IE 3186: Approximate Dynamic Programming **Fall 2018**

Lecture 7: VFA, Fitted V.I., and the LP Approach

Lecturer: Daniel Jiang *Scribes: Shaoning Han, Mingyuan Xu*

References:

D. P. Bertsekas, J. N. Tsitsiklis *Neuro-dynamic programming*, Athena Scientific, Belmont MA, 1996. (§6.5)

D. P. De Farias, B. Van Roy. *The linear programming approach to approximate dynamic programming.* Operations Research, 51(6), pp. 850-865, 2003.

D. P. De Farias, B. Van Roy. *On constraint sampling in the linear programming approach to approximate dynamic programming.* Mathematics of Operations Research 29(3), pp. 462-478, 2004.

7.1 Approximating the Value Function

We start with a series of simple theorems related to quantifying errors in a value function approximation (VFA) setting.

Proposition 7.1 (ϵ–V.I.). *Consider the approximation V.I. algorithm with*

$$\|J_{k+1} - TJ_k\|_\infty \leq \epsilon, \quad \forall\, k.$$

Then:

$$J^* - \frac{\epsilon}{1-\gamma}e \leq \liminf_k J_k \leq \limsup_k J_k \leq J^* + \frac{\epsilon}{1-\gamma}e,$$

where e is a vector with elements all ones.

Proof. We know

$$-\epsilon e \leq J_1 - TJ_0 \leq \epsilon e$$

Apply T^{k-1}:

$$-\gamma^{k-1}\epsilon e \leq T^{k-1}J_1 - T^k J_0 \leq \gamma^{k-1}\epsilon e.$$

Generally, we can get

$$-\gamma^{k-i}\epsilon \leq T^{k-i}J_i - T_{k-i+1}J_{i-1} \leq \gamma^{k-i}\epsilon, \;\; 1 \leq i \leq k.$$

Sum such inequalities up, we get

$$-\sum_{i=1}^{k}\gamma^{k-i}\epsilon \le \sum_{i=1}^{k}(T^{k-i}J_i - T^{k-i+1}J_{i-1}) \le \sum_{i=1}^{k}\gamma^{k-i}\epsilon$$

i.e.

$$-\epsilon\sum_{i=0}^{k-1}\gamma^i \le J_k - T^k J_0 \le \epsilon\sum_{i=0}^{k-1}\gamma^i$$

Take limit to conclude. □

If the value function approximate well, the limiting value function is close to optimal value. But what about policies?

Proposition 7.2 (Error in VFA → Performance)**.** *Let* $\|J^* - J\|_\infty = \epsilon$ *and let* μ *be policy greedy with respect to* J*:*

$$\mu(i) = \arg\min_u[g(i,u) + \gamma\mathbf{E}[J(f(i,u,w))]].$$

Then

$$\|J^\mu - J^*\|_\infty \le \frac{2\gamma\epsilon}{1-\gamma}$$

Proof.

$$\begin{aligned}
\|J^\mu - J^*\|_\infty &= \|T_\mu J_\mu - J^*\|_\infty \\
&\le \|T_\mu J^\mu - T_\mu J + T_\mu J - J^*\|_\infty \\
&\le \|T_\mu J^\mu - T_\mu J\|_\infty + \|T_\mu - J^*\|_\infty \\
&\le \gamma\|J^\mu - J\|_infty + \|TJ - TJ^*\|_\infty \\
&\le \gamma\|J^\mu - J\|_\infty + \gamma\|J - J^*\|_\infty \\
&\le \gamma\|J^\mu - J^* + J^* - J\|_\infty + \gamma\epsilon
\end{aligned}$$

It follows that $(1-\gamma)\|J^\mu - J^*\|_\infty \le 2\epsilon\gamma$. □

Proposition 7.3. *When* J^* *is approximated closely enough, the greedy policy of the VFA becomes optimal.*

