17-740 Spring 2025
Algorithmic Foundations of Interactive Learning

Hybrid RL

Lecturer : Yuda Song Scribe: Apurva Gandhi, Sangyun Lee, Ryan Schuerkamp, Grace Liu

16.1 Introduction

16.1.0.1 Offline data

Some examples of offline data are YouTube videos and teleoperation data for robotics.

We sample $s, a \sim \mu$, $r = R(s, a)$ (assume deterministic reward), $s' \sim P(\cdot|s, a)$

Unlike imitation learning, we do not place any guarantee on the quality of the distribution μ and the samples – they can be meaningfully suboptimal. However, we do see rewards.

16.1.0.2 Value learning

Notation. We consider finite horizon Markov Decision Process $M = \{S, A, H, R, P, d_0\}$. We define a policy π where $\pi_h : S \mapsto \Delta(A)$ and let d^π denotes the visitation distribution induced by π at step h. Let $V^\pi(s) = \mathbb{E}[\sum_{\tau=0}^{H-1} r_\tau|\pi, s_h = s]$ and $Q_h^\pi(s, a) = \mathbb{E}[\sum_{\tau=0}^{H-1} r_\tau|\pi, s_h = s, a_h = a]$ be value functions and let $Q^\star$ and $V^\star$ denote the optimal value functions. We will use the following pieces of notation repeatedly:

Definition 1 *Let π_Q to be the greedy policy w.r.t. a state-action value function Q.*

$$\pi^Q(s) = \arg\max_{a \in \mathcal{A}} Q(s, a) \tag{16.1}$$

Definition 2 *We define the Bellman operator $\mathcal{T}$ such that for any $f : S \times A \mapsto \mathbb{R}$,*

$$\mathcal{T} f(s, a) = \mathbb{E}[r(s, a)] + \mathbb{E}_{s' \sim P(s,a)} \max_{a'} f(s', a'). \tag{16.2}$$

Recall that $\mathcal{T}Q^\star = Q^\star$ – the optimal Q function is a *fixed point* of the Bellman Operator. Thus, one way to learn a good Q function is to minimize the **Bellman error** – the difference between the two sides of the fixed point condition. More formally,

Algorithm 1 Q-Value Iteration

Require: MDP $M = \{S, A, H, R, P, d_0\}$
  Initialize $Q_H(s, a) = 0$ for all $(s, a) \in S \times A$
  for $h = H - 1, H - 2, \ldots, 0$ **do**
    for each $(s, a) \in S \times A$ **do**
      $Q_h(s, a) \leftarrow \mathcal{T} Q_{h-1}$
    end for
  end for
  Return $\{Q_0, Q_1, \ldots, Q_{H-1}\}$ =0

Definition 3 *Bellman Error is defined as* $f - \mathcal{T} f$.

Per the above, the Bellman Error of the optimal Q function is $Q^\star - \mathcal{T} Q^\star = 0$.

Recall the Q-Value iteration algorithm (Alg. 1). Observe that this algorithm is attempting to minimize the Bellman Error by iteratively applying the Bellman Operator. Once we have near optimal Q-value estimates ($\hat{Q} \simeq Q^\star$), we can get a near optimal policy via a simple action-level argmax: $\pi^{Q^\star} = \pi^\star$. However, these sorts of tabular algorithms assume we have access to the transition dynamics ($P(\cdot|s, a)$) *everywhere*. If we have to learn the dynamics from data, we have to answer the question of where it is most important to do so to ensure value learning leads to a strong policy.

16.2 Where should Bellman error be minimized?

16.2.1 Naive case, assuming we have data everywhere

If we have data for every state-action pair, we do not need to explore online.

Assumption 1 (Full Coverage) *A distribution $\mu(s, a)$ is said to satisfy* full coverage *if*

$$\frac{1}{\mu(s, a)} \leq C \quad \textit{for all } (s, a).$$

This condition implies that the density ratio $\frac{d^\pi(s,a)}{\mu(s,a)}$ is bounded for any policy π, where $d^\pi(s, a)$ is the *visitation distribution* defined as the probability of encountering the state-action pair (s, a) when following policy π starting from the initial state distribution.

