Lecture 11: Structural conditions for nonlinear function approximation

Akshay Krishnamurthy
akshay@cs.umass.edu

April 11, 2022

In the last few lecture, we focused on linear function approximation and found that the question of statistical efficiency is quite subtle. In particular, we require some *structural conditions* beyond simply realizability of the optimal value function to obtain efficient algorithms. While the linear setting is a good place to start, of course we would like to understand what can be done with more general non-linear functions. Since the linear setting is a special case, we will still require structural conditions here. In this lecture we will cover some conditions that allow us to use non-linear classes.

1 Review of the linear Bellman complete setting

The best place to start is from an algorithmic perspective. Here it is worthwhile to review the algorithm we studied for the linear Bellman complete setting, which we will see can be substantially generalized. The reason to start with this algorithm is that the "global optimism" approach is much more flexible then the "local optimism" approach, since demanding pointwise optimism places quite strong constraints on the function class. But global optimism is much easier to achieve.

As usual, we have a finite horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \mu)$ with episode length H. We also have a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and consider using a linear function to fit $Q^\star_h$. To this end we assume realizability, or the existence of $\theta^\star_{0:H}$ such that $Q^\star_h(s,a) = \langle \theta^\star_h, \phi(s,a)\rangle$ and we assume *bellman completeness*. Recall that we define the bellman backup operator as

$$\mathcal{T} : (\mathcal{S} \times \mathcal{A} \to \mathbb{R}) \to (\mathcal{S} \times \mathcal{A} \to \mathbb{R}) \text{ via } (\mathcal{T}f)(s,a) = \mathbb{E}[r + \max_{a'} f(s',a') \mid s,a].$$

With linear functions we sometimes change the domain of the bellman operator to be d-dimensional vectors, i.e., $(\mathcal{T}\theta)(s,a) = \mathbb{E}[r + \max_{a'} \langle \theta, \phi(s',a')\rangle \mid s,a]$. Bellman completeness requires that for any θ there exists a $\bar\theta$ such that $\forall (s,a) : \langle \bar\theta, \phi(s,a)\rangle = (\mathcal{T}\theta)(s,a)$.

The algorithm we studied for the linear Bellman complete setting maintained "nested" confidence ellipsoids and used global optimism to choose the policy to deploy in each episode. Specifically, at iteration t the algorithm computes the following:

$$\begin{aligned}
R_h^{t-1}(\theta, \tilde\theta) &:= \sum_{i=1}^{t-1} \left(\langle \phi(s_h^i, a_h^i), \theta\rangle - r_h^i - \max_a \left\langle \phi(s_{h+1}^i, a), \tilde\theta \right\rangle \right)^2 + \lambda \|\theta\|_2^2, \\
\textsc{Ball}^t &:= \left\{ (\theta_0, \ldots, \theta_H) : \theta_H = 0, \forall h : R_h^{t-1}(\theta_h, \theta_{h+1}) \le \min_\theta R_h^{t-1}(\theta, \theta_{h+1}) + \beta^2 \right\} \\
\vec\theta^t &\leftarrow \operatorname*{argmax}_{(\theta_0,\ldots,\theta_H) \in \textsc{Ball}^t} \mathbb{E}_{s_0} \max_a \langle \phi(s_0,a), \theta_0\rangle .
\end{aligned}$$

Then we deploy the greedy policy with respect to $\vec\theta^t$ to collect an episode. This is a version space algorithm that keeps plausible hypotheses $\vec\theta$ that satisfy the constraints imposed by the confidence ball. These constraints can be interpreted as demanding that $\vec\theta$ is "consistent" on the states and actions previously visited. Indeed, we can show

$$\vec\theta \in \textsc{Ball}^t \Rightarrow \forall h : \sum_{i=1}^{t-1} \left(\langle \theta_h, \phi(s_h^i, a_h^i)\rangle - (\mathcal{T}\theta_{h+1})(s_h^i, a_h^i) \right)^2 \le O(\beta^2).$$

This connects to the regret analysis via the regret decomposition for globally optimistic algorithms.

Lemma 1 (Global optimistic regret decomposition)**.** *Suppose we have a Q function $(Q_0, \ldots, Q_{H-1})$ such that $\mathbb{E}_{s_0} \max_a Q_0(s_0, a) \geq \mathbb{E}_{s_0} \max_a Q_0^\star(s_0, a)$ and we set π to be the greedy policy with respect to $\vec{Q}$. Then*

$$J(\pi^\star) - J(\pi) \leq \sum_{h=0}^{H-1} \mathbb{E}_{(s,a)\sim d_h^\pi} \left[Q_h(s,a) - (\mathcal{T} Q_{h+1})(s,a) \right].$$

Now the regret of π is upper bounded by its "inconsistency" on its own visitation distribution, while the version space eliminates functions that are inconsistent on previously seen state-action pairs. Therefore, if the policy that we deploy (which is in the version space) has high regret, intuitively it must find a new and very different constraint. Finally, the linear structure allows us to use the elliptical potential to argue that the latter cannot happen too often.

