Importance Sampling

(ref: notes on course website;
not all contents in notes are covered in class)

Motivating scenario: off-policy evaluation

- Given π, estimate $J(\pi) := \mathbb{E}_{s \sim d_0}[V^{\pi}(s)]$
- Alg outputs some scalar v; accuracy measured by $|v - J(\pi)|$
- Previously we solved this problem by on-policy MC
- What if we have data collected using some other policy π_0?
    - Likely the case when we try to evaluate a trained policy using historical data (only meaningful for "real-life" app of RL)
- There are approaches you can already take from what we have learned so far
    - e.g., run expected Sarsa on the off-policy data, and output as $v = \mathbb{E}_{s \sim d_0}[\hat{Q}^{\pi}(s, \pi(s))]$ the estimate
    - requires function approximation, and is in general biased
- Is there an unbiased estimator?

Introduction to Importance Sampling (IS)

- Suppose we are interested in estimating $\mathbb{E}_{x \sim p}[f(x)]$
- If we have $x \sim p$, $f(x)$ would be an unbiased MC estimate
- What if we can only sample $x \sim q$, but still want a "MC-style" estimator?
- IS (or importance weighted, or inverse propensity score (IPS) estimator): $\frac{p(x)}{q(x)} f(x)$
- Unbiasedness:
$$\mathbb{E}_{x \sim q}\left[\frac{p(x)}{q(x)} f(x)\right] = \sum_x q(x)\left(\frac{p(x)}{q(x)} f(x)\right) = \sum_x p(x) f(x) = \mathbb{E}_{x \sim p}[f(x)]$$
- $\frac{p(x)}{q(x)}$: Importance weight (ratio), which "converts" the distribution from q (the data distribution) to p
- $\mathbb{E}_{x \sim q}\left[\frac{p(x)}{q(x)}\right] \equiv 1$: always holds!

Application in contextual bandit (CB)

- CB: episodic MDP with $H = 1$. Actions have no long-term effects. Just optimize the immediate reward.
  - $x \sim d_0$: context distribution (corresponds to initial state distribution of the MDP)
  - agent takes an action a based on x
  - agent observes reward $r \sim R(x, a)$
  - (episode terminates; no next-state)
- The off-policy evaluation problem
  - We have collected a dataset (a bag of (x, a, r) tuples), where $a \sim \pi_b(s)$ (π_b is stochastic)
  - want to know $J(\pi) := \mathbb{E}_{\pi}[r]$
    - The π in the subscript is short for $x \sim d_0, a \sim \pi, r \sim R(x, a)$
    - Let π be also stochastic (can be deterministic)

Application in contextual bandit (CB)

- The data point is a tuple (x, a, r)
- The function of interest is $(x, a, r) \mapsto r$

π: target policy

- The distribution of interest is $x \sim d_0$, $a \sim \pi$, $r \sim R(x, a)$
  - Let the joint density be $p(x, a, r)$

π_b: behavior/logging policy

- The data distribution is $x \sim d_0$, $a \sim \pi_b$, $r \sim R(x, a)$
  - Let the joint density be $q(x, a, r)$
- IS estimator: $\dfrac{p(x,a,r)}{q(x,a,r)} \cdot r$ $\quad \boxed{= \dfrac{\pi(a\,|\,x)}{\pi_b(a\,|\,x)} \cdot r}$
- Write down the densities
  - $p(x, a, r) = d_0(x) \cdot \pi(a \mid x) \cdot R(r \mid x, a)$
  - $q(x, a, r) = d_0(x) \cdot \pi_b(a \mid x) \cdot R(r \mid x, a)$
  - To compute importance weight, you don't need knowledge of μ or R! You just need π_b (or even just $\pi_b(a \mid x)$, "proposal prob.")

Application in contextual bandit (CB)

- Let ρ be a shorthand for $\dfrac{\pi(a|x)}{\pi_b(a|x)}$, so estimator is $\rho \cdot r$
- π_b need to "cover" π
    - i.e., whenever $\pi(a \mid x) > 0$, we need $\pi_b(a \mid x) > 0$
- A special case:
    - π is deterministic, and π_b is uniformly random ($\pi_b(a \mid x) \equiv 1/|A|$)
    - $\dfrac{\mathbb{I}[a = \pi(x)]}{1/|A|} \cdot r$
        - only look at actions that match what π wants to take, and discard other data points
        - If match, $\rho = |A|$; mismatch: $\rho = 0$
    - On average: only $1/|A|$ portion of the data is useful
    - Variance of ρ is $O(|A|)$

A note about using IS

- We know that shifting rewards do not matter (for planning purposes) for fixed-horizon problems
- However, when you apply IS, shifting rewards *do* impact the variance of the estimator
- Special case:
  - deterministic π, uniformly random π_b ,
  - reward is deterministic and constant: regardless of (x,a), reward is always 1 (without any randomness)
  - We know the value of any policy is 1
  - On-policy MC has 0 variance
  - IS still has high variance!

