Lecture 7: Policy Gradients and Imitation learning

Emma Brunskill

CS234 Reinforcement Learning.

Spring 2024

- Monotonic improvement slides and several PPO slides from Joshua Achiam

Refresh Your Understanding L7N1

Which of the following are true about REINFORCE? In the following options, PG stands for policy gradient.

a. Adding a baseline term can help to reduce the variance of the PG updates true
b. It will converge to a global optima false
c. It can be initialized with a sub-optimal, deterministic policy and still converge to a local optima, given the appropriate step sizes false
d. If we take one step of PG, it is possible that the resulting policy gets worse (in terms of achieved returns) than our initial policy true

Refresh Your Understanding L7N1

Which of the following are true about REINFORCE? In the following options, PG stands for policy gradient.

a. Adding a baseline term can help to reduce the variance of the PG updates
b. It will converge to a global optima
c. It can be initialized with a sub-optimal, deterministic policy and still converge to a local optima, given the appropriate step sizes
d. If we take one step of PG, it is possible that the resulting policy gets worse (in terms of achieved returns) than our initial policy

Class Structure

- Last time: Advanced Policy Search
- This time: Policy search continued and Imitation Learning

Today

- Proximal policy optimization (PPO) (will implement in homework)
  - Generalized Advantage Estimation (GAE)
  - Theory: Monotonic Improvement Theory
- Imitation Learning
  - Behavior cloning
  - DAGGER
  - Max entropy inverse RL

Recall Problems with Policy Gradients

Policy gradient algorithms try to solve the optimization problem

$$\max_{\theta} J(\pi_\theta) \doteq \mathop{\mathrm{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$$

by taking stochastic gradient ascent on the policy parameters θ, using the *policy gradient*

$$g = \nabla_\theta J(\pi_\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t) \right].$$

Limitations of policy gradients:

- Sample efficiency is poor
- Distance in parameter space $\neq$ distance in policy space!
  - What is policy space? For tabular case, set of matrices

$$\Pi = \left\{ \pi \ : \ \pi \in \mathbb{R}^{|S| \times |A|}, \ \sum_a \pi_{sa} = 1, \ \ \pi_{sa} \geq 0 \right\}$$

  - Policy gradients take steps in parameter space
  - Step size is hard to get right as a result

Recall Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a family of methods that approximately enforce KL constraint Two variants:

- Adaptive KL Penalty
  - Policy update solves unconstrained optimization problem

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta || \theta_k)$$

  - Penalty coefficient β_k changes between iterations to approximately enforce KL-divergence constraint
- Clipped Objective
  - New objective function: let $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_k}(a_t|s_t)$. Then

$$\mathcal{L}^{CLIP}_{\theta_k}(\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_k}\left[\sum_{t=0}^{T}\left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_k})\right]\right]$$

    where ϵ is a hyperparameter (maybe $\epsilon = 0.2$)
  - Policy update is $\theta_{k+1} = \arg\max_\theta \mathcal{L}^{CLIP}_{\theta_k}(\theta)$

Recall Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a family of methods that approximately enforce KL constraint **without computing natural gradients**. Two variants:

- Adaptive KL Penalty
  - Policy update solves unconstrained optimization problem

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta || \theta_k)$$

  - Penalty coefficient β_k changes between iterations to approximately enforce KL-divergence constraint
- Clipped Objective
  - New objective function: let $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_k}(a_t|s_t)$. Then

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_k} \left[\sum_{t=0}^{T} \left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_t^{\pi_k}) \right] \right]$$

    where ϵ is a hyperparameter (maybe $\epsilon = 0.2$)
  - Policy update is $\theta_{k+1} = \arg\max_\theta \mathcal{L}_{\theta_k}^{CLIP}(\theta)$
- **How do we estimate the advantage function inside the policy update?**

$$\nabla_\theta V(\theta) \approx (1/m) \sum_{i=1}^{m} \sum_{t=0}^{T-1} A_{ti} \nabla_\theta \log \pi_\theta(a_{ti}|s_{ti})$$

- Recall the N-step advantage estimators

$$\hat{A}_t^{(1)} = r_t + \gamma V(s_{t+1}) - V(s_t)$$

$$\hat{A}_t^{(2)} = r_t + \gamma r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$

$$\hat{A}_t^{(\text{inf})} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \cdots - V(s_t)$$

- Define $\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$. Then

$$\hat{A}_t^{(1)} = \delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$$

$$\hat{A}_t^{(2)} = \delta_t^V + \gamma \delta_{t+1}^V = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$$

