Offline Reinforcement Learning

Policy Evaluation

Amit.K.Singh

Self learning

December 31, 2022

Outline

Problem definition

Let,

- State space $S = \{s_1, s_2, s_3 ...\}$
- Random variable X s.t. $X(s_i) \to R$
- Probability distribution over state space is $p = \{p_{s1}, p_{s2}, p_{s3}, ...\}$
- Dataset D_p is collected from S with distribution p. For example $D_p = \{s_1, s_5, s_3, s_1, ...\}$

The objective is to find $\mu_S(p; D_p)$.

$$\mu_S(p; D_p) = \frac{1}{|D_p|} \sum_{s_i \in D_p} X(s_i) \approx \mathbb{E}_p[X(s) : s \in S] = \sum_{s \in S} X(s) \cdot p(s)$$

Inverse Propensity Score

- How to compute $\mu_S(p_e; D_{p0})$?
- $\mu_S(p_e; D_{p0}) = \sum_{s \in S} X(s) \cdot p_e(s) = \sum_{s \in S} X(s) \cdot \frac{p_e(s)}{p_0(s)} \cdot p_0(s)$
- $\mu_S(p_e; D_{p0}) = \mathbb{E}_{p0}[X(s) \cdot \frac{p_e(s)}{p_0(s)}]$

$$\mu_S(p_e; D_{p0}) = \frac{1}{|D_{p0}|} \sum_{s_i \in D_{p0}} X(s_i) \cdot \frac{p_e(s)}{p_0(s)}$$

- Unbiased estimator
- Has high variance due to $\frac{p_e(s)}{p_0(s)}$ term.
- p_0 is not known and approximated by dataset D_{p0}.

Clipped Inverse Propensity Score (CIPS)

$$\mu_S(p_e; D_{p0}) = \frac{1}{|D_{p0}|} \sum_{s_i \in D_{p0}} X(s_i) \cdot \min\{\lambda, \frac{p_e(s)}{p_0(s)}\}$$

- Additional hyperparameter tuning required.
- Low variance but biased.

Self-Normalized IPS (SNIPS)

$$\mu_S(p_e; D_{p0}) = \frac{\frac{1}{|D_{p0}|}\sum_{s_i \in D_{p0}} X(s_i) \cdot \frac{p_e(s)}{p_0(s)}}{\frac{1}{|D_{p0}|}\sum_{s_i \in D_{p0}} \frac{p_e(s)}{p_0(s)}} = \frac{\sum_{s_i \in D_{p0}} X(s_i) \cdot \frac{p_e(s)}{p_0(s)}}{\sum_{s_i \in D_{p0}} \frac{p_e(s)}{p_0(s)}}$$

- SNIPS is enough for easy settings.
- It works well without any hyperparameters
- It fails when the deviation between p_0 and p_e is large.

Multi-armed bandit (MAB)

MAB can be seen as a single state Reinforcement learning (RL) formulation. Mathematically we can define MAB as follows:
Let,

- $S = \{s_0, s_1, ...\}$ is the state space.
- $A = \{a_0, a_1, ...\}$ is action space and $|A| = K$.
- $R = \{r_0, r_1, ...\}$ is associated with each action $a_k \in A$, representing mean of reward distribution.
- $r(s,a) \sim \mathcal{N}(r_k, \sigma^2)$.
- Policy $\pi = p(a|s)$ represents a distribution over action space conditioned on input state.
- Goal of a learning algorithm is to find $\pi = \underset{\pi \in \prod}{\mathrm{argmax}} \mathbb{E}[r]$.

Policy Evaluation

To evaluate a policy,

- Dataset $D_\pi = \{(s_i, a_i, r_i)\}_{i=1}^{n}$ is generated by n interactions of MAB using policy π.
- Expected reward is defined as $\mathbb{E}[r] = \frac{1}{n}\sum_{i=1}^{n} r_i = \hat{V}(\pi; D_\pi) \approx V(\pi)$.
- Our goal is to evaluate the policy π_e using a dataset D_{π_0} generated using behavior policy π_0.
- Hypothesis is $\hat{V}(\pi_e; D_{\pi_0}) \approx V(\pi_e)$.

Experiments

Evaluate optimal policy π^* using dataset collected from random policy D_π i.e. $V(\pi^*) \approx \hat{V}(\pi^*; D_{\pi \in \prod})$

- $\hat{V}_{IPS} = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi^*(a_i|s_i)}{\pi(a_i|s_i)} * r_i$.
- V_{IPS} is theoretically an unbiased.
- Variance in $\hat{V}_{IPS}(\pi^*; D_\pi, n)$ reduces as n increases.

