Policy Gradient Methods

Define

s_0 = initial state (could use $\mu(s)$ = initial distribution)

π_θ = Parameter policy (e.g., neural net). Will write π sometimes for simplicity

τ = A trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_H, a_H)$

Note: we will truncate at H here. not sure it that matters in general

$R(\tau)$ = Return along trajectory τ $= \sum_{t=0}^{H} \gamma_t r_t$, where r_t is reward at time t.

$$J(\theta) := \mathbb{E}_{\pi_\theta}(R(\tau)) = \sum_\tau \underbrace{P(\tau;\theta)}_{\text{Probability of trajectory } \tau} R(\tau)$$

Observe

$$\nabla_\theta J(\theta) = \sum_\tau \nabla_\theta P(\tau;\theta) R(\tau)$$

Want to write as expectation so we can use WLLN / Sampling

$$= \sum_\tau P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta)}{P(\tau;\theta)} R(\tau)$$

$$= \mathbb{E}_{\pi_\theta}\left(\nabla_\theta \ln[P(\tau;\theta)] R(\tau) \right)$$

Hence,

$$\hat{g} := \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \ln[P(\tau^{(i)};\theta)] R(\tau^{(i)})$$

is an unbiased estimator of $\nabla_\theta J(\theta)$

Lets look more carefully at $\nabla_\theta \ln P(\tau;\theta)$

$$\nabla_\theta \ln P(\tau;\theta) = \nabla_\theta \ln \boxed{\prod_{t=0}^{H}} P(s_{t+1}|s_t, a_t)\,\pi_\theta(a_t|s_t) \;]$$

$$= \nabla_\theta \sum_{t=0}^{H} \ln P(s_{t+1}|s_t,a_t) + \nabla_\theta \sum_{t=0}^{H} \ln \pi_\theta(a_t|s_t)$$

We can compute this for a specified parameter policy class

$$= \sum_{t=0}^{H} \nabla_\theta \ln \pi_\theta(a_t|s_t)$$

$$= \sum_{t=0}^{H} \underbrace{\frac{1}{\pi_\theta(a_t|s_t)}}_{\text{"Correction" or Normalization for low probability event?}} \underbrace{\nabla_\theta \pi_\theta(a_t|s_t)}_{\text{This is a vector pointing in direction that causes greatest increase for probability of playing action } a_t \text{ from state } s_t}$$

This part is confusing

Recall:

$$\hat{g} = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{t=0}^{H} \frac{1}{\pi_\theta(a_t|s_t)} \nabla_\theta \pi(a_t|s_t)\right) R(\tau)$$

If a Reward for a trajectory is large, then you increase the probability of repeating that trajectory.

In practice, no one does exactly this. Variance is too high.

We may exploit temporal structure of MDP to reduce the variance of the estimator. Let

$$\tau_t = \{ s_0, a_0, \dots, s_t, a_t \}$$

be a truncation of τ. Note

$$J(\theta) = \mathbb{E}_\tau(R) = \mathbb{E}_\tau\left(\sum_{t=0}^{H} r_t \right)$$

$$\cancel{= \sum_{\tau} \left(\sum_{t=0}^{H} r_t \right) P(\tau;\theta)}$$

~~Lazy expectation, but should exploit fact that r_t is independent of $\{s_{t+1}, a_{t+1}, \dots\}$~~

$$\cancel{= \sum_{\tau} \sum_{t=0}^{H} r_t \, P(\tau;\theta)}$$

$$\cancel{= \sum_{\tau} \sum_{t=0}^{H} r_t \, P(\tau_t;\theta)}$$

$$= \sum_{t=0}^{H} \mathbb{E}_{\tau}(r_t)$$

$$= \sum_{t=0}^{H} \sum_{\tau} r_t(\tau) P(\tau_t;\theta)$$

$$= \sum_{\tau} \sum_{t=0}^{H} r_t(\tau) P(\tau_t;\theta)$$

Hence,

$$\nabla_\theta J(\theta) = \sum_{\tau} \sum_{t=0}^{H} r_t(\tau) \nabla P(\tau_t;\theta)$$

$$= \sum_{\tau} \sum_{t=0}^{H} r_t(\tau) P(\tau_t;\theta) \nabla \ln P(\tau_t;\theta)$$

$$= \sum_{\tau} \sum_{t=0}^{H} P(\tau_t;\theta)\, r_t(\tau) \sum_{h=0}^{t} \nabla \ln \pi_\theta(a_h|s_h)$$

can drop the t by given outside sums

$$= \mathbb{E}\left(\sum_{t=0}^{H} r_t(\tau) \sum_{h=0}^{t} \nabla \ln \pi_\theta(a_h|s_h) \right)$$

$$= \mathbb{E}\left(\sum_{t=0}^{H} \sum_{h=0}^{t} r_t \nabla \ln \pi_\theta(a_h|s_h) \right) \quad (*)$$