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t)$$

- Note the above is an instance of a **telescoping sum**

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \quad (1)$$

- GAE is an exponentially-weighted average of k-step estimators

$$\begin{aligned} \hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots) \\ &= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots) \end{aligned}$$

δ

$= (1-\lambda)(\delta_t^V(1+\lambda+\lambda^2+\lambda^3\dots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\lambda^3+\dots)$
$+ \gamma\delta_{t+2}^V(\lambda^2+\lambda^3+\dots$

$= (1-\lambda)\left[\frac{\delta_t^V}{1-\lambda} + \right.$

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \quad (2)$$

- GAE is an exponentially-weighted average of k-step estimators

$$\begin{aligned}
\hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots) \\
&= (1-\lambda)(\delta_t^V(1+\lambda+\lambda^2+\dots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\dots) \\
&\quad +\gamma^2\delta_{t+2}^V(\lambda^2+\lambda^3+\dots)+\dots) \\
&= (1-\gamma)(\delta_t^V \frac{1}{1-\lambda} + \gamma\lambda\delta_{t+1}^V \frac{1}{1-\lambda} + \gamma^2\lambda^2\delta_{t+2}^V \frac{1}{1-\lambda} + \dots) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}$$

geometric

- Introduced in "High-Dimensional Continuous Control Using Generalized Advantage Estimation" ICLR 2016 by Schulman et al.
- Our derivation follows the derivation presented in the paper

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \quad (3)$$

- GAE is an exponentially-weighted average of k-step estimators

$$\begin{aligned}
\hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \ldots) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \ldots) \\
&= (1-\lambda)(\delta_t^V(1+\lambda+\lambda^2+\ldots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\ldots) \\
&\quad + \gamma^2\delta_{t+2}^V(\lambda^2+\lambda^3+\ldots)+\ldots) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}$$

$\lambda = 0$ look at 1st line

γ, λ

- What are the properties of GAE(γ,0) and GAE(γ,1)? (select all)
- (a) GAE(γ,1) is the advantage function using a TD(0) return
- (b) GAE(γ,0) is the advantage function using a TD(0) return
- (c) The variance of GAE(γ,0) is likely to be larger than GAE(γ,1)
- (d) The bias of GAE(γ,0) is likely to be larger than GAE(γ,1)
- (e) Not sure

Check Your Understanding L7N2: GAE Solution

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \tag{4}$$

- GAE is an exponentially-weighted average of k-step estimators

$$\begin{aligned}
\hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma \delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma \delta_{t+1}^V + \gamma^2 \delta_{t+2}^V) + \dots) \\
&= (1-\lambda)(\delta_t^V(1 + \lambda + \lambda^2 + \dots) + \gamma \delta_{t+1}^V(\lambda + \lambda^2 + \dots) \\
&\quad + \gamma^2 \delta_{t+2}^V(\lambda^2 + \lambda^3 + \dots) + \dots) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}$$

- What are the properties of GAE(γ,0) and GAE(γ,1)? (select all)
- (a) GAE(γ,1) is the advantage function using a TD(0) return
- (b) GAE(γ,0) is the advantage function using a TD(0) return
- (c) The variance of GAE(γ,0) is likely to be larger than GAE(γ,1)
- (d) The bias of GAE(γ,0) is likely to be larger than GAE(γ,1)
- (e) Not sure

b and d are true

Generalized Advantage Estimator (GAE) Balance

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \quad (5)$$

- GAE is an exponentially-weighted average of k-step estimators

$$\begin{aligned}
\hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots) \\
&= (1-\lambda)(\delta_t^V(1+\lambda+\lambda^2+\dots) + \gamma\delta_{t+1}^V(\lambda+\lambda^2+\dots) \\
&\quad +\gamma^2\delta_{t+2}^V(\lambda^2+\lambda^3+\dots)+\dots) \\
&= (1-\gamma)(\delta_t^V \frac{1}{1-\lambda} + \gamma\lambda\delta_{t+1}^V \frac{1}{1-\lambda} + \gamma^2\lambda^2\delta_{t+2}^V \frac{1}{1-\lambda} + \dots) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}$$

- Introduced in "High-Dimensional Continuous Control Using Generalized Advantage Estimation" ICLR 2016 by Schulman et al.
- In general will prefer $\lambda \in (0,1)$ to balance bias and variance