Under full coverage, our dataset is sufficiently rich so that we do not need to explore online. Instead, we can learn a model, plan using it, and extract a greedy policy. This approach

is known as **certainty-equivalent model-based RL**. Given a fixed dataset, the method computes the policy as follows:

$$\hat{P}(s'|s,a) = \frac{\text{count}(s,a,s')}{\text{count}(s,a)}$$

$$\hat{\pi}(s) = \arg\max_a \hat{Q}^{\hat{P}}(s,a) \text{ , where } Q^{\hat{P}} \text{ is obtained via Q-value iteration inside } \hat{P} \text{ (MBRL)}$$

A standard uniform convergence argument tells us that with enough samples from μ, the learned dynamics/transitions are close to the true ones. Moreover, the more data you have, the lower the error will be. More formally, a Hoeffding + union bound (out of scope) tells us that

$$||\hat{P}(\cdot|s,a) - P(\cdot|s,a)||_{\text{TV}} \leq \epsilon \coloneqq |S|\sqrt{\frac{C}{h}} \quad \forall s,a.$$

Recall that under the assumption that we have a model that is good everywhere, i.e.

$$\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\|_{\text{TV}} \leq \epsilon \quad \forall\,(s,a),$$

the Simulation Lemma implies that for any state-action pair,

$$\left|Q(s,a) - \hat{Q}(s,a)\right| \leq H^2\epsilon.$$

Proof: For any state s, let $\pi^\star$ be the optimal policy for the true MDP M and let $\hat{\pi}$ be the greedy policy with respect to $\hat{Q}$. Then, via an add and subtract trick,

$$\begin{aligned} Q(s,\pi^*(s)) - Q(s,\hat{\pi}(s)) &= \underbrace{Q(s,\pi^*(s)) - \hat{Q}(s,\pi^*(s))}_{\leq H^2\epsilon} + \underbrace{\hat{Q}(s,\pi^*(s)) - Q(s,\hat{\pi}(s))}_{\leq H^2\epsilon} \\ &\leq H^2\epsilon + H^2\epsilon \\ &= 2H^2\epsilon. \end{aligned}$$

Applying the simulation lemma to both terms yields the final bound.

■

Note: in translating between finite horizon and discounted infinite horizon setting, $\frac{1}{1-\gamma} \simeq H$. We will interchange throughout the lecture for convenience of analysis.

16.2.2 All-π concentrability

All-π concentrability: $\forall\pi$, $\max_{s,a} \frac{d^\pi(s,a)}{\mu(s,a)} \leq C$. In other words, instead of needing to cover every state-action pair, we only need to cover state and actions where any policies could visit.

However, there may be policies that visit every state-action pair, making this condition not that much weaker than full coverage.

Let us now introduce function approximation – i.e., going beyond the tabular setting. We'll search over a class of candidate value functions $\mathcal{F}$ for some $f \in \mathcal{F}$ such that $f \simeq Q^\star$. This function class is defined as $\mathcal{F} \subseteq S \times A \mapsto [0, V_{\max}]$. The natural offline algorithm for this setting is called Fitted Q-Iteration (FQI), and it minimizes the squared Bellman error over your offline dataset (Algorithm 2).

Algorithm 2 Fitted Q-iteration (FQI)

for h: H, ..., 0 **do**

$\mathrm{f}_h \leftarrow \arg\min_{f \in \mathcal{F}} \hat{\mathbb{E}}_{s,a}[(f(s,a) - \mathcal{T} f_{h+1}(s,a))^2]$

To prove guarantees for the above algorithm, we'll need the **Bellman Completeness Assumption**: $\forall f \in \mathcal{F}$, $\mathcal{T} f \in \mathcal{F}$ – i.e. the Bellman backup of an arbitrary function in $\mathcal{F}$ is contained in $\mathcal{F}$. This is a very strong assumption, which is satisfied by limited settings such as the tabular setting (the setting considered in this course). For instance, this doesn't hold for the deep neural net policies. The reason this is strong is that because adding a function can break the assumption, unlike expert realizability in imitation learning where expanding the policy class can only help. Given the strength of these assumptions, we will now consider a weaker set of assumptions and a corresponding algorithm that works under them.

16.2.3 Single-Policy Coverage (SPC)

Definition 4 (Single-Policy Coverage) *For some policy π, a distribution $\mu(s,a)$ satisfies* single-policy coverage *if*

$$\max_{s,a} \frac{d^\pi(s,a)}{\mu(s,a)} \leq C^\pi < \infty$$

where $d^\pi(s,a)$ is the visitation distribution under policy π and C^π is the coverage coefficient.

Moving from all-π concentratability to SPC is similar to moving from for all to there exists. Intuitively, we should be able to compete against policies where we've seen what they'll do in our dataset.