Proof. There are a finite number of polices. Let $\bar\mu \ne \mu^*$ be the policy for which $J^{\bar\mu}$ is closest to J^* in $\|\cdot\|_\infty$. Suppose $\|J^{\bar\mu} - J^*\| = \delta$. Then if ϵ is small such that $\frac{2\gamma\epsilon}{1-\gamma} < \delta$, μ must be optimal. □

Corollary 7.4. *Let* μ_k *be the policy greedy to* J_k*. For* k *sufficiently large in* ϵ*–AVI, we have*

$$\|J^{\mu_k} - J^*\|_\infty \le \frac{2\gamma}{1-\gamma}\left(\frac{\epsilon}{1-\gamma}\right) = \frac{2\gamma\epsilon}{(1-\gamma)^2}.$$

In the above, we needed to know $\epsilon = \|J - J^*\|$. Given some J, how can we tell if it is good when we don't have access to J^*? We can compute the Bellman error.

Proposition 7.5. *Suppose Bellman error is bounded by ϵ in $\|\cdot\|_\infty$, i.e., $\|J - TJ\|_\infty \leq \epsilon$. Then, it holds that*

$$\|J - J^*\|_\infty \leq \frac{\epsilon}{1-\gamma}.$$

Proof. Note that

$$\begin{aligned}\|J - J^*\|_\infty &\leq \|J - TJ + TJ - J^*\|_\infty \\ &\leq \|T - TJ\|_\infty + \|TJ - TJ^*\|_\infty \\ &\leq \epsilon + \gamma\|J - J^*\|_\infty.\end{aligned}$$

It follows that $(1-\gamma)\|J - J^*\|_\infty \leq \epsilon$. □

7.2 Fitted Value Iteration

How do we approximate the value function, either J^* or J^μ, in practice?

- A parametric class:

  $$\tilde{J}(i; r) \approx J^*(i) \text{ or } J^\mu(i),$$

  where $r = (r_1, \ldots, r_m) \in \mathbb{R}^m$ where m should be much smaller than n. Then we have a possibly tractable problem.

Linear architecture:

$$\tilde{J}(\cdot; r) = \Phi \cdot r = \begin{bmatrix} \Phi_1(1) & \Phi_2(1) & \cdots & \Phi_k(1) & \cdots & \Phi_m(1) \\ \Phi_1(2) & \Phi_2(2) & \cdots & \Phi_k(2) & \cdots & \Phi_m(2) \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \Phi_1(n) & \Phi_2(n) & \cdots & \Phi_k(n) & \cdots & \Phi_m(n) \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}$$

i.e. $\tilde{J}(i; r) = \sum_{k=1}^m \Phi_k(i) r_k$, where $\Phi_k(\cdot)$ is basis function k.

Example 7.6 (Polynomials)**.** *Set $i = (i_1, i_2, \ldots, i_\lambda)$. Quadratic basis function*

$$\tilde{J}(i; r) = r_0 + \sum_k i_k r_k + \sum_\ell \sum_k i_\ell i_k r_{\ell k}$$

.

Example 7.7 (Radial Basis)**.** *Radial basis function:*

$$\Phi_k(i) = \exp(-\|i - \mu_k\|_2^2)/\sigma_k$$

and consider a linear combination.

Example 7.8 (Tetris game)**.** *For Tetris game, every square has two states 0 or 1. If the screen has* 10×20 *sqaures, there are* 2^{200} *states. Successful features were of the form:*

$$\Phi(screen) = \begin{bmatrix} \textit{heights of columns(10)} \\ \textit{height differences(9)} \\ \textit{max height} \\ \textit{number of holes} \\ 1 \end{bmatrix}$$

What is fitted V.I.?

First, select a small sample of states. Let $\tilde{J}_k(\cdot) = \tilde{J}(\cdot; r_k)$ (Here r_k is an m–dimensional iterative vector rather than k^{th} component of vector r as before) be the approximation at iteration k. For each state $i \in$ sample, compute $(T\tilde{J}_k)(\cdot)$.

