \wedge a_t = a]\right] A_{s,a}(\theta)\right] \\
&= \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right] \pi_\theta(a|s) A_{s,a}(\theta)\right] \\
&= \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right]\right] \pi_\theta(a|s) A_{s,a}(\theta) \\
&= \frac{1}{1-\gamma} d_\rho^\theta(s) \pi_\theta(a|s) A_{s,a}(\theta)
\end{aligned}$$

The second term is

$$\begin{aligned}
\mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right] \pi_\theta(a|s) A_{s_t,a_t}(\theta)\right] &= \pi_\theta(a|s) \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right] A_{s_t,a_t}(\theta)\right] \\
&= \pi_\theta(a|s) \mathop{\mathbf{E}}_{s_0 \sim \rho}\left[\sum_{t=0}^{\infty} \gamma^t \left[[s_t = s]\right] \mathop{\mathbf{E}}_{a \sim \pi_\theta(\cdot|s_t)}\left[A_{s_t,a}(\theta)\right]\right] \\
&= 0
\end{aligned}$$

This shows the first equality. For the second equality, for any $w \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $s, a \in \mathcal{S} \times \mathcal{A}$, we can write

$$w^\top \nabla l_{s,a}(\theta) = w_{s,a} - c_s^w$$

where $c_s^w := \mathbf{E}_{a' \sim \pi_\theta(\cdot|s)}[w_{s,a'}]$. Thus

$$I_\rho(\theta) w = \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} \left[\nabla l_{s,a}(\theta)\, (w_{s,a} - c_s^w)\right]$$

Using (9) again, we have for any $s', a' \in \mathcal{S} \times \mathcal{A}$:

$$\begin{aligned}
[I_\rho(\theta) w]_{s',a'} &= \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} \left[\frac{\partial}{\partial \theta_{s',a'}} l_{s,a}(\theta)\, (w_{s,a} - c_s^w)\right] \\
&= \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^\theta \\ a \sim \pi_\theta(\cdot|s)}} \left[[[s = s' \wedge a = a']]\, (w_{s,a} - c_s^w)\right] - \pi_\theta(a'|s') \mathop{\mathbf{E}}_{s \sim d_\rho^\theta}\left[[[s = s']] \underbrace{\mathop{\mathbf{E}}_{a \sim \pi_\theta(\cdot|s)}\left[(w_{s,a} - c_s^w)\right]}_{0}\right] \\
&= d_\rho^\theta(s') \pi_\theta(a'|s') (w_{s',a'} - c_{s'}^w)
\end{aligned}$$

By Lemma F.1, $I_\rho(\theta)^+\nabla V_\rho(\theta) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the unique minimizing w of

$$\begin{aligned}
||\nabla V_\rho(\theta) - I_\rho(\theta)w||^2 &= \sum_{s\in\mathcal{S},\ a\in\mathcal{A}} \left([\nabla V_\rho(\theta)]_{s,a} - [I_\rho(\theta)w]_{s,a}\right)^2 \\
&= \sum_{s\in\mathcal{S},\ a\in\mathcal{A}} \left(\frac{1}{1-\gamma} d_\rho^\theta(s)\pi_\theta(a|s)A_{s,a}(\theta) - d_\rho^\theta(s)\pi_\theta(a|s)(w_{s,a} - c_s^w)\right)^2 \\
&= \sum_{s\in\mathcal{S},\ a\in\mathcal{A}} \left(d_\rho^\theta(s)\pi_\theta(a|s)\left(\frac{1}{1-\gamma}A_{s,a}(\theta) + c_s^w - w_{s,a}\right)\right)^2 \geq 0
\end{aligned}$$

Hence $I_\rho(\theta)^+\nabla V_\rho(\theta)$ satisfies: for every $s, a \in \mathcal{S}\times\mathcal{A}$,

$$[I_\rho(\theta)^+\nabla V_\rho(\theta)]_{s,a} = \frac{1}{1-\gamma}A_{s,a}(\theta) + c_s^{I_\rho(\theta)^+\nabla V_\rho(\theta)}$$

■

Corollary A.1. Let π_θ be a softmax policy. For any distribution ρ over $\mathcal{S}$, the natural gradient update at $\theta^{(i)}$,

$$\theta^{(i+1)} = \theta^{(i)} + \eta I_\rho(\theta^{(i)})^+\nabla V_\rho(\theta^{(i)})$$

is equivalent to the multiplicative update

$$\pi_{\theta^{(i+1)}}(a|s) = \pi_{\theta^{(i)}}(a|s)\frac{1}{Z_s(\theta^{(i)})}\exp\left(\frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)})\right) \tag{10}$$

where $Z_s(\theta^{(i)}) = \sum_{a'\in\mathcal{A}}\pi_{\theta^{(i)}}(a'|s)\exp(\eta/(1-\gamma)A_{s,a'}(\theta^{(i)}))$.

