important dates / deadlines:

1. 15/1 : deadline to register for final presentations

2. 23/1 : deadline to submit final projects

3. 25/1 : begin of review process
   DEADLINE: 1/2 to submit peer review reports

4. 1/2 : DEADLINE to submit slides (in pdf format) for final presentations

remaining lectures:

- 9/1 : lecture advanced policy gradients
- 15/1 : Q/A session at 6 pm (instead of coding session)
- 16/1 : lecture : RL w/ continuous actions
- 22/1 : no coding session
- 23/1 : no lecture

Advanced Policy Gradient Methods

recap: REINFORCE algo:

policy π_θ, θ variational params

RL objective $J(\theta)$

1. sample $\{\tau_j\}$ from π_θ

2. estimate policy gradient

$$\nabla_\theta J \approx \frac{1}{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^j | s_t^j) \underbrace{\sum_{t'=t}^{T} r(s_{t'}^j, a_{t'}^j)}_{= \hat{Q}_{j,t}^{\pi}}$$

reward-to-go

3. update policy:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

why does policy gradient work?

recall: using baseline naturally leads to estimating advantages $\hat{A}^\pi$

high-level view of PG:

1. estimate $\hat{A}^{\pi}(s_t, a_t)$ for current policy π
2. use $\hat{A}^{\pi}$ to get an improved policy π'

→ looks like policy iteration → is it?

recall:

$$J(\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}}\left[\sum_t \gamma^t r(s_t, a_t)\right]$$

$$\tau = (s_0, a_0, s_1, a_1, \ldots)$$

$$p_{\pi_\theta}(\tau) = p(s_0) \prod_{t=1}^{T} \pi_\theta(a_t | s_t) \underbrace{p(s_t | s_{t-1}, a_{t-1})}_{= p_\theta(s_t)}$$

τ θ-dep. b/c $a_{t-1} \sim \pi_\theta$

want to show that PG is a form of policy iteration.

- consider policy improvement objective

$$J(\theta) - J(\theta')$$

(θ: new parameters, θ': old params)

claim:

$$J(\theta) - J(\theta') = \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_t \gamma^t A^{\pi_\theta}(s_t, a_t)\right]$$

proof:

$$J(\theta') - J(\theta) = J(\theta') - \mathbb{E}_{s_0 \sim p(s_0)}\left[V^{\pi_\theta}(s_0)\right]$$

s_0 part of all trajectories; $p(s_0)$ indep. of θ

$$= J(\theta') - \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[V^{\pi_\theta}(s_0)\right]$$

telescopic sum

$$= J(\theta') - \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_{t=0}^{\infty} \gamma^t V^{\pi_\theta}(s_t) - \sum_{t=1}^{\infty} \gamma^t V^{\pi_\theta}(s_t)\right]$$

$$= J(\theta') + \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_{t=0}^{\infty} \gamma^t \left(\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)\right)\right]$$

def $J(\theta')$

$$= \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

$$+ \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_{t=0}^{\infty} \gamma^t \left(\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)\right)\right]$$

$$= \mathbb{E}_{\tau \sim p_{\pi_{\theta'}}}\left[\sum_{t=0}^{\infty} \gamma^t \underbrace{\left(r(s_t, a_t) + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)\right)}_{= A^{\pi_\theta}(s_t, a_t)}\right]$$

↑ def

$$= \mathbb{E}_{\tau \sim p_{\pi_\theta'}} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \Rightarrow \text{claim } \checkmark$$

new policy $\pi_{\theta'}$ ↑ old policy π_θ ↑

→ policy improvement objective is the expectation of the advantage of previous policy π_θ under trajectory distr. of new policy $\pi_{\theta'}$.

