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

Emma Brunskill

CS234 Reinforcement Learning.

Winter 2022

[1]With slides from Katerina Fragkiadaki and Pieter Abbeel

Refresh Your Understanding: Batch RL Policy Evaluation

- Select all that are true:
  1. Batch RL refers to when we have many agents acting in a batch
  2. In batch RL we generally care more about sample efficiency than computational efficiency
  3. Importance sampling can be used to get an unbiased estimate of policy performance
  4. Q-learning can be used in batch RL and will generally provide a better estimate than importance sampling in Markov environments for any function approximator used for the Q
  5. Not sure

Refresh Your Understanding: Batch RL Policy Evaluation

- Select all that are true:
  1. Batch RL refers to when we have many agents acting in a batch
  2. In batch RL we generally care more about sample efficiency than computational efficiency
  3. Importance sampling can be used to get an unbiased estimate of policy performance
  4. Q-learning can be used in batch RL and will generally provide a better estimate than importance sampling in Markov environments for any function approximator used for the Q
  5. Not sure

Answer. F. T. T. F.

Class Structure

- Last time: Learning from offline data, overview and policy evaluation
- This time: Learning from offline data, policy evaluation and imitation learning
- Next time: Learning from offline data, policy optimization / learning

Today

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

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

$$\mathbb{E}_p[r] = \sum_x p(x) r(x) \tag{1}$$

reweighing

$$\approx \frac{1}{N} \sum_{i=1, x \sim q}^{N} \frac{p(x_i)}{q(x_i)} r(x_i) \tag{2}$$

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

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 is a Bernoulli where with probability 0.5 it gets 1, 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

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 is a Bernoulli where with probability 0.5 it gets 1, else it gets 0. Select all that are true.

- Arm 1 has a mean reward μ_1 of 2, Arm 2 has a mean reward μ_2 of 1.1 and Arm 3 has a mean reward μ_3 of .5
- pi_1 selects a2 with probability 0.2 and a3 with probability 0.8. Its mean reward is: $0.2 * \mu_2 + 0.8\mu_3$
- pi_2 selects a1 with probability 0.5 and a2 with probability 0.5. Its mean reward is: $0.5 * \mu_1 + 0.5\mu_2$

$V^{\pi_2} > V^{\pi_1}$

Importance Sampling (IS) for RL Policy Evaluation

- Let h_j be episode j (history) of states, actions and rewards

$$h_j = (s_{j,1}, a_{j,1}, r_{j,1}, s_{j,2}, a_{j,2}, r_{j,2}, \ldots, s_{j,L_j(terminal)})$$

length

Importance Sampling (IS) for Policy Evaluation

- Let h_j be episode j (history) of states, actions and rewards

$$h_j = (s_{j,1}, a_{j,1}, r_{j,1}, s_{j,2}, a_{j,2}, r_{j,2}, \ldots, s_{j,L_j(terminal)})$$

$$\begin{aligned} p(h_j|\pi, s = s_{j,1}) =& p(a_{j,1}|s_{j,1})p(r_{j,1}|s_{j,1}, a_{j,1})p(s_{j,2}|s_{j,1}, a_{j,1}) \\ & p(a_{j,2}|s_{j,2})p(r_{j,2}|s_{j,2}, a_{j,2})p(s_{j,3}|s_{j,2}, a_{j,2}) \ldots \\ =& \prod_{t=1}^{L_j-1} p(a_{j,t}|s_{j,t})p(r_{j,t}|s_{j,t}, a_{j,t})p(s_{j,t+1}|s_{j,t}, a_{j,t}) \\ =& \prod_{t=1}^{L_j-1} \pi(a_{j,t}|s_{j,t})p(r_{j,t}|s_{j,t}, a_{j,t})p(s_{j,t+1}|s_{j,t}, a_{j,t}) \end{aligned}$$