Proof of Corollary A.1 By Lemma 2.7,

$$\theta_{s,a}^{(i+1)} = \theta_{s,a}^{(i)} + \frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)}) + \eta c_s(\theta^{(i)})$$

for some $c_s(\theta^{(i)})$. Exponentiating both sides and diving by the sum over $\mathcal{A}$,

$$\begin{aligned}
\pi_{\theta^{(i+1)}}(a|s) = \frac{\exp\left(\theta_{s,a}^{(i+1)}\right)}{\sum_{a'\in\mathcal{A}}\exp\left(\theta_{s,a'}^{(i+1)}\right)} &= \frac{\exp\left(\theta_{s,a}^{(i)}\right)\exp\left(\frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)})\right)\exp\left(\eta c_s(\theta^{(i)})\right)}{\sum_{a'\in\mathcal{A}}\exp\left(\theta_{s,a'}^{(i)}\right)\exp\left(\frac{\eta}{1-\gamma}A_{s,a'}(\theta^{(i)})\right)\exp\left(\eta c_s(\theta^{(i)})\right)} \\
&= \frac{\exp\left(\theta_{s,a}^{(i)}\right)\exp\left(\frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)})\right)}{\sum_{a'\in\mathcal{A}}\exp\left(\theta_{s,a'}^{(i)}\right)\exp\left(\frac{\eta}{1-\gamma}A_{s,a'}(\theta^{(i)})\right)} \\
&= \frac{\frac{\exp\left(\theta_{s,a}^{(i)}\right)}{\sum_{a''\in\mathcal{A}}\exp\left(\theta_{s,a''}^{(i)}\right)}\exp\left(\frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)})\right)}{\sum_{a'\in\mathcal{A}}\frac{\exp\left(\theta_{s,a'}^{(i)}\right)}{\sum_{a''\in\mathcal{A}}\exp\left(\theta_{s,a''}^{(i)}\right)}\exp\left(\frac{\eta}{1-\gamma}A_{s,a'}(\theta^{(i)})\right)} \\
&= \pi_{\theta^{(i)}}(a|s)\frac{1}{Z_s(\theta^{(i)})}\exp\left(\frac{\eta}{1-\gamma}A_{s,a}(\theta^{(i)})\right)
\end{aligned}$$

■

Lemma A.2. Let π_θ be a softmax policy. For any distribution ρ over $\mathcal{S}$, if $\theta^{(i+1)} = \theta^{(i)} + \eta I_\rho(\theta^{(i)})^+\nabla V_\rho(\theta^{(i)})$, then

$$V_\rho(\theta^{(i+1)}) - V_\rho(\theta^{(i)}) \geq \frac{1-\gamma}{\eta}\mathop{\mathbf{E}}_{s\sim\rho}\left[\log Z_s(\theta^{(i)})\right] \geq 0$$

where $Z_s(\theta^{(i)}) = \sum_{a'\in\mathcal{A}}\pi_{\theta^{(i)}}(a'|s)\exp(\eta/(1-\gamma)A_{s,a'}(\theta^{(i)}))$.

Proof of Lemma A.2 The nonnegativity follows since for any $s \in \mathcal{S}$ and θ,

$$\log Z_s(\theta) = \log \mathop{\mathbf{E}}_{a' \sim \pi_\theta(\cdot|s)} \left[\exp\left(\frac{\eta}{1-\gamma} A_{s,a'}(\theta) \right) \right] \geq \frac{\eta}{1-\gamma} \mathop{\mathbf{E}}_{a' \sim \pi_\theta(\cdot|s)} \left[A_{s,a'}(\theta) \right] = 0$$

For the first inequality, note that by (10)

$$A_{s,a}(\theta^{(i)}) = \frac{1-\gamma}{\eta} \log \frac{\pi_{\theta^{(i+1)}}(a|s) Z_s(\theta^{(i)})}{\pi_{\theta^{(i)}}(a|s)} \tag{11}$$

Thus by the performance difference lemma (Lemma 2.3),

$$\begin{aligned}
V_\rho(\theta^{(i+1)}) - V_\rho(\theta^{(i)}) &= \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^{\theta^{(i+1)}} \\ a \sim \pi_{\theta^{(i+1)}}(\cdot|s)}} \left[A_{s,a}(\theta^{(i)}) \right] \\
&= \frac{1}{\eta} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^{\theta^{(i+1)}} \\ a \sim \pi_{\theta^{(i+1)}}(\cdot|s)}} \left[\log \frac{\pi_{\theta^{(i+1)}}(a|s)}{\pi_{\theta^{(i)}}(a|s)} \right] + \frac{1}{\eta} \mathop{\mathbf{E}}_{s \sim d_\rho^{\theta^{(i+1)}}} \left[\log Z_s(\theta^{(i)}) \right] \\
&\geq \frac{1}{\eta} \mathop{\mathbf{E}}_{s \sim d_\rho^{\theta^{(i+1)}}} \left[\log Z_s(\theta^{(i)}) \right] \\
&\geq \frac{1-\gamma}{\eta} \mathop{\mathbf{E}}_{s \sim \rho} \left[\log Z_s(\theta^{(i)}) \right]
\end{aligned}$$

where the last inequality follows by the definition of occupancy probability: for any ρ and θ,

$$d_\rho^\theta(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_{\substack{s_0 \sim \rho \\ a_t \sim \pi_\theta(\cdot|s_t)}} (s_t = s) \geq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \rho(s) = \rho(s)$$

■

The following corolloary is due to the fact that $V_\rho(\theta^{(i+1)}) \geq V_\rho(\theta^{(i)})$ for all t by Lemma A.2.