→ problem: samples generated from π_θ cannot be used to estimate expectation wrt $\pi_{\theta'}$

$$\mathbb{E}_{\tau \sim p_{\pi_{\theta'}}(\tau)} \left[\sum_t \gamma^t A^{\pi_\theta}(s_t, a_t) \right]$$

$$= \sum_t \mathbb{E}_{s_t \sim p_{\theta'}(s_t)} \left[\mathbb{E}_{a_t \sim \pi_{\theta'}(a_t|s_t)} \left[\gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right]$$

$$= \sum_t \mathbb{E}_{s_t \sim p_{\theta'}(s_t)} \left[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)} \left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right]$$

↑ importance sampling trick

↑ outer expectation

↑ inner expectation wrt π_θ ✓

still prevents us from using π_θ-samples

→ distribution mismatch problem

• can we ignore it? : $\mathbb{E}_{s_t \sim p_{\theta'}(s_t)}[\cdot] \approx \mathbb{E}_{s_t \sim p_\theta(s_t)}[\cdot]$

<u>def</u>:

$$\bar{A}(\theta') := \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)}\left[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}\left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t)\right]\right]$$

$\Rightarrow J(\theta') - J(\theta) \approx \bar{A}(\theta')$

policy improvement: $\theta' \leftarrow \arg\max_{\theta'} \bar{A}(\theta')$

<u>claim</u>: $p_\theta(s_t)$ is close to $p_{\theta'}(s_t)$
whenever π_θ is close to $\pi_{\theta'}$
(proof → see Berkeley lectures)

what does "close" mean?
Kullback-Leibler (KL) divergence:
$D_{KL}(p_1(x) \,\|\, p_2(x)) = \mathbb{E}_{x \sim p_1(x)}\left[\log \frac{p_1(x)}{p_2(x)}\right]$

policy improvement step:

$$\theta' \leftarrow \arg\max_{\theta'} \underbrace{\sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)} \left[\mathbb{E}_{a_t \sim \pi(a_t|s_t)} \left[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right]}_{= \bar{A}(\theta')}$$

such that

$$D_{KL}(\pi_{\theta'}(a_t|s_t) \,||\, \pi_\theta(a_t|s_t)) \leq \varepsilon$$

- for ε small enough, this is guaranteed to improve $J(\theta') - J(\theta)$.

→ how do we enforce the constraint?

1) introduce Lagrange multiplier λ

$$L(\theta', \lambda) = \bar{A}(\theta') - \lambda (D_{KL}(\pi_{\theta'} || \pi_\theta) - \varepsilon)$$

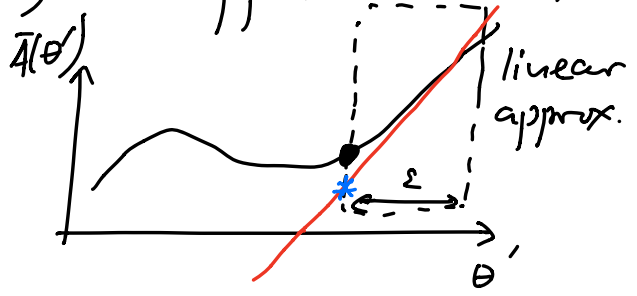
- algorithm:

1. maximize $L(\theta', \lambda)$ wrt. θ' ← no need to optimize until convergence
2. $\lambda \leftarrow \lambda + \alpha (D_{KL}(\pi_{\theta'} || \pi_\theta) - \varepsilon)$

intuition: increase λ if constraint is violated too much; else, we decrease λ

(→ dual gradient descent)

2) use approximation, i.e. Taylor expansion



optimum of linear function is close to optimum of the original function, if the box size is small enough

$$\bar{A}(\theta') \approx \text{const} + \nabla_\theta \bar{A}^t(\theta) \cdot (\theta' - \theta)$$

=> policy improvement:

$$\theta' \leftarrow \arg\max_{\theta'} \nabla_\theta \bar{A}^t(\theta) \cdot (\theta' - \theta)$$

$$\text{s.t. } D_{KL}(\pi_{\theta'} \| \pi_\theta) \le \varepsilon$$

$$\& \quad \|\theta' - \theta\|^2 \le \varepsilon$$

} can use same ε

note: $\nabla_\theta \bar{A}(\theta)$ evaluated at $\theta' = \theta$ is equivalent to policy gradient (importance sampling ratio is unity as $\theta' \to \theta$):

$$\nabla_{\theta'} \bar{A}(\theta') = \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)} \Big[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)} \Big[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \cdot \nabla_{\theta'} \log \pi_{\theta'}(a_t|s_t) \gamma^t A^{\pi_\theta}(s_t, a_t) \Big] \Big]$$

$$\nabla_{\theta'} \bar{A}\big|_{\theta'=\theta} = \mathbb{E}_{\tau \sim p_{\pi_\theta}} \Big[\sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) \gamma^t A^{\pi_\theta}(s_t, a_t) \Big]$$

ordinary PG

what does policy gradient actually do?