Importance Sampling (IS) for Policy Evaluation

- Let h_j be episode j (history) of states, actions and rewards, where the actions are sampled from π_2

$$h_j = (s_{j,1}, a_{j,1}, r_{j,1}, s_{j,2}, a_{j,2}, r_{j,2}, \ldots, s_{j,L_j(terminal)})$$

$$V^{\pi_1}(s) \approx \frac{1}{n} \sum_{j=1}^{n} \frac{p(h_j|\pi_1, s)}{p(h_j|\pi_2, s)} G(h_j)$$

$$= \frac{1}{n} \sum_{\substack{j=1 \\ h \sim \pi_2}}^{N} \prod_{t=1}^{L_j} \frac{p(a_t|\pi, s)\, p(s_t|\lambda_j)\, p(r|h_j)}{p(a_t|\pi_2, s)} G(h_j)$$

$$= \frac{1}{N} \sum_{j=1}^{N} \prod \frac{\pi_1(a_t|s_t)}{\pi_2(a_t|s_t)} G(h_j)$$

don't need explicit dyn or reward

Importance Sampling (IS) for Policy Evaluation

- Let h_j be episode j (history) of states, actions and rewards, where the actions are sampled from π_2

$$h_j = (s_{j,1}, a_{j,1}, r_{j,1}, s_{j,2}, a_{j,2}, r_{j,2}, \ldots, s_{j,L_j(terminal)})$$

$$V^{\pi_1}(s) \approx \frac{1}{n}\sum_{j=1}^{n}\frac{p(h_j|\pi_1, s)}{p(h_j|\pi_2, s)}G(h_j)$$

$$= \frac{1}{n}\sum_{j=1}^{n}\prod_{t=1}^{L_j-1}\frac{\pi_1(a_{j,t}|s_{j,t})p(r_{j,t}|s_{j,t}, a_{j,t})p(s_{j,t+1}|s_{j,t}, a_{j,t})}{\pi_2(a_{j,t}|s_{j,t})p(r_{j,t}|s_{j,t}, a_{j,t})p(s_{j,t+1}|s_{j,t}, a_{j,t})}G(h_j)$$

$$= \frac{1}{n}\sum_{j=1}^{n}\prod_{t=1}^{L_j-1}\frac{\pi_1(a_{j,t}|s_{j,t})}{\pi_2(a_{j,t}|s_{j,t})}G(h_j)$$

Importance Sampling for Policy Evaluation

- Aim: estimate $V^{\pi_1}(s)$ given episodes generated under policy π_2
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$ where the actions are sampled from π_2
- Have access to $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$ in MDP M under policy π_2
- Want $V^{\pi_1}(s) = \mathbb{E}_{\pi_1}[G_t | s_t = s]$
- IS = Monte Carlo estimate given off policy data
- Model-free method
- Does not require Markov assumption
- Under mild assumptions (coverage and no confounding), unbiased & consistent estimator of V^{π_1}
- Can be used when agent is interacting with environment to estimate value of policies different than agent's control policy

Leveraging Future Can't Influence Past Rewards: PDIS

- Importance sampling (IS):

$$IS(D) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{t=1}^{L}\frac{\pi_e(a_t \mid s_t)}{\pi_b(a_t \mid s_t)}\right)\left(\sum_{t=1}^{L}\gamma^t r_t^i\right)$$

- Per-decision importance sampling (PDIS)

$$PSID(D) = \sum_{t=1}^{L}\gamma^t \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{\tau=1}^{t}\frac{\pi_e(a_\tau \mid s_\tau)}{\pi_b(a_\tau \mid s_\tau)}\right) r_t^i$$

- PDIS is unbiased estimator of true value under the same (minimal) assumptions that IS is an unbiased estimator

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
- Variance can generally scale exponentially with the horizon
  1. Concentration inequalities like Hoeffding scale with the largest range of the variable
  2. The largest range of the variable depends on the product of importance weights
  3. **Optional 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?