Generalized Advantage Estimator (GAE) in PPO

- GAE is an exponentially-weighted average of k-step estimators

$$
\begin{aligned}
\hat{A}_t^{(k)} &= \sum_{l=0}^{k-1} \gamma^l r_{t+l} + \gamma^k V(s_{t+k}) - V(s_t) \\
\delta_t^V &= r_t + \gamma V(s_{t+1}) - V(s_t) \\
\hat{A}_t^{GAE(\gamma,\lambda)} &= (1-\lambda)(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}
$$

- PPO uses a truncated version of a GAE

$$
\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}^V
$$

- Benefits: Only have to run policy in environment for T timesteps before updating, improved estimate of gradient

Monotonic Improvement Theory

Return to Approximation Bound for Difference Between Two Policies

In last lecture used $d^{\pi'}$ as approximation of d^{π} (Why?)

$$J(\pi') - J(\pi) \approx \frac{1}{1-\gamma} \mathop{\mathrm{E}}_{\substack{s\sim d^{\pi} \\ a\sim\pi}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^{\pi}(s,a) \right]$$

$$\doteq \mathcal{L}_{\pi}(\pi')$$

This approximation is good when π' and π are close in KL-divergence

Relative policy performance bounds: [1]

$$\left| J(\pi') - \left(J(\pi) + \mathcal{L}_{\pi}(\pi') \right) \right| \leq C \sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi}} [D_{KL}(\pi' || \pi)[s]]} \tag{6}$$

[1] Achiam, Held, Tamar, Abbeel, 2017

Monotonic Improvement Theory

$J(\pi) = V(\pi)$

From the bound on the previous slide, we get

$$J(\pi') - J(\pi) \geq \mathcal{L}_\pi(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^\pi}\left[D_{KL}(\pi'||\pi)[s]\right]}.$$

- If we maximize the right hand side (RHS) with respect to π', we are **guaranteed to improve over π**.
  - This is a *majorize-maximize* algorithm w.r.t. the true objective, the LHS.
- And $\mathcal{L}_\pi(\pi')$ & the KL-divergence term *can both be estimated with samples from* π!

Monotonic Improvement Theory

Proof of improvement guarantee: Suppose π_{k+1} and π_k are related by

$$\pi_{k+1} = \arg\max_{\pi'} \underbrace{\mathcal{L}_{\pi_k}(\pi')}_{①} - C \underbrace{\sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi_k}} [D_{KL}(\pi'||\pi_k)[s]]}}_{②}.$$

π_k feasible

$$\underset{①}{\mathcal{L}_{\pi_k}(\pi_k)} = \frac{1}{1-\gamma} \mathop{E}_{\substack{s\sim d^{\pi_k} \\ a\sim\pi_k}} \left[\frac{\pi_k(a|s)}{\pi_k(a|s)} A^{\pi_k}(s,a) \right]$$

$$\left[Q^{\pi_k}(s,a) - V^{\pi_k}(s) \right]$$

$$\sum_a \pi_k(a|s) Q^{\pi_k}(s,a) = V^{\pi_k}(s)$$

② $= 0$

$D_{KL}(\pi_k/\pi_k) = 0$

① - ② ≥ 0 because argmax is at least as good as π_k

$J(\pi_{k+1}) - J(\pi_k) \geq$ ① - ② ≥ 0

Monotonic Improvement Theory

Proof of improvement guarantee: Suppose π_{k+1} and π_k are related by

$$\pi_{k+1} = \arg\max_{\pi'} \mathcal{L}_{\pi_k}(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi_k}}[D_{KL}(\pi'||\pi_k)[s]]}.$$

- π_k is a feasible point, and the objective at π_k is equal to 0.
  - $\mathcal{L}_{\pi_k}(\pi_k) \propto \mathop{\mathrm{E}}_{s,a\sim d^{\pi_k},\pi_k}[A^{\pi_k}(s,a)] = 0$
  - $D_{KL}(\pi_k||\pi_k)[s] = 0$
- $\Longrightarrow$ optimal value ≥ 0
- $\Longrightarrow$ by the performance bound, $J(\pi_{k+1}) - J(\pi_k) \geq 0$

This proof works even if we restrict the domain of optimization to an arbitrary class of parametrized policies Π_θ, as long as $\pi_k \in \Pi_\theta$.

Approximate Monotonic Improvement

$$\pi_{k+1} = \arg\max_{\pi'} \mathcal{L}_{\pi_k}(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi_k}}[D_{KL}(\pi'||\pi_k)[s]]}. \quad (7)$$

Problem:

- C provided by theory is quite high when γ is near 1
- $\implies$ steps from Equation (7) are too small.