Next, choose r_{k+1} to "fit" the function $\tilde{J}_{k+1}$ using the "observed" values $(T\tilde{J}_k)$ at sampled states. Fitted V.I. for evaluation: replace T with T_μ.

Example 7.9 (Error Amplification)**.** *Consider the following MDP.*

- *2 states with* $i \in \{1, 2\}$.
- *Just one action with cost 0 and transitions are deterministic: transition probability matrix*
$$P = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$
Notice 2 is an absorbing state.
- $J^*(1) = J^*(2) = 0$.
- *A single basis function* $\Phi_1(i) = i$*. So* $\Phi = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ *and* $\Phi r = \begin{bmatrix} r \\ 2r \end{bmatrix}$.

Exact fitted V.I. (compute T *at all states) using least squares fit:*

$$\begin{aligned} r_{k+1} &= \arg\min_r \left[\sum_{i=1}^{2} (\tilde{J}(i; r) - (T\tilde{J}_k)(i))^2 \right] \\ &= \arg\min_r \left[(r - \underbrace{(T\tilde{J}_k)(1)}_{0+\gamma\cdot 2r_k})^2 + (2r - \underbrace{(T\tilde{J}_k)(2)}_{0+\gamma\cdot 2r_k})^2 \right] \\ &= \arg\min_r \left[(r - \gamma \cdot 2r_k)^2 + (2r - \gamma \cdot 2r_k)^2 \right] \end{aligned}$$

By taking derivative, we can get $r_{k+1} = \frac{6}{5}\gamma r_k$*. It diverges for* $\gamma > \frac{5}{6}$*! One way to explain this is that state 2 is much more important than state 1 and we need to weight state 2 much more, but by using least squares fit, we don't take it into account.*

Notation: let Π be a projection operator onto linear space $S = \{\Phi r : r \in \mathbb{R}^m\}$ with respect to $\|\cdot\|_2$. The "fit" step can be written as $\Pi J = \Phi r^*$ where $r^* = \arg\min_r \|\Phi r - J\|_2^2$. Thus fitted V.I. can be done by:

$$\tilde{J}_k \xrightarrow{\text{Bellman V.I.}} T_\mu \tilde{J}_k \xrightarrow{\text{fit w.r.t. some norm}} \tilde{J}_{k+1} = \Pi T_\mu \tilde{J}_k$$

The problem is that thew operator ΠT_μ may be not a contraction.

Let's focus on weighted projection. Define a weighted Euclidean norm (think of this as weighing states)

$$\|v\|_\xi = \sqrt{\sum_i \xi_i (v(i))^2},$$

where $\xi = (\xi_1, \ldots, \xi_n)$ is a distribution.

Example 7.10. *Let's revisit the divergent example with weighted projection $\|\cdot\|_\xi$ with $\xi = (\xi_1, \xi_2)$.*

$$r_{k+1} = \arg\min_r \left[\xi_1 (r - \gamma \cdot 2r_k)^2 + \xi_2 (2r - \gamma \cdot 2r_k)^2\right].$$

By taking the derivative, it's easy to get

$$r_{k+1} = \left(\frac{\xi_1}{\xi_1 + 4\xi_2} + 1\right) \gamma\, r_k.$$

We can see that if ξ_2 is large enough, then r_k converges. Notice that state 2 is occupied by system most of the time, so this makes intuitive sense.