Off-policy policy evaluation (revisited)

RL this is $\prod_{t=1}^{L} \frac{\pi_e(a_t|s_t)}{\pi_b(a_t|s_t)}$

- Importance sampling (IS):

$$IS(D) = \frac{1}{n} \sum_{i=1}^{n} w_i \left(\sum_{t=1}^{L} \gamma^t r_t^i \right)$$

- Weighted importance sampling (WIS)

$$WIS(D) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \left(\sum_{t=1}^{L} \gamma^t r^i \right)$$

Weighted Importance Sampling)

- Weighted importance sampling (WIS)

$$WIS(D) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \left(\sum_{t=1}^{L} \gamma^t r_t^i \right)$$

- Often has much lower variance than IS or PDIS
- Biased. When $n = 1, \mathbb{E}[WIS] = V(\pi_b)$
- Strongly consistent estimator of V^{π_e}
  - i.e. $\Pr(\lim_{n \to \infty} WIS(D) = V^{\pi_e}) = 1$
  - If
    - Finite horizon
    - One behavior policy, or bounded rewards

$\frac{1}{w_i} \cdot w_i = 1$

and overlap no confounding

Offline Policy Evaluation: What You Should Know

- Be able to define and apply importance sampling for off policy policy evaluation
- Define some limitations of IS (variance)
- Know alternatives that reduce variance without impacting bias (PDIS) or reduce variance while incurring bias (WIS)
- Define why we might want to do batch offline RL policy evaluation and potential applications
- Be aware of the main potential limitations of model and model free methods

Today

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

robotics

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

1999/2000
Stuart Russell
Andrew Ng

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 (no reward) $s_1, a, s', a', \ldots$
- 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?