Potential Solution:

- Tune the KL penalty ($\implies$ PPO)
- Use KL constraint (called **trust region**).

PPO Summary

- Improves data efficiency: can take several gradient steps before gathering more data from new policy
- Uses clipping (or KL constraint) to help increase likelihood of monotonic improvement
    - Conservative policy updating is an influential idea in RL, stemming at least from early 2000s
- Converges to local optima
- Very popular method, easy to implement, used in ChatGPT tuning

Policy Gradient Summary

- Extremely popular and useful algorithms, many beyond this class
- Can be used when the reward function is not differentiable
- Often used in conjunction with model-free value methods: actor-critic methods

Today

- Proximal policy optimization (PPO) (will implement in homework)
    - Generalized Advantage Estimation (GAE)
    - Theory: Monotonic Improvement Theory
- **Imitation Learning**[2]
    - Behavior cloning
    - DAGGER
    - Max entropy inverse RL

[2] With slides from Katerina Fragkiadaki and slides from Pieter Abbeel

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

$s, a, s', a', \ldots$)

- 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?

Behavioral Cloning

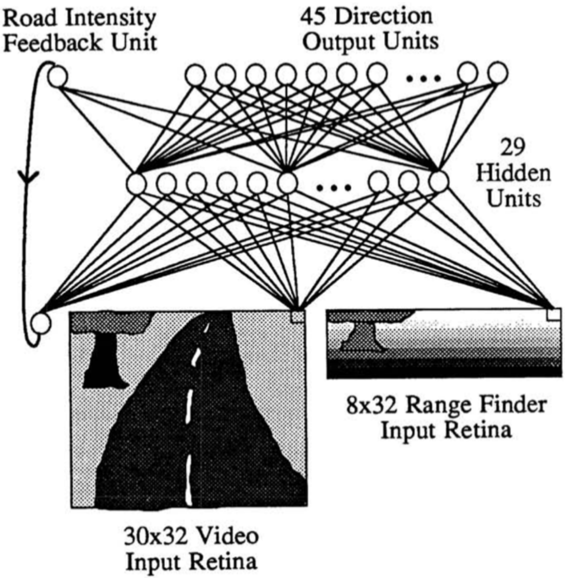
Behavioral Cloning

$s_0, a_0, s_1 \longrightarrow a_1$

$s_0, a_0, s_1, a_1, s_2 \ldots \longrightarrow a$

- Reduce problem to a standard supervised 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 early 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

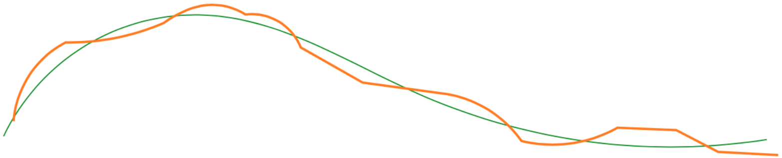
Behavioral cloning

- 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
- Extensively used in practice

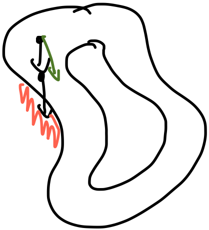
DAGGER

$s_0, a_0 \quad s_1$

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



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



Modified after class, deleted incorrect image

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

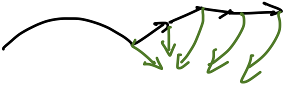
Problem: Compounding Errors

Modified after class, deleted incorrect image

- 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? human has to supervise constantly

Reward Learning

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 expert'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?

expert's

- 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.

teacher = expert

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

O

Check Your Understanding L7N3: 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 $\boldsymbol{w} \in \mathbb{R}^n, x : S \to \mathbb{R}^n$ features are $x(s)$
- Goal: identify the weight vector $\boldsymbol{w}$ given a set of demonstrations
- The resulting value function for a policy π can be expressed as

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

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

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

$$= w^T \mu(\pi)$$ ← state distrib under π discounted

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

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

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

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

- 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 $w \in \mathbb{R}^n, x : S \to \mathbb{R}^n$
- 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.

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

$$W^T \mu(\pi^*) \geq W^T \mu(\pi) \quad \forall \pi$$

experts/observed

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 $w \in \mathbb{R}^n, x : S \to \mathbb{R}^n$
- 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.

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

- 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 $\|w\|_\infty \leq 1$ (uses Holder's inequality):

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

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)