Proposition 7.11 (Projections are nonexpensive)**.** *Using $\|\cdot\|$,*

$$\|\Pi J - \Pi J'\|_\xi \le \|J - J'\|_\xi$$

Proof.

$$\|\Pi J - \Phi r\|_\xi^2 + \|J - \Pi J\|_x i^2 = \|J - \Phi r\|_\xi^2, \quad \forall\ \Phi r \in S.$$

It follows that

$$\|\Pi(J - J')\|_\xi^2 \le \|\Pi(J - J')\|^2 + \|(I - \Pi)(J - J')\|_\xi^2 = \|J - J'\|_\xi,$$

which concludes the proof. □

Now, in $\tilde{J}_{k+1} = \Pi J_\mu \tilde{J}_k$, operator Π is nonexpansive with respect to $\|\cdot\|_\xi$ and operator J_μ is a contraction with respect to $\|\cdot\|_\infty$. If these two operators are with respect to the same norm, ΠJ_μ would be a good operator. Unfortunately, we face the "norm mismatch" problem. To be continued.

7.3 Paper Discussion (Mingyuan)

7.3.1 Exact LP Reformulation

Consider the following linear programming problem:

$$\begin{aligned} \max_{J(x)} \quad & c^T J = \sum_{x \in S} c(x) J(x) \\ \text{s.t.} \quad & J(x) \leq TJ(x) \\ & \forall\, x \in S \end{aligned} \tag{7.1}$$

- Linearity:

  $J(x) \leq TJ(x) \Leftrightarrow (T_\mu J)(x) = \mathbf{E}\left[g(x, \mu, w) + \gamma J\left(f(x, \mu, w)\right] \geq J(x), \forall\, \mu \in \mathcal{U}(x)\right.$

- Dimensionality: $|S|$ variables, $|S| \times |A|$ constraints
- Feasibility and Optimility:
  - $J^* = TJ^* \leq TJ^*$
  - $J \leq TJ \Rightarrow J \leq TJ \leq T^2 J \leq \cdots \leq J^*$
- State-relevance weights: $(c(x) \geq 0, \forall\, x \in S)$. Note that the choice of state-relevance weights does not influence the solution of (7.1).

7.3.2 Approximate/Reduced LP Approach

7.3.2.1 Parameterization

Given pre-selected K basis functions $\phi_k(x)$ ($\phi_k : S \rightarrow \mathbb{R}^1, K \ll |S|$), define a matrix:

$$\Phi_{|S|*K} = [\phi_1, \phi_2, \cdots, \phi_K]_{|S| \times K} \tag{7.2}$$

The aim is to generate a weight vector $\tilde{r} \in \mathbb{R}^K$:

$$\tilde{J}(x) \approx \Phi \tilde{r}(x) = \sum_{k=1}^{K} \phi_k(x) \tilde{r}_k \tag{7.3}$$

Then we have the following linear programming problem:

$$\begin{aligned} \max_{r} \quad & c^T \Phi r = \sum_{x \in S} c(x) \sum_{k=1}^{K} \phi_k(x) r_k \\ \text{s.t.} \quad & \Phi r \leq T \Phi r \\ & \forall\, x \in S \end{aligned} \tag{7.4}$$

- Dimensionality: K variables, $|S| \times |A|$ constraints
- Φ^* : the optimal cost-to-go function lies within the span of the basis functions. In practice, basis functions should be chosen based on heuristics and perhaps some simplified analysis of the problem.
- Feasibility: depends on Φ
- State-relevance weights:
  - Consider c to be a probability distribution $\sum_{x \in S} c(x) = 1$. Then the objective can be viewed as an expected value where x is sampled according to the distribution c.
  - (7.4) is equivalent to the programming with weighted norm based on c:

$$\begin{aligned} \min_r \quad & ||J^* - \Phi r||_{1,c} = \sum_{x \in S} c(x) \; ||J^*(x) - \Phi r(x)||_1 \\ \text{s.t.} \quad & \Phi r(x) \leq T\Phi r(x) \\ & \forall \, x \in S \end{aligned} \tag{7.5}$$

- Error Bound:

$$||J^* - \Phi \tilde{r}||_{1,c} \leq \frac{2}{1-\gamma} \min_r ||J^* - \Phi r||_\infty \tag{7.6}$$