Corollary A.3. Let π_θ be a softmax policy. For any distribution ρ over $\mathcal{S}$, if $\theta^{(i+1)} = \theta^{(i)} + \eta I_\rho(\theta^{(i)})^+ \nabla V_\rho(\theta^{(i)})$, then for any $R > 0$,

$$V_\rho(\theta^{(R)}) \geq V_\rho(\theta^{(R-1)}) \geq \frac{1}{R} \sum_{i=0}^{R-1} V_\rho(\theta^{(i)})$$

Theorem A.4. Let π_θ be a softmax policy. Let ρ be any distribution over $\mathcal{S}$. If $\theta^{(R)}$ is the parameter value after taking R natural gradient steps from an arbitrary $\theta^{(0)}$ with some step size $\eta > 0$, for any $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$

$$V_\rho(\theta^{(R)}) \geq V_\rho(\theta) - \frac{\log |\mathcal{A}|}{\eta R} - \frac{1}{(1-\gamma)^2 R}$$

Proof of Theorem A.4 We upper bound $V_\rho(\theta) - V_\rho(\theta^{(R)})$. Using Corollary A.3,

$$V_\rho(\theta) - V_\rho(\theta^{(R)}) \leq \frac{1}{R} \sum_{i=0}^{R-1} V_\rho(\theta) - V_\rho(\theta^{(i)}) \tag{12}$$

For each difference in the sum, apply the performance difference lemma (Lemma 2.3) and (11):

$$\begin{aligned}
V_\rho(\theta) - V_\rho(\theta^{(i)}) &= \frac{1}{1-\gamma} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^{\theta} \\ a \sim \pi_\theta(\cdot|s)}} \left[A_{s,a}(\theta^{(i)}) \right] \\
&= \frac{1}{\eta} \mathop{\mathbf{E}}_{\substack{s \sim d_\rho^{\theta} \\ a \sim \pi_\theta(\cdot|s)}} \left[\log \frac{\pi_{\theta^{(i+1)}}(a|s)}{\pi_{\theta^{(i)}}(a|s)} \right] + \frac{1}{\eta} \mathop{\mathbf{E}}_{s \sim d_\rho^{\theta}} \left[\log Z_s(\theta^{(i)}) \right]
\end{aligned}$$

Plugging this in (12), the upper bound is the sum of two terms. The first term is

$$\begin{aligned}
\frac{1}{\eta R}\sum_{i=0}^{R-1}\mathop{\mathbf{E}}_{\substack{s\sim d_\rho^\theta\\ a\sim\pi_\theta(\cdot|s)}}\left[\log\frac{\pi_{\theta^{(i+1)}}(a|s)}{\pi_{\theta^{(i)}}(a|s)}\right] &= \frac{1}{\eta R}\sum_{i=0}^{R-1}\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_{\theta^{(i)}}(a|s))\right]-\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_{\theta^{(i+1)}}(a|s))\right]\\
&= \frac{1}{\eta R}\left(\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_{\theta^{(0)}}(a|s))\right]-\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_{\theta^{(R)}}(a|s))\right]\right)\\
&\le \frac{1}{\eta R}\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot|s)||\pi_{\theta^{(0)}}(a|s))\right]\\
&\le \frac{\log|\mathcal{A}|}{\eta R}
\end{aligned}$$

The second term is, by Lemma A.2,

$$\begin{aligned}
\frac{1}{\eta R}\sum_{i=0}^{R-1}\mathop{\mathbf{E}}_{s\sim d_\rho^\theta}\left[\log Z_s(\theta^{(i)})\right] &\le \frac{1}{(1-\gamma)R}\sum_{i=0}^{R-1}V_{d_\rho^\theta}(\theta^{(i+1)})-V_{d_\rho^\theta}(\theta^{(i)})\\
&= \frac{1}{(1-\gamma)R}\left(V_{d_\rho^\theta}(\theta^{(R)})-V_{d_\rho^\theta}(\theta^{(0)})\right)\\
&\le \frac{1}{(1-\gamma)R}V_{d_\rho^\theta}(\theta^{(R)})\\
&\le \frac{1}{(1-\gamma)^2R}
\end{aligned}$$

■

What is mind-bending about the proof of Theorem A.4 is that we are not using any special property of the target θ. The convergence follows from the fact that the natural gradient updates never decrease the value function (Lemma A.2), and the specific form of the updates allows us to (1) write the advantage (which we get by invoking the performance difference lemma) as a difference between consecutive KL terms plus a log partition function, (2) upper bound the log partition function as a difference between consecutive values—no matter what state distribution we consider (Lemma A.2). The difference of consecutive terms leads to telescoping cancellation.

Proof of Lemma 2.5. The objective

$$J(w)=\mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta\\ a\sim\pi_\theta(\cdot|s)}}\left[\left(\frac{1}{1-\gamma}A_{s,a}(\theta)-w^\top\nabla l_{s,a}(\theta)\right)^2\right]$$

is convex in w, so we can solve for a stationary point that satisfies

$$\begin{aligned}
\nabla J(w) &= -2\mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta\\ a\sim\pi_\theta(\cdot|s)}}\left[\left(\frac{1}{1-\gamma}A_{s,a}(\theta)-w^\top\nabla l_{s,a}(\theta)\right)\nabla l_{s,a}(\theta)\right]\\
&= -2\left(\underbrace{\frac{1}{1-\gamma}\mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta\\ a\sim\pi_\theta(\cdot|s)}}\left[A_{s,a}(\theta)\nabla l_{s,a}(\theta)\right]}_{\nabla V_\mu(\theta)}-\underbrace{\mathop{\mathbf{E}}_{\substack{s\sim d_\mu^\theta\\ a\sim\pi_\theta(\cdot|s)}}\left[\nabla l_{s,a}(\theta)\nabla l_{s,a}(\theta)^\top\right]}_{I_\mu(\theta)}w\right)=0_d
\end{aligned}$$

where the first underbrace is by (2). Solving for w gives

$$w=I_\mu(\theta)^+\nabla V_\mu(\theta)$$

■

B Maximum Likelihood Estimators (MLEs)