A note about using IS

- Where does variance come from?
  - $$\frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}[a^{(i)}=\pi(x^{(i)})]}{1/|A|}\cdot r^{(i)}=\sum_{i=1}^{n}\frac{\mathbb{I}[a^{(i)}=\pi(x^{(i)})]\cdot r^{(i)}}{n/|A|}$$
    $$=\frac{1}{n/|A|}\sum_{i:a^{(i)}=\pi(x^{(i)})}r^{(i)}$$
  - Find all “matched” data points, sum their rewards, then…
  - normalize by the *expected* # matched data points $n/|A|$
- You might think we should normalize by the actual # matched data points observed in data…
  - This is what weighted IS does (not required)
  - Generally a biased (but consistent) estimator, but much lower variance in some cases

Example Application: Off-policy TD(0)

- Recall that TD(0) is on-policy
- How to derive its off-policy version?
- Data: (s, a, r, s') where $a \sim \pi_b(s)$, but we want to learn V^{π}
- TD(0) target: $r + \gamma V(s')$ => learns V^{π_b}
- Off-policy TD(0) target: $\frac{\pi(a \mid s)}{\pi_b(a \mid s)}(r + \gamma V(s'))$

Multi-step IS in MDPs

- Data: trajectories starting from $s_1 \sim \mu$ using π_b (i.e., $a_t \sim \pi_b(s_t)$)
  $$\{(s_1^{(i)}, a_1^{(i)}, r_1^{(i)}, s_2^{(i)}, \ldots, s_H^{(i)}, a_H^{(i)}, r_H^{(i)})\}_{i=1}^n$$
  (for simplicity, assume process terminates in H time steps)
- Want to estimate $J(\pi) := \mathbb{E}_{s \sim d_0}[V^{\pi}(s)]$
- Same idea as in bandit: apply IS to the entire trajectory

Application in MDPs

- The data point is $\tau := (s_1, a_1, r_1, \ldots, s_H, a_H, r_H)$
- The function of interest is $\tau \mapsto \sum_{t=1}^{H} \gamma^{t-1} r_t$
- Let the distribution of trajectory induced by π be $p(\tau)$
- Let the distribution of trajectory induced by π_b be $q(\tau)$
- IS estimator: $\dfrac{p(\tau)}{q(\tau)} \cdot \displaystyle\sum_{t=1}^{H} \gamma^{t-1} r_t$
- Write down the densities (assume deterministic reward for simplicity)
    - $p(\tau) = d_0(s_1) \cdot \pi(a_1 | s_1) \cdot P(s_2 | s_1, a_1) \cdot \pi(a_2 | s_2) \cdots P(s_H | s_{H-1}, a_{H-1}) \cdot \pi(a_H | s_H)$
    - $q(\tau) = d_0(s_1) \cdot \pi_b(a_1 | s_1) \cdot P(s_2 | s_1, a_1) \cdot \pi_b(a_2 | s_2) \cdots P(s_H | s_{H-1}, a_{H-1}) \cdot \pi_b(a_H | s_H)$
    - Let $\rho_t = \dfrac{\pi(a_t | s_t)}{\pi_b(a_t | s_t)}$, then $\dfrac{p(\tau)}{q(\tau)} = \displaystyle\prod_{t=1}^{H} \rho_t =: \rho_{1:H}$

Examine the special case again

- π is deterministic, and π_b is uniformly random ($\pi_b(a \mid x) \equiv 1/|A|$)
- $\rho_t = \frac{\mathbb{I}[a_t = \pi(s_t)]}{1/|A|}$
- only look at trajectories where **all actions** happen to match what π wants to take
  - If match, $\rho = |A|^H$; mismatch: $\rho = 0$
- On average: only $1/|A|^H$ portion of the data is useful
  - (When state space is unboundedly large, can prove that $|A|^H$ is inevitable; a version of "curse of horizon" in RL)
- When horizon is long, mostly applied when π and π_b are close to each other

An obvious improvement: step-wise IS

- "trajectory-wise" IS: $\rho_{1:H}\left(\sum_{t=1}^{H}\gamma^{t-1}r_t\right)$
- Idea: estimate the expected reward for each time step t, and then add them up
  - i.e., $J(\pi) = \sum_{t=1}^{H}\gamma^{t-1}\mathbb{E}[r_t \mid s_1 \sim d_0, \pi]$
  - When estimating $\mathbb{E}[r_t \mid s \sim d_0, \pi]$, we know that decisions made after time step t are irrelevant; truncate at time step t
  - Improved estimator: $\sum_{t=1}^{H}\gamma^{t-1}\cdot\rho_{1:t}\cdot r_t$
  - Equivalent to trajectory-wise IS when intermediate rewards are all 0