There are lower bounds showing that even in the realizable setting with Bellman completeness, SPC is insufficient to guarantee strong performance in the offline setting—online algorithms can surpass these lower bounds. In practice, pure offline RL methods often face severe distribution shift, making it challenging to determine when to stop training or which checkpoint to deploy. Furthermore, model selection in offline RL often relies on some form of

online testing (e.g., testing the performance of different checkpoints online and then choosing the best one), so directly incorporating online interactions may not be too much of a leap. The hybrid RL framework addresses these issues by using limited online data to correct for distribution mismatch, thereby enabling more reliable policy improvement—if we have access to an online environment, then SPC can be made to work.

Hybrid RL - You are given offline data but also access to online environment access. Note that Agnostic System Identification [1] which we covered in the model-based RL lecture (Lecture 15) was the first example of this setting. Here we introduce a model-free counterpart.

Algorithm 3 Hybrid Q-iteration (HyQ) [2]

We have offline dataset $\mathcal{D}_{\text{off}}$, initialize f^0
for $t = 1, \ldots, T$ **do**
  $\pi_h^t(s) \leftarrow \arg\max_a f_h^{t-1}(s,a);\ \forall h \in [1, ..., H]$
  $\mathcal{D}_{\text{on}} \leftarrow \mathcal{D}_{\text{on}} \cup (s, a, r, s') \sim \pi^t$ (online data)
  $f^t \leftarrow \text{FQI}(\mathcal{D}_{\text{off}} \cup \mathcal{D}_{\text{on}})$
end for=0

Note, that h above indexes the timesteps of the MDP and the inner FQI loop, while t indexes the iterations of the outer loop. In each outer-loop step, we learn a sequence of H policies in the finite horizon setting. Observe that we aggregate online data like in DAgger.

In Hybrid RL, the goal is to compete against the best π with $C^\pi \leq \infty$, meaning we aim to compete against any policy covered by the offline dataset. This is the same fundamental objective as offline RL, but the ability to interact with the environment in Hybrid RL allows us to circumvent the lower bounds that restrict purely offline methods. This goal is also quite reasonable—if a policy is represented in the dataset, we should be able to imitate or even surpass it. While we are typically limited to competing with the best policy contained in the offline dataset, there are variations of this setting that allow us to get closer to the optimal policy $\pi^\star$, often by incorporating an exploration bonus to guide the online interactions.

16.2.3.1 Why Should HyQ Work?

Intuitively, running FQI on the hybrid data guarantees low Bellman Error on states from π^e and the policy induced by HyQ π^f. We now argue this is sufficient to guarantee we learn a strong policy. We begin with a lemma analogous to the Performance Difference Lemma.

Lemma 2 *Given any comparator policy π^e, for any $f \in \mathcal{F}$ and corresponding greedy policy*

π^f, *we have*

$$\mathbb{E}_{s_0\sim d_0}\Big[V_0^{\pi^e}(s_0) - V_0^{\pi^f}(s_0)\Big] \leq \underbrace{\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_e}}\Big[\mathcal{T}f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big]}_{\textit{offline error}} + \underbrace{\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi^f}}\Big[f_h(s_h,a_h) - \mathcal{T}f_{h+1}(s_h,a_h)\Big]}_{\textit{online error}}.$$

Proof: We can consider the following decomposition:

$$\mathbb{E}_{s_0\sim d_0}\Big[V_0^{\pi^e}(s_0) - V_0^{\pi_f}(s_0)\Big] = \mathbb{E}_{s_0\sim d_0}\Big[V_0^{\pi^e}(s_0) - \max_a f_0(s_0,a) + \max_a f_0(s_0,a) - V_0^{\pi_f}(s_0)\Big].$$

We bound the second difference using a variant of the Performance Difference Lemma:

$$\begin{aligned}
\mathbb{E}_{s\sim d_0}\Big[\max_a f_0(s,a) - V^{\pi^f}(s)\Big] &\overset{(1)}{=} \mathbb{E}_{s\sim d_0}\Big[\mathbb{E}_{a\sim\pi_0^f(s)}\big(f_0(s,a) - V_0^{\pi^f}(s)\big)\Big] \\
&\overset{(2)}{=} \mathbb{E}_{s\sim d_0}\Big[\mathbb{E}_{a\sim\pi_0^f(s)}\big(f_0(s,a) - \mathcal{T}f_1(s,a)\big)\Big] \\
&\quad + \mathbb{E}_{s\sim d_0}\Big[\mathbb{E}_{a\sim\pi_0^f(s)}\big(\mathcal{T}f_1(s,a) - V_0^{\pi^f}(s)\big)\Big] \\
&\overset{(3)}{=} \mathbb{E}_{(s,a)\sim d_0^{\pi^f}}\Big[f_0(s,a) - \mathcal{T}f_1(s,a)\Big] \\
&\quad + \mathbb{E}_{s\sim d_0}\Bigg[\mathbb{E}_{a\sim\pi_0^f(s)}\Bigg(R(s,a) + \gamma\,\mathbb{E}_{s'\sim P(s,a)}\max_{a'} f_1(s',a') \\
&\qquad\qquad - R(s,a) + \mathbb{E}_{s'\sim P(s,a)}V_1^{\pi^f}(s')\Bigg)\Bigg] \\
&\overset{(4)}{=} \mathbb{E}_{(s,a)\sim d_0^{\pi^f}}\Big[f_0(s,a) - \mathcal{T}f_1(s,a)\Big] + \mathbb{E}_{s\sim d_1^{\pi^f}}\Big[\max_a f_1(s,a) - V_1^{\pi^f}(s)\Big].
\end{aligned}$$

(1) follows from the definition of the greedy policy. (2) follows from an add and subtract trick. (3) follows from expanding out the Bellman operator and usimg the definition of the value function. Lastly, (4) follows from canceling terms, leaving us with what we started with except one step further in the future. Then, we we can solve the recurrence relation for the overall bound. We can similarly bound the first term, which is done in [2].

■

16.2.3.2 Completing the Proof

To complete the performance bound, note that offline error term can be further bounded by a change of measure. Specifically, noting that

$$\mathbb{E}_{(s,a)\sim d^{\pi_e}}[h(s,a)] = \mathbb{E}_{(s,a)\sim\mu}\left[\frac{d^{\pi_e}(s,a)}{\mu(s,a)}\, h(s,a)\right],$$

with $h(s,a) = \mathcal{T} f_{h+1}(s,a) - f_h(s,a)$ (i.e. the Bellman Error at (s,a)), we have

$$\sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_e}}\Big[\mathcal{T} f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big] = \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\mu}\left[\frac{d^{\pi_e}(s_h,a_h)}{\mu(s_h,a_h)}\Big(\mathcal{T} f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big)\right]$$

Next, we can apply the Cauchy-Shwartz inequality:

$$\begin{aligned}
&\leq \sum_{h=1}^{H} \sqrt{\mathbb{E}_{(s_h,a_h)\sim\mu}\left[\left(\frac{d^{\pi_e}(s_h,a_h)}{\mu(s_h,a_h)}\right)^2\right] \cdot \mathbb{E}_{(s_h,a_h)\sim\mu}\left[\Big(\mathcal{T} f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big)^2\right]} \\
&\leq \sum_{h=1}^{H} \sqrt{C \cdot \mathbb{E}_{(s_h,a_h)\sim\mu}\left[\Big(\mathcal{T} f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big)^2\right]},
\end{aligned}$$

where the final inequality uses the SPC assumption w.r.t. π_e, i.e. that

$$\frac{d^{\pi_e}(s,a)}{\mu(s,a)} \leq C \quad \text{for all } (s,a).$$

This gives us an overall bound of

$$\mathbb{E}_{s_0\sim d_0}\Big[V_0^{\pi^e}(s_0) - V_0^{\pi^f}(s_0)\Big] \leq \sum_{h=1}^{H} \sqrt{C \cdot \mathbb{E}_{(s_h,a_h)\sim\mu}\left[\Big(\mathcal{T} f_{h+1}(s_h,a_h) - f_h(s_h,a_h)\Big)^2\right]} \tag{16.3}$$

$$+ \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim d_h^{\pi^f}}\left[f_h(s_h,a_h) - \mathcal{T} f_{h+1}(s_h,a_h)\right]. \tag{16.4}$$

Observe that minimizing squared Bellman Error on samples from μ makes the first term small. Under certain assumptions we don't discuss here, an analogous bound can be proved for the second term in terms of squared Bellman Error – see [2] for the full proof. The key take-away from the above proof is that hybrid RL allows us to compete against a policy with just SPC, which purely offline algorithms provably cannot achieve. Furthermore, hybrid RL is efficient, in the sense we can avoid explicit optimism / pessimism procedures.

References

[1] Stephane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

[2] Yuda Song, Yifei Zhou, Ayush Sekhari, Drew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. In *The Eleventh International Conference on Learning Representations*.