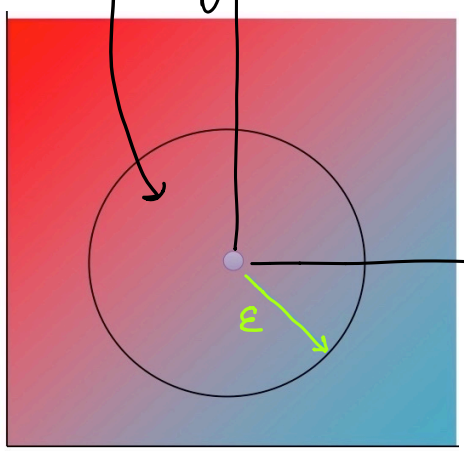
$J(\theta') \approx \text{const} + \nabla_\theta J(\theta) \cdot (\theta' - \theta)$ Taylor expand the RL objective

$$\theta' \leftarrow \arg\max_{\theta'} \nabla_\theta J(\theta) \cdot (\theta' - \theta) \quad \text{s.t.} \quad \|\theta' - \theta\|^2 \le \varepsilon$$

← linear function! → colorbar

- visualize:

find the bluest point inside circle $\|\theta' - \theta\|^2 \le \varepsilon$



(Berkeley lecture)

optimum taken at edge of circle:

$$\|\theta' - \theta\|^2 = \varepsilon^2$$

$$\Rightarrow \underset{\theta'}{\operatorname{argmax}} \nabla_\theta J^t(\theta) \cdot (\theta' - \theta) \Big|_{\text{s.t. } \|\theta' - \theta\|^2 = \varepsilon} = \overbrace{\frac{\nabla_\theta J(\theta)}{\|\nabla_\theta J(\theta)\|}}^{\text{has unit length}} \cdot \varepsilon \quad \left(= \|\theta' - \theta\| \right)$$

$$\Rightarrow \theta' = \theta + \underbrace{\sqrt{\frac{\varepsilon}{\|\nabla_\theta J(\theta)\|^2}}}_{\text{gives correct value for learning rate / step size}} \nabla_\theta J(\theta)$$

→ ADAM, etc.

<u>but</u>: we have $D_{KL}(\pi_{\theta'} \| \pi_\theta) \leq \varepsilon$ in addition to $\|\theta' - \theta\|^2 \leq \varepsilon$

<u>ideal</u>: Taylor-expand D_{KL}:

$$D_{KL}(\pi_{\theta'} \| \pi_\theta) \approx \frac{1}{2} (\theta' - \theta)^t F (\theta - \theta) + \dots$$

- has no zeroth & first order terms

F: Fisher information matrix (in param space)

$\theta \in \mathbb{R}^n \Rightarrow F \in \mathbb{R}^{n \times n}$

$$F := \mathbb{E}_{\pi_\theta}\left[\underbrace{\nabla_\theta \log \pi_\theta}_{= D} \cdot (\nabla_\theta \log \pi_\theta)^\top\right]$$

