Batch / Offline RL

Emma Brunskill
CS234

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**

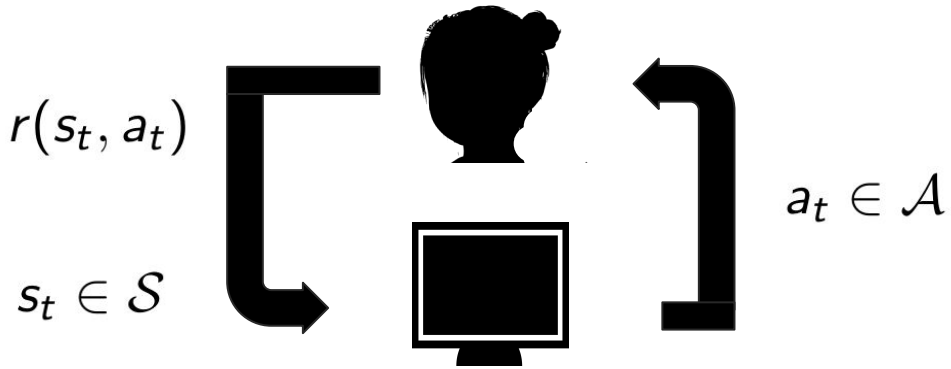
Outline for Today

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

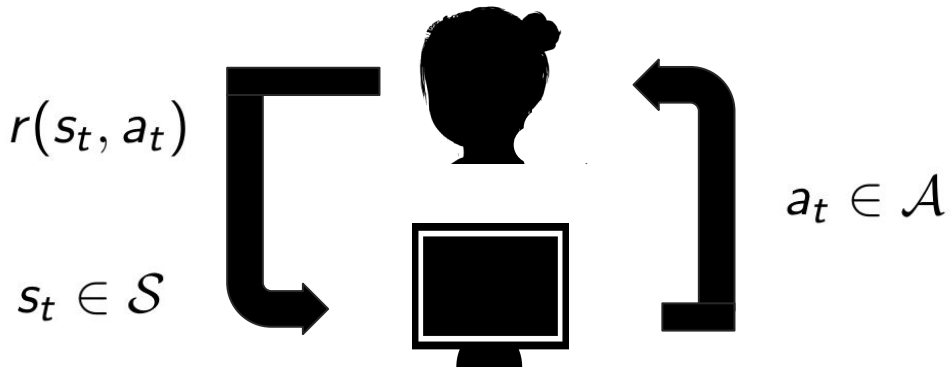
Reinforcement Learning



$$\underbrace{V^{\pi}(s)}_{\text{Value func.}} = \underbrace{r(s, \pi(s))}_{\text{Reward}} + \gamma \sum_{s'} \underbrace{p(s'|s, a)}_{\text{Dynamics}} V^{\pi}(s')$$

Only observed through samples (experience)

Today: Counterfactual / Batch RL



$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$

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  Outcome: 92

Patient group 2  Outcome: 91

Patient group 1 →   → Outcome: 92

Patient group 2 →   → Outcome: 91

 → ?

“What If?” Reasoning Given Past Data

Patient group 1 →   → Outcome: 92

Patient group 2 →   → Outcome: 91

 → ?

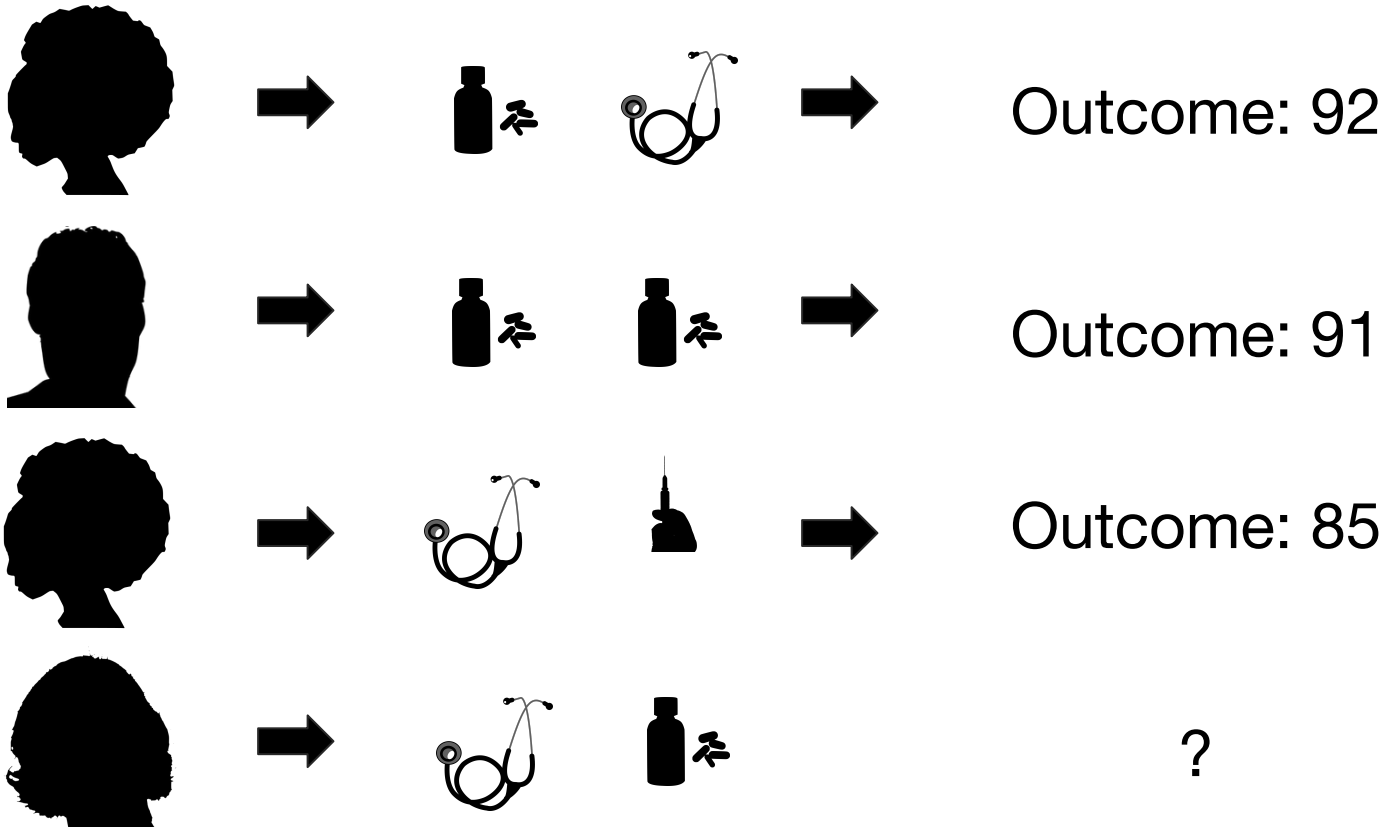
Data Is Censored in that Only Observe Outcomes for Decisions Made

Patient group 1 →   → Outcome: 92

Patient group 2 →   → Outcome: 91

 → ?

Need for Generalization



Off Policy Reinforcement Learning

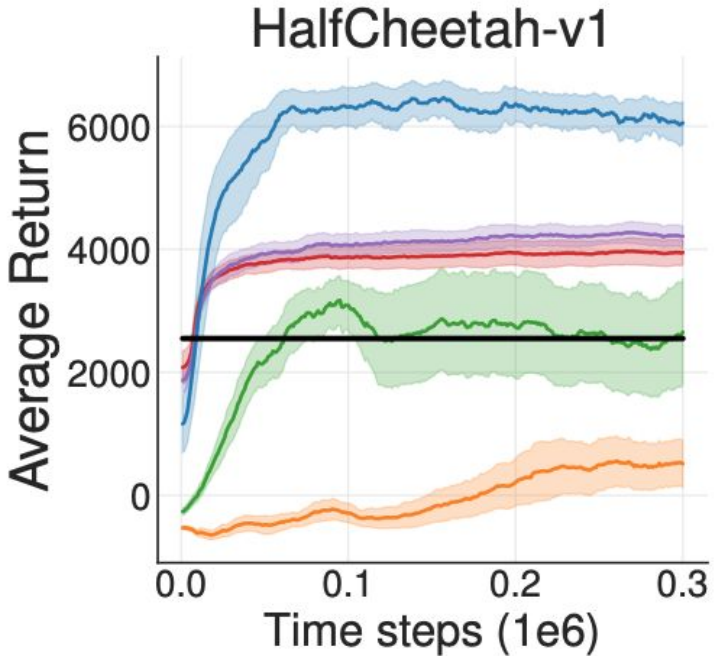
Watkins 1989
Watkins and Dayan 1992
Precup et al. 2000
Lagoudakis and Parr 2002
Murphy 2005
Sutton, Szepesvari and Maei 2009
Shortreed, Laber, Lizotte, Stroup, Pineau, & Murphy 2011
Degirs, White, and Sutton 2012
Mnih et al. 2015
Mahmood et al. 2014
Jiang & Li 2016
Hallak, Tamar and Mannor 2015
Munos, 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

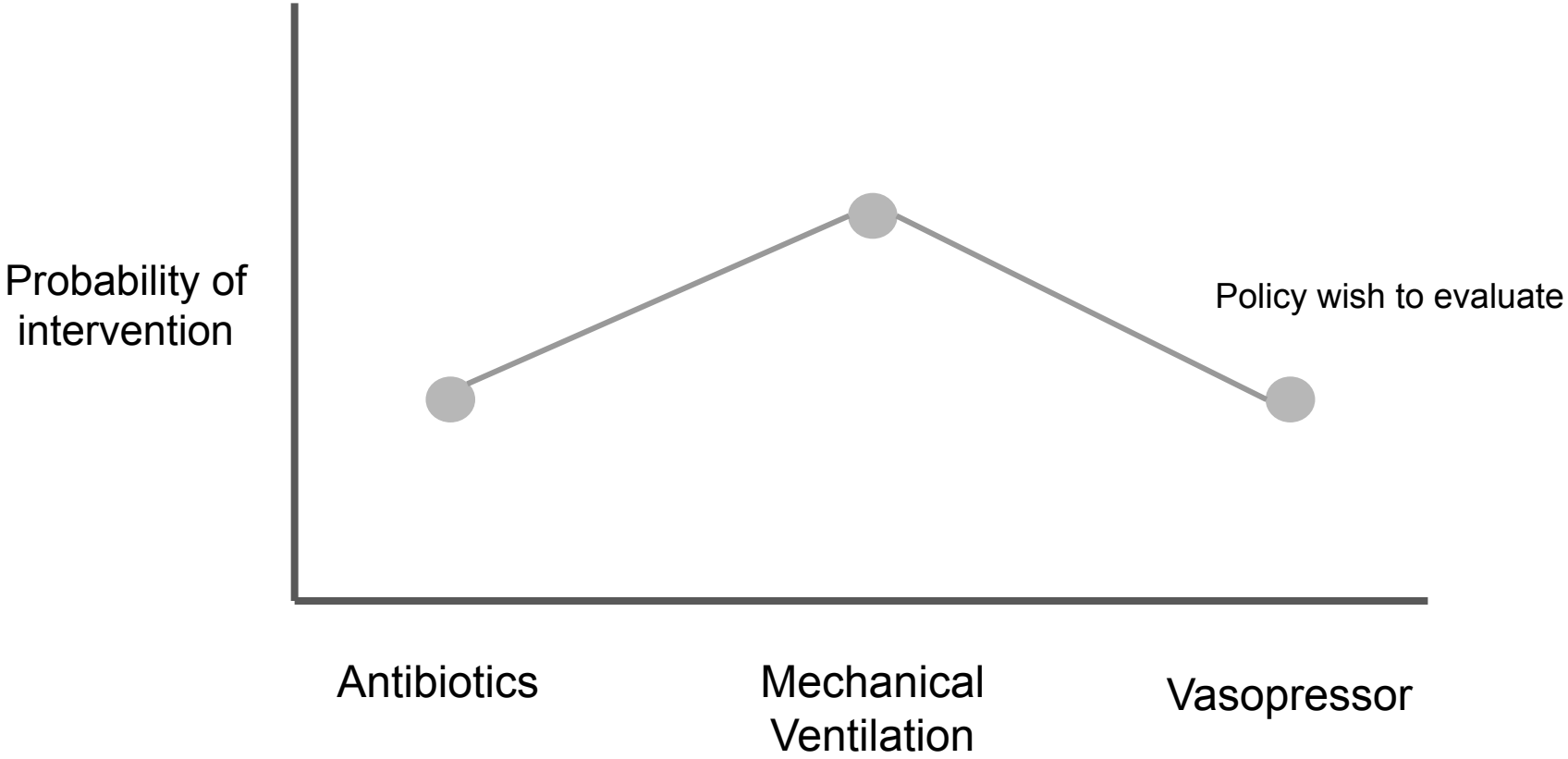
- But deadly triad of bootstrapping, function approximation and off policy, and can fail

