RL with Linear Features: When Does It Work & When Doesn't It Work?

Part 1: The Assumption Ladder & Bellman Completeness

CS 2284: Foundations of Reinforcement Learning

Kianté Brantley & Sham Kakade

Agenda

Announcements

- Second reading assignment is out.

Recap++

- Wrap up our tabular sample complexity analysis (Chapter 2).
- **Minimax Result:** We established the fundamental limits of tabular learning.

Today

- Function Approximation. **We'll move beyond tabular RL!**

Motivation: Beyond Tabular RL

Recap: Tabular MDPs

- State space $\mathcal{S}$, Action space $\mathcal{A}$.
- Sample complexity scales with $|\mathcal{S}||\mathcal{A}|$.
- **Problem:** In many real-world applications (robotics, games, healthcare), $|\mathcal{S}|$ is enormous or continuous.

Function Approximation

- Introduce a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.
- Approximate values using linear functions:

$$f_\theta(s, a) = \theta^\top \phi(s, a)$$

- **Goal:** Sample complexity polynomial in d (and H), independent of $|\mathcal{S}|$.

Finite-horizon dynamic programming (reminder)

$$Q^*_h(s,a) = r_h(s,a) + \mathbb{E}_{s' \sim P^h_{sa}} \Big[\underbrace{\max_{a'} Q^*_{h+1}(s',a')}_{V^*_{h+1}(s')} \Big]$$

Stage-h **optimal Bellman operator**:

$$(\mathcal{T}_h f)(s,a) \;:=\; r_h(s,a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \Big[\max_{a' \in \mathcal{A}} f(s',a') \Big].$$

Optimal Q satisfies backward recursion:

$$Q^\star_H \equiv 0, \qquad Q^\star_h = \mathcal{T}_h Q^\star_{h+1} \quad \text{for } h = H-1, \ldots, 0.$$

Least-Squares Value Iteration (LSVI)

Setting: Finite Horizon H, Generative Model (Simulator).

Offline setting

Algorithm: Backward Induction via Regression.

1. Initialize $\widehat{V}_H(s) = 0$.
2. For $h = H - 1, \ldots, 0$:
   - **Collect Data:** Generate dataset $D_h = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$.

Least-Squares Value Iteration (LSVI)

Setting: Finite Horizon H, Generative Model (Simulator).

Algorithm: Backward Induction via Regression.

1. Initialize $\widehat{V}_H(s) = 0$.
2. For $h = H-1, \ldots, 0$:
   - **Collect Data:** Generate dataset $D_h = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$.
   - **Form Targets:** Compute regression targets using the *next* value function:

$$y_i = r_h(s_i, a_i) + \widehat{V}_{h+1}(s_i')$$

Least-Squares Value Iteration (LSVI)

Setting: Finite Horizon H, Generative Model (Simulator).

Algorithm: Backward Induction via Regression.

1. Initialize $\widehat{V}_H(s) = 0$.
2. For $h = H-1, \ldots, 0$:
   - **Collect Data:** Generate dataset $D_h = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$.
   - **Form Targets:** Compute regression targets using the *next* value function:

$$y_i = r_h(s_i, a_i) + \widehat{V}_{h+1}(s_i')$$

   - **Regression:** Solve for parameters $\widehat{\theta}_h$:

$$E[\theta^* \cdot \phi \mid s, a] \approx Q^*_h(s, a)$$

$$\widehat{\theta}_h \in \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{N} \left(\theta^\top \phi(s_i, a_i) - y_i\right)^2$$

Least-Squares Value Iteration (LSVI)

Setting: Finite Horizon H, Generative Model (Simulator).

Algorithm: Backward Induction via Regression.

1. Initialize $\widehat{V}_H(s) = 0$.
2. For $h = H-1, \ldots, 0$:
   - **Collect Data:** Generate dataset $D_h = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$.
   - **Form Targets:** Compute regression targets using the *next* value function:

$$y_i = r_h(s_i, a_i) + \widehat{V}_{h+1}(s_i')$$

$$\max_{a'} \widehat{\theta}_{h+1} \cdot \phi(s', a')$$

   - **Regression:** Solve for parameters $\widehat{\theta}_h$:

$$\widehat{\theta}_h \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \sum_{i=1}^{N} \left(\theta^\top \phi(s_i, a_i) - y_i\right)^2$$

   - **Update:** Set $\widehat{Q}_h(s, a) = \widehat{\theta}_h^\top \phi(s, a)$ and $\widehat{V}_h(s) = \max_a \widehat{Q}_h(s, a)$.