2 Generalizing to non-linear functions

In fact, most of the above argument does not require linear functions at all. The only place where linearity is used is when we apply the elliptial potential lemma to bound the number of times we can find new constraints. But there could be many other ways to do this. For example, maybe the dynamics have a "bottleneck" structure where the distribution over (s_h, a_h) we obtain when deploying a policy π is very simple (or lies in a low-dimensional set). If this were true, perhaps we can argue that once we visit enough of these distributions we will eliminate all of the inconsistent functions, without appealing to linear function approximation. We will see many examples shortly, but let us first describe how to generalize the above algorithm.

The new algorithm will have essentially three differences. First, we'll generalize to an abstract function class $\mathcal{F} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Second we'll avoid the completeness assumption which puts us at risk of facing the double sampling issue. We'll avoid this by collecting many samples and using averaging. Finally we'll measure the Bellman errors in a slightly different way.

The key quantity is the *average Bellman error* of function $\vec{f}$ witnessed by policy π. This is defined as

$$\begin{aligned}
\mathcal{E}_h(\pi, \vec{f}) &:= \mathbb{E}\left[f_h(s_h, a_h) - r_h - \max_{a'} f_{h+1}(s_{h+1}, a') \mid s_h \sim d_h^\pi, a_h = \pi_f(s_h) \right] \\
&= \mathbb{E}\left[f_h(s_h, a_h) - (\mathcal{T} f_{h+1})(s_h, a_h) \mid s_h \sim d_h^\pi, a_h = \pi_f(s_h) \right]
\end{aligned}$$

This is the analog of the constraints we used in the linear setting but there are two important differences. First we are not imposing constraints on single state-action pairs, but rather on *distributions* induced by the policies. Second the action a_h here is chosen by the (policy induced by the) Q-function we are trying to evaluate, rather than the policy we used to roll in. This provides a decoupling effect that is important in some applications but not fundamental to the broader approach.

Given policy π we can easily estimate $\mathcal{E}_h(\pi, \vec{f})$ for all $\vec{f} \in \mathcal{F}$ simultaneously by collecting a single dataset of n samples where we roll-in to $s_h \sim \pi$ and take a_h uniformly at random. Given the dataset $\{(s_h^i, a_h^i, r_h^i, s_{h+1}^i)\}_{i=1}^n$ sampled from this process we form the estimator using importance weighting:

$$\widehat{\mathcal{E}}_h(\pi, \vec{f}) := \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{1}\{a_h^i = \pi_f(s_h^i)\}}{1/A} \cdot \left(f_h(s_h^i, a_h^i) - r_h^i - \max_{a'} f_{h+1}(s_{h+1}^i, a') \right)$$

A elementary concentration argument using Bernstein's inequality shows that with probability at least $1 - \delta$

$$\forall f \in \mathcal{F} : \left| \widehat{\mathcal{E}}_h(\pi, \vec{f}) - \mathcal{E}_h(\pi, \vec{f}) \right| \leq O\left(\sqrt{\frac{A \log(|\mathcal{F}|/\delta)}{n}} \right)$$

The idea is to construct the confidence ball by asking that $\left| \widehat{\mathcal{E}}_h(\pi^i, \vec{f}) \right| \leq \epsilon$ for each previous policy π^i that we have deployed. This is justified by the Bellman consistency intuition and the fact that $\mathcal{E}_h(\pi, \vec{Q}^\star) = 0$ for all policies π.

We summarize the algorithm in Algorithm 2.1. The idea is very similar to the linear Bellman complete case. In each iteration we select the globally optimistic value function $\vec{f}^t$ subject to the confidence set constraints. We