Important in Practice

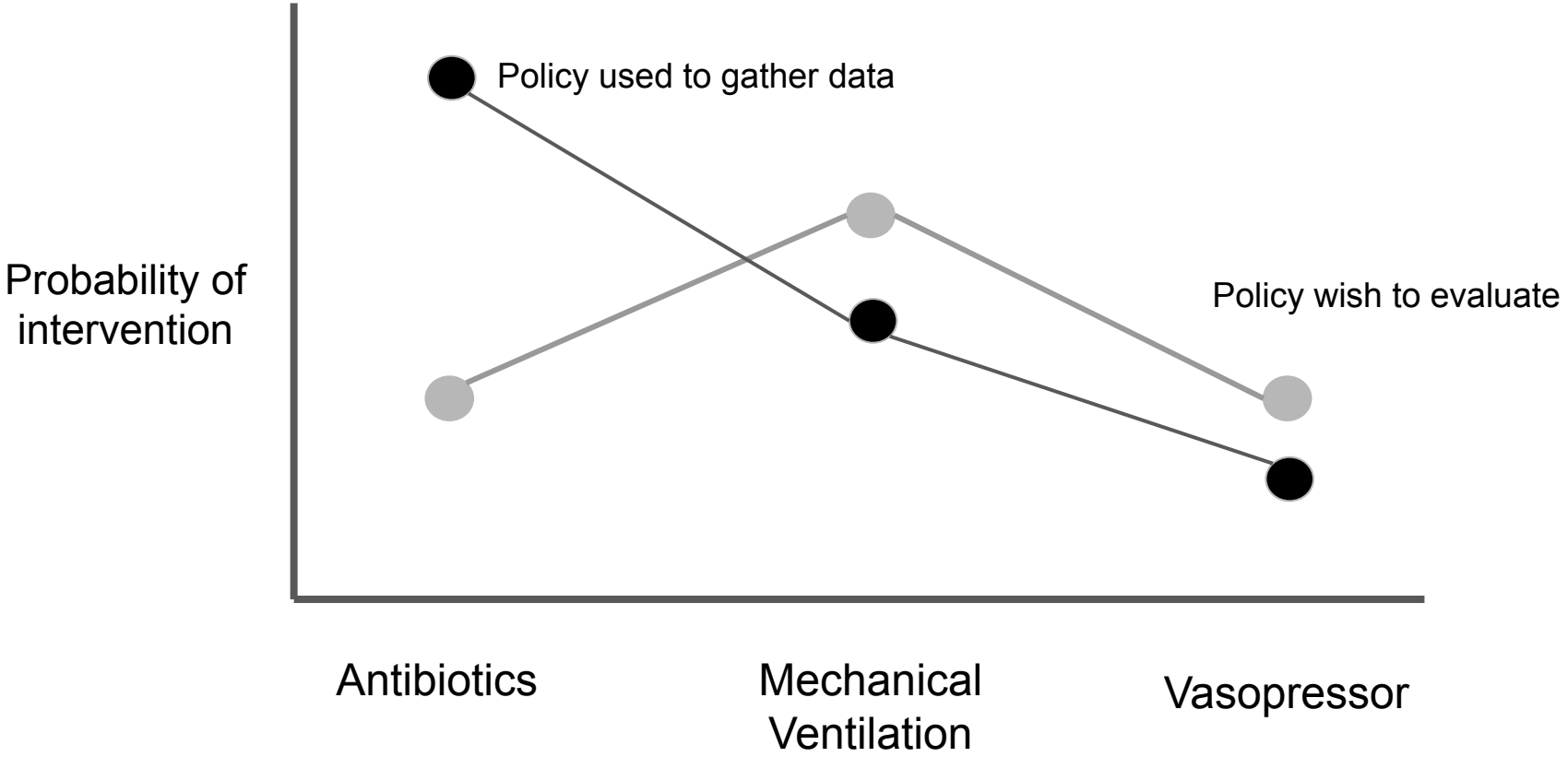


BCQ figure from Fujimoto, Meger, Precup ICML 2019

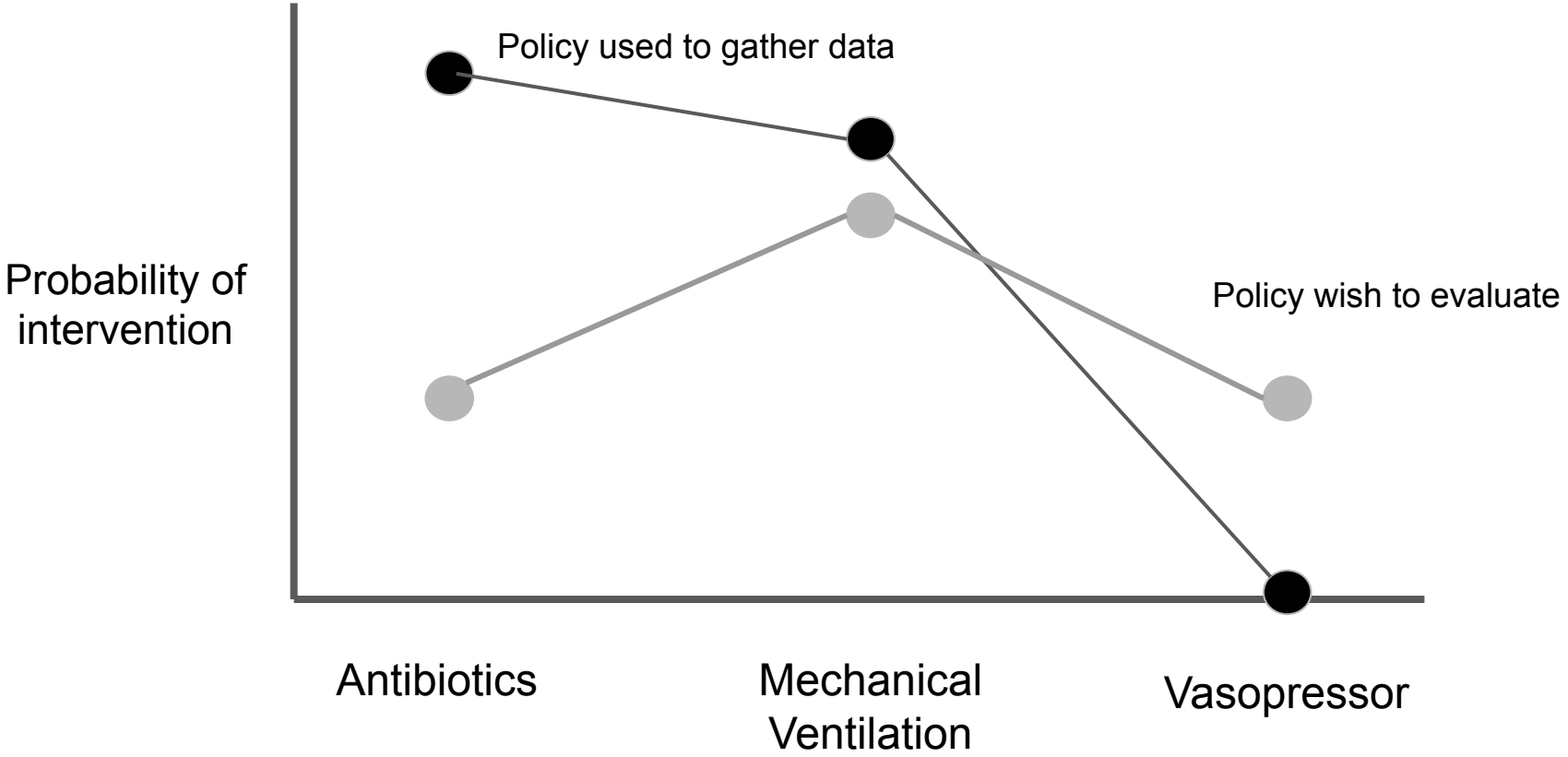
Challenge: Overlap Requirement



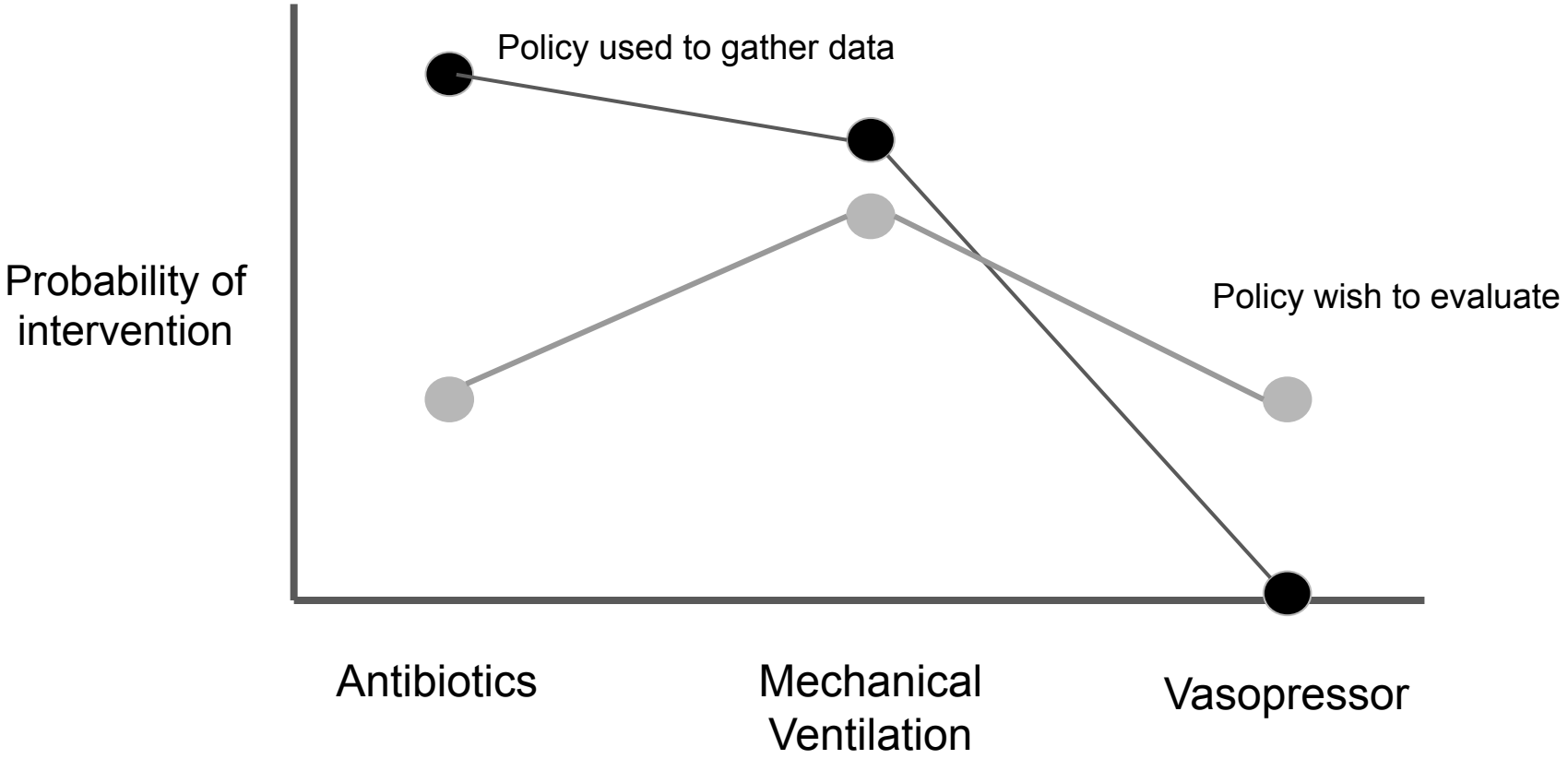
Overlap Requirement: Data Must Support Policy Wish to Evaluate