Let $\Theta \subseteq \mathbb{R}^d$ denote a parameter space specifying an associated set of distributions p_θ over $\mathcal{X}$ for $\theta \in \Theta$. The usual statistical setting is that there is some target distribution p_{θ_0} we want to estimate by drawing samples from it. Given N iid samples $\underline{x} \sim p_{\theta_0}^N$, an MLE is defined as

$$\hat{\theta}_{\text{MLE}}(\underline{x}) \in \arg\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i)$$

An immediate property from cross entropy minimization is that as $N \to \infty$ we have $p_{\hat{\theta}_{\text{MLE}}(\underline{x})} = p_{\theta_0}$. However, we can endow stronger properties by assuming certain conditions, for example an identifiability condition that says $\theta = \theta'$ iff $p_\theta = p_{\theta'}$ and other regularity conditions. With such conditions, it is possible to give the following extremely strong statement:

Theorem B.1 (Asymptotic optimality of MLEs)**.** With mild regularity conditions, as N tends to infinity in $\underline{x} \sim p_{\theta_0}^N$

$$\hat{\theta}_{\text{MLE}}(\underline{x}) \xrightarrow{d} \mathcal{N}(\theta_0, N^{-1} I^{-1}(\theta))$$

Thus an MLE is normally distributed with mean θ_0 and covariance $N^{-1}I^{-1}(\theta)$ where the randomness is with respect to N iid samples, as $N \to \infty$. This in particular means that it is unbiased and efficient (Theorem 1.2). Non-asymptotically this is not necessarily the case. For example it can be biased with finite sample (Example B.2). But an MLE is usually difficult to outperform even with finite sample unless we exploit a special property of the model structure (e.g., method of moments).

Example B.1. Let $\Theta = (0, 1)$ and $p_\theta = \text{Ber}(\theta)$ over $\mathcal{X} = \{0, 1\}$, that is

$$p_\theta(x) = \theta^x (1-\theta)^{1-x}$$

Denoting $l_x(\theta) := \log p_\theta(x) = x \log \theta + (1-x) \log(1-\theta)$, we have

$$l_x'(\theta) = x\theta^{-1} - (1-x)(1-\theta)^{-1}$$
$$l_x''(\theta) = -x\theta^{-2} - (1-x)(1-\theta)^{-2}$$

and thus the Fisher information "matrix" is given by

$$I(\theta) := \mathop{\mathbf{E}}_{x \sim p_\theta} [-l_x''(\theta)] = -\theta \times l_0''(\theta) - (1-\theta) \times l_1''(\theta) = \frac{1}{1-\theta} + \frac{1}{\theta} = \frac{1}{\theta(1-\theta)}$$

Now consider the MLE objective given $\underline{x} \in \{0, 1\}^N$:

$$J(\theta) = \sum_{i=1}^{N} x_i \log \theta + (1 - x_i) \log(1-\theta)$$

This is a sum of concave functions[4] and hence concave, so to find a maximizer we can just fine a stationary point satisfying $J'(\theta) = 0$ or

$$\frac{\sum_{i=1}^N x_i}{\theta} = \frac{\sum_{i=1}^N 1 - x_i}{1-\theta} \Leftrightarrow \frac{1-\theta}{\theta} = \frac{\sum_{i=1}^N 1 - x_i}{\sum_{i=1}^N x_i} \Leftrightarrow \frac{1}{\theta} = \frac{\sum_{i=1}^N 1 - x_i + \sum_{i=1}^N x_i}{\sum_{i=1}^N x_i} \Leftrightarrow \theta = \frac{\sum_{i=1}^N x_i}{N}$$

Thus $\hat{\theta}_{\text{MLE}}(\underline{x}) = (1/N)(\sum_{i=1}^N x_i)$. It is non-asymptotically unbiased; for any target $\theta_0 \in (0, 1)$

$$\mathop{\mathbf{E}}_{\underline{x} \sim p_{\theta_0}^N} \left[\hat{\theta}_{\text{MLE}}(\underline{x})\right] = \mathop{\mathbf{E}}_{\underline{x} \sim p_{\theta_0}^N} \left[\frac{\sum_{i=1}^N x_i}{N}\right] = \mathop{\mathbf{E}}_{x \sim p_{\theta_0}} [x] = \theta_0$$

It is also non-asymptotically efficient:

$$\mathop{\text{Var}}_{\underline{x} \sim p_{\theta_0}^N} \left(\hat{\theta}_{\text{MLE}}(\underline{x})\right) = \mathop{\text{Var}}_{\underline{x} \sim p_{\theta_0}^N} \left(\frac{\sum_{i=1}^N x_i}{N}\right) = \frac{1}{N} \mathop{\text{Var}}_{x \sim p_{\theta_0}} (x) = \frac{\theta(1-\theta)}{N} = \frac{1}{NI(\theta)}$$

[4] $\log(1-\theta)$ is concave it is a composition of a concave (log) and an affine function.