$F_{ij} = \langle D_i D_j \rangle_{\pi_\theta}$

$F_{ij} v_j = \langle D_i D_j v_j \rangle_{\pi_\theta}$

↑ approximate using MC-sampled trajectories

we can use samples from π_θ to estimate F

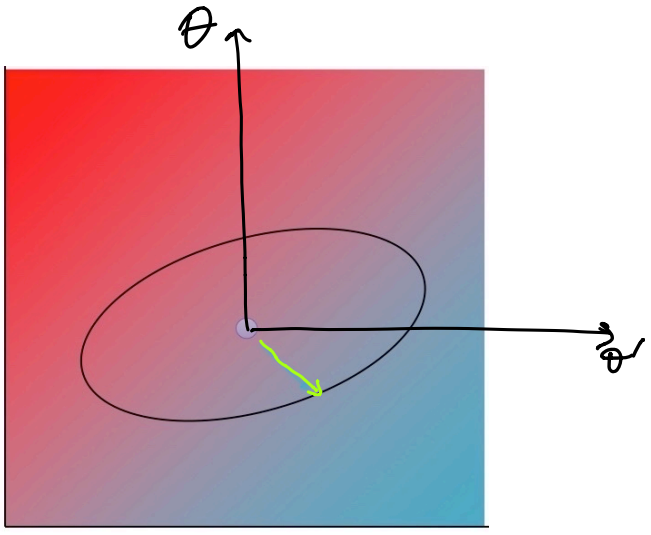
→ replace KL constraint with a generalized quadratic constraint:

$$\underset{\theta'}{\operatorname{argmax}} \; \nabla_\theta J(\theta)^\top \cdot (\theta' - \theta) \quad \Big| \; (\theta' - \theta)^\top \cdot F \cdot (\theta' - \theta) \le 2\varepsilon$$

$F \neq \mathbb{1}$

circle → ellipse

look for bluest point on the ellipse!



⇒ reshape ellipse to circle by applying F^{-1}; then use formula for quadratic constraint.

- Natural Policy Gradient:

natural gradient:

$$\boxed{\theta' = \theta + \alpha F^{-1} \nabla_\theta J(\theta)}$$

- insensitive to parametrization of the policy distribution
- all params affect distribution according to curvature matrix F
  → geometry of parameter space

- Trust Region Policy Optimization (TRPO)

$$\theta' = \theta + \alpha F^{-1} \nabla_\theta J(\theta)$$

choose: $$\boxed{\alpha = \sqrt{\frac{2\varepsilon}{\nabla_\theta J^t \cdot F \cdot \nabla_\theta J}}}$$

ε: hyperparam. to set size of trust region

issue: F, F^{-1} hard/infeasible to compute
(but there exist tricks to still apply the algorithm in practice)

3) Proximal Policy Optimization (PPO)

$$\arg\max_{\theta'} \mathbb{E}_{\tau \sim P_{\pi_\theta}} \left[\sum_t \gamma^t \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} A^{\pi_\theta}(s_t, a_t) \right]$$

$$\text{s.t. } D_{KL}(\pi_{\theta'} \,||\, \pi_\theta) \leq \varepsilon$$

<u>want</u>: first-order PG method (i.e. no second derivatives)

<u>def</u>: $\rho_t(\theta') := \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}$ importance sampling ratio

note: $\rho_t(\theta' = \theta) = 1$

- consider unconstrained objective:

$$\mathbb{E}_{\tau \sim P_{\pi_\theta}} \left[\rho_t(\theta') \hat{A}^{\pi_\theta}(s_t, a_t) \right]$$

→ lack of KL-div. constraint leads to excessively large policy updates, b/c importance sampling ratio ρ_t is unbounded

ideas:

(i): clip updates within small ε-range around $\rho = 1$

$$\text{clip}(\rho; a, b) = \min(\max(\rho, a), b)$$

clips ρ within $[a, b]$

$$\rho(\theta') A^{\pi_\theta} \longrightarrow \text{clip}(\rho(\theta'); 1-\varepsilon, 1+\varepsilon) A^{\pi_\theta}$$

(ii) use a pessimistic estimate (i.e. lower bound) of the unclipped objective: this ignores changes in ρ when it improves the objective but it'll keep the changes if it makes objective worse:

→ use following objective instead:

$$\mathbb{E}_{\tau \sim p_{\pi_\theta}}\left[\min\left\{\rho_t(\theta') A^{\pi_\theta}(s_t, a_t);\ \text{clip}(\rho_t(\theta'); 1-\varepsilon, 1+\varepsilon) A^{\pi_\theta}(s_t, a_t)\right\}\right]$$

(iii) repeat each update step (for the same π_θ-sample) for K epochs.