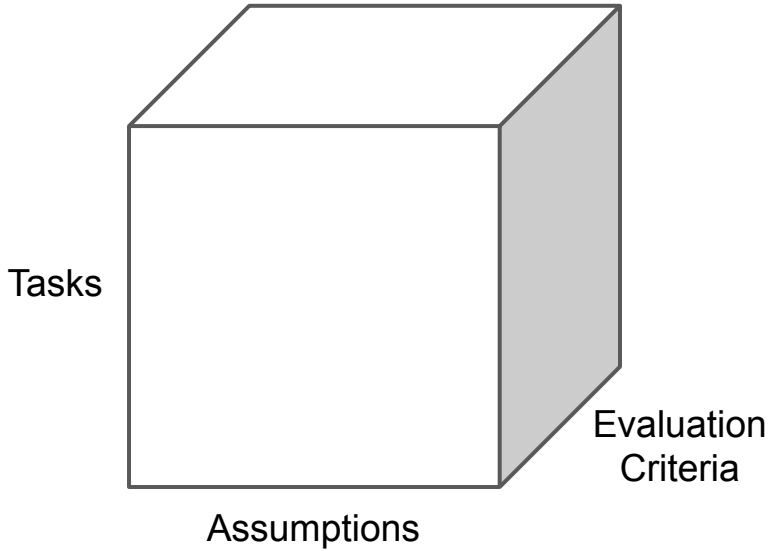
No Overlap for Vasopressor⇒ Can't Do Off Policy Estimation for Desired Policy



How to Evaluate Sufficient Overlap in Real Data?



Offline / Batch Reinforcement Learning



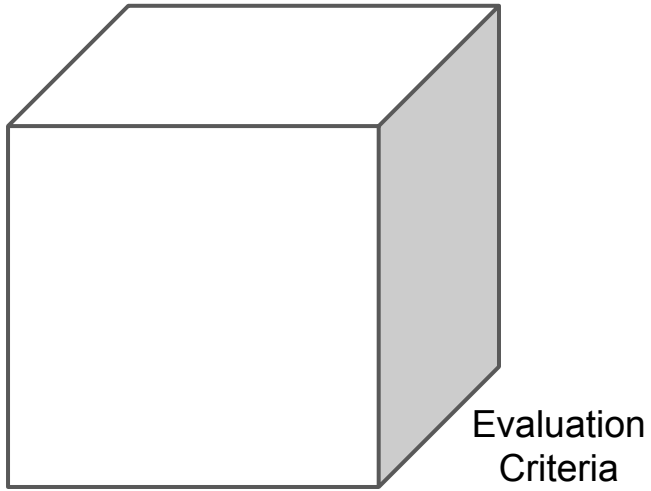
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Common Tasks: Off Policy Evaluation & Optimization

Tasks

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

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



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

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)

$$\{Y(A_{1:(t-1)}, a_{t:T}), S_{t'}(A_{1:(t-1)}, a_{t:(t'-1)})\}_{t'=t+1}^{T} \perp\!\!\!\perp A_t \,\big|\, \mathcal{F}_t$$

- Overlap

$$\forall(s,a)\ \mu_e(s,a) > 0 \quad \rightarrow \mu_b(s,a) > 0$$

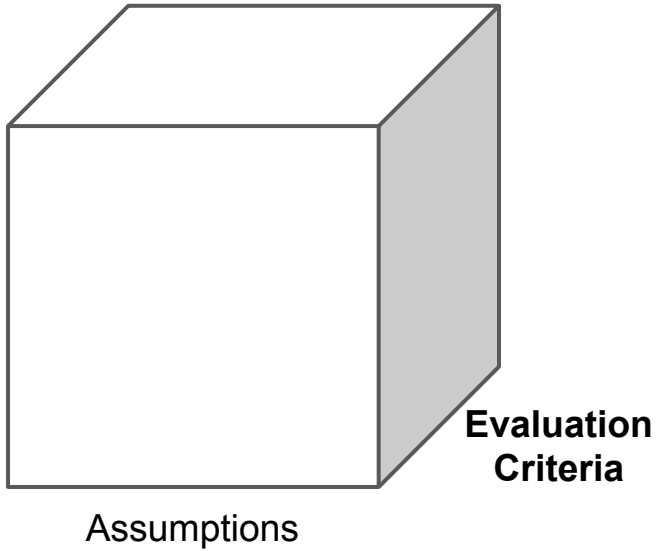
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Common Tasks: Off Policy Evaluation & Optimization

Tasks

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

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



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

Off Policy Reinforcement Learning

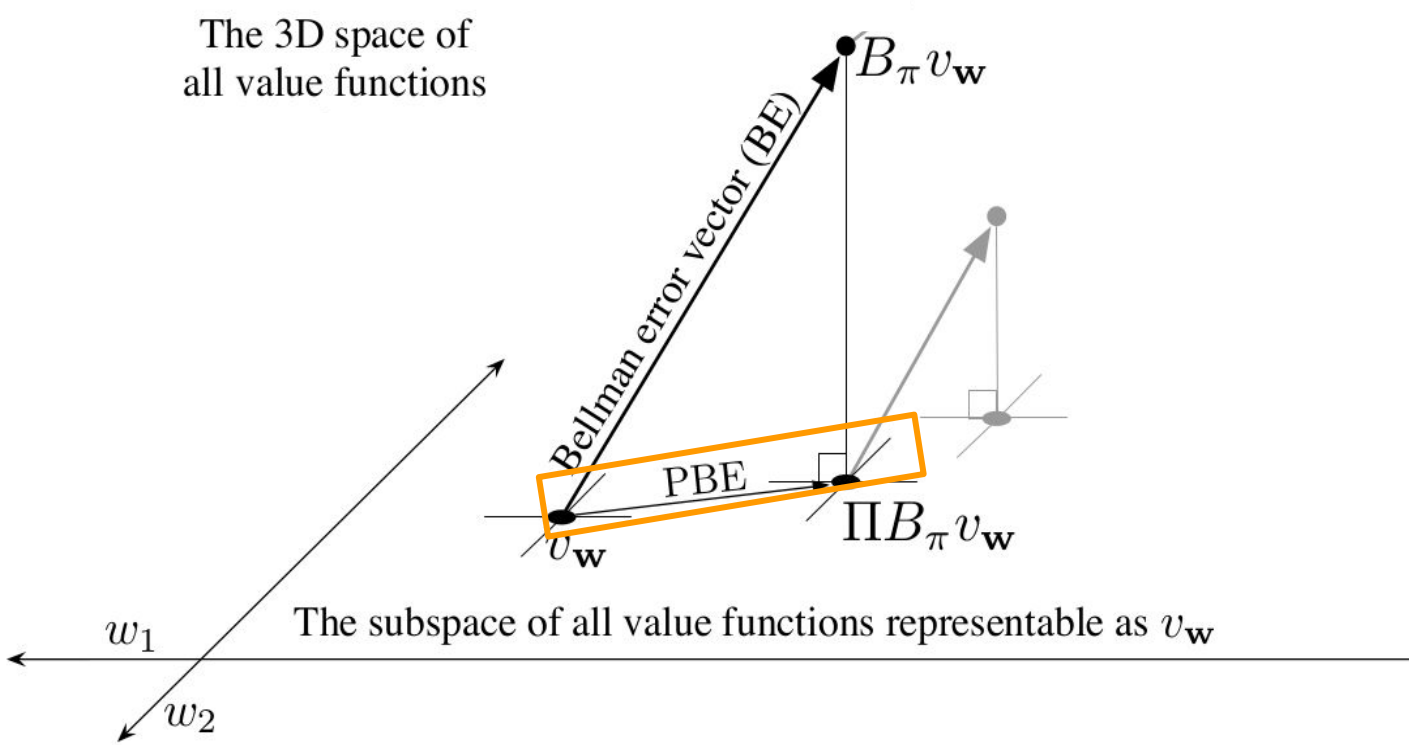
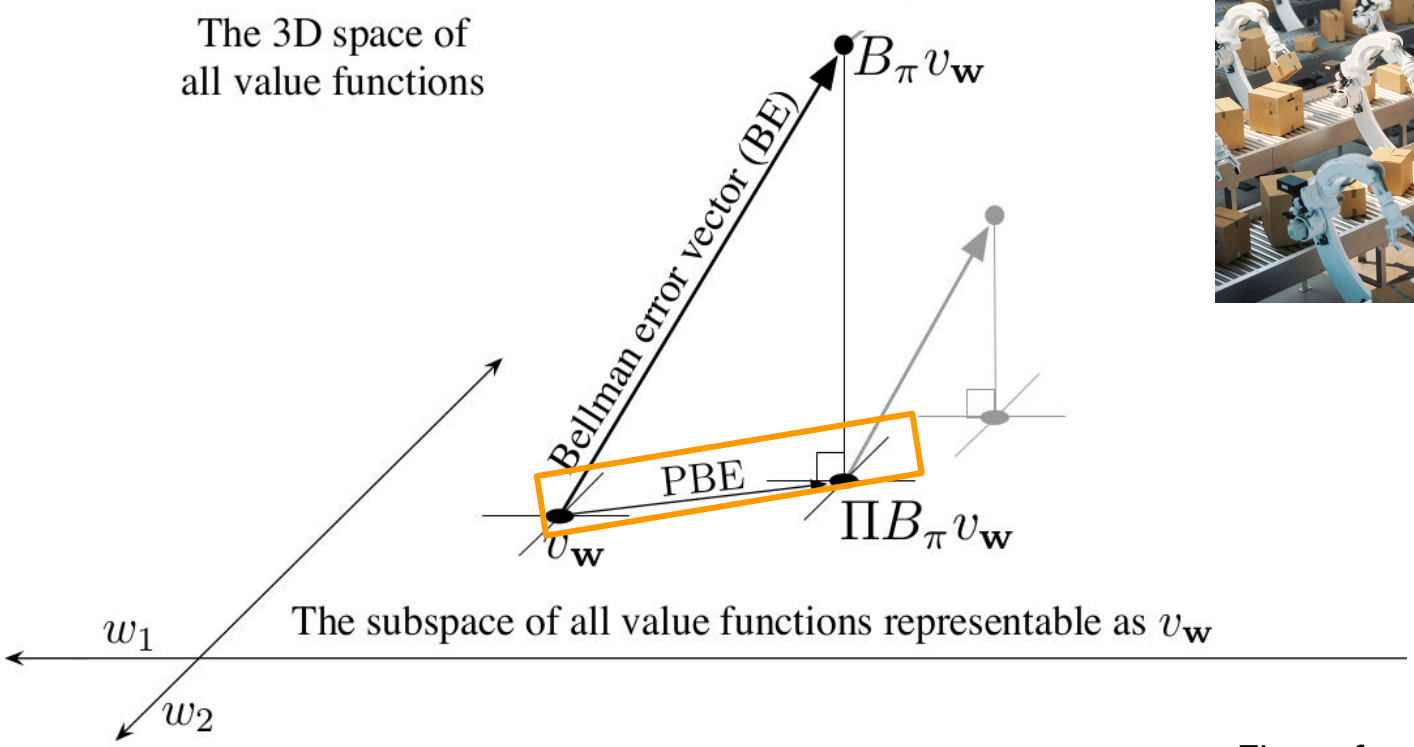


