Batch / Offline RL

Emma Brunskill
CS234
Winter 2021

Thanks to Phil Thomas for some figures
Refresh Your Understanding: Fast RL

Select all that are true:

- In Thompson sampling for MDPs, the posterior over the dynamics can be updated after each transition
- When using a Beta prior for a Bernoulli reward parameter for an (s,a) pair, the posterior after N samples of that pair time steps can be the same as after N+2 samples
- The optimism bonuses discussed for MBIE-EB depend on the maximum reward but not on the maximum value function
- In class we discussed adding a bonus term to an update for a (s,a,r,s’) tuple using Q-learning with function approximation. Adding this bonus term will ensure all Q estimates used to make decisions online using DQN are optimistic with respect to Q*
- Not sure
Refresh Your Understanding: Fast RL  Solutions

Select all that are true:

- In Thompson sampling for MDPs, the posterior over the dynamics can be updated after each transition (True)
- When using a Beta prior for a Bernoulli reward parameter for an (s,a) pair, the posterior after N samples of that pair time steps can be the same as after N+2 samples (False)
  - Beta(alpha,beta) could be Beta(alpha+2,beta), Beta(alpha+1,beta+1), Beta(alpha,beta+2)
- The optimism bonuses discussed for MBIE-EB depend on the maximum reward but not on the maximum value function (False)
  - The optimism bonuses depend on the max value
- In class we discussed adding a bonus term to an update for a (s,a,r,s’) tuple using Q-learning with function approximation. Adding this bonus term will ensure all Q estimates used to make decisions online using DQN are optimistic with respect to Q* (False)
  - Function approximation may mean that the resulting estimate is not always optimistic
Outline for Today

1. Introduction and Setting
2. Offline batch evaluation using models
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
Class Progress

- Last time: Fast RL III
- This time: Batch RL
- Next time: Guest lecture with Professor Doshi-Velez
Reinforcement Learning

\[ r(s_t, a_t) \]
\[ s_t \in S \]
\[ \pi_t(s_t) \rightarrow a_t \]
\[ a_t \in A \]

\[ V^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, a) V^\pi(s') \]

Value func.
Reward
Dynamics

Only observed through samples (experience)
Today: Counterfactual / Batch RL

\[ r(s_t, a_t) \]
\[ s_t \in S \]
\[ \pi_t(s_t) \rightarrow a_t \]
\[ a_t \in A \]

\[ D: \text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b \]
Outline for Today

1. Introduction and Setting
2. Offline batch evaluation using models
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
Patient group 1 → 

Patient group 2 →
Patient group 1 → Medicine 1 → Doctor → Outcome: 92

Patient group 2 → Medicine 2 → Medicine 3 → Outcome: 91

?
“What If?” Reasoning Given Past Data

Patient group 1 →  bottle  →  stethoscope  →  Outcome: 92

Patient group 2 →  bottle  →  bottle  →  Outcome: 91

?
Data Is Censored in that Only Observe Outcomes for Decisions Made

Patient group 1 →  Bottle  Stethoscope  →  Outcome: 92

Patient group 2 →  Bottle  Bottle  →  Outcome: 91

?
Need for Generalization

Outcome: 92
Outcome: 91
Outcome: 85

?
Off Policy Reinforcement Learning

Watkins 1989
Watkins and Dayan 1992
Precup et al. 2000
Lagoudakis and Parr 2002
Murphy 2005
Sutton, Szepesvari and Maei 2009
Shortreed, Laber, Lizotte, Stroup, Pineau, & Murphy 2011
Degirs, White, and Sutton 2012
Mnih et al. 2015
Mahmood et al. 2014
Jiang & Li 2016
Hallak, Tamar and Mannor 2015
Munos, Stepleton, Harutyunyan and Bellemare 2016
Sutton, Mahmood and White 2016
Du, Chen, Li, Ziao, and Zhou 2016
Why Can’t We Just Use Q-Learning?