- PPO algorithm :

1. sample trajectories $\{\tau^j\}_{j=1}^{N}$, using π_θ

$\times K$

2. $\theta' \leftarrow \arg\max_{\theta'} \frac{1}{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \min \Big\{ \rho_t^j(\theta') A^{\pi_\theta}(s_t^j, a_t^j),$

$\text{clip}\big(\rho_t^j(\theta'); 1-\varepsilon, 1+\varepsilon\big) A^{\pi_\theta}(s_t^j, a_t^j) \Big\}$

3. $\theta \leftarrow \theta'$ (update old policy)

note : no $\log \pi$ here !

Remark : extensions adding a critic can be used to further reduce variance of policy gradient

• at iteration step 1 of PPO : $\theta' = \theta$

but $\nabla_{\theta'}\big(\rho(\theta') A^{\pi_\theta}\big) = \frac{\nabla_{\theta'} \pi_{\theta'}}{\pi_\theta} A^{\pi_\theta}$

$\theta' \to \theta$ $\longrightarrow$ $\frac{\nabla_\theta \pi_\theta}{\pi_\theta} A^{\pi_\theta} = \nabla_\theta \log \pi_\theta \times A^{\pi_\theta}$

$\to$ recover ordinary PG

- Summary & Practical considerations

- Natural Policy Gradient: stabilizes PG training

$$\theta' = \theta + \alpha F^{-1} \nabla_\theta J(\theta)$$

issue: F^{-1} hard to compute for a NN with many params

→ requires efficient F·v products

idea: avoid computing F itself

only compute F·v

→ use, e.g. conjugate gradient (CG)

$$A x = b$$

solution $x = A^{-1} b$

CG: find x iteratively, if we know F·v for an arbitrary v

JAX: allows to do F·v efficiently

see Appendix of arXiv: 1502.05477 (Schulman et al.)

- Trust Region Policy Optimization (TRPO)

$$\theta' = \theta' + \alpha F^{-1} \nabla_\theta J(\theta)$$

$$\& \quad \alpha = \sqrt{\frac{2\varepsilon}{\nabla_\theta J(\theta)^t \cdot F \cdot \nabla_\theta J(\theta)}}$$

requires same tricks as natural policy gradient

- Proximal Policy Optimization (PPO)
  - uses importance sampling objective directly + clipping

    arXiv: 1707.06347
    (Schulman et al)

in practice:

- a clipping param of $\varepsilon \sim 0.1 \div 0.2$ typically works fine
- typically, one repeats the θ'-max step (step 2) for $K \sim 4$ PPO epochs.

Entropy-based Exploration

- policy $\pi(a|s)$ is a probability distr.

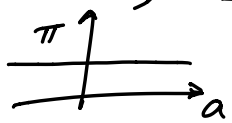
$$\sum_{a \in A} \pi(a|s) = 1$$

- def: information entropy for $\pi(a|\cdot)$

$$S_\pi(s) = \sum_{a \in A} - \pi(a|s) \log \pi(a|s)$$

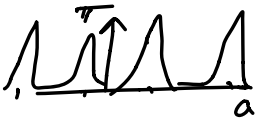
• if $\pi(a|s)$ is uniform: $\pi(a|s) = \frac{1}{|A|}$

$\Rightarrow S(s) = \log |A|$ maximum possible entropy



• if $\pi(a|s)$ is delta function, i.e. deterministic

$\Rightarrow S(s) = 0$

$\Rightarrow$ S measures how broad π is: 