Figure from Sutton & Barto 2018

Off Policy Reinforcement Learning



ATARI

Figure from Sutton & Barto 2018

Batch Off Policy Reinforcement Learning

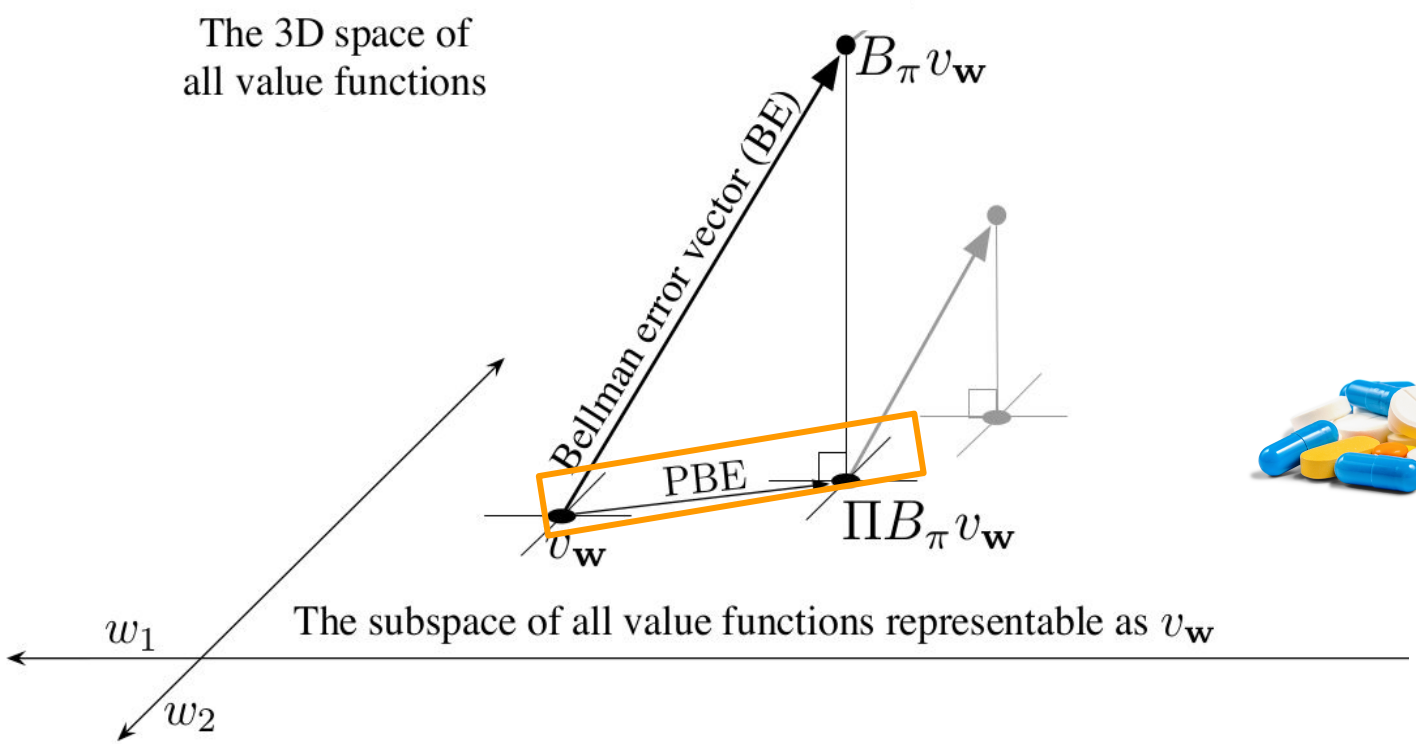


Figure from Sutton & Barto 2018

Batch Off Policy Reinforcement Learning

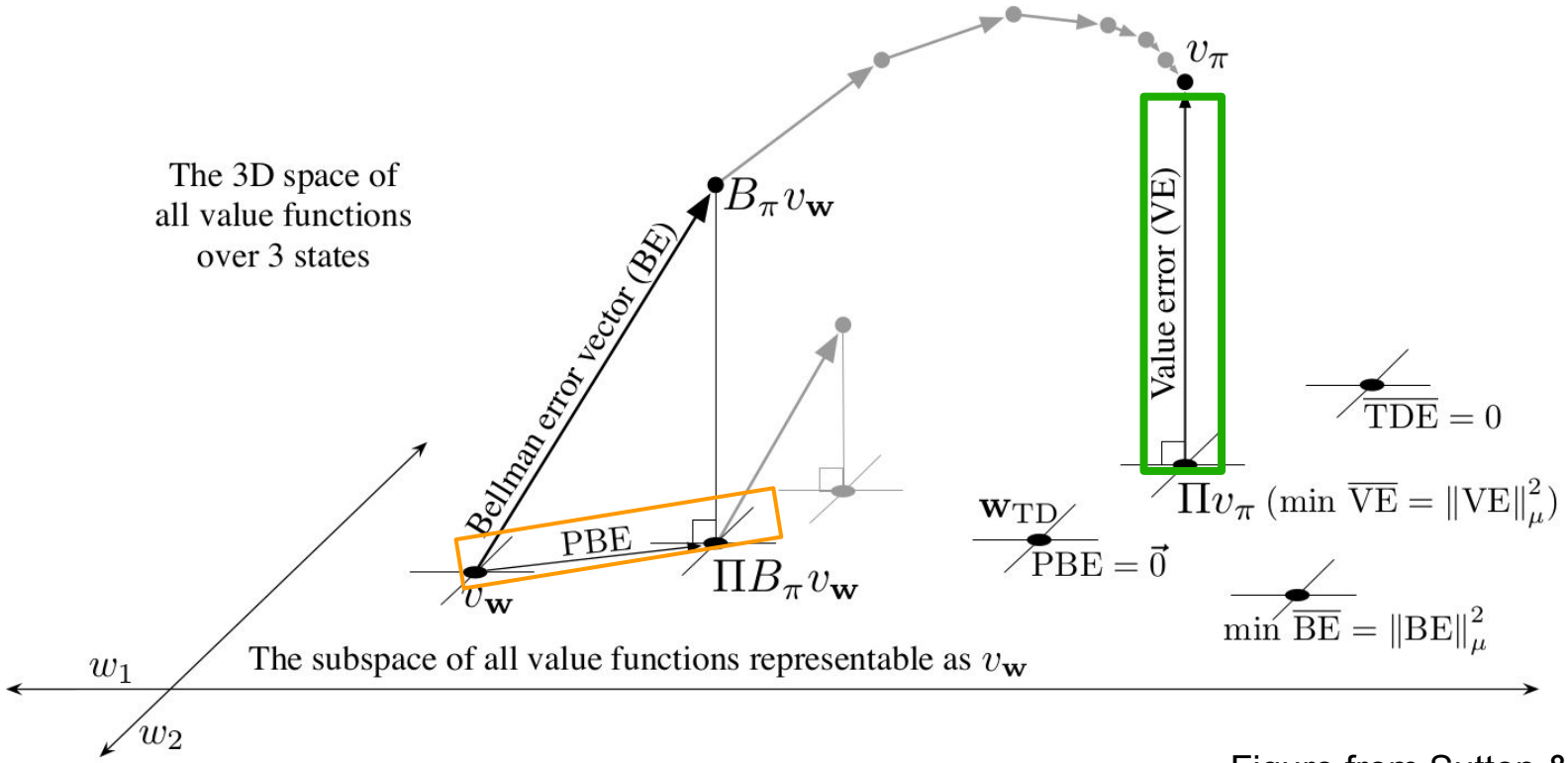


Figure from Sutton & Barto 2018

Common Evaluation Criteria for Off Policy Evaluation

- Computational efficiency
- Performance accuracy

$$\forall \mathcal{D}_i \in \{\mathcal{D}_1 \sim \mathcal{M}_1, \mathcal{D}_2 \sim \mathcal{M}_2, \ldots, \mathcal{D}_K \sim \mathcal{M}_K\} \quad \frac{1}{|\rho|} \sum_{s_0 \in \rho} (\hat{V}^{\pi}_{\mathcal{M}_i}(s_0, \mathcal{D}_i) - V^{\pi}_{\mathcal{M}_i}(s_0))^2$$

$$\lim_{|\mathcal{D}| \to \infty} \frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}^{\pi}(s_0, \mathcal{D}) \to \frac{1}{|\rho|} \sum_{s_0 \in \rho} V^{\pi}(s_0)$$

$$\frac{1}{|\rho|} \sum_{s_0 \in \rho} \hat{V}^{\pi}(s_0, \mathcal{D}) \leq \frac{1}{|\rho|} \sum_{s_0 \in \rho} V^{\pi}(s_0) - f(n, \ldots)$$

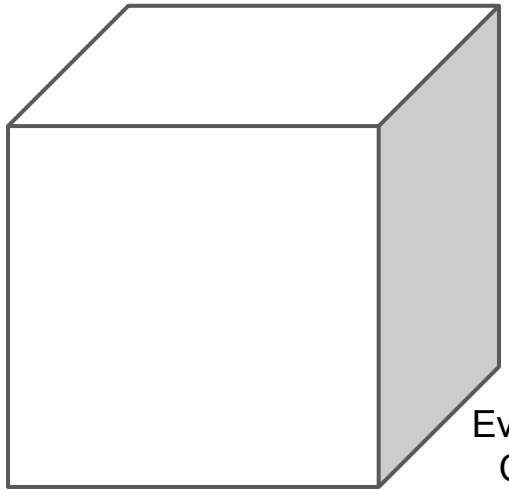
$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \to a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ w/dataset $\mathcal{D}$

Offline / Batch Reinforcement Learning

Tasks

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

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



Assumptions

- Markov?
- Overlap?
- Sequential ignorability?