The Central Question & The Intuition Trap

$\text{poly}(d, H)$

When do linear features actually buy you $\text{poly}(d)$ *sample complexity —*
and when do they fundamentally not?

The Central Question & The Intuition Trap

When do linear features actually buy you $\mathrm{poly}(d)$ *sample complexity — and when do they fundamentally not?*

$y = w \cdot x + \varepsilon$

The Intuition Trap:

- Standard Supervised Learning: "If the target function is linear, we can learn it with $O(d)$ samples."

The Central Question & The Intuition Trap

When do linear features actually buy you $\mathrm{poly}(d)$ *sample complexity — and when do they fundamentally not?*

The Intuition Trap:

- Standard Supervised Learning: "If the target function is linear, we can learn it with $O(d)$ samples."
- **But RL is different:** LSVI is a **composition** of regressions.
  - The target for stage h depends on our *own estimate* at $h+1$:

$$\mathrm{Target}_h(s, a) \approx r(s, a) + \mathbb{E}\Big[\max_{a'} \widehat{Q}_{h+1}(s', a')\Big]$$

The Central Question & The Intuition Trap

When do linear features actually buy you $\mathrm{poly}(d)$ *sample complexity — and when do they fundamentally not?*

The Intuition Trap:

- Standard Supervised Learning: "If the target function is linear, we can learn it with $O(d)$ samples."
- **But RL is different:** LSVI is a **composition** of regressions.
  - The target for stage h depends on our *own estimate* at $h+1$:

$$\text{Target}_h(s,a) \approx r(s,a) + \mathbb{E}\left[\max_{a'} \widehat{Q}_{h+1}(s',a')\right]$$

  - **Crucial Question:** Even if $Q^\star$ is linear, does the target defined by $\widehat{Q}_{h+1}$ remain learnable (linear)?
- If the target "falls off the manifold", how do errors compound?

The Assumption Ladder

We can organize linear RL assumptions from weakest (hardest) to strongest (easiest).

The Assumption Ladder

We can organize linear RL assumptions from weakest (hardest) to strongest (easiest).

(A) **Agnostic Approximation**
No realizability. $Q^\star$ is "close" to linear.
Status: Hard. Requires strong distribution assumptions.

The Assumption Ladder

We can organize linear RL assumptions from weakest (hardest) to strongest (easiest).

(A) **Agnostic Approximation**
No realizability. $Q^\star$ is "close" to linear.
Status: Hard. Requires strong distribution assumptions.

(B) **Linear $Q^\star$ Realizability**
$Q_h^\star(s, a) = (\theta_h^\star)^\top \phi(s, a)$.
Status: **Insufficient.** *Exponential lower bounds exist.*

The Assumption Ladder

We can organize linear RL assumptions from weakest (hardest) to strongest (easiest).

(A) **Agnostic Approximation**
No realizability. $Q^\star$ is "close" to linear.
Status: Hard. Requires strong distribution assumptions.

(B) **Linear $Q^\star$ Realizability**
$Q_h^\star(s, a) = (\theta_h^\star)^\top \phi(s, a)$.
Status: **Insufficient.** *Exponential lower bounds exist.*

(C) **All-Policies Realizability**
$Q_h^\pi(s, a) = (\theta_h^\pi)^\top \phi(s, a)$ for **all** π.
Status: **Subtle.** *Fails offline even with perfect coverage.*

The Assumption Ladder

We can organize linear RL assumptions from weakest (hardest) to strongest (easiest).

(A) **Agnostic Approximation**
No realizability. $Q^\star$ is "close" to linear.
Status: Hard. Requires strong distribution assumptions.