Algorithm 2.1 Bilin-UCB

Input: function class $\mathcal{F}$, accuracy and failure parameters (ϵ, δ).
Define $\mathcal{F}_0 \leftarrow \mathcal{F}$.
for $t = 1, \ldots, T$ **do**
  Let $\vec{f}^t = \operatorname{argmax}_{\vec{f} \in \mathcal{F}_{t-1}} \mathbb{E}_{s_0}\left[\max_a f_0(s_0, a)\right]$ (Estimate from samples if necessary)
  for each $h \in [H]$ **do**
    Collect n_{est} samples $\{(s_h^i, a_h^i, r_h^i, s_{h+1}^i)\}_{i=1}^{n_{\mathrm{est}}}$ where $s_h^i \sim d_h^{\pi^t}, a_h^i \sim \mathrm{Unif}(\mathcal{A})$ and form $\widehat{\mathcal{E}}_h(\pi^t, \vec{f})$ for each $f \in \mathcal{F}$.
  end for
  Define $\mathcal{F}_t$ as

$$\mathcal{F}_t = \left\{ \vec{f} \in \mathcal{F} : \forall h : \sum_{i=1}^{t} (\widehat{\mathcal{E}}_h(\pi^i, \vec{f}))^2 \leq \beta^2 \right\}$$

end for
Output the best π^t found, according to estimates of $J(\pi^t)$.

will use the greedy policy induced by $\vec{f}^t$, which we call π^t for data collection in this iteration. Then we collect data for each time step h to estimate $\mathcal{E}_h(\pi^t, \vec{f})$ and we incorporate this new constraint into our confidence set. The confidence set should be viewed as the analog of $\textsc{Ball}^t$ from the previous algorithm.

This algorithm is analyzed assuming *realizability*, meaning that $\vec{Q}^\star \in \mathcal{F}$. The analysis is based on the following steps. First a concentration analysis for $\widehat{\mathcal{E}}$ results in a choice for β in terms of the number of iterations T to ensure that $Q^\star \in \mathcal{F}_{t-1}$ for all t. This also results in a bound on the "constraint violations" for any candidate $\vec{f} \in \mathcal{F}_{t-1}$. Then we use the regret decomposition to argue that either $\vec{f}^t$ is near optimal or it provides a significantly novel constraint. The key will be to identify a new potential function to show that the latter cannot happen too often.

Let us describe the concentration part in more detail. First observe that the algorithm uses $O(n_{\mathrm{est}} HT)$ episodes, where both n_{est} and T will be set later. Taking a union bound over all $h \in [H]$ and $t \in [T]$ we know that

$$\forall h, t, \vec{f} : \left| \widehat{\mathcal{E}}_h(\pi^t, \vec{f}) - \mathcal{E}_h(\pi^t, \vec{f}) \right| \leq O\left(\sqrt{\frac{A \log(|\mathcal{F}|HT/\delta)}{n_{\mathrm{est}}}} \right) =: \epsilon_{\mathrm{est}}$$

We want to use this bound to set β. Since we know that $\mathcal{E}_h(\pi^t, \vec{Q}^\star) = 0$ we can be sure that

$$\sum_{i=1}^{T} (\widehat{\mathcal{E}}_h(\pi^i, \vec{Q}^\star))^2 \leq \sum_{i=1}^{T} (\widehat{\mathcal{E}}_h(\pi^i, \vec{Q}^\star) - \mathcal{E}_h(\pi^i, \vec{Q}^\star))^2 \leq T\epsilon_{\mathrm{est}}^2$$

So if we set $\beta^2 = T\epsilon_{\mathrm{est}}^2$ we can be sure that $\vec{Q}^\star \in \mathcal{F}_t$ for all $t \in [T]$. We also know that for any $\vec{f} \in \mathcal{F}^{t-1}$ we have

$$\sum_{i=1}^{t-1} (\mathcal{E}_h(\pi^i, \vec{f}))^2 \leq \sum_{i=1}^{t-1} 2(\widehat{\mathcal{E}}_h(\pi^i, \vec{f}))^2 + 2t\epsilon_{\mathrm{est}}^2 \leq 4\beta^2 \tag{1}$$

By the optimistic regret decomposition, if $\vec{f}^t$ is ϵ_{opt} sub-optimal, then we have

$$\epsilon_{\mathrm{opt}} = J(\pi^\star) - J(\pi^t) \leq \sum_h \mathbb{E}_{s,a \sim d_h^{\pi^t}}[f_h(s,a) - (\mathcal{T} f_{h+1})(s,a)] = \sum_h \mathcal{E}_h(\pi^t, \vec{f}^t) \tag{2}$$

This implies that there must exist some h for which

$$(\mathcal{E}_h(\pi^t, \vec{f}^t))^2 \geq \epsilon_{\mathrm{opt}}^2 / \sqrt{H} \tag{3}$$

Now we can more clearly see why π^t is providing a significantly new constraint. Since $\vec{f}^t \in \mathcal{F}_{t-1}$ we know that $\mathcal{E}_h(\pi^i, \vec{f}^t)$ is small for all previous policies π^i, by (1). On the other hand, (3) shows that if π^t is highly suboptimal, then $\mathcal{E}_h(\pi^t, \vec{f}^t)$ must be large. So π^t must be providing us with new information. The question is how long can we continue to acquire new information?