Evaluation Criteria

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

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

Batch Policy Optimization: Find a Good Policy That Will Perform Well in the Future

$$\underbrace{\arg\max_{\pi \in \mathcal{H}_i} \quad \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}}}_{\text{Policy Optimization}} \quad \underbrace{\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds}_{\text{Policy Evaluation}}$$

$$\mathcal{H} = \mathcal{M}, \mathcal{V}, \Pi \ ?$$

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

Batch Policy Evaluation: Estimate the Performance of a Particular Decision Policy

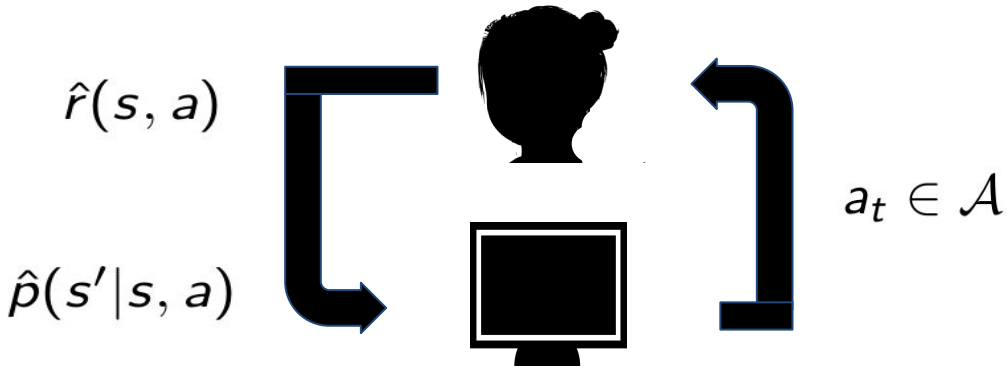
$$\underbrace{\arg\max_{\pi \in \mathcal{H}_i} \quad \max_{\mathcal{H}_i \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots\}}}_{\text{Policy Optimization}} \qquad \underbrace{\int_{s \in S_0} \hat{V}^{\pi}(s, \mathcal{D}) ds}_{\text{Policy Evaluation}}$$

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

Learn Dynamics and Reward Models from Data, Evaluate Policy



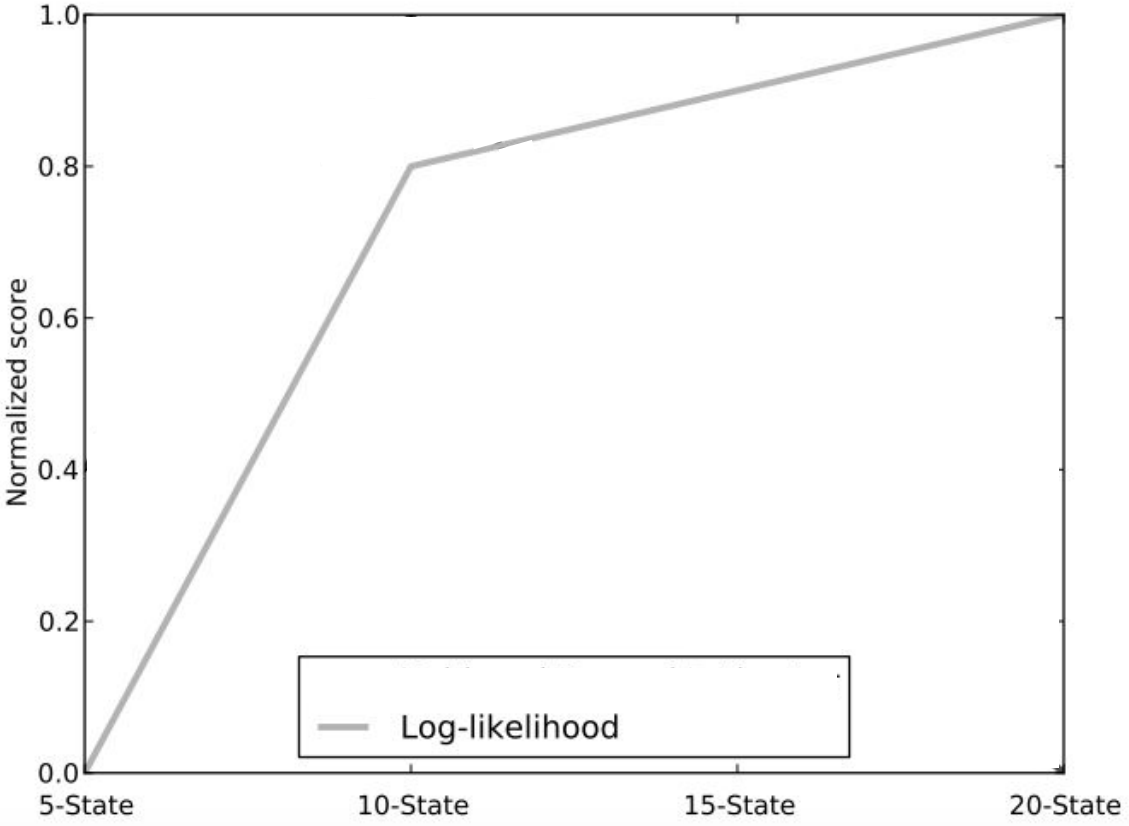
$$V^{\pi} \approx (I - \gamma \hat{P}^{\pi})^{-1} \hat{R}^{\pi}$$

$$P^{\pi}(s'|s) = p(s'|s, \pi(s))$$

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

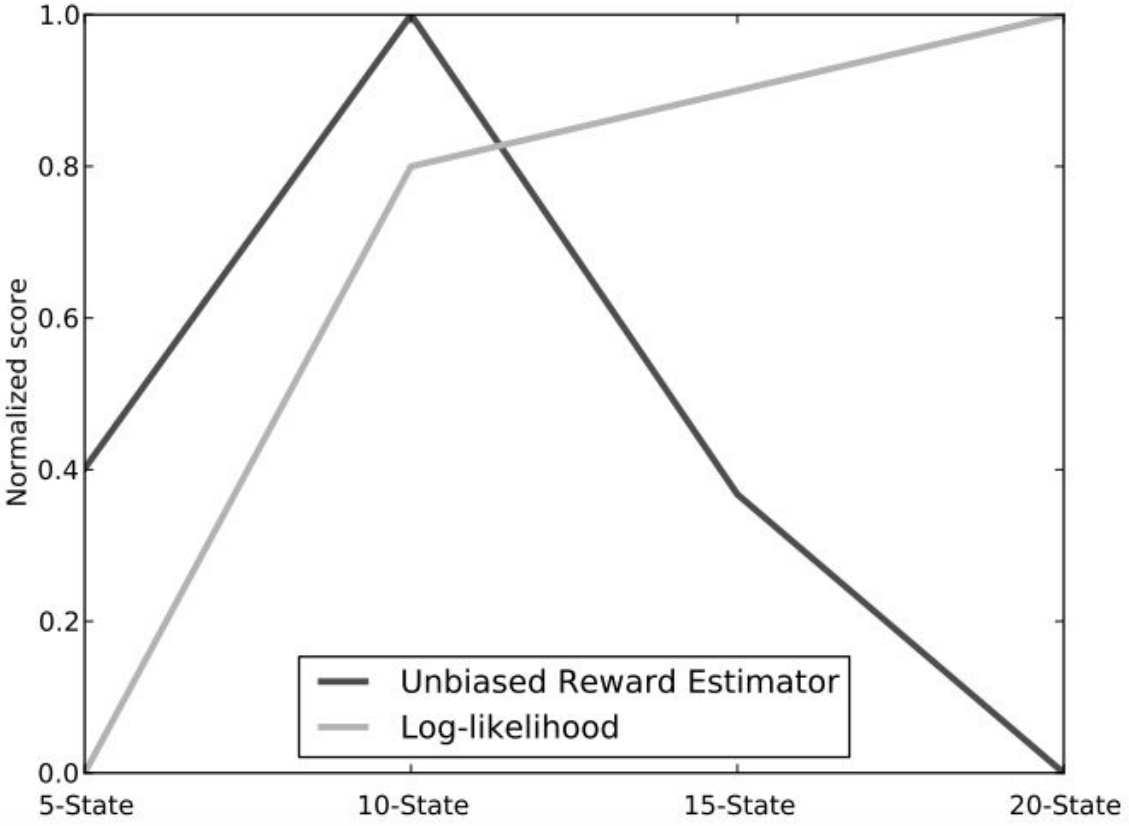
- Mannor, Simster, Sun, Tsitsiklis 2007

Better Dynamics/Reward Models for Existing Data (Improve likelihood)



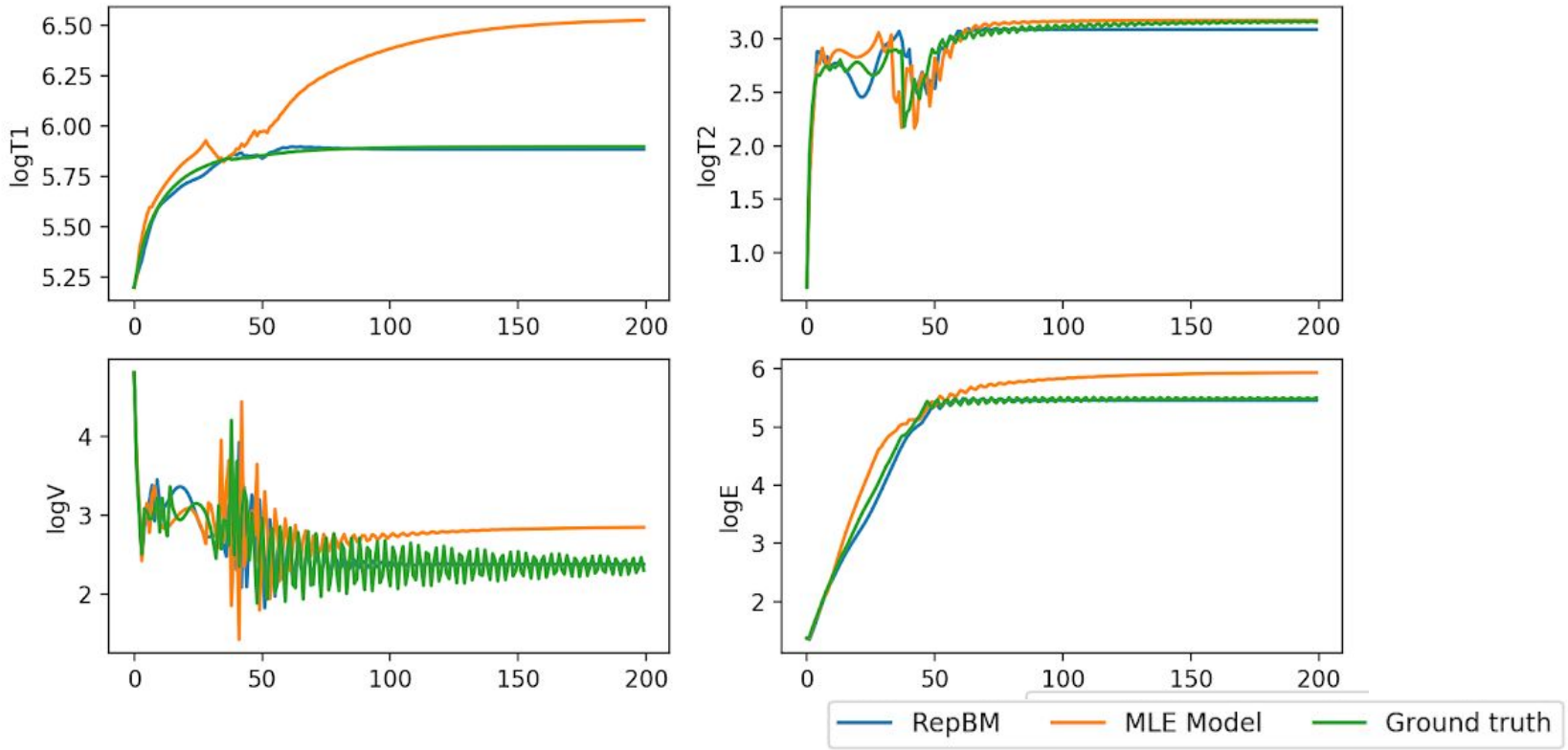
Mandel, Liu, Brunskill, Popovic
AAMAS 2014

