CS292F StatRL Lecture 11
Exploration in Linear MDP & Introduction to offline RL

Instructor: Yu-Xiang Wang
Spring 2021
UC Santa Barbara
Logistics

• Project midterm milestone due
  • Important as I need to allocate space for student presentation

• For those who haven’t submitted HW1
  • You don’t have to solve everything, just submit what you have
  • HW1 is long I am thinking of adjusting grading criteria

• HW2 is not as long
  • Don’t wait
Recap: Lecture 10

- Exploration in Linear MDPs
- Properties of Linear MDPs
- Algorithm: UCB-VI for Linear MDPs
- Regret analysis
Recap: Impossibility results

• What are the assumptions to make?
  • $Q^*(s,a)$ approximately linear?
  • $Q^\pi(s,a)$ is approximately linear for all $\pi$?
  • $Q^*(s,a)$ is exactly linear?
  • $Q^\pi(s,a)$ is exactly linear for all $\pi$?

Exponential sample complexity / regret lower bounds for the approximate case...

(Du, Kakade, Wang, Yang, 2019) Is a good representation sufficient for sample efficient reinforcement learning?

Weisz et al (ALT-2020):
Recap: Linear MDPs

- Exists feature map $\phi : S \times A \mapsto \mathbb{R}^d$
  - Such that:
    $$r_h(s, a) = \theta_h^* \cdot \phi(s, a), \quad P_h(\cdot|s, a) = \mu_h^* \phi(s, a), \forall h$$

(Jin et al., 2020) Provably efficient reinforcement learning with linear function approximation
Recap: UCB-VI for Linear MDPs

• In every round:
  1. Run Ridge regression for estimating the model
     \[
     \hat{\mu}_h^n = \arg\min_{\mu \in \mathbb{R}^{|s| \times d}} \sum_{i=0}^{n-1} \| \mu \phi(s_h^i, a_h^i) - \delta(s_{h+1}^i) \|^2_2 + \lambda \|\mu\|^2_F.
     \]
     \[
     \hat{\mu}_h^n = \sum_{i=0}^{n-1} \delta(s_{h+1}^i) \phi(s_h^i, a_h^i) \top (\Lambda_h^n)^{-1}
     \]
  2. Construct the exploration bonuses
     \[
     b_h^n(s, a) = \beta \sqrt{\phi(s, a) \top (\Lambda_h^n)^{-1} \phi(s, a)},
     \]
  3. Run optimistic value iterations, and update greedy policy
Recap: Regret bound

- Choose
  \[ \beta = H d \left( \sqrt{\ln \frac{H}{\delta}} + \sqrt{\ln(W + H)} + \sqrt{\ln B} + \sqrt{\ln d} + \sqrt{\ln N} \right) \]
  \[ \lambda = 1 \]
  \[ b^n_h(s, a) = \beta \sqrt{\phi(s, a)^\top (\Lambda^n_h)^{-1} \phi(s, a)} \]

- Regret
  \[ \tilde{O} \left( H^2 \sqrt{d^3 N} \right) \]
Recap: Regret analysis

• Regret of episode $t$

\[
V^* - V^{\pi_n} = V^*_0(s_0) - V^{\pi_n}_0(s_0) \leq V^*_0(s_0) - V^{\pi_n}_0(s_0)
\]

Simulation lemma \[
\leq \sum_{n=0}^{t-1} \mathbb{E} \left[ \beta_h^n(s_n a_n) + (P_h^n \mathbb{L}(S_h A_h) - P_h \mathbb{L}(S_h A_h)) \right]
\]

• Optimism / simulation lemma

• Sum them up to get total regret

Lemma (Information gain bound)

\[
A S_h A_h \text{ sequence } \sum_{n=0}^{N} \mathbb{P}(S_h^n A_h^n) (|\Lambda^n_h|)^{-1} \mathbb{E}(S_h^n A_h^n) = O(d \log N)
\]

• Same information-gain bound from linear bandits
Recap: It remains to prove

• 1. Uniform convergence bound

• 2. “Optimism”  
  The same induction argument as in the UCB-VI for tabular MDP  
  (Read Lemma 7.10 in AJKS)

• 3. “Information gain” bound
  The same argument as in the Linear Bandits case. 
  (Read Lemma 7.12 in AJKS)
Recap: Bound for a fixed $V$

- **Lemma 7.3** AJKS
  \[
  \hat{\mu}_h^n - \mu_h^* = -\lambda \mu_h^* (\Lambda_h^n)^{-1} + \sum_{i=1}^{n-1} \epsilon_h^i \phi(s_h^i, a_h^i)^\top (\Lambda_h^n)^{-1}.
  \]

- The quantity of interest is a inner product with this:
  \[
  \left( (\hat{\mu}_h^n - \mu_h^n) \cdot \phi(s,a) \right)^\top V = \phi(s,a)^\top \underbrace{(\hat{\mu}_h^n - \mu_h^n)}^\bullet \cdot V
  \]
Challenge: we cannot use union bound because we have an infinite number of value functions

• A covering number argument.