3 Structural conditions

In general we can acquire new information for an exponentially long time. But in many more-structured settings, this is not the case. To capture these favorable conditions, we now introduce the main structural assumption.

Definition 2 (Bellman rank)**.** *Let $\mathcal{F}$ be given and let Π be the induced policy class, $\Pi = \{\pi_f : \vec{f} \in \mathcal{F}\}$. For each h we assume there exists embedding functions $w_h : \Pi \to \mathbb{R}^d$ and $v_h(\vec{f}) : \mathcal{F} \to \mathbb{R}^d$ such that the average Bellman error factorizes as $\mathcal{E}_h(\pi, \vec{f}) = \left\langle w_h(\pi), v_h(\vec{f}) \right\rangle$. Here d is the* Bellman rank *of the problem. We also assume that $\|w_h(\pi)\|_2 \leq W$ and $\|v_h(\vec{f})\|_2 \leq V$, for normalization.*

Many interesting models admit low Bellman rank. However, note that although we are assuming some linear structure in the problem, this is much more general than linear function approximation. Here the structure is not directly on the class $\mathcal{F}$, but rather on how the class *interacts* with the MDP. Before we turn to the examples let us finish the analysis of the algorithm. As we have linear structure, it is natural to use the elliptical potential.

Lemma 3. *Let $x_1, \ldots, x_T$ be a sequence of vectors with $\|x_t\|_2 \leq B$ and define $\Sigma_0 = \lambda I$, $\Sigma_t = \Sigma_{t-1} + x_t x_t^\top$. Then*

$$\sum_{t=1}^{T} \min(1, x_t^\top \Sigma_{t-1}^{-1} x_t) \leq d \log\left(1 + \frac{TB^2}{d\lambda}\right)$$

To finish the proof, we write (1) and (2) in terms of the Bellman rank embeddings. Fix $\lambda > 0$ and let $\Sigma_{0,h} = \lambda I$ and $\Sigma_{t,h} = \Sigma_{t-1,h} + w_h(\pi^t) w_h(\pi^t)^\top$ Then summing up the bound from the elliptical potential over h gives

$$\sum_h \sum_t \min(1, w_h(\pi^t)^\top \Sigma_{t-1,h}^{-1} w_h(\pi^t)) \leq Hd \log\left(1 + TW^2/(d\lambda)\right)$$

Since all the terms are positive, this implies that there exists a t such that:

$$\forall h : \min(1, w_h(\pi^t)^\top \Sigma_{t-1,h}^{-1} w_h(\pi^t)) \leq \frac{Hd}{T} \cdot \log\left(1 + TW^2/(d\lambda)\right)$$

If T is large enough that the RHS is strictly less than 1, we can drop the min on the LHS. Under this condition, we will show that policy π^t is near optimal. Indeed, we know that

$$J(\pi^\star) - J(\pi^t) \leq \sum_h \mathcal{E}_h(\pi^t, f^t) = \sum_h \left\langle w_h(\pi^t), v_h(f^t) \right\rangle \leq \sum_h \|w_h(\pi_t)\|_{\Sigma_{t-1,h}^{-1}} \cdot \|v_h(f^t)\|_{\Sigma_{t-1,h}}$$

We just bounded the first term and the second term can be bounded using the version space constraint in (1)

$$\|v_h(f^t)\|_{\Sigma_{t-1,h}}^2 \leq \lambda V^2 + \sum_{i=0}^{t-1} \left\langle w_h(\pi^i), v_h(f^t) \right\rangle^2 = \lambda V^2 + \sum_{i=0}^{i-1} (\mathcal{E}_h(\pi^i, f^t))^2 \leq \lambda V^2 + 4\beta^2$$

Putting everything together, we have shown that there exists a t such that

$$J(\pi^\star) - J(\pi^t) \leq H \sqrt{\frac{Hd}{T} \cdot \log(1 + TW^2/(d\lambda))} \cdot \sqrt{\lambda V^2 + \frac{4TA \log(|\mathcal{F}|TH/\delta)}{n_{\text{est}}}}.$$

Theorem 4. *Consider any problem with Bellman rank d and suppose that $\vec{Q}^\star \in \mathcal{F}$. Set:*

$$\lambda = \frac{TA \log(|\mathcal{F}|TH/\delta)}{n_{\text{est}} V^2}, \quad T \geq Hd \log(1 + TW^2/(d\lambda)), \quad n_{\text{est}} \geq \frac{H^3 dA \log(|\mathcal{F}|TH/\delta) \log(1 + TW^2/(d\lambda))}{\epsilon^2}$$

Then with probability at least $1 - \delta$, Bilin-UCB outputs a policy $\hat{\pi}$ satisfying $J(\pi^\star) - J(\hat{\pi}) \leq \epsilon$ while using at most $O(HTn_{\text{est}})$ samples.