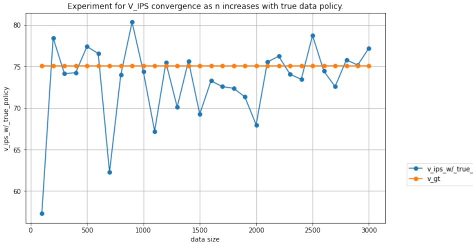

Figure: $\hat{V}_{IPS}(\pi^*; D_\pi, n)$

- Offline Reinforcement Learning
- Offline Multi-Armed Bandit (MAB)

Observations

- Data policy π is not known and estimated using a dataset $\hat{\pi}$.
- $\hat{V}_{IPS} = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi^*(a_i|s_i)}{\hat{\pi}(a_i|s_i)} * r_i$.
- Variance in $\hat{V}_{IPS}(\pi^*; D_\pi, n)$ significantly larger than previous case.

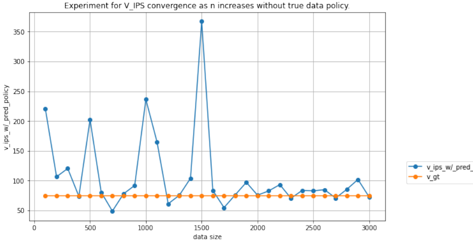

Figure: $\hat{V}_{IPS}(\pi^*; D_\pi, n)$

Observations

- Variance in V_{SNIPS} reduces as n increases.
- V_{SNIPS} is less sensitive to data policy approximation.
- Practically biased.

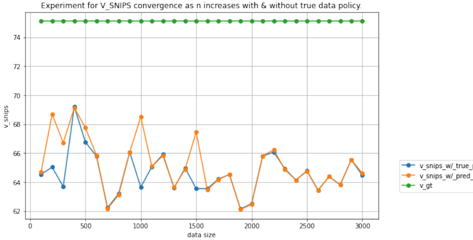

Figure: $\hat{V}_{SNIPS}(\pi^*; D_\pi, n)$

Observations

- V_{SNIPS} is almost always biased for any data policy $\pi \in \prod$.
- Major cause of the error is a deviation between the data policy and evaluation policy.

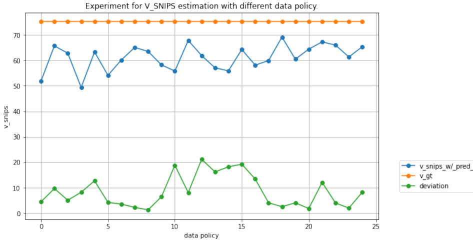

Figure: $\hat{V}_{SNIPS}(\pi^*; D_{\pi \in \prod})$

Distribution correction via resampling

- Sample state $s \in \mathrm{D}_\pi(s, a, r)$.
- Get action using evaluation policy $a \sim \pi_e(s)$.
- $\hat{r}(s, a)$ is estimated with D.
- $\hat{D}_{\pi_e} = \{(s, a, \hat{r}(s, a))\}$
- $\hat{r}(s, a)$ is Counterfactual Evaluation. For this experiment, KNN is used.

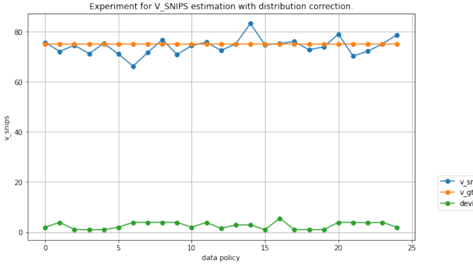

Figure: $\hat{V}_{SNIPS}(\pi^*; D_{\pi \in \prod})$

References

- Off-Policy Deep Reinforcement Learning without Exploration.
- PLAS: Latent Action Space for Offline Reinforcement Learning.
- Addressing Extrapolation Error in Deep Offline Reinforcement Learning.
- ConQUR: Mitigating Delusional Bias in Deep Q-learning.
- Distilled Thompson Sampling: Practical and Efficient Thompson Sampling via Imitation Learning.
- A Contextual-Bandit Approach to Personalized News Article Recommendation.
- Counterfactual risk minimization.
- Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction.
- Conservative Q-Learning for Offline Reinforcement Learning.
- COG: Connecting New Skills to Past Experience with Offline Reinforcement Learning.
- Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems.
- A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems.
- Counterfactual Learning and Evaluation for Recommender Systems.