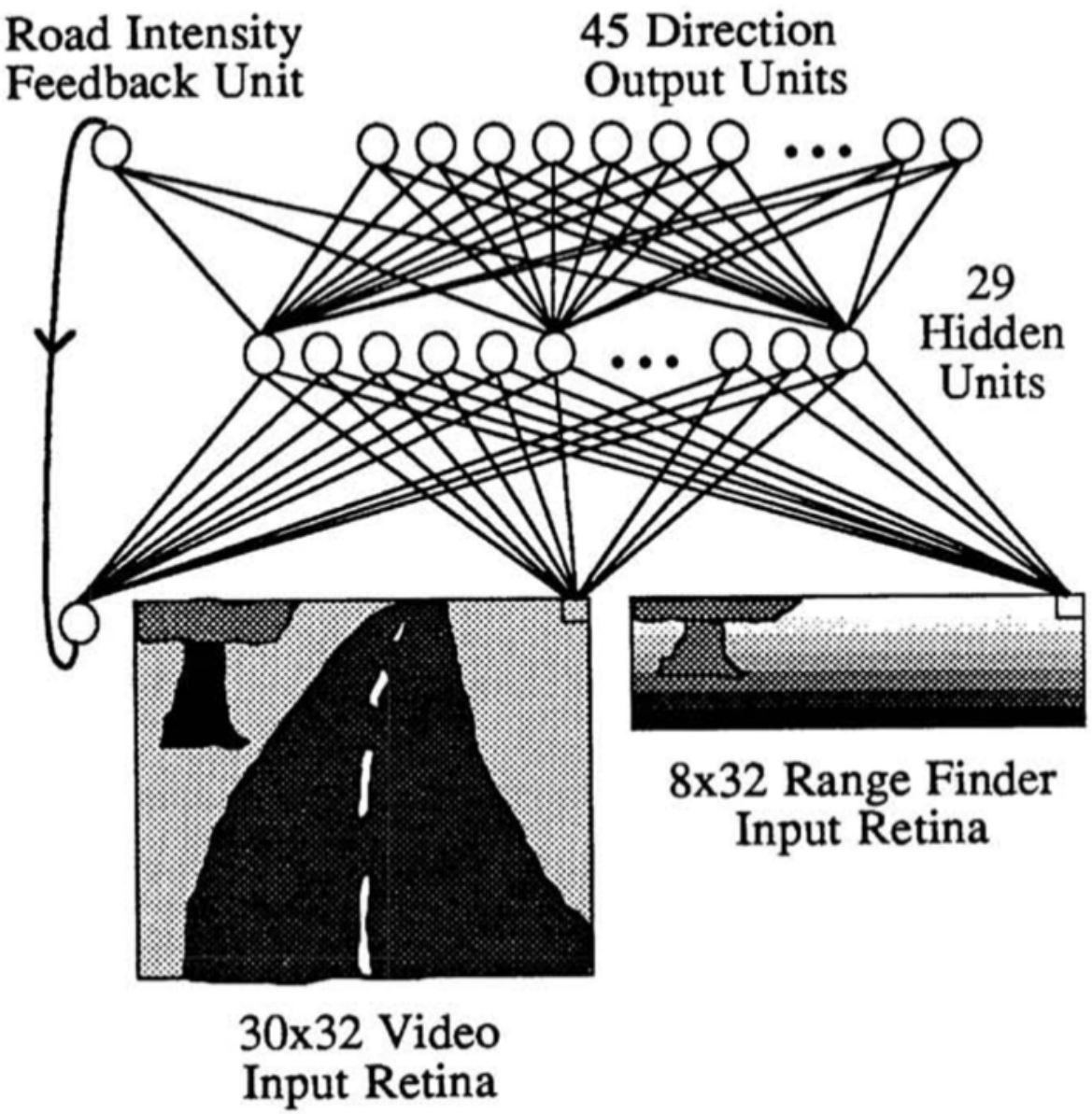
Table of Contents

1 Behavioral Cloning

Behavioral Cloning

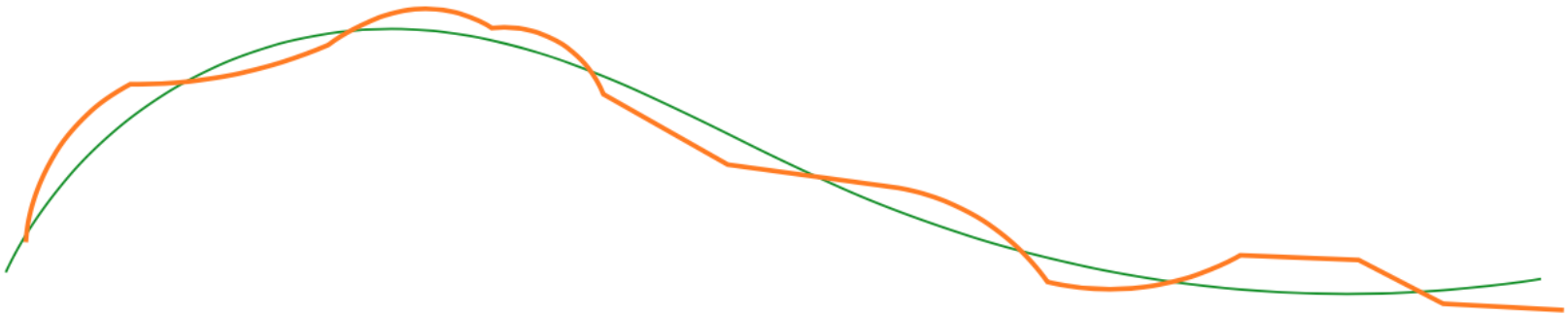
- Formulate problem as a standard machine learning problem:
  - Fix a policy class (e.g. neural network, decision tree, etc.)
  - Estimate a policy from training examples $(s_0, a_0), (s_1, a_1), (s_2, a_2), \ldots$
- Two notable success stories:
  - Pomerleau, NIPS 1989: ALVINN
  - Summut et al., ICML 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