• Covering number: the number of balls with radius $\varepsilon$ that is needed to cover all points in a set.
Family of value functions we consider

\[ f_{w,\beta,\Lambda}(s) = \min \left\{ \max_a \left( w^\top \phi(s, a) + \beta \sqrt{\phi(s, a)^\top \Lambda^{-1} \phi(s, a)} \right), \ H \right\}, \forall s \in S. \]

\[ \mathcal{F} = \{ f_{w,\beta,\Lambda} : \|w\|_2 \leq L, \beta \in [0, B], \sigma_{\min}(\Lambda) \geq \lambda \}. \]

What is a finite set to cover this class such that for every f in this set, there is a function in the finite set, such that they are \( \varepsilon \)-close in sup-norm?
Covering number calculations
From covering number to a uniform convergence bound
Final notes about linear MDPs

• A semi-parametric model
  • The number of parameters to describe the model can be exponentially large: $d S$
  • Efficient algorithm with regret independent to $S$

• Still very strong assumption on the feature map
  • Interesting open problems:
    • Representation learning
    • Nonlinear parametric models
    • Suboptimal rates when naively applying to the tabular case
Remainder of the lecture

• Introduction to offline reinforcement learning

• Off-policy evaluation in contextual bandits
Recap: RL is among the hottest area of research in ML!
An RL agent learns **interactively** through the **feedbacks** of an environment.

- Learning how the world works (dynamics) and how to maximize the long-term reward (control) at the same time.
Applications of RL in the real life

- RL for robotics.
- RL for dialogue systems.
- RL for personalized medicine.
- RL for self-driving cars.
- RL for new material discovery.
- RL for sustainable energy.
- RL for feature-based dynamic pricing.
- RL for maximizing user satisfaction.
- RL for QoE optimization in networking.
- ...

19
Challenges of Reinforcement in the real life

• No access to a simulator
• Every data point is costly.
• Legal, safety issues associated with exploration.
• Large / complex state-space, action space.
• Long horizon
• Limited adaptivity (cannot run too many iterations)
From an Applied ML Scientist point of view, the starting point of a project is often:
Online RL vs Offline RL

**Online Reinforcement Learning**

Agent

Environment

**Offline Reinforcement Learning**

Agent

Logged data

Exploration is often **expensive, unsafe, unethical** or **illegal** in practice, e.g., in self-driving cars, or in medical applications.

Can we learn a policy from already **logged interaction data**?
Off-Policy learning: an example

- How to evaluate a new algorithm without actually running it live?
- How to learn a better system than the one that is deployed.
Offline Reinforcement Learning, aka. Batch RL

- Task 1: Offline Policy Evaluation. (OPE)

- Task 2: Offline Policy Learning. (OPL)

Via Uniform OPE
Contextual bandits model

• Contexts:
  • \( x_1, \ldots, x_n \sim \lambda \) drawn iid, possibly infinite domain

• Actions:
  • \( a_i \sim \mu(a|x_i) \) Taken by a randomized “Logging” policy

• Reward:
  • \( r_i \sim D(r|x_i, a_i) \) Revealed only for the action taken

• Value:
  • \( \mathcal{V}^{\mu} = \mathbb{E}_{x \sim \lambda} \mathbb{E}_{a \sim \mu(\cdot|x)} \mathbb{E}_{D}[r|x, a] \)

• We collect data \( (x_i, a_i, r_i)_{i=1}^{n} \) by the above processes.
Off-policy Evaluation and Learning

**Off-policy evaluation**

Estimate the value of a fixed target policy $\pi$

$$v_\pi := \mathbb{E}_\pi [\text{Reward}]$$

**Off-policy learning**

Find $\pi^*$ that maximizes $U_\pi$

- Using data $(x_i, a_i, r_i)_{i=1}^n$
- Often the policy $\mu$ or logged propensities $(\mu_i)_{i=1}^n$
ATE estimation is a special case of off-policy evaluation

- **a**: Action $\leftrightarrow$ T: Treatment $\{0,1\}$
- **r**: Reward $\leftrightarrow$ Y: Response variable
- **x**: Contexts $\leftrightarrow$ X: covariates
Direct Method / Regression-estimator

• Fit a regression model of the reward
  \[ \hat{r}(x, a) \approx \mathbb{E}(r|x, a) \]
  using the data

• Then for any target policy
  \[ \hat{v}^\pi_{\text{DM}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in A} \hat{r}(x_i, a) \pi(a|x_i) \]

Pros:
• Low-variance.
• Can evaluate on unseen contexts

Cons:
• Often high bias
• The model can be wrong/hard to learn
Inverse propensity score / Importance sampling

(Horvitz & Thompson, 1952)

\[ \hat{\nu}^{\pi}_{\text{IPS}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i \mid x_i)}{\mu(a_i \mid x_i)} r_i \]

Pros:
- No assumption on rewards
- Unbiased
- Computationally efficient

Cons:
- High variance when the weight is large
Next lecture: OPE for reinforcement learning

• Importance sampling

• Marginalized importance sampling