4 Examples, generalizations, history

The Bellman rank can be small even in problems with high-dimensional/complex observations and even when $\mathcal{F}$ is an arbitrary set of functions (satisfying realizability). We can think of the embedding property as capturing two things simultaneously: whether $\mathcal{F}$ supports extrapolation (like linear functions), or whether there are not to many distinct roll-in distributions d_h^π. We have seen the first feature in the linear Bellman complete setting and you should verify for yourself that the Bellman rank is d. The next example highlights the other property.

Block MDPs. A block MDP is a problem with high dimensional inputs but where the dynamics are governed by a finite *latent* state space $\mathcal{Z}$. Specifically, there is a latent dynamics operator $P : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z})$ and an emission operator $q : \mathcal{Z} \to \Delta(\mathcal{S})$ and a trajectory is $(z_0, s_0, a_0, r_0, z_1, s_1, a_1, r_1, \ldots)$ where the latent states $z_{0:H-1}$ are unobserved. In this setting we may want to use non-linear function approximation but we can use the "bottleneck structure" of the dynamics to argue that we cannot find too many constraints. Specifically, for a Q-function $\vec{f}$ and a policy π we can write

$$\mathcal{E}_h(\pi, \vec{f}) = \mathbb{E}_{\substack{s_h \sim \pi, \\ a_h \sim \pi_f}} [f_h(s_h, a_h) - (\mathcal{T} f_{h+1})(s_h, a_h)] = \sum_z \Pr[z_h = z \mid \pi] \cdot \mathbb{E}_{\substack{s_h \sim q(z_h), \\ a_h \sim \pi_f}} [f_h(s_h, a_h) - (\mathcal{T} f_{h+1})(s_h, a_h)].$$

(It is important that $a_h \sim \pi_f$ here, since otherwise the bellman error may not decouple across latent states.) With this derivation we take $w_h(\pi) \in \mathbb{R}^{|\mathcal{Z}|}$ to have entries $[w_h(\pi)]_z = \Pr[z_h = z \mid \pi]$ and we take $[v_h(\vec{f})]_z$ to be the average Bellman error for $\vec{f}$ from latent state z (the second term above). The calculation shows that all roll-in distributions can be concisely described in terms of the latent states. Intuitively, the algorithm measures coverage over the latent states rather than the observations, which allows us to explore efficiently. This verifies that the Bellman rank of the block MDP is at most $|\mathcal{Z}|$ and allows for efficient algorithm as long as $\mathcal{F}$ satisfies realizability.

Other examples. All of the models we have derived upper bounds for so far have small Bellman rank: tabular MDPs have rank $|\mathcal{S}|$, linear MDPs have rank d, linear Bellman complete has rank d (but for the latter two the above algorithm, as stated pays for the number of actions, which we saw is unnecessary). In the homework, you will also show that the linear MDP has rank d even if the features are not known in advance, a setting that that may be a nice theoretical framework for studying representation learning in RL. Beyond this, many other examples have been documented in the literature, but the Block MDP may be the best illustration of how nonlinear function approximation may be possible.

History and generalizations. The Bellman rank as we have defined it was originally developed by Jiang et al. (2017) who provided a different algorithm, called OLIVE, to demonstrate tractability. The intuition for the algorithm is the same, but the proof uses a different potential function argument that is based on the concept of "deep cuts" in the convex programming literature. While they documented many models with low Bellman rank (according to the definition above), the subsequent years saw the development of several models that do not exactly fit into the definition above. However we found that the definition, algorithm, and analysis can be modified slightly to accommodate essentially all of these models.

All of these modifications/generalizations are captured in the paper of Du et al. (2021), who propose the "Bilinear class" framework and develop essentially the algorithm we presented here. One key generalization is the introduction of a "loss function" which upper bounds the on-policy Bellman error $\mathcal{E}(\pi_f, \vec{f})$ and factorizes as in the Bellman rank definition. This loss function takes the place of the average Bellman error that we estimate. There is also a technique for removing the dependence on the number of actions in the linear MDP, based on using a different "one-step" policy in lieu of randomizing uniformly. With these generalizations, the framework captures almost all models that are known to be statistically tractable. One notable exception is the deterministic setting with linear $Q^\star$ that we saw in the last lecture.