- Q-learning is an off policy RL algorithm
  - Can be used with data different than the state--action pairs would visit under the optimal Q state action values

- But deadly triad of bootstrapping, function approximation and off policy, and can fail
Important in Practice

BCQ figure from Fujimoto, Meger, Precup ICML 2019
Challenge: Overlap Requirement

Policy wish to evaluate

Probability of intervention

Antibiotics  Mechanical Ventilation  Vasopressor
Overlap Requirement: Data Must Support Policy Wish to Evaluate

Probability of intervention

- Antibiotics
- Mechanical Ventilation
- Vasopressor

Policy used to gather data
Policy wish to evaluate
No Overlap for Vasopressor ⇒ Can’t Do Off Policy Estimation for Desired Policy
How to Evaluate Sufficient Overlap in Real Data?

Policy used to gather data

Policy wish to evaluate

Probability of intervention

Antibiotics  Mechanical Ventilation  Vasopressor
Offline / Batch Reinforcement Learning

Assumptions

Evaluation Criteria

Tasks

\( D \): Dataset of \( n \) trajectories \( \tau, \tau \sim \pi_b \)

\( \pi \): Policy mapping \( s \rightarrow a \)

\( S_0 \): Set of initial states

\( \hat{V}^\pi(s, D) \): Estimate \( V(s) \) with dataset \( D \)
Common Tasks: Off Policy Evaluation & Optimization

\[ \int_{s \in S_0} \hat{V}^\pi(s, D) \, ds \]

\[ \arg \max_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^\pi(s, D) \, ds \]

\( \mathcal{D} \): Dataset of \( n \) trajectories \( \tau \), \( \tau \sim \pi_b \)

\( \pi \): Policy mapping \( s \rightarrow a \)

\( S_0 \): Set of initial states

\( \hat{V}^\pi(s, D) \): Estimate \( V(s) \) w/dataset \( D \)
Common Assumptions

- **Stationary process**: Policy will be evaluated in or deployed in the same stationary decision process as the behavior policy operated in to gather data.
- **Markov**
- **Sequential ignorability (no confounding)**
  \[
  \{Y(A_{1:(t-1)}, a_{t:T}), S_{t'}(A_{1:(t-1)}, a_{t:(t'-1)})\}_{t'=t+1}^T \perp A_t \mid \mathcal{F}_t
  \]
- **Overlap**
  \[
  \forall (s, a) \mu_e(s, a) > 0 \rightarrow \mu_b(s, a) > 0
  \]

\[D\]: Dataset of \( n \) trajectories \( \tau, \tau \sim \pi_b \)

\(\pi\): Policy mapping \( s \rightarrow a \)

\(S_0\): Set of initial states

\(\hat{V}^\pi(s, D)\): Estimate \( V(s) \) with dataset \( D \)
Common Tasks: Off Policy Evaluation & Optimization

Tasks

\[
\int_{s \in S_0} \hat{V}^\pi(s, D) \, ds
\]

arg max \limits_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^\pi(s, D) \, ds

Assumptions

Evaluation Criteria

\(D\): Dataset of \(n\) trajectories, \(\tau\), \(\tau \sim \pi_b\)

\(\pi\): Policy mapping \(s \rightarrow a\)

\(S_0\): Set of initial states

\(\hat{V}^\pi(s, D)\): Estimate \(V(s)\) with dataset \(D\)
Off Policy Reinforcement Learning

The 3D space of all value functions

The subspace of all value functions representable as $v_w$

Figure from Sutton & Barto 2018
Off Policy Reinforcement Learning

The 3D space of all value functions

Bellman error vector (BE)

$B_\pi v_w$

$\Pi B_\pi v_w$

The subspace of all value functions representable as $v_w$

Figure from Sutton & Barto 2018
Batch Off Policy Reinforcement Learning

The 3D space of all value functions

Bellman error vector (BE)

\( B_\pi v_w \)

The subspace of all value functions representable as \( v_w \)