Note:

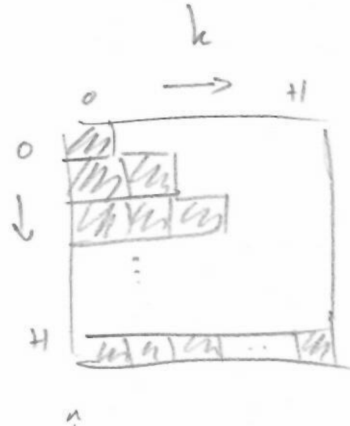
Suppose you want to sum

$$(**)\ \sum_{t=0}^{H} \sum_{k=0}^{t} a_t b_k$$

Instead of summing rows,

Sum columns:

$$(**) = \sum_{k=0}^{H} \sum_{t=k}^{H} a_t b_k$$



Hence, $(*)$

$$= \mathbb{E}\left(\sum_{k=0}^{H} \sum_{t=k}^{H} r_t \nabla \ln \pi_\theta (a_k | s_k) \right)$$

$$= \mathbb{E}\left(\sum_{k=0}^{H} \nabla_\theta \ln \pi_\theta (a_k | s_k) \underbrace{\sum_{t=k}^{H} r_t}_{=: G_k \text{ = return after time } k} \right)$$

$$= \mathbb{E}\left(\sum_{k=0}^{H} \nabla_\theta \ln \pi_\theta (a_k | s_k)\, G_k \right)$$

REINFORCE algorithm (Williams, '92)

Initialize: θ

Loop forever

Generate episode following π_θ: $s_0, a_0, r_0, s_1, a_1, r_1, \ldots s_H, a_H, r_H$

Loop for each step of the episode $t = 0, 1, \ldots, H$

$$G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$$

$$\theta \leftarrow \theta + \alpha \gamma^t G_t \nabla \ln \pi (a_t | s_t, \theta)$$

<u>Note:</u> It doesn't totally make sense why we're not summing up all the terms $t = 0, \ldots, H$ first & then update θ. That's the estimator, right? We really derived a different algorithm, that I'm not sure what pros & cons are. But, perspective is valuable. And will derive REINFORCE for real, next.

An alternate perspective: The Policy Gradient thrm

Define

- $q_\pi(s,a)$ = Value of taking action a from state s & following π thereafter

Thrm Let $J(\theta) = v_\pi(s_0)$ Then

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)$$

Pf (Next page)

$$\nabla v_\pi(s) = \nabla \sum_a \pi(a|s)\, q_\pi(s,a) \qquad \forall s$$

$$= \sum_a \nabla \pi(a|s)\, q_\pi(s,a) + \pi(a,s)\, \nabla q_\pi(s,a)$$

$$= \sum_a \nabla \pi(a|s)\, q_\pi(s,a) + \pi(a,s)\, \nabla \sum_{s'} P(s'|s,a)\,(r + v_\pi(s'))$$

$$= \sum_a \nabla \pi(a|s)\, q_\pi(s,a) + \pi(a,s) \sum_{s'} P(s'|s,a)\,\Big(\overset{\nabla r = 0}{\cancel{r+}^{\,0}}\ \underbrace{\nabla v_\pi(s')}_{\text{Unroll this in same way}}\Big)$$

$$= \underbrace{\sum_a \nabla \pi(a|s)\, q_\pi(s,a)}_{①} + \pi(a,s) \sum_{s'} P(s'|s,a)$$

$$\Big[\underbrace{\sum_a \nabla \pi(a|s')\, q_\pi(s',a)}_{②} + \pi(a,s') \sum_{s''} P(s''|s',a)\, \nabla v_\pi(s'') \Big]$$

Note:

- We start in state s
- ② is multiplied by prob of transitioning to s' in exactly k time steps
- If we keep unrolling, the pattern continues

$$= \sum_{x \in S} \sum_{k=0}^{\infty} \underbrace{\mathbb{P}(s \to x, k, \pi)}_{\text{Prob of transiting from } s \text{ to } x \text{ in exactly } k \text{ time step under } \pi} \sum_a \nabla \pi(a|x)\, q_\pi(x,a)$$

Hence,

$$\nabla J(\theta) = \nabla v_\pi(s_0)$$

$$= \sum_{x \in S} \underbrace{\Big(\sum_{k \geq 0} \mathbb{P}(s \to x, k, \pi) \Big)}_{\approx\, \eta(s)} \sum_a \nabla \pi(a|s)\, q_\pi(s,a)$$

$$= \sum_s \Big(\sum_{s'} \eta(s')\Big) \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s)\, q_\pi(s,a)$$

, where $\eta(s)$ = # time steps spent on avg in state s in single episode

$$= \sum_{s'} \eta(s') \sum_s \mu(s) \sum_a \nabla \pi(a|s)\, q_\pi(s,a)$$

$$\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s)\, q_\pi(s,a)$$

$\blacksquare$

How to use this result?