Example B.2. Let $\Theta = \mathbb{R} \times (0, \infty)$ and $p_\theta = \mathcal{N}(\theta_1, \theta_2)$ Gaussian with mean θ_1 and variance θ_2 over $\mathcal{X} = \mathbb{R}$:

$$p_\theta(x) = \frac{1}{\sqrt{2\pi\theta_2}} \exp\left(-\frac{(x-\theta_1)^2}{2\theta_2}\right)$$

We have

$$l_x(\theta_1, \theta_2) = -\frac{(x-\theta_1)^2}{2\theta_2} - \frac{1}{2}\log\theta_2 - \frac{1}{2}\log(2\pi)$$

This is clearly concave in θ_1. It is *not* concave in θ_2.[5] But we can use the change of variable $z := 1/\theta_2$ which yields

$$l_x(\theta_1, z) = -\frac{(x-\theta_1)^2 z}{2} + \frac{1}{2}\log z - \frac{1}{2}\log(2\pi)$$

This is now clearly concave in z. Once we estimate z, we can recover $\theta_2 = 1/z$. With this in mind, let us consider the MLE objective given $\underline{x} \in \mathbb{R}^N$ using the change of variable and ignoring the constant:

$$J(\theta_1, z) = \frac{N}{2}\log z - \sum_{i=1}^N \frac{(x_i-\theta_1)^2 z}{2}$$

Since this is concave in θ_1 and z, we can identify maximizers by finding stationary points:

$$\frac{\partial}{\partial\theta_1} J(\theta_1, z) = \sum_{i=1}^N (x_i - \theta_1) z = 0 \qquad \Leftrightarrow \qquad \theta_1 = \frac{\sum_{i=1}^N x_i}{N}$$

$$\frac{\partial}{\partial z} J(\theta_1, z) = \frac{N}{2z} - \sum_{i=1}^N \frac{(x_i-\theta_1)^2}{2} = 0 \qquad \Leftrightarrow \qquad z = \frac{N}{\sum_{i=1}^N (x_i-\theta_1)^2}$$

The stationary point $\theta_1 = (1/N)\sum_{i=1}^N x_i$ does not depend on z, thus the MLE for $\theta_2 = 1/z$ is

$$\hat{\theta}_{2,\mathrm{MLE}}(\underline{x}) = \frac{1}{N}\sum_{i=1}^N \left(x_i - \frac{\sum_{j=1}^N x_j}{N}\right)^2$$

Direct calculation shows that

$$\mathop{\mathbf{E}}_{\underline{x}\sim p_\theta}\left[\hat{\theta}_{2,\mathrm{MLE}}(\underline{x})\right] = \left(\frac{N-1}{N}\right)\theta_2$$

Thus the MLE is non-asymptotically biased. However, we see that the bias vanishes as N becomes larger.

C Adam is Not Natural Gradient

It is now standard to use Adam (Kingma and Ba, 2014) to estimate a model p_θ of **pop** by minimizing cross entropy $J(\theta) := \mathbf{E}_{x\sim\mathbf{pop}}[-l_x(\theta)]$. The Adam update is

$$\theta_t = \theta_{t-1} - \eta \mathrm{diag}\left(\mathop{\widehat{\mathbf{E}}}_{x\sim\mathbf{pop}}[\nabla l_x(\theta) \odot \nabla l_x(\theta)]\right)^{-1/2} \mathop{\widehat{\mathbf{E}}}_{x\sim\mathbf{pop}}[\nabla l_x(\theta)]$$

where $\odot$ is the element-wise multiplication and the hatted expectations are empircal estimates based on moving average. Specifically, at step t with $x_t \sim \mathbf{pop}$ we estimate $\mathbf{E}_{x\sim\mathbf{pop}}[\nabla l_x(\theta) \odot \nabla l_x(\theta)]$ by

$$\hat{v}_t = \frac{1}{1-0.999^t}\left(0.999\hat{v}_{t-1} + 0.001\nabla l_{x_t}(\theta) \odot \nabla l_{x_t}(\theta)\right) + 0.00000001$$

[5] $l_x(\theta_1, \theta_2)$ is concave at θ_2 iff

$$\frac{\partial^2}{\partial^2\theta_2} l_x(\theta_1, \theta_2) = -\frac{(x-\theta_1)^2}{\theta_2^2} + \frac{1}{2\theta_2^2} \le 0 \qquad \Leftrightarrow \qquad \theta_2^2 \le 2(x-\theta_1)^2$$

otherwise convex.

where $\hat{v}_0 = 0_d$. The estimate for $\mathbf{E}_{x\sim\mathbf{pop}}[\nabla l_x(\theta)]$ is similar. Despite superficial similarity to the natural gradient $I(\theta)^{-1}\nabla J(\theta)$, there are many differences such as using a diagonal approximation of the outer product and taking the inverse square root. But the most important distinction is

$$\mathop{\mathbf{E}}_{x\sim\mathbf{pop}}\left[\nabla l_x(\theta)\nabla l_x(\theta)^\top\right] \neq \mathop{\mathbf{E}}_{x\sim p_\theta}\left[\nabla l_x(\theta)\nabla l_x(\theta)^\top\right] = I(\theta)$$

Thus the preconditioning term cannot be called a diagonal approximation of the Fisher matrix; see Kunstner *et al.* (2019) for more discussions. It seems best to view Adam simply as what it is: for each parameter θ_i, subtract a bias-corrected moving average of its gradient normalized by a bias-corrected (and smoothed) moving average of its uncentered standard deviation.