Figure from Sutton & Barto 2018
Batch Off Policy Reinforcement Learning

The 3D space of all value functions over 3 states

The subspace of all value functions representable as $v_w$

Figure from Sutton & Barto 2018
Common Evaluation Criteria for Off Policy Evaluation

- Computational efficiency
- Performance accuracy

\[ \forall D_i \in \{ D_1 \sim M_1, D_2 \sim M_2, \ldots, D_K \sim M_K \} \quad \frac{1}{|\rho|} \sum_{s_0 \in \rho} (\hat{V}_{\hat{M}_i}(s_0, D_i) - V_{\hat{M}_i}(s_0))^2 \]

\[ \lim_{|D| \to \infty} \frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}_\pi(s_0, D) \to \frac{1}{|\rho|} \sum_{s_0 \in \rho} V_\pi(s_0) \]

\[ \frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}_\pi(s_0, D) \leq \frac{1}{|\rho|} \sum_{s_0 \in \rho} V_\pi(s_0) - f(n, \ldots) \]

\( D \): Dataset of \( n \) traj.s \( \tau, \tau \sim \pi_b \)
\( \pi \): Policy mapping \( s \to a \)
\( S_0 \): Set of initial states
\( \hat{V}_\pi(s, D) \): Estimate \( V(s) \) w/dataset \( D \)
Offline / Batch Reinforcement Learning

Tasks
\[
\int_{s \in S_0} \hat{V}^\pi(s, \mathcal{D}) ds
\]
\[
\text{arg max}_{\pi \in \mathcal{H}_i} \int_{s \in S_0} \hat{V}^\pi(s, \mathcal{D}) ds
\]

Evaluation Criteria
- Empirical accuracy
- Consistency
- Robustness
- Asymptotic efficiency
- Finite sample bounds
- Computational cost

Assumptions
- Markov?
- Overlap?
- Sequential ignorability?

\[\mathcal{D}: \text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b\]
\[\pi: \text{Policy mapping } s \to a\]
\[S_0: \text{Set of initial states}\]
\[\hat{V}^\pi(s, \mathcal{D}): \text{Estimate } V(s) \text{ w/dataset } \mathcal{D}\]
Batch Policy Optimization: Find a Good Policy That Will Perform Well in the Future

\[
\arg \max_{\pi \in \mathcal{H}_i} \max_{\mathcal{H}_i \in \mathcal{H}_1, \mathcal{H}_2, \ldots} \int_{s \in S_0} \hat{V}^\pi(s, D) ds
\]

Policy Optimization

Policy Evaluation

\[\mathcal{H} = M, V, \Pi?\]

\[D: \text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b \]
\[\pi: \text{Policy mapping } s \rightarrow a \]
\[S_0: \text{Set of initial states} \]
\[\hat{V}^\pi(s, D): \text{Estimate } V(s) \text{ w/dataset } D \]
Batch Policy Evaluation: Estimate the Performance of a Particular Decision Policy

\[
\arg \max_{\pi \in \mathcal{H}_i} \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}} \int_{s \in S_0} \hat{V}^\pi(s, \mathcal{D}) ds
\]

Policy Optimization

Policy Evaluation

\(\mathcal{D}\): Dataset of \(n\) trajectories, \(\tau \sim \pi_b\)
\(\pi\): Policy mapping \(s \rightarrow a\)
\(S_0\): Set of initial states
\(\hat{V}^\pi(s, \mathcal{D})\): Estimate \(V(s)\) with dataset \(\mathcal{D}\)
Outline

1. Introduction and Setting
2. **Offline batch evaluation using models**
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
5. Safe batch RL
Learn Dynamics and Reward Models from Data, Evaluate Policy

\[ \hat{r}(s, a) \]
\[ \hat{p}(s'|s, a) \]
\[ \pi_t(s_t) \rightarrow a_t \]

\[ V^\pi \approx (I - \gamma \hat{P}^\pi)^{-1} \hat{R}^\pi \]

\[ P^\pi(s'|s) = p(s'|s, \pi(s)) \]