Better Dynamics/Reward Models for Existing Data, May **Not** Lead to Better Policies for Future Use → Bias due to Model Misspecification



Mandel, Liu, Brunskill, Popovic
AAMAS 2014

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

$$\mathcal{D} = (s_i, a_i, r_i, s_{i+1}) \; \forall i$$

$$\tilde{Q}^{\pi}(s_i, a_i) \quad = \quad r_i + \gamma V_{\theta}^{\pi}(s_{i+1})$$

$$\arg\min_{\theta} \sum_i (Q_{\theta}^{\pi}(s_i, a_i) - \tilde{Q}^{\pi}(s_i, a_i))^2$$

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

- 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 $n = O\big(\frac{\overline{C}^4}{\epsilon^2}(\log \frac{K}{\delta} + \dim_F \log \frac{\overline{C}^2}{\epsilon^2} + \log \dim_F)\big)$, we have with probability $1-\delta$:*

$$\left| \int_{s_0 \in \rho} \hat{V}^\pi(s_0) - V^\pi(s_0) \right| \leq \frac{\gamma^{.5}}{(1-\gamma)^{1.5}} \left(\sqrt{\beta_u}(2d_F^\pi + \epsilon) + \frac{2\gamma^{K/2}\bar{C}}{(1-\gamma)^{.5}} \right)$$

Le, Voloshin, Yue ICML 2019

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

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

	Arm 1	Arm 2	Arm 3	Arm 4
Gaussian mean	10	1	0	0.5
Behavior policy q	0.2	0.5	0.15	0.15
Evaluation policy p	0.8	0.2	0	0
Num samples from behavior q	20	50	15	15

Why Did This Fail?

	Arm 1	Arm 2	Arm 3	Arm 4
Gaussian mean	10	1	0	0.5
Behavior policy q	0.2	0.5	0.15	0.15
Evaluation policy p	0.8	0.2	0	0
Num samples from behavior q	20	50	15	15

Importance Sampling

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

Importance Sampling: Can Compute Expected Value Under An Alternate Distribution!

$$
\begin{aligned}
\mathbb{E}_p[r] &= \sum_x p(x) r(x) \\
&= \sum_x \frac{p(x) q(x)}{q(x)} r(x) \\
&\approx \frac{1}{N} \sum_{i=1, x \sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i)
\end{aligned}
$$

Importance Sampling is an Unbiased Estimator of True Expectation Under Desired Distribution If

$$
\begin{aligned}
\mathbb{E}_p[r] &= \sum_x p(x) r(x) \\
&= \sum_x \frac{p(x) q(x)}{q(x)} r(x) \\
&\approx \frac{1}{N} \sum_{i=1, x \sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i)
\end{aligned}
$$

- The sampling distribution q(x) > 0 for all x s.t. p(x) > 0 (Coverage / overlap)
- No hidden confounding

Importance Sampling (IS) Example

$$
\begin{aligned}
\mathbb{E}_p[r] &= \sum_x p(x) r(x) \\
&= \sum_x \frac{p(x) q(x)}{q(x)} r(x) \\
&\approx \frac{1}{N} \sum_{i=1, x \sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i)
\end{aligned}
$$

	Arm 1	Arm 2	Arm 3	Arm 4
Gaussian mean	10	1	0	0.5
Behavior policy q	0.2	0.5	0.15	0.15
Evaluation policy p	0.8	0.2	0	0
Num samples from behavior q	20	50	15	15

Importance Sampling (IS) Example

$$\begin{aligned}\mathbb{E}_p[r] &= \sum_x p(x)r(x) \\ &= \sum_x \frac{p(x)q(x)}{q(x)} r(x) \\ &\approx \frac{1}{N} \sum_{i=1, x\sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i)\end{aligned}$$

	Arm 1	Arm 2	Arm 3	Arm 4
Gaussian mean	10	1	0	0.5
Behavior policy q	0.2	0.5	0.15	0.15
Evaluation policy p	0.8	0.2	0	0
Num samples from behavior q	20	50	15	15

X = arms

Expected reward for following behavior policy? 0.2*10 + 0.5*1 + 0*0.15 + 0.15*.5

Expected reward for target policy p? 0.8*10 + 0.2*1 = 8.2

Computing expected reward for p using IS: (20/100) * (.8/.2) * 10 + (50/100)*(.2/.5)*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 .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. **(True)**
- We cannot use pi1 to get an unbiased estimate of the average reward pi2 using importance sampling. **(True, pi1 never pulls arm 1 which is taken by pi2)**
- If rewards can be positive or negative, we can still get a lower bound on pi2 using data from pi1 using importance sampling **(False, only if rewards are positive)**
- 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. (True)
- 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. (False)

Importance Sampling for RL Policy Evaluation

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

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

Importance Sampling for RL Policy Evaluation

$$
\begin{aligned}
V^{\pi}(s) &= \sum_{\tau} p(\tau|\pi, s) R(\tau) \\
&= \sum_{\tau} p(\tau|\pi_b, s) \frac{p(\tau|\pi, s)}{p(\tau|\pi_b, s)} R_{\tau} \\
&\approx \sum_{i=1, \tau_i \sim \pi_b}^{N} \frac{p(\tau_i|\pi, s)}{p(\tau_i|\pi_b, s)} R_{\tau_i}
\end{aligned}
$$

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

Importance Sampling for RL Policy Evaluation

$$
\begin{aligned}
V^{\pi}(s) &= \sum_{\tau} p(\tau|\pi, s) R(\tau) \\
&= \sum_{\tau} p(\tau|\pi_b, s) \frac{p(\tau|\pi, s)}{p(\tau|\pi_b, s)} R_{\tau} \\
&\approx \sum_{i=1, \tau_i \sim \pi_b}^{N} \frac{p(\tau_i|\pi, s)}{p(\tau_i|\pi_b, s)} R_{\tau_i} \\
&= \sum_{i=1, \tau_i \sim \pi_b}^{N} R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(s_{i,t+1}|s_{it}, a_{it}) p(a_{it}|\pi, s_{it})}{p(s_{i,t+1}|s_{it}, a_{it}) p(a_{it}|\pi_b, s_{it})} \\
&= \sum_{i=1, \tau_i \sim \pi_b}^{N} R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(a_{it}|\pi, s_{it})}{p(a_{it}|\pi_b, s_{it})}
\end{aligned}
$$

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

Importance Sampling for RL Policy Evaluation: Don't Need to Know Dynamics Model!

$$
\begin{aligned}
V^{\pi}(s) &= \sum_{\tau} p(\tau|\pi, s) R(\tau) \\
&= \sum_{\tau} p(\tau|\pi_b, s) \frac{p(\tau|\pi, s)}{p(\tau|\pi_b, s)} R_{\tau} \\
&\approx \sum_{i=1, \tau_i \sim \pi_b}^{N} \frac{p(\tau_i|\pi, s)}{p(\tau_i|\pi_b, s)} R_{\tau_i} \\
&= \sum_{i=1, \tau_i \sim \pi_b}^{N} R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(s_{i,t+1}|s_{it}, a_{it}) p(a_{it}|\pi, s_{it})}{p(s_{i,t+1}|s_{it}, a_{it}) p(a_{it}|\pi_b, s_{it})} \\
&= \sum_{i=1, \tau_i \sim \pi_b}^{N} R_{\tau_i} \prod_{t=1}^{H_i} \frac{p(a_{it}|\pi, s_{it})}{p(a_{it}|\pi_b, s_{it})}
\end{aligned}
$$

$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$
π: Policy mapping $s \rightarrow a$
S_0: Set of initial states
$\hat{V}^{\pi}(s, \mathcal{D})$: Estimate $V(s)$ 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

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

- 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**

REPORT

Preventing undesirable behavior of intelligent machines

Philip S. Thomas[1,*], Bruno Castro da Silva[2], Andrew G. Barto[1], Stephen Giguere[1], Yuriy Brun[1], Emma Brunskill[3]
+ See all authors and affiliations

Science 22 Nov 2019:
Vol. 366, Issue 6468, pp. 999-1004
DOI: 10.1126/science.aag3311

Making well-behaved algorithms

Machine learning algorithms are being used in an ever-increasing number of applications, and many of these applications affect quality of life. Yet such algorithms often exhibit undesirable behavior, from various types of bias to causing financial loss or delaying medical diagnoses. In standard machine learning approaches, the burden of avoiding this harmful behavior is placed on the user of the algorithm, who most often is not a computer scientist. Thomas *et al.* introduce a general framework for algorithm design in which this burden is shifted from the user to the designer of the algorithm. The researchers illustrate the benefits of their approach using examples in gender fairness and diabetes management.

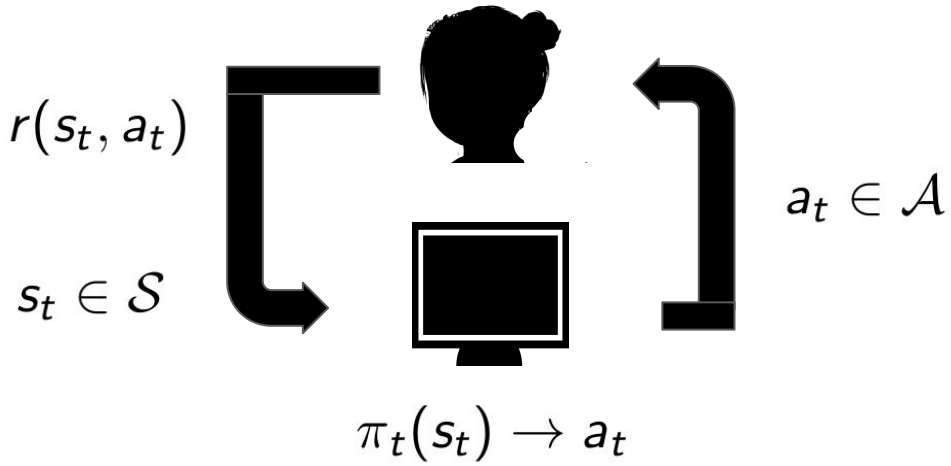
Science, this issue p. **999**

Optimizing while Ensuring Solution Won't, in the Future, Exhibit Undesirable Behavior

$$\arg\max_{a \in \mathcal{A}} f(a)$$

$$\text{s.t. s.t. } \forall i \in \{1, \ldots, n\}, \Pr\Big(\underbrace{g_i(a(D)) \leq 0}_{\text{Constraints}}\Big) \geq 1 - \delta_i$$