D Classical Reinforcement Learning

Classical reinforcement learning considers the problem of finding an optimal deterministic policy π^* in a finite MDP (so the only randomness, if any, comes from state transitions). We can then reduce the problem to estimating a table of maximal expected future rewards, either $V^* \in \mathbb{R}^{|\mathcal{S}|}$ or $Q^* \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ where

$$\begin{aligned} V^*_s &:= \max_{\pi:\mathcal{S}\to\mathcal{A}} \mathbf{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0 = s\right] &&= \max_{a\in\mathcal{A}} r(s,a) + \gamma \mathop{\mathbf{E}}_{s'\sim\tau(\cdot|s,a)}[V^*_{s'}] \\ Q^*_{s,a} &:= \max_{\pi:\mathcal{S}\to\mathcal{A}} \mathbf{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0 = s, a_0 = a\right] &&= r(s,a) + \gamma \mathop{\mathbf{E}}_{s'\sim\tau(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}} Q^*_{s',a'}\right] \end{aligned}$$

because we can easily extract π^* from these tables by $\pi^*(s) = \arg\max_a Q^*_{s,a} = \arg\max_a r(s,a) + \gamma\mathbf{E}_{s'\sim\tau(\cdot|s,a)}[V^*_{s'}]$. Each of the second equalities follows from the definition of optimality and provides a *fixed-point* characterization of the optimal table under some transformation Φ. The transformation is γ-contracting (easy to check): $||\Phi(U) - \Phi(V)||_\infty \leq \gamma ||U - V||_\infty$. If U^* is a fixed-point under a γ-contracting transformation Φ, we can recover U^* by arbitrarily initializing $U^{(0)}$ and iterating $U^{(t+1)} = \Phi(U^{(t)})$ since

$$\left|\left|U^{(t)} - U^*\right|\right|_\infty = \left|\left|\Phi(U^{(t-1)}) - \Phi(U^*)\right|\right|_\infty \leq \gamma \left|\left|U^{(t-1)} - U^*\right|\right|_\infty \leq \gamma^t \left|\left|U^{(0)} - U^*\right|\right|_\infty$$

Thus $U^{(t)} \to U^*$ as $t \to \infty$. A similar iterative scheme can be used to recover the value function associated with a particular policy π since

$$V_s(\pi) := \mathbf{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)|s_0 = s\right] = \mathop{\mathbf{E}}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \mathop{\mathbf{E}}_{s'\sim\tau(\cdot|s,a)}[V_{s'}(\pi)]\right] \tag{13}$$

and thus $V(\pi) \in \mathbb{R}^{|\mathcal{S}|}$ is again a stationary point of some γ-contracting transformation Φ.[6]

D.1 Value Iteration and Q-Iteration

Given the above discussion, it is clear that we can recover V^* by initializing $V^{(0)} = 0_{|\mathcal{S}|}$ and setting

$$V^{(t+1)}_s = \max_{a\in\mathcal{A}} r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \tau(s'|s,a) V^{(t)}_{s'} \qquad \forall s \in \mathcal{S} \tag{14}$$

We can also recover Q^* by initializing $Q^{(0)} = 0_{|\mathcal{S}|\times|\mathcal{A}|}$ and setting

$$Q^{(t+1)}_{s,a} = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \tau(s'|s,a)\left(\max_{a'\in\mathcal{A}} Q^{(t)}_{s',a'}\right) \qquad \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{15}$$

[6]The iterative update can be seen as simply expanding the definition step by step. To see this more clearly, consider a deterministic policy $\pi : s \mapsto a$ and deterministic state transition $\tau : (s,a) \mapsto s'$. Then the iteration (13) at step t is

$$V^{(t)}_s(\pi) = r(s_0,a_0) + \gamma V^{(t-1)}_{s_1}(\pi) = r(s_0,a_0) + \gamma\left(r(s_1,a_1) + \gamma V^{(t-1)}_{s_2}(\pi)\right) = \sum_{t'=0}^{\infty}\gamma^{t'} r(s_{t'},a_{t'})$$

where $s_0 = s$, $a_t = \pi(s_t)$, and $s_{t+1} = \tau(s_t,a_t)$. So clearly $V^{(t)}(\pi) \to V(\pi)$.

D.2 Q-Learning

Both (14) and (15) need access to the reward function and/or state transition probabilities ("model-based"). In contrast, Q-learning is model-free. In Q-learning, at each iteration t we choose a state $s_t \in \mathcal{S}$ randomly and then (potentially for many times) select action $a_t \in \mathcal{A}$ by ϵ-greedy exploration using the current Q[7], observe reward $r(s_t, a_t)$ and the next state $s_{t+1} \sim \tau(\cdot|s_t, a_t)$ from the unknown MDP, and update

$$Q_{s_t,a_t} \leftarrow (1-\eta)Q_{s_t,a_t} + \eta\left(r(s_t,a_t) + \gamma \max_{a\in\mathcal{A}} Q_{s_{t+1},a}\right)$$

Note that setting $\eta = 1$ corresponds to estimating (15) with a single sample. Q-learning is an example of "off-policy" learning since the update does not depend on the current policy.[8] Standard tools in stochastic approximation can be used to show that Q converges to Q^* if we anneal $\eta \to 0$ at an appropriate rate. Since the update is equivalent to

$$Q_{s_t,a_t} \leftarrow Q_{s_t,a_t} + \eta\left(r(s_t,a_t) + \gamma \max_{a\in\mathcal{A}} Q_{s_{t+1},a} - Q_{s_t,a_t}\right)$$