- Mannor, Simster, Sun, Tsitsiklis 2007

\[ \mathcal{D} \]: Dataset of \( n \) traj.s \( \tau, \tau \sim \pi_b \)
\[ \pi \]: Policy mapping \( s \rightarrow a \)
\[ S_0 \]: Set of initial states
\[ \hat{V}^\pi(s, \mathcal{D}) \]: Estimate \( V(s) \) w/dataset \( \mathcal{D} \)
Better Dynamics/Reward Models for Existing Data (Improve likelihood)
Better Dynamics/Reward Models for Existing Data, May **Not** Lead to Better Policies for Future Use → Bias due to Model Misspecification
Models Fit for Off Policy Evaluation Can Result in Better Estimates When Trained Under a **Different Loss Function**

Liu, Gottesman, Raghu, Komorowski, Faisal, Doshi-Velez, Brunskill NeurIPS 2018
Outline

1. Introduction and Setting
2. Offline batch evaluation using models
3. **Offline batch evaluation using Q functions**
4. Offline batch evaluation using importance sampling
Model Free Value Function Approximation

\[ \mathcal{D} = (s_i, a_i, r_i, s_{i+1}) \quad \forall i \]

\[ \tilde{Q}^\pi(s_i, a_i) = r_i + \gamma V^\pi_\theta(s_{i+1}) \]

\[ \arg \min_\theta \sum_i (Q^\pi_\theta(s_i, a_i) - \tilde{Q}^\pi(s_i, a_i))^2 \]

- Fitted Q evaluation, LSTD, ...
Example Fitted Q Evaluation Guarantees

\[ d_{F}^{\pi} = \sup_{g \in F} \inf_{f \in F} \| f - B^{\pi} g \|_{\pi} \]

**Theorem 4.2** (Generalization error of FQE). Under Assumption 1, for \( \epsilon > 0 \) & \( \delta \in (0, 1) \), after \( K \) iterations of Fitted Q Evaluation (Algorithm 3), for \( n = O \left( \frac{C^4}{\epsilon^2} \left( \log \frac{K}{\delta} + \dim_{F} \log \frac{C^2}{\epsilon^2} + \log \dim_{F} \right) \right) \), we have with probability \( 1 - \delta \):

\[
\left| \int_{s_{0} \in \rho} \hat{V}_{\pi}(s_{0}) - V_{\pi}(s_{0}) \right| \leq \frac{\gamma^{5}}{(1 - \gamma)^{1.5}} \left( \sqrt{\beta_{u} (2d_{F}^{\pi} + \epsilon) + \frac{2\gamma^{K/2} C}{(1 - \gamma)^{5}}} \right)
\]

\( \mathcal{D} \): Dataset of \( n \) traj.s \( \tau, \tau \sim \pi_{b} \)
\( \pi \): Policy mapping \( s \rightarrow a \)
\( S_{0} \): Set of initial states
\( \hat{V}_{\pi}(s, \mathcal{D}) \): Estimate \( V(s) \) w/dataset \( \mathcal{D} \)

Le, Voloshin, Yue ICML 2019
Model Free Policy Evaluation

- Challenge: still relies on Markov assumption
- Challenge: still relies on models being well specified or have no computable guarantees if there is misspecification
Outline

1. Introduction and Setting
2. Offline batch evaluation using models
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
Off Policy Evaluation With Minimal Assumptions

- Would like a method that doesn’t rely on models being correct or Markov assumption
- Monte Carlo methods did this for online policy evaluation
- We would like to do something similar
- Challenge: data distribution mismatch
Computing Expected Return Under a Distribution

$$\mathbb{E}_p[r] = \sum_x p(x)r(x)$$
Computing Expected Return Under a Alternate Distribution: Simple Idea

\[ \mathbb{E}_p[r] = \sum_{x} p(x)r(x) \]
## Computing Expected Return Under a Alternate Distribution:
### Simple Idea, Worked Example

<table>
<thead>
<tr>
<th></th>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian mean</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Behavior policy q</td>
<td>0.2</td>
<td>0.5</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Evaluation policy p</td>
<td>0.8</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Num samples from behavior q</td>
<td>20</td>
<td>50</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Why Did This Fail?

