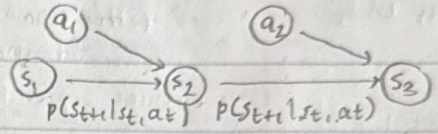
CS 182 Lecture 15: Policy Gradients

7/14

Without ground truth: how do we know which action is better or worse?
↳ $r(s,a)$ = reward function. Which states/actions are better?
↳ Greedy system: always chooses high reward short-term. Not effective, we want long-term.

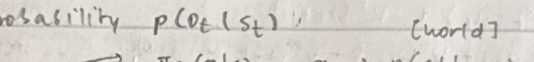
Markov decision process (MDP) - extension of Markov chain.
$M = \{S, A, T, r\}$
S - state space, states $s \in S$ (discrete / continuous)
A - action space, actions $a \in A$ (discrete / continuous)
r - reward function $r: S \times A \to \mathbb{R}$



← transition operator ($T(s_{t+1}|s_t, a_t)$)

partially observed MDP - $M = \{S, A, O, \gamma, \mathcal{E}, r\}$
O - observation space $o \in O$ (discrete / continuous)
↳ refer to Bayes Net diagram in Lecture 14.
$\mathcal{E}$ - emission probability $p(o_t|s_t)$

$p_\theta(\tau)$ ↓ trajectory — chain rule of probability
$p_\theta(s_1, a_1, \dots, s_T, a_T) = p(s_1) \prod_{t=1}^{T} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t)$

The goal of RL: 

"$J(\theta)$" $\theta^* = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(s_t, a_t) \right]$ - pick the policy parameters that give highest expected reward.
↳ exponentially large, fix via unbiased estimator (sample)

$J(\theta) \approx \frac{1}{N} \sum_i \sum_t r(s_{i,t}, a_{i,t})$ ← sum over samples from π_θ

direct policy differentiation (gradient ascent) preparation ↓

$J(\theta) = E_{\tau \sim \pi_\theta(\tau)}[r(\tau)] = \int \pi_\theta(\tau) r(\tau) \, d\tau$ ← sum of rewards

$\nabla_\theta J(\theta) = \int \nabla_\theta \pi_\theta(\tau) r(\tau) \, d\tau$ ← $\pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) = \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} = \nabla_\theta \pi_\theta(\tau)$

$= \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) r(\tau) \, d\tau$ ← $\log \pi_\theta(\tau) = \log p(s_1) + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)$

$= E_{\tau \sim \pi_\theta(\tau)} [\nabla_\theta \log \pi_\theta(\tau) r(\tau)]$

$= E_{\tau \sim \pi_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right) \left(\sum_{t=1}^{T} r(s_t, a_t) \right) \right]$

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

$\approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t}) \right) \left(\sum_{t=1}^{T} r(s_{i,t}, a_{i,t}) \right)$ → similar to maximum likelihood

flaw: not sequence specific → $\approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau_i) r(\tau_i)$ "good stuff is made more likely" aka "assisted" trial/error.

REINFORCE algorithm:
1. sample $\{\tau^i\}$ from $\pi_\theta(a_t|s_t)$
2. evaluate $\nabla_\theta J(\theta)$
3. gradient ascent

Causality: policy at time t' cannot affect reward at time t when $t < t'$.
↳ small fix: $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t}) \left(\sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'}) \right)$ ← less "noise" (variance).
"reward to go" $= \hat{Q}_{i,t}$

Baselines:
→ we want to "normalize" trajectories
$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau) [r(\tau) - b]$, $b = \frac{1}{N} \sum_{i=1}^{N} r(\tau)$. Proof: $E[\nabla_\theta \log \pi_\theta(\tau) b] = \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) b \, d\tau$

off-policy policy gradient

Policy gradient = on-policy (we must generate new samples every time) (gradient step)
↳ if we don't have samples from $\pi_\theta(\tau)$, but $\bar{\pi}(\tau)$: importance sampling

$E_{x \sim p(x)}[f(x)] = E_{x \sim q(x)} \left[\frac{p(x)}{q(x)} f(x) \right]$

$J(\theta) = E_{\tau \sim \bar{\pi}(\tau)} \left[\frac{\pi_\theta(\tau)}{\bar{\pi}(\tau)} r(\tau) \right]$ ← importance weight $= \frac{\prod_{t=1}^{T} \pi_\theta(a_t|s_t)}{\prod_{t=1}^{T} \bar{\pi}(a_t|s_t)}$

$\nabla_{\theta'} J(\theta') = E_{\tau \sim \pi_\theta(\tau)} \left[\frac{\pi_{\theta'}(\tau)}{\pi_\theta(\tau)} \nabla_{\theta'} \log \pi_{\theta'}(\tau) r(\tau) \right]$, $\theta \neq \theta'$.

$\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\pi_{\theta'}(s_{i,t}, a_{i,t})}{\pi_\theta(s_{i,t}, a_{i,t})} \nabla_{\theta'} \log \pi_{\theta'}(a_{i,t}|s_{i,t}) \hat{Q}_{i,t}$

$\tilde{J}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \log \pi_\theta(a_{i,t}|s_{i,t}) \hat{Q}_{i,t}$ ← reward

For backprop: same NLL loss, except we weigh by $\hat{Q}_{i,t}$. Minimal code change.