Counterfactual RL
with Constraints on Future Performance of Policy



$\mathcal{D}$: Dataset of n traj.s τ, $\tau \sim \pi_b$

Related Work in Decision Making

$$\arg\max_{a \in \mathcal{A}} f(a)$$

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

- 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

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

Default policy

Potential policy

- r’(H) is a function of the trajectory H

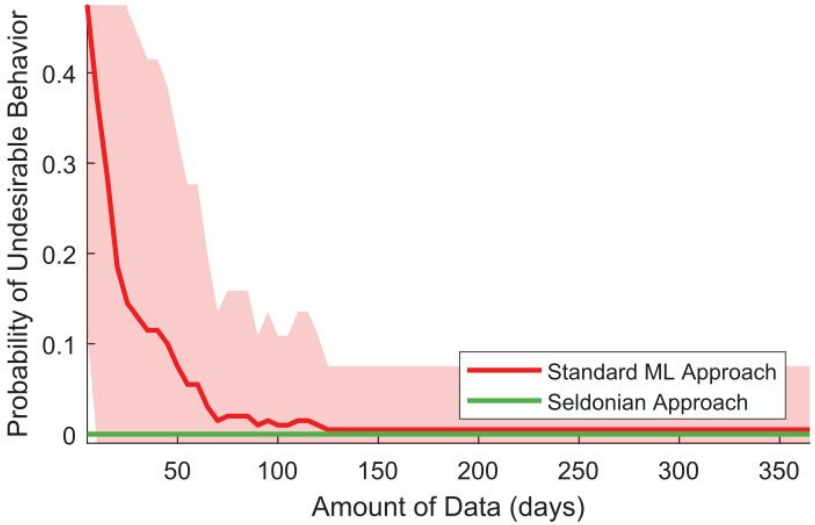
1 Algorithm for Batch RL with Safety Constraints

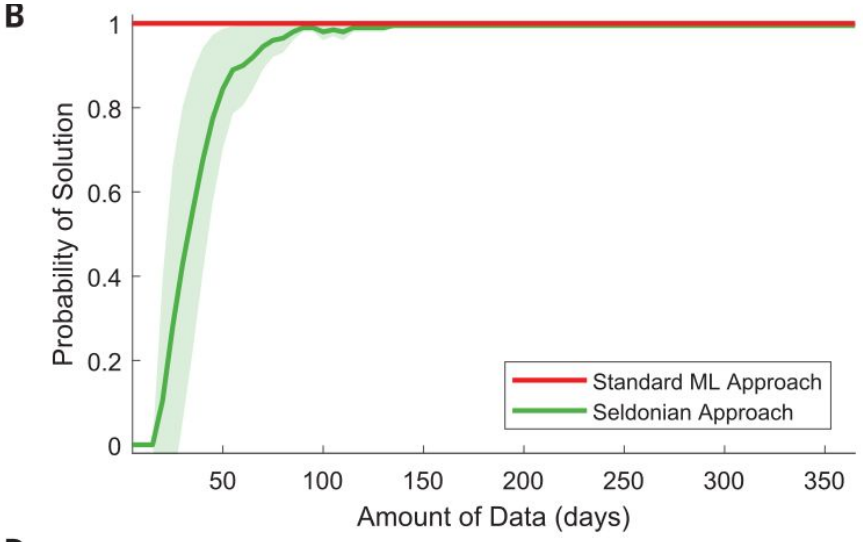
- Take in desired behavior constraints g and confidence level & data
- Given a finite set of decision policies, for each policy i
  - Compute generalization bound for each constraint
  - If passes all with desired confidence*, Safe(i) = true
- Estimate performance f of all policies that are safe
- Return 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

Lecture 14: Imitation Learning in Large State Spaces[1]

Emma Brunskill

CS234 Reinforcement Learning.

[1] With slides from Katerina Fragkiadaki and Pieter Abbeel

Today

- Importance sampling
- Imitation learning
  - Behavior cloning
  - Inverse RL

Learning from Past Decisions and Outcomes

In some settings there exist very good decision policies and we would like to automate them

- One idea: humans provide reward signal when RL algorithm makes decisions
- Good: simple, cheap form of supervision
- Bad: High sample complexity

Alternative: imitation learning

Reward Shaping

Rewards that are **dense in time** closely guide the agent. How can we supply these rewards?

- **Manually design them**: often brittle
- **Implicitly specify them through demonstrations**

Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain, Silver et al. 2010

Examples

- Simulated highway driving [Abbeel and Ng, ICML 2004; Syed and Schapire, NIPS 2007; Majumdar et al., RSS 2017]
- Parking lot navigation [Abbeel, Dolgov, Ng, and Thrun, IROS 2008]

Learning from Demonstrations

- Expert provides a set of **demonstration trajectories**: sequences of states and actions
- Imitation learning is useful when it is easier for the expert to demonstrate the desired behavior rather than:
  - Specifying a reward that would generate such behavior,
  - Specifying the desired policy directly

Problem Setup

- Input:

  State space, action space
  Transition model $P(s' \mid s, a)$
  No reward function R
  Set of one or more teacher's demonstrations $(s_0, a_0, s_1, s_0, \ldots)$
  (actions drawn from teacher's policy π^*)
- Behavioral Cloning:

  Can we directly learn the teacher's policy using supervised learning?
- Inverse RL:

  Can we recover R?
- Apprenticeship learning via Inverse RL:

  Can we use R to generate a good policy?

Table of Contents

Behavioral Cloning

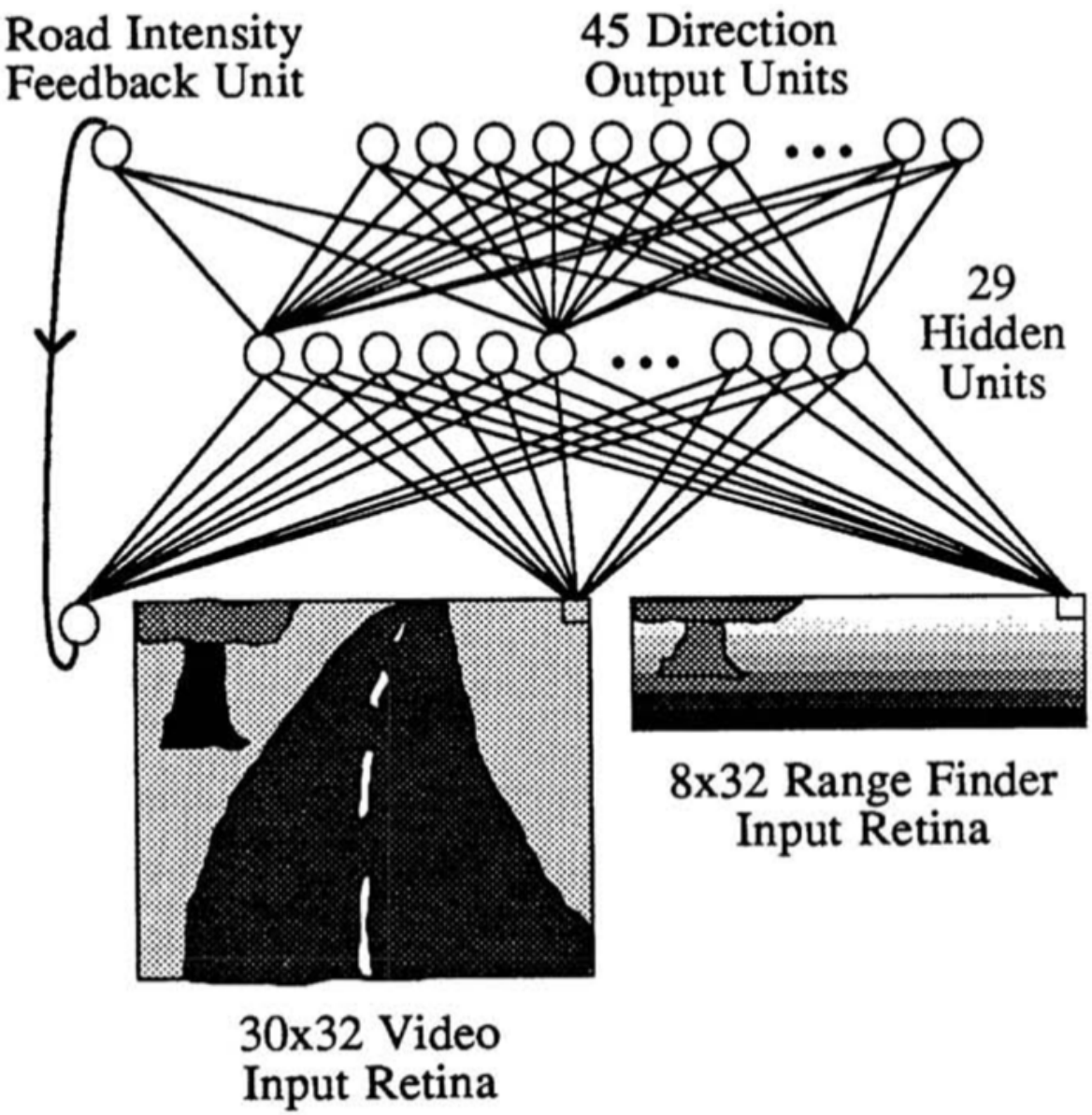
- Formulate problem as a standard machine learning problem:

  Fix a policy class (e.g. neural network, decision tree, etc.)
  Estimate a policy from training examples $(s_0, a_0), (s_1, a_1), (s_2, a_2), \ldots$

- Two notable success stories:

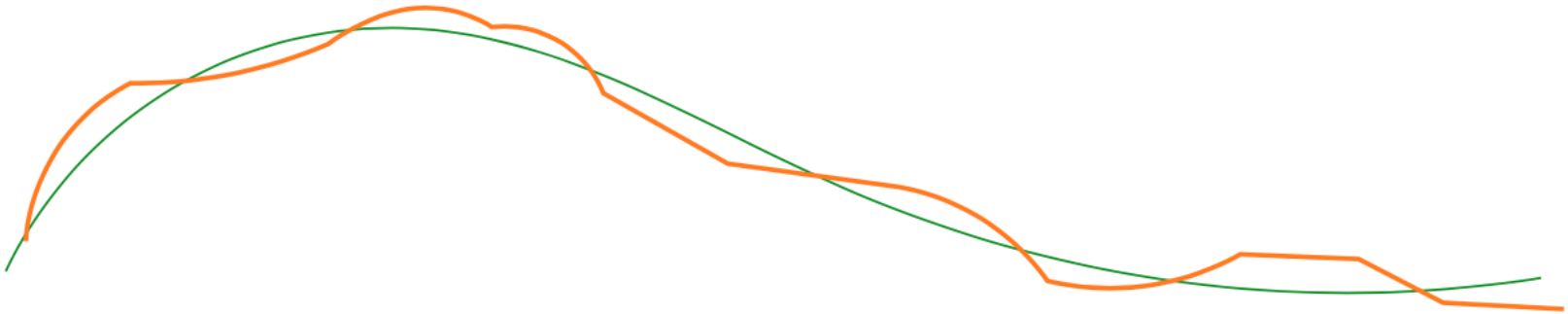
  Pomerleau, NIPS 1989: ALVINN
  Summut et al., ICML 1992: Learning to fly in flight simulator

ALVINN


Road Intensity
Feedback Unit
45 Direction
Output Units
29
Hidden
Units
8x32 Range Finder
Input Retina
30x32 Video
Input Retina