<table>
<thead>
<tr>
<th></th>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian mean</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Behavior policy q</td>
<td>0.2</td>
<td>0.5</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Evaluation policy p</td>
<td>0.8</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Num samples from behavior q</td>
<td>20</td>
<td>50</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Importance Sampling

$$\mathbb{E}_p[r] = \sum_x p(x) r(x)$$
Importance Sampling: Can Compute Expected Value Under An Alternate Distribution!

\[ \mathbb{E}_p[r] = \sum_x p(x) r(x) = \sum_x \frac{p(x)q(x)}{q(x)} r(x) \approx \frac{1}{N} \sum_{i=1, x \sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i) \]
Importance Sampling is an Unbiased Estimator of True Expectation Under Desired Distribution If

\[ \mathbb{E}_p[r] = \sum_x p(x) r(x) \]

\[ = \sum_x \frac{p(x) q(x)}{q(x)} r(x) \]

\[ \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i)}{q(x_i)} r(x_i) \]

- The sampling distribution \( q(x) > 0 \) for all \( x \) s.t. \( p(x) > 0 \) (Coverage\slash overlap)
- No hidden confounding
Importance Sampling (IS) Example

$$\mathbb{E}_p[r] = \sum_x p(x) r(x)$$

$$= \sum_x \frac{p(x) q(x)}{q(x)} r(x)$$

$$\approx \frac{1}{N} \sum_{i=1, x \sim q}^N \frac{p(x_i)}{q(x_i)} r(x_i)$$

<table>
<thead>
<tr>
<th>Evaluation policy p</th>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian mean</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Behavior policy q</td>
<td>0.2</td>
<td>0.5</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>Num samples from behavior q</td>
<td>20</td>
<td>50</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

$$q = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.15 \\ 0.15 \end{bmatrix}$$

$$p = \begin{bmatrix} 0.8 \\ 0.2 \\ 0 \\ 0 \end{bmatrix}$$
Importance Sampling (IS) Example

\[
\mathbb{E}_p[r] = \sum_x p(x) r(x)
\]

\[
= \sum_x \frac{p(x)q(x)}{q(x)} r(x)
\]

\[
\approx \frac{1}{N} \sum_{i=1, x \sim q}^N p(x_i) \frac{r(x_i)}{q(x_i)}
\]

\(X = \text{arms}\)

Expected reward for following behavior policy? \(0.2 \times 10 + 0.5 \times 1 + 0 \times 0.15 + 0.15 \times 0.5\)

Expected reward for target policy p? \(0.8 \times 10 + 0.2 \times 1 = 8.2\)

Computing expected reward for p using IS: \(\frac{20}{100} \times \frac{0.8}{0.2} \times 10 + \frac{50}{100} \times \frac{0.2}{0.5} \times 1 = 8.2\)
Check Your Understanding: Importance Sampling

We can use importance sampling to do batch bandit policy evaluation. Consider we have a dataset for pulls from 3 arms. Consider that arm 1 is a Bernoulli where with probability .98 we get 0 and with probability 0.02 we get 100. Arm 2 is a Bernoulli where with probability 0.55 the reward is 2 else the reward is 0. Arm 3 has a probability of yielding a reward of 1 with probability 0.5 else it gets 0. Select all that are true.