idea: add the entropy to the RL objective to enhance exploration

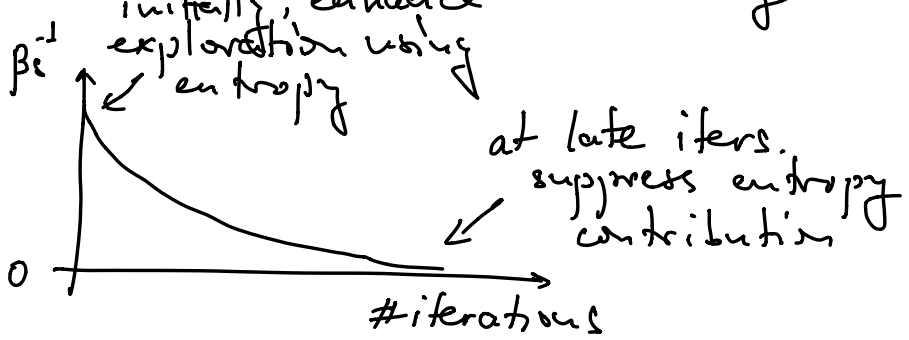
$$J(\theta) \longrightarrow J(\theta) + \beta_S^{-1} \times \text{Entropy}$$

β_s^{-1} : "temperature" : controls the weight of the entropy term

$\beta_s^{-1} = 0$ $\Rightarrow$ back to old PG objective $J(\theta)$

$\beta_s^{-1} \to \infty$ $\Rightarrow$ completely entropy-based objective

$\rightarrow$ schedule :



$$J(\theta) + \beta_s^{-1} \times \text{Entropy} =$$

$$= \sum_{\{\tau_j\}} \left\{ \sum_{t=1}^{T} \pi_\theta(a_t^j | s_t^j) \sum_{t'=t}^{T} \left[r(s_{t'}^j, a_{t'}^j) - b(s_t^j) \right. \right.$$

$$\left. \left. - \beta_s^{-1} \sum_{a \in A} \pi_\theta(a | s_t^j) \log \pi_\theta(a | s_t^j) \right] \right\}$$

<u>want</u> : maximize expected return $J(\theta)$
& maximize entropy (to keep up exploration)

- can we compute the gradient contribution coming from the entropy term?
- can we estimate it using MC sampling?

- contribution to $\nabla_\theta J$ due to entropy

$$\nabla_\theta \sum_{a \in A} \pi_\theta(a|s) \log \pi_\theta(a|s) =$$

$$= \sum_{a \in A} \left\{ (\nabla_\theta \pi_\theta) \cdot \log \pi_\theta + \pi_\theta \underbrace{\nabla_\theta \log \pi_\theta}_{= \frac{\nabla_\theta \pi_\theta}{\pi_\theta}} \right\}$$

$$= \sum_a \underbrace{\nabla_\theta \pi_\theta}_{= \pi_\theta \nabla_\theta \log \pi_\theta} \cdot \log \pi_\theta + \underbrace{\sum_a \nabla_\theta \pi_\theta}_{= \nabla_\theta \sum_a \pi_\theta(a|s) = \nabla_\theta (1) = 0}$$

$\Rightarrow$ vanishes (or, becomes part of baseline)

$$= \sum_a \pi_\theta \cdot \nabla_\theta \log \pi_\theta \times \log \pi_\theta$$

$$= \sum_a \pi_\theta \nabla_\theta \left[\frac{1}{2} (\log \pi_\theta)^2 \right] = \mathbb{E}_{\pi_\theta} \left[\frac{1}{2} \nabla_\theta (\log \pi_\theta)^2 \right]$$

$\rightarrow$ use MC to estimate gradient contribution

$\Rightarrow$ pseudo entropy cost/reward function:

$$\frac{1}{2} \mathbb{E}_{a \sim \pi_\theta} \left[(\log \pi_\theta)^2 \right]$$

- total pseudo reward function for PG with entropy-based exploration:

$$\mathbb{E}_{\tau \sim P_{\pi_\theta}} \left[\log \pi_\theta(\tau) \times \{ r(\tau) - b \} - \frac{\beta s^{-1}}{2} \left(\log \pi_\theta(\tau) \right)^2 \right]$$

$$= \mathbb{E}_{\tau \sim P_{\pi_\theta}} \left[\log \pi_\theta(\tau) \{ r(\tau) - b \underbrace{- \frac{\beta s^{-1}}{2} \log \pi_\theta(\tau)}_{\text{"entropy bonus"}} \} \right]$$

note: implementing the so-called entropy bonus comes for free

more on RL as inference in arXiv: 1805.00909 (S. Levine)

note: similarly, we can add the entropy bonus to actor-critic methods, etc.