Q-learning can also be viewed as gradient descent on a per-step squared error loss: at each step, for a given state-action pair (s, a) a sample of the next state $s' \sim \tau(\cdot|s, a)$ is drawn to define

$$J(Q_{s,a}) = \frac{1}{2}\left(\left(r(s,a) + \gamma \max_{a\in\mathcal{A}} Q_{s',a}\right) - Q_{s,a}\right)^2 \tag{16}$$

D.2.1 Deep Q-Learning

The per-step objective formulation of Q-learning (16) is particularly useful as it trivially admits neural extensions. Now Q_θ is a differentiable function which maps $s \in \mathcal{S}$ to a vector of action values $Q_\theta(s) \in \mathbb{R}^{|\mathcal{A}|}$. For example, in deep Q-learning for Atari games (Mnih *et al.*, 2013), each state $s \in \mathbb{R}^{84\times84\times4}$ represents gray-scale images of the playing area of the past 4 frames and each action $a \in \mathcal{A}$ is one of 4–18 available game-specific actions. The game environment provides an initial state s_0 and a deterministic mapping from the current state-action pair (s_t, a_t) to a reward r_t (1 if the game score increased, -1 if decreased, 0 otherwise) and the next state s_{t+1}. The model is learned by playing the game with an experience replay mechanism; starting from $t = 0$, we get (s_t, a_t, r_t, s_{t+1}) at each step (the action a_t is selected by ϵ-greedy exploration using the current model where ϵ is annealed from 1 to 0.1) and store it in a bounded leaky buffer, repeating this until we get a terminal state for s_{t+1} (then restarting the game). Then for an actual update we sample a random minibatch B of recent memories from the memory buffer and take a gradient step on

$$\sum_{(s_t,a_t,r_t,s_{t+1})\in B}\left(\left(r_t + \gamma \max_{a\in\mathcal{A}}[Q_\theta(s_{t+1})]_a\right) - [Q_\theta(s_t)]_{a_t}\right)^2$$

with respect to θ. Here Q_θ is parameterized by a CNN (it actually uses the previous state-action pair as well as the current state as input). The update is made every 3 or 4 frames.

D.3 Policy Iteration

The performance difference lemma (Lemma 2.3) says that for any policies π, π':

$$V_\mu(\pi') = V_\mu(\pi) + \frac{1}{1-\gamma}\mathop{\mathbf{E}}_{\substack{s\sim d_\mu^{\pi'}\\ a\sim\pi'(\cdot|s)}}[A_{s,a}(\pi)]$$

In particular, if π' satisfies $\mathbf{E}_{a\sim\pi'(\cdot|s)}[A_{s,a}(\pi)] \geq 0$ for every $s \in \mathcal{S}$, we have $V_\mu(\pi') \geq V_\mu(\pi)$. One way to achieve this is to set $\pi'(\cdot|s)$ a point mass distribution such that $\pi'(a^*_{\pi,s}|s) = 1$ where $a^*_{\pi,s} = \arg\max_a A_{s,a}(\pi)$ (ties broken arbitrarily) since then

$$\mathop{\mathbf{E}}_{a\sim\pi'(\cdot|s)}[A_{s,a}(\pi)] = \max_{a\in\mathcal{A}} A_{s,a}(\pi) \geq 0$$

[7] That is, we set $a_t = \arg\max_{a\in\mathcal{A}} Q_{s_t,a}$ with probability $1-\epsilon$ and we set $a_t \in \mathcal{A}$ randomly with probability ϵ, for some $\epsilon \in [0, 1]$.

[8] If we replace $\max_{a\in\mathcal{A}} Q_{s_{t+1},a}$ with $Q_{s_{t+1},a_{t+1}}$ where a_{t+1} is drawn using the current learning policy, the algorithm is on-policy and called SARSA.

Thus as long as there exists a reachable state s with an action a that has a positive advantage under π, we have $V_\mu(\pi') > V_\mu(\pi)$. This yields policy iteration: initialize $\pi^{(0)}$ and for $t = 0, 1, \ldots$

1. **Policy evaluation**: Calculate $V(\pi^{(t)}) \in \mathbb{R}^{|\mathcal{S}|}$ by (13).

2. **Policy improvement**: Define $\pi^{(t+1)}$ to be a deterministic mapping from each $s \in \mathcal{S}$ to

$$\arg\max_{a \in \mathcal{A}} \; r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \tau(s'|s,a) V_{s'}(\pi^{(t)})$$

E Single-Step REINFORCE

Consider a degenerate "MDP" $(\mathcal{S}, \mathcal{A}, r, \mu)$ in which there is no state transition. That is, we sample a state $s \in \mathcal{S}$ from μ, sample an action $a \in \mathcal{A}$ from $\pi_\theta(\cdot|s)$, and get a reward $r(s,a)$. In this case the objective and its REINFORCE gradient with control variate $f : \mathbb{R}^d \times \mathcal{S} \to \mathbb{R}$ are

$$\begin{aligned} V_\mu(\theta) &= \mathop{\mathbf{E}}_{\substack{s\sim\mu \\ a\sim\pi_\theta(\cdot|s)}} [r(s,a)] \\ \nabla V_\mu(\theta) &= \mathop{\mathbf{E}}_{\substack{s\sim\mu \\ a\sim\pi_\theta(\cdot|s)}} [(r(s,a) - f(\theta,s)) \nabla l_{s,a}(\theta)] \end{aligned}$$

There is no agent exploring different states for long-term rewards, but we simply wish to optimize an objective which is an expectation with respect to the model itself. This is a common setting. For instance, in unsupervised parsing μ might be a distribution over sentences, $\pi_\theta(\cdot|s)$ is a parser that proposes a parse tree for a given sentence s, and the reward is the log likelihood of sentence-tree pairs under some model (Kim *et al.*, 2019). A multi-sample version of REINFORCE is given below for concreteness where we use $f(\theta, s) = V_s(\theta)$.