- Data is sampled from pi1 where with probability 0.8 it pulls arm 3 else it pulls arm 2. The policy we wish to evaluate, pi2, pulls arm 2 with probability 0.5 else it pulls arm 1. pi2 has higher true reward than pi1.
- We cannot use pi1 to get an unbiased estimate of the average reward pi2 using importance sampling.
- If rewards can be positive or negative, we can still get a lower bound on pi2 using data from pi1 using importance sampling.
- Now assume pi1 selects arm1 with probability 0.2 and arm2 with probability 0.8. We can use importance sampling to get an unbiased estimate of pi2 using data from pi1.
- Still with the same pi1, it is likely with N=20 pulls that the estimate using IS for pi2 will be higher than the empirical value of pi1.
- Not Sure
Check Your Understanding: Importance Sampling Answers

We can use importance sampling to do batch bandit policy evaluation. Consider we have a dataset for pulls from 3 arms. Consider that arm 1 is a Bernoulli where with probability 0.98 we get 0 and with probability 0.02 we get 100. Arm 2 is a Bernoulli where with probability 0.55 the reward is 2 else the reward is 0. Arm 3 has a probability of yielding a reward of 1 with probability 0.5 else it gets 0. Select all that are true.

- Data is sampled from π₁ where with probability 0.8 it pulls arm 3 else it pulls arm 2. The policy we wish to evaluate, π₂, pulls arm 2 with probability 0.5 else it pulls arm 1. π₂ has higher true reward than π₁. (True)
- We cannot use π₁ to get an unbiased estimate of the average reward π₂ using importance sampling. (True, π₁ never pulls arm 1 which is taken by π₂)
- If rewards can be positive or negative, we can still get a lower bound on π₂ using data from π₁ using importance sampling (False, only if rewards are positive)
- Now assume π₁ selects arm 1 with probability 0.2 and arm 2 with probability 0.8. We can use importance sampling to get an unbiased estimate of π₂ using data from π₁. (True)
- Still with the same π₁, it is likely with N=20 pulls that the estimate using IS for π₂ will be higher than the empirical value of π₁. (False)
Importance Sampling for RL Policy Evaluation

$$V^\pi(s) = \sum_{\tau} p(\tau | \pi, s) R(\tau)$$

$\mathcal{D}$: Dataset of $n$ trajectories $\tau$, $\tau \sim \pi_b$

$\pi$: Policy mapping $s \rightarrow a$

$S_0$: Set of initial states

$\hat{V}^\pi(s, \mathcal{D})$: Estimate $V(s)$ with dataset $\mathcal{D}$
Importance Sampling for RL Policy Evaluation

\[ V^{\pi}(s) = \sum_{\tau} p(\tau | \pi, s) R(\tau) \]

\[ = \sum_{\tau} p(\tau | \pi_b, s) \frac{p(\tau | \pi, s)}{p(\tau | \pi_b, s)} R_{\tau} \]

\[ \approx \sum_{i=1, \tau_i \sim \pi_b}^{N} \frac{p(\tau_i | \pi, s)}{p(\tau_i | \pi_b, s)} R_{\tau_i} \]

\[ \mathcal{D}: \text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b \]
\[ \pi: \text{Policy mapping } s \rightarrow a \]
\[ S_0: \text{Set of initial states} \]
\[ \hat{V}^{\pi}(s, \mathcal{D}): \text{Estimate } V(s) \text{ w/dataset } \mathcal{D} \]
Importance Sampling for RL Policy Evaluation

\[
V^\pi(s) = \sum_\tau p(\tau | \pi, s) R(\tau)
\]

\[
= \sum_\tau p(\tau | \pi_b, s) \frac{p(\tau | \pi, s)}{p(\tau | \pi_b, s)} R_\tau
\]

\[
\approx \sum_{i=1, \tau_i \sim \pi_b}^N \frac{p(\tau_i | \pi, s)}{p(\tau_i | \pi_b, s)} R_{\tau_i}
\]

\[
= \sum_{i=1, \tau_i \sim \pi_b}^N \prod_{t=1}^{H_i} p(s_{i,t+1}|s_{i,t}, a_{i,t}) p(a_{i,t}|\pi, s_{i,t})
\frac{p(s_{i,t+1}|s_{i,t}, a_{i,t}) p(a_{i,t}|\pi_b, s_{i,t})}{p(a_{i,t}|\pi_b, s_{i,t})}
\]

\[
= \sum_{i=1, \tau_i \sim \pi_b}^N \prod_{t=1}^{H_i} \frac{p(a_{i,t}|\pi, s_{i,t})}{p(a_{i,t}|\pi_b, s_{i,t})}
\]