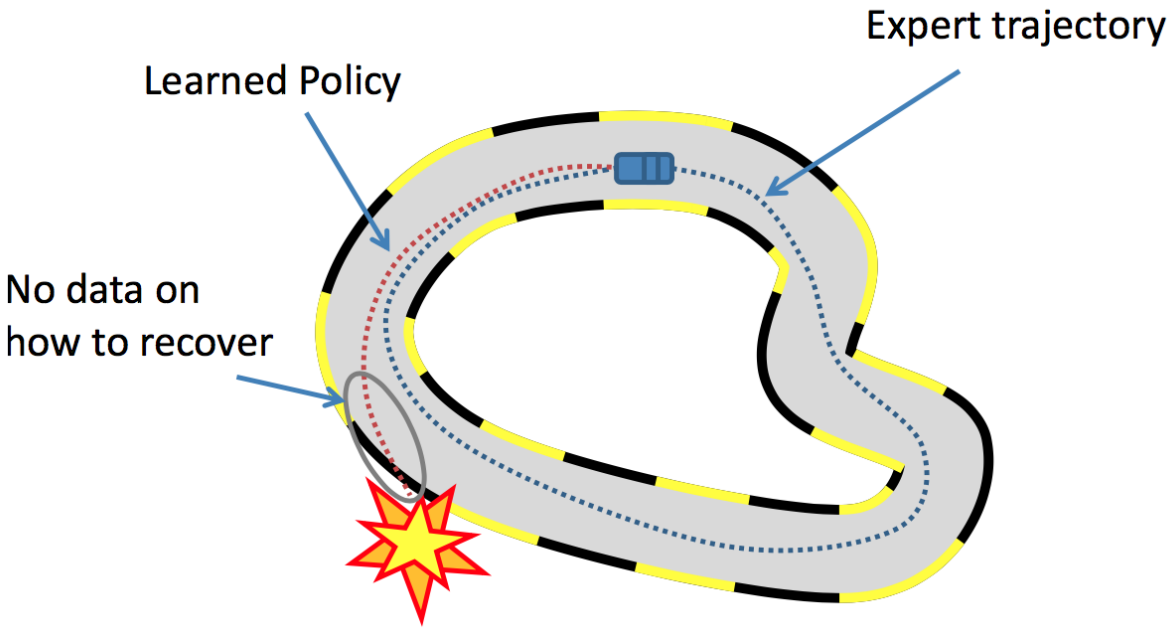
Problem: Compounding Errors

Supervised learning assumes iid. (s, a) pairs and ignores temporal structure
Independent in time errors:



Error at time t with probability $<= \epsilon$
$\mathbb{E}$[Total errors] $<= \epsilon T$

Problem: Compounding Errors



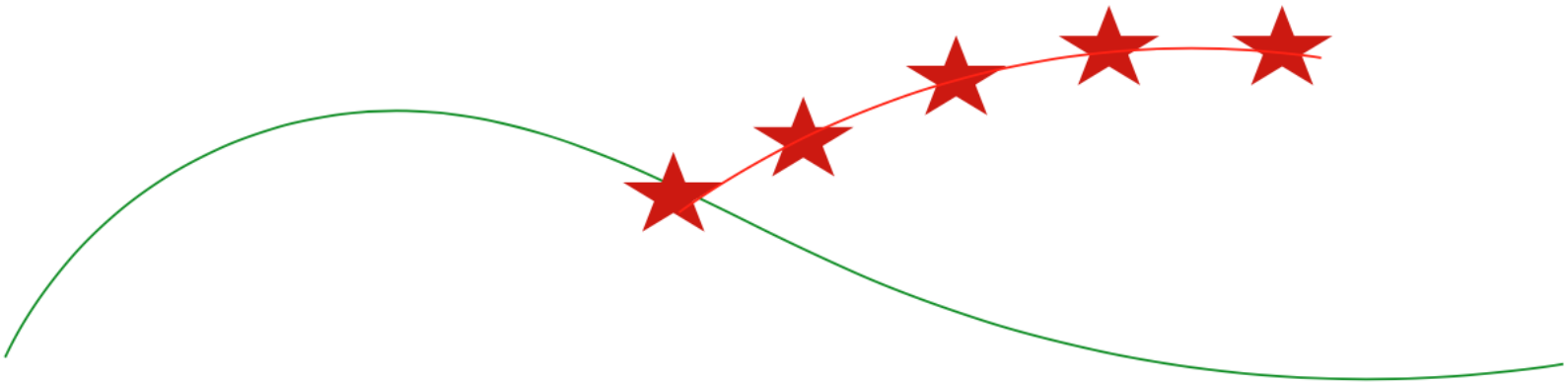
Data distribution mismatch!

In supervised learning, $(x, y) \sim D$ during train **and** test. In MDPs:

- Train: $s_t \sim D_{\pi^*}$
- Test: $s_t \sim D_{\pi_\theta}$

A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross et al. 2011

Problem: Compounding Errors



- Error at time t with probability ϵ
- Approximate intuition: $\mathbb{E}[\text{Total errors}]$
  $\leq \epsilon(T + (T-1) + (T-2)\ldots + 1) \propto \epsilon T^2$
- Real result requires more formality. See Theorem 2.1 in http://www.cs.cmu.edu/~sross1/publications/Ross-AIStats10-paper.pdf with proof in supplement: http://www.cs.cmu.edu/~sross1/publications/Ross-AIStats10-sup.pdf

---

A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Ross et al. 2011

DAGGER: Dataset Aggregation

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i
  and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

- Idea: Get more labels of the expert action along the path taken by the policy computed by behavior cloning
- Obtains a stationary deterministic policy with good performance under its induced state distribution
- Key limitation?

Behavioral cloning

- Note: despite these potential limitations, often behavior cloning in practice can work very well, especially if use BCRNN
- See What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. Mandlekar et al. CORL 2021

Today

- Importance sampling
- Imitation learning
  - Behavior cloning
  - Inverse RL

Feature Based Reward Function

- Given state space, action space, transition model $P(s' \mid s, a)$
- No reward function R
- Set of one or more teacher's demonstrations $(s_0, a_0, s_1, s_0, \ldots)$ (actions drawn from teacher's policy π^*)
- Goal: infer the reward function R
- Assume that the teacher's policy is optimal. What can be inferred about R?

Check Your Understanding: Feature Based Reward Function

- Given state space, action space, transition model $P(s' \mid s, a)$
- No reward function R
- Set of one or more teacher's demonstrations $(s_0, a_0, s_1, s_0, \ldots)$ (actions drawn from teacher's policy π^*)
- Goal: infer the reward function R
- Assume that the teacher's policy is optimal.

1. There is a single unique R that makes teacher's policy optimal
2. There are many possible R that makes teacher's policy optimal
3. It depends on the MDP
4. Not sure

Check Your Understanding: Feature Based Reward Function

- Given state space, action space, transition model $P(s' \mid s, a)$
- No reward function R
- Set of one or more teacher's demonstrations $(s_0, a_0, s_1, s_0, \ldots)$ (actions drawn from teacher's policy π^*)
- Goal: infer the reward function R
- Assume that the teacher's policy is optimal.

1. There is a single unique R that makes teacher's policy optimal
2. There are many possible R that makes teacher's policy optimal
3. It depends on the MDP
4. Not sure

.

Linear Feature Reward Inverse RL

- Recall linear value function approximation
- Similarly, here consider when reward is linear over features

  $R(s) = \boldsymbol{w}^T x(s)$ where $\boldsymbol{w} \in \mathbb{R}^n, x : S \rightarrow \mathbb{R}^n$
- Goal: identify the weight vector $\boldsymbol{w}$ given a set of demonstrations
- The resulting value function for a policy π can be expressed as

$$V^{\pi}(s_0) = \mathbb{E}_{s \sim \pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0]$$

Linear Feature Reward Inverse RL

- Recall linear value function approximation
- Similarly, here consider when reward is linear over features

  $$R(s) = \boldsymbol{w}^T x(s) \text{ where } w \in \mathbb{R}^n, x : S \rightarrow \mathbb{R}^n$$

- Goal: identify the weight vector $\boldsymbol{w}$ given a set of demonstrations
- The resulting value function for a policy π can be expressed as

$$V^\pi(s_0) = \mathbb{E}_{s\sim\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0] = \mathbb{E}_{s\sim\pi}[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{w}^T x(s_t) \mid s_0]$$

$$= w^T \mathbb{E}_{s\sim\pi}[\sum_{t=0}^{\infty} \gamma^t x(s_t) \mid s_0]$$

$$= \boldsymbol{w}^T \mu(\pi)$$

- where $\mu(\pi)(s)$ is defined as the discounted weighted frequency of state features under policy π, starting in state s_0.

Relating Frequencies to Optimality

- Assume $R(s) = \boldsymbol{w}^T x(s)$ where $w \in \mathbb{R}^n, x : S \rightarrow \mathbb{R}^n$
- Goal: identify the weight vector $\boldsymbol{w}$ given a set of demonstrations
- $V^\pi = \mathbb{E}_{s\sim\pi}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \mid \pi] = \boldsymbol{w}^T \mu(\pi)$ where
  $\mu(\pi)(s) =$ discounted weighted frequency of state s under policy π.

$$V^* \geq V^\pi$$

Relating Frequencies to Optimality

- Recall linear value function approximation
- Similarly, here consider when reward is linear over features
  $R(s) = \boldsymbol{w}^T x(s)$ where $w \in \mathbb{R}^n, x : S \rightarrow \mathbb{R}^n$
- Goal: identify the weight vector $\boldsymbol{w}$ given a set of demonstrations
- The resulting value function for a policy π can be expressed as

$$V^\pi = \boldsymbol{w}^T \mu(\pi)$$

- $\mu(\pi)(s) =$ discounted weighted frequency of state s under policy π.

$$\mathbb{E}_{s\sim\pi^*}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \mid \pi^*] = V^* \geq V^\pi = \mathbb{E}_{s\sim\pi}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \mid \pi] \quad \forall \pi$$

- Therefore if the expert's demonstrations are from the optimal policy, to identify $\boldsymbol{w}$ it is sufficient to find w^* such that

$$w^{*T} \mu(\pi^*) \geq w^{*T} \mu(\pi), \forall \pi \neq \pi^*$$

Feature Matching

- Want to find a reward function such that the expert policy outperforms other policies.
- For a policy π to be guaranteed to perform as well as the expert policy π^*, sufficient if its discounted summed feature expectations match the expert's policy [Abbeel & Ng, 2004].
- More precisely, if

$$\|\mu(\pi) - \mu(\pi^*)\|_1 \leq \epsilon$$

then for all w with $\|w\|_\infty \leq 1$:

$$|w^T \mu(\pi) - w^T \mu(\pi^*)| \leq \epsilon$$

Ambiguity

- There is an infinite number of reward functions with the same optimal policy.
- There are infinitely many stochastic policies that can match feature counts
- Which one should be chosen?

Learning from Demonstration / Imitation Learning Pointers

- Many different approaches
- Two of the key papers are:

  Maximumum Entropy Inverse Reinforcement Learning (Ziebart et al. AAAI 2008)

  Generative adversarial imitation learning (Ho and Ermon, NeurIPS 2016)

Summary

- Imitation learning can greatly reduce the amount of data need to learn a good policy
- Challenges remain and one exciting area is combining inverse RL / learning from demonstration and online reinforcement learning
- For a look into some of the theory between imitation learning and RL, see Sun, Venkatraman, Gordon, Boots, Bagnell (ICML 2017)

Imitation learning: What You Should Know

- Define behavior cloning and how it differs from reinforcement learning
- Understand when behavior cloning might be worse than offline reinforcement learning
-