REINFORCE-DEGENERATE
Input: distribution μ over states $\mathcal{S}$, differentiable policy π_θ that defines a distribution over actions $\mathcal{A}$ given a state, reward $r : \mathcal{A} \to [0,1]$, number of gradient steps R, number of samples K, step size η
Output: gradient-based estimation of $\theta^* \in \arg\max_{\theta \in \mathbb{R}^d} \; \mathbf{E}_{s\sim\mu,\, a\sim\pi_\theta(\cdot|s)}[r(s,a)]$

1. Initialize $\theta^{(0)}$.
2. For $i = 0, 1, \ldots R-1$,
   (a) Sample $s \sim \mu$ and $a_1 \ldots a_K \sim \pi_{\theta^{(i)}}(\cdot|s)$.
   (b) Calculate

$$\theta^{(i+1)} = \theta^{(i)} + \eta \left(\frac{1}{K} \sum_{k=1}^{K} \left(r(s,a_k) - \frac{1}{K-1} \sum_{k'=1:\; k' \neq k}^{K} r(s,a_{k'}) \right) \nabla l_{s,a_k}(\theta^{(i)}) \right)$$

3. Return $\theta^{(R)}$.

F Least Squares and Pseudo-Inverse

Definition F.1. The **pseudo-inverse** of $A \in \mathbb{R}^{m \times n}$ is the unique matrix $A^+ \in \mathbb{R}^{n \times m}$ such that $AA^+ \in \mathbb{R}^{m \times m}$ is the orthogonal projection onto range$(A) \subset \mathbb{R}^m$ and $A^+A \in \mathbb{R}^{n \times n}$ is the orthogonal projection onto row$(A) \subset \mathbb{R}^n$.

The pseudo-inverse always exists and can be constructed by $A^+ = \overline{V}\overline{\Sigma}^{-1}\overline{U}^\top$ where $\overline{U}\overline{\Sigma}\overline{V}^\top$ is a thin SVD of A.

Lemma F.1. Let $A = U\Sigma V^\top \in \mathbb{R}^{m \times n}$ be a matrix with $r = \text{rank}\,(A) > 0$ and let $b \in \mathbb{R}^m$. The unique solution

$$x^* = \arg\min_{x \in \mathbb{R}^n} \; ||Ax - b||^2$$

is given by $x^* = A^+ b$ with error $||Ax - b||^2 = \sum_{i>r} (u_i^\top b)^2$.

Proof.

$$\begin{aligned}
||Ax-b||^2 &= \left|\left|U^\top Ax - U^\top b\right|\right|^2 && (U \in \mathbb{R}^{m\times m} \text{ is a rotation}) \\
&= \left|\left|U^\top AVV^\top x - U^\top b\right|\right|^2 && (VV^\top \in \mathbb{R}^{n\times n} \text{ is the OP onto range}\,(A)) \\
&= \left|\left|\Sigma V^\top x - U^\top b\right|\right|^2 \\
&= \sum_{i\le r}\left(\sigma_i v_i^\top x - u_i^\top b\right)^2 + \sum_{i>r}\left(u_i^\top b\right)^2 \\
&\ge \sum_{i>r}\left(u_i^\top b\right)^2
\end{aligned}$$

with equality iff $\sigma_i v_i^\top x = u_i^\top b$ for every $i = 1 \dots r$. Using the orthogonality of V, we achieve this equality by

$$x = \sum_{i\le r}\frac{u_i^\top b}{\sigma_i}v_i = \overline{V}\overline{\Sigma}^{-1}\overline{U}^\top b = A^+ b$$

□

G Matrix Form of Cauchy-Schwarz

We write $A \succeq B$ to mean that $A - B$ is positive semidefinite.

Theorem G.1 (Tripathi, 1999)**.** Let $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^p$ be random vectors with $\mathbf{E}\left[||u||^2\right] < \infty$ and $\mathbf{E}\left[||v||^2\right] < \infty$. If $\mathbf{E}\left[vv^\top\right]$ is invertible, then

$$\mathbf{E}\left[uu^\top\right] \succeq \mathbf{E}\left[uv^\top\right]\mathbf{E}\left[vv^\top\right]^{-1}\mathbf{E}\left[vu^\top\right]$$

Corollary G.2. Let $u \in \mathbb{R}^d$ be a random vector with $\mathbf{E}\left[||u||^2\right] < \infty$. For any $v \in \mathbb{R}^p$ with $\mathbf{E}\left[||v||^2\right] < \infty$ such that $\operatorname{Cov}(v, v)$ is invertible,

$$\operatorname{Cov}(u, u) \succeq \operatorname{Cov}(u, v)\operatorname{Cov}(v, v)^{-1}\operatorname{Cov}(v, u)$$