- **D**: Dataset of \( n \) traj.s \( \tau \), \( \tau \sim \pi_b \)
- \( \pi \): Policy mapping \( s \rightarrow a \)
- \( S_0 \): Set of initial states
- \( \hat{V}^\pi(s, D) \): Estimate \( V(s) \) w/dataset \( D \)
Importance Sampling for RL Policy Evaluation: Don’t Need to Know Dynamics Model!

\[
V^\pi(s) = \sum_{\tau} p(\tau | \pi, s) R(\tau)
\]

\[
= \sum_{\tau} p(\tau | \pi_b, s) \frac{p(\tau | \pi, s)}{p(\tau | \pi_b, s)} R(\tau)
\]

\[
\approx \sum_{i=1, \tau_i \sim \pi_b}^N \frac{p(\tau_i | \pi, s)}{p(\tau_i | \pi_b, s)} R_{\tau_i}
\]

\[
= \sum_{i=1, \tau_i \sim \pi_b}^N R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(s_{i,t+1} | s_{it}, a_{it}) p(a_{it} | \pi, s_{it})}{p(s_{i,t+1} | s_{it}, a_{it}) p(a_{it} | \pi_b, s_{it})}
\]

\[
= \sum_{i=1, \tau_i \sim \pi_b}^N R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(a_{it} | \pi, s_{it})}{p(a_{it} | \pi_b, s_{it})}
\]

\[\mathcal{D}:\text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b\]
\[\pi: \text{ Policy mapping } s \rightarrow a\]
\[S_0: \text{ Set of initial states}\]
\[\hat{V}^\pi(s, \mathcal{D}): \text{Estimate } V(s) \text{ w/dataset } \mathcal{D}\]

- First used for RL by Precup, Sutton & Singh 2000. Recent work includes: Thomas, Theocharous, Ghavamzadeh 2015; Thomas and Brunskill 2016; Guo, Thomas, Brunskill 2017; Hanna, Niekum, Stone 2019
Importance Sampling

- Does not rely on Markov assumption
- Requires minimal assumptions
- Provides unbiased estimator
- Similar to Monte Carlo estimator but corrects for distribution mismatch
Check Your Understanding: Importance Sampling 2

Select all that you’d guess might be true about importance sampling

- It requires the behavior policy to visit all the state--action pairs that would be visited under the evaluation policy in order to get an unbiased estimator
- It is likely to be high variance
- Not Sure
Check Your Understanding: Importance Sampling 2 Answers

Select all that you’d guess might be true about importance sampling

- It requires the behavior policy to visit all the state--action pairs that would be visited under the evaluation policy in order to get an unbiased estimator (True)
- It is likely to be high variance (True)
- Not Sure
Per Decision Importance Sampling (PDIS)

- Leverage temporal structure of the domain (similar to policy gradient)

\[ IS(D) = \frac{1}{n} \sum_{i=1}^{n} \left( \prod_{t=1}^{L} \frac{\pi_e(a_{t}\mid s_{t})}{\pi_b(a_{t}\mid s_{t})} \right) \left( \sum_{t=1}^{L} \gamma^t R_t^i \right) \]

\[ PSID(D) = \sum_{t=1}^{L} \gamma^t \frac{1}{n} \sum_{i=1}^{n} \left( \prod_{\tau=1}^{t} \frac{\pi_e(a_{\tau}\mid s_{\tau})}{\pi_b(a_{\tau}\mid s_{\tau})} \right) R_t^i \]
Importance Sampling Variance