(B) **Linear $Q^\star$ Realizability**
$Q_h^\star(s, a) = (\theta_h^\star)^\top \phi(s, a)$.
Status: **Insufficient.** *Exponential lower bounds exist.*

$\mathcal{F} = \{ w \cdot \phi \mid w \in \mathbb{R}^d \}$

(C) **All-Policies Realizability**
$Q_h^\pi(s, a) = (\theta_h^\pi)^\top \phi(s, a)$ for **all** π.
Status: **Subtle.** *Fails offline even with perfect coverage.*

(D) **Linear Bellman Completeness**
$\mathcal{T}_h f \in \mathcal{F}$ for all $f \in \mathcal{F}$.
Status: **Sufficient!** *(This Lecture)*

Assumption (D): Linear Bellman Completeness

Definition (Bellman Completeness)
For any linear function $f(s, a) = w^\top \phi(s, a)$, applying the Bellman optimality operator $\mathcal{T}_h$ yields a function that is also linear in ϕ.

$$\forall w \in \mathbb{R}^d, \exists \theta \in \mathbb{R}^d \text{ such that}$$

$$\underbrace{r_h(s, a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[\max_{a'} w^\top \phi(s', a') \right]}_{(\mathcal{T}_h f_w)(s,a)} = \theta^\top \phi(s, a)$$

Key Implication:

- If we run LSVI with finite samples, the **target** is always realizable.
- This reduces RL to a sequence of well-specified regression problems.

Examples of Completeness

When does Linear Bellman Completeness hold?

1. **Tabular MDPs**
   $\phi(s, a)$ is a one-hot encoding of size $|\mathcal{S}||\mathcal{A}|$.
   Any function over $\mathcal{S} \times \mathcal{A}$ is linear in ϕ.

Examples of Completeness

When does Linear Bellman Completeness hold?

1. **Tabular MDPs**
   $\phi(s, a)$ is a one-hot encoding of size $|\mathcal{S}||\mathcal{A}|$.
   Any function over $\mathcal{S} \times \mathcal{A}$ is linear in ϕ.
2. **Linear MDPs (Low-Rank Transition)**

$$P_h(s'|s, a) = \sum_{i=1}^{d} \phi_i(s, a)\mu_i(s')$$

   Here, the transition dynamics themselves are linear in features.

Examples of Completeness

When does Linear Bellman Completeness hold?

1. **Tabular MDPs**
   $\phi(s,a)$ is a one-hot encoding of size $|\mathcal{S}||\mathcal{A}|$.
   Any function over $\mathcal{S} \times \mathcal{A}$ is linear in ϕ.

2. **Linear MDPs (Low-Rank Transition)**

$$P_h(s'|s,a) = \sum_{i=1}^{d} \phi_i(s,a)\mu_i(s')$$

   Here, the transition dynamics themselves are linear in features.

3. **Linear Quadratic Regulators (LQR)?**
   Yes, with quadratic features: $\phi(s,a) = [s^\top, a^\top, s^\top s, \dots]^\top$.
   (Value functions are quadratic in s, a).

Warning: Adding "junk" features can break completeness!

The Error Propagation Question (Intuition)

Suppose we run approximate dynamic programming (like LSVI) where we force every estimate $\widehat{Q}_h$ to be a linear function:

$\widehat{Q}_{H-1} \approx V_{H-1}$

$$\widehat{Q}_H \equiv 0, \qquad \widehat{Q}_h \leftarrow \text{Project}_{\mathcal{F}}\left(\mathcal{T}_h \widehat{Q}_{h+1}\right).$$

The Subtle Danger:

- Even if $Q^\star$ is linear, the **target** $\mathcal{T}_h \widehat{Q}_{h+1}$ might *not* be linear!
- If the target falls "off-manifold" (outside $\mathcal{F}$), we incur a projection error (bias) at step h.
- **The Crucial Question:** How do these errors compound?
  - Does the error at $h+1$ get amplified when we back up to h?
  - Without **Completeness** (closure), this error can grow exponentially with H.

Realizability of $Q^\star$ alone does not guarantee the intermediate targets remain learnable.

Refresher: LSVI Regression Step

To analyze the algorithm, let's focus on exactly what happens at stage h.

We have a fixed next-stage value $\widehat{V}_{h+1}$. We gather data D_h and solve:

$$\widehat{\theta}_h = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \sum_{i=1}^{N} \left(\underbrace{\theta^\top \phi(s_i, a_i)}_{\text{Prediction}} - \underbrace{(r_i + \widehat{V}_{h+1}(s_i'))}_{\text{Target } y_i} \right)^2$$

Why does this work?

- **Completeness** implies the *true expected target* is linear:

$$\mathbb{E}[y_i \mid s_i, a_i] = (\mathcal{T}_h \widehat{Q}_{h+1})(s_i, a_i) = \theta_h^\star \phi(s_i, a_i)$$

- So this is **well-specified** linear regression!
- Bias is zero; we only need to control variance.