Note, thrm implies

$$\nabla J(\theta) \propto \mathbb{E}_{s\sim\mu}\left[\sum_a q_\pi(s,a)\,\nabla\pi(a|s,\theta)\right]$$

We know that $s_t \sim \mu$ as t large. So, for t large this is same as

$$\approx \mathbb{E}_\pi\left[\sum_a q_\pi(s,a)\,\nabla\pi(a|s,\theta)\right]$$

Could use estimator

$$\hat{g} = \sum_a \hat{q}(s_t,a,w)\,\nabla\pi(a|s_t,\theta) \quad \text{for } s_t \text{ generated by } \pi.$$

But, this requires $\hat{q}$. Can we replace sum over actions w/ an expectation somehow & do MC trick? Backing up, again we have

$$\nabla J(\theta) \propto \mathbb{E}_{s\sim\mu}\Big[\underbrace{\sum_a q_\pi(s,a)\,\nabla\pi(a|s,\theta)}_{\text{Want to write this as } \mathbb{E}}\Big]$$

$$= \mathbb{E}_{s\sim\mu}\left[\sum_a \pi(a|s,\theta)\,q_\pi(s,a)\,\frac{\nabla\pi(a|s,\theta)}{\pi(a|s,\theta)}\right]$$

$$= \mathbb{E}_{s\sim\mu}\left[\mathbb{E}_{a\sim\pi}\left[q(s,a)\,\frac{\nabla\pi(a|s,\theta)}{\pi(a|s,\theta)}\right]\right]$$

Estimator: Let $s_0, a_0, \ldots$ be trajectory generated by π & let $t =$ "large"

Let

$$\hat{g} = q(s_t,a_t)\,\frac{\nabla\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)} = \overset{\text{standard return}}{G_t}\,\frac{\nabla\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)}$$

This yields the exact same REINFORCE algorithm from before.

- Note:
  - Differences from prev. derivation
  - Big assumption that everyone completely ignores / glosses over: Need t to be large for $S_t \overset{\text{approx}}{\sim} \mu$. Yet, we sample only the beginning of a trajectory, not its tail.

REINFORCE w/ Baseline

Let $b(s)$ be some arbitrary function depending only on state s.

Claim:

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a \left(q_\pi(s,a) - b(s) \right) \nabla \pi(a|s,\theta)$$

This follows from the Policy gradient theorem and the fact that we're actually subtracting zero

$$\sum_a b(s) \nabla \pi(a|s,\theta) = b(s) \nabla \sum_a \pi(a|s,\theta) = b(s) \nabla 1 = 0$$

A good choice of baseline can yield a lower variance unbiased estimator of $\nabla J(\theta)$. Standard choice: $b(s) \approx V(s)$ = Value s

REINFORCE w/ baseline:

Input: parametrize $\pi(a|s, \theta)$, $\hat{v}(s; w)$
Algorithm params: step sizes α^w, α^θ
Initialize θ & w.

Loop forever:

Generate episode: $s_0, a_0, r_1, s_1, a_1, r_2, \ldots s_{H}, a_H, r_H$

Loop for each step of episode $t = 0, \ldots, H$

$$G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$$

$$\delta \leftarrow G - \hat{v}(s_t, w)$$

$$(*) \quad w \leftarrow w + \alpha^w \delta \nabla \hat{v}(s_t, w)$$

$$\theta \leftarrow \theta + \alpha^\theta \gamma^t \delta \nabla \ln \pi(a_t | s_t, \theta)$$

Question: why do we modulate $\hat{v}$ update by δ in $(*)$?

Ans: We're doing least squares fitting. Want to take a step in ∇ of $\frac{1}{2}(\hat{v}(s,w) - G)^2$

$$\nabla_w \tfrac{1}{2}(\hat{v}(s,w) - G)^2 = (\hat{v} - G)\nabla_w \hat{v}(s,w) = \delta \nabla_w \hat{v}(s,w)$$