- Importance sampling, like Monte Carlo estimation, is generally high variance.
- Importance sampling is particularly high variance for estimating the return of a policy in a sequential decision process.

\[ R_{\tau_i} = \sum_{i=1, \tau_i \sim \pi_b}^{N} \prod_{t=1}^{H_i} \frac{p(a_{it} | \pi, s_{it})}{p(a_{it} | \pi_b, s_{it})} \]

- Variance can generally scale exponentially with the horizon.
  a. Concentration inequalities like Hoeffding scale with the largest range of the variable.
  b. The largest range of the variable depends on the product of importance weights.
  c. Check your understanding: for a H step horizon with a maximum reward in a single trajectory of 1, and if \( p(a|s, \pi_b) = .1 \) and \( p(a|s, \pi) = 1 \) for each time step, what is the maximum importance-weighted return for a single trajectory?

\[ R_{\tau_i} = \prod_{t=1}^{H_i} \frac{p(a_{it} | \pi, s_{it})}{p(a_{it} | \pi_b, s_{it})} \]
Outline

1. Introduction and Setting
2. Offline batch evaluation using models
3. Offline batch evaluation using Q functions
4. Offline batch evaluation using importance sampling
5. Example application & What You Should Know
Preventing undesirable behavior of intelligent machines

Philip S. Thomas¹, Bruno Castro da Silva², Andrew G. Barto¹, Stephen Giguere¹,
Yuriv Brun¹, Emma Brunskill³

Science November 2019
Optimizing while Ensuring Solution Won’t, in the Future, Exhibit Undesirable Behavior

\[
\begin{align*}
\text{arg max } & \quad f(a) \\
\text{s.t. } & \quad \forall i \in \{1, \ldots, n\}, \Pr\left(g_i(a(D)) \leq 0\right) \geq 1 - \delta_i
\end{align*}
\]
Counterfactual RL
with Constraints on Future Performance of Policy

\[ r(s_t, a_t) \]
\[ s_t \in S \]
\[ a_t \in A \]
\[ \pi_t(s_t) \rightarrow a_t \]

\[ D: \text{Dataset of } n \text{ traj.s } \tau, \tau \sim \pi_b \]
Related Work in Decision Making

\[
\arg \max_{a \in A} f(a)
\]

\[
\text{s.t. } \forall i \in \{1, \ldots, n\}, \Pr\left( g_i(a(D)) \leq 0 \right) \geq 1 - \delta_i
\]

- Chance constraints, data driven robust optimization have similar aims
- Most of this work has focused on ensuring computational efficiency for \( f \) and/or constraints \( g \) with certain structure (e.g. convex)
- Also need to be able to capture broader set of aims & constraints
Batch RL with Safety Constraints

$$g(\theta) = \mathbb{E}[r'(H)|\theta_0] - \mathbb{E}[r'(H)|\theta]$$

- $r'(H)$ is a function of the trajectory $H$
1 Algorithm for Batch RL with Safety Constraints

- Take in desired behavior constraints $g$ and confidence level & data
- Given a finite set of decision policies, for each policy $i$
  - Compute generalization bound for each constraint
  - If passes all with desired confidence*, $\text{Safe}(i) = \text{true}$
- Estimate performance $f$ of all policies that are safe
- Return best policy that is safe, or no solution if safe set is empty
Diabetes Insulin Management

- Blood glucose control
- Action: insulin dosage
- Search over policies
- Constraint: hypoglycemia
- Very accurate simulator: approved by FDA to replace early stage animal trials
Personalized Insulin Dosage: Safe Batch Policy Improvement
What You Should Know

- Be able to define and apply importance sampling for off policy policy evaluation
- Define some limitations of IS (variance)
- Define why we might want to do batch offline RL
Class Progress

- Last time: Fast Reinforcement Learning
- This time: Batch RL
- Next time: Guest Lecture by Professor Doshi-Velez