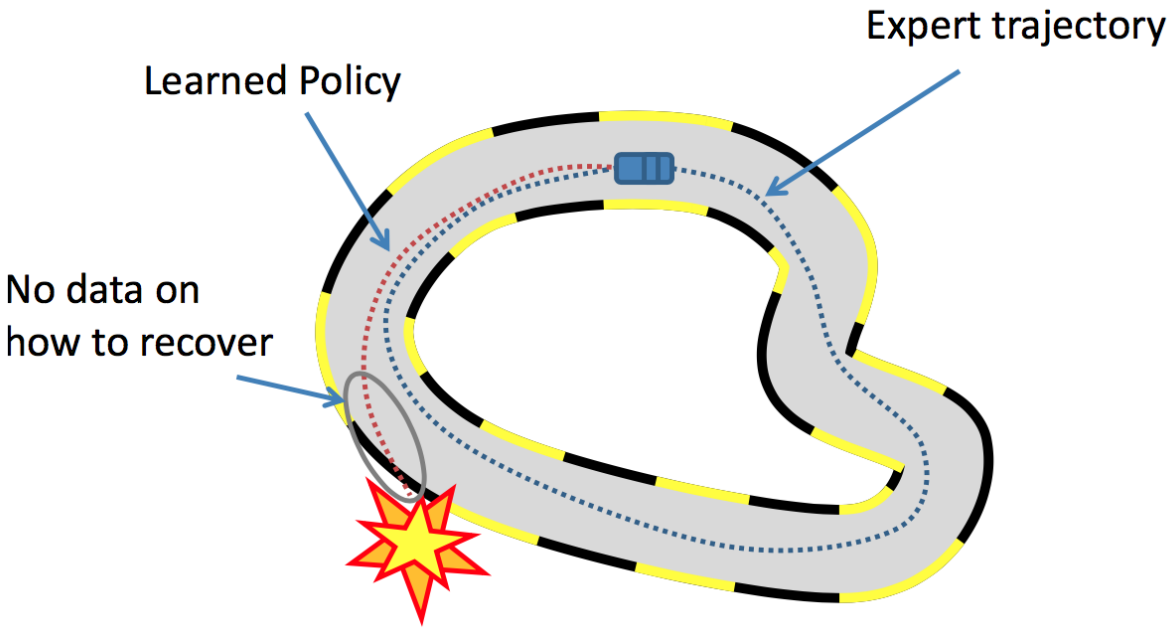
Problem: Compounding Errors

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



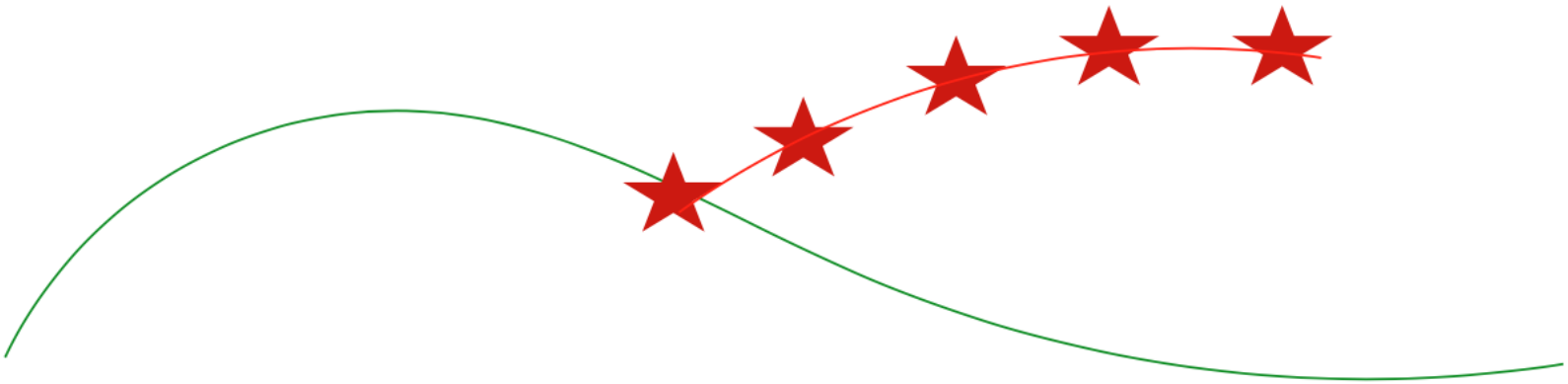
Data distribution mismatch!

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** ignore for now
  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

Answer: There are an infinite set of R .

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
  - $R(s) = \boldsymbol{w}^T x(s)$ 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.

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

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

- 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
-

Class Structure

- Last time: Learning from offline data, overview and policy evaluation
- This time: Learning from offline data, policy evaluation and imitation learning
- Next time: Learning from offline data, policy optimization / learning