Fixed Design OLS (The Tool)

Consider the standard linear regression setting:

$$y_i = x_i^\top \theta^\star + \xi_i, \quad \text{with } \mathbb{E}[\xi_i | x_i] = 0 \text{ (sub-Gaussian).}$$

The OLS estimator is $\widehat{\theta} = \Lambda^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} x_i y_i \right)$, where $\Lambda = \frac{1}{N} \sum_i x_i x_i^\top$.

Fixed Design OLS (The Tool)

Consider the standard linear regression setting:

new x

$$y_i = x_i^\top \theta^\star + \xi_i, \quad \text{with } \mathbb{E}[\xi_i | x_i] = 0 \text{ (sub-Gaussian).}$$

$E[y|x] = \theta^\star \cdot x$? $\hat{\theta} \cdot x$

The OLS estimator is $\widehat{\theta} = \Lambda^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} x_i y_i\right)$, where $\Lambda = \frac{1}{N}\sum_i x_i x_i^\top$.

Fixed-design OLS Bound

With probability at least $1-\delta$, the prediction error is bounded in the Λ-norm:

$$\|\widehat{\theta} - \theta^\star\|_\Lambda \lesssim \sigma\sqrt{\frac{d\log(1/\delta)}{N}}.$$

Fixed Design OLS (The Tool)

Consider the standard linear regression setting:

$$y_i = x_i^\top \theta^\star + \xi_i, \quad \text{with } \mathbb{E}[\xi_i | x_i] = 0 \text{ (sub-Gaussian)}.$$

The OLS estimator is $\widehat{\theta} = \Lambda^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} x_i y_i\right)$, where $\Lambda = \frac{1}{N}\sum_i x_i x_i^\top$.

Fixed-design OLS Bound

With probability at least $1-\delta$, the prediction error is bounded in the Λ-norm:

$$\|\widehat{\theta} - \theta^\star\|_\Lambda \lesssim \sigma\sqrt{\frac{d\log(1/\delta)}{N}}.$$

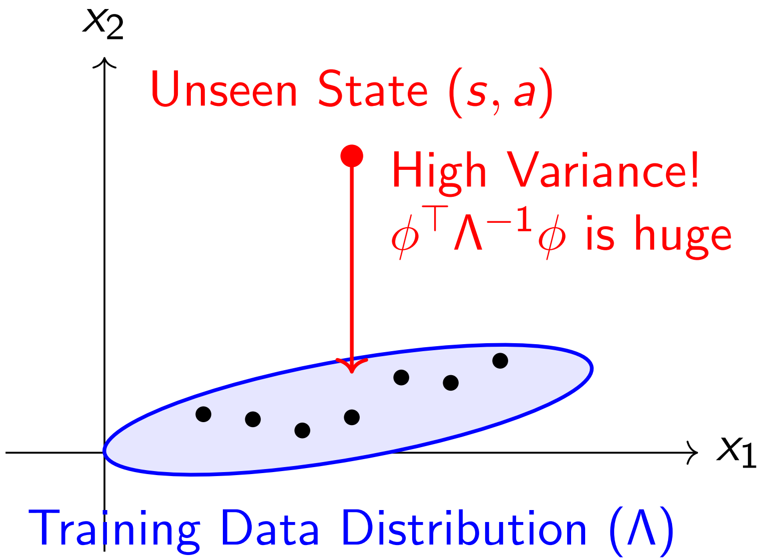
This translates to a *pointwise* bound using leverage scores:

$$|(\widehat{\theta} - \theta^\star)^\top \phi(s,a)| \leq \|\widehat{\theta} - \theta^\star\|_\Lambda \sqrt{\phi(s,a)^\top \Lambda^{-1} \phi(s,a)}$$

The Hidden Failure Mode of OLS

The Problem:

- The OLS bound depends on $\Lambda = \frac{1}{N}\sum \phi(s_i, a_i)\phi(s_i, a_i)^\top$.
- It bounds the **average** prediction error (weighted by training data).
- RL requires **Uniform (ℓ_∞)** error bounds. We must predict well at *any* state the optimal policy might visit.



Backward induction may query *outside* the ellipse, causing huge expansion.

D-optimal design: the leverage-minimizing geometry

$\mathbb{E}_{\phi\sim\rho}[\phi^\top \Sigma^{-1} \phi]$

To guarantee uniform bounds, we must choose our training data carefully.

The feature set is:

$$\Phi := \{\phi(s, a) : (s, a) \in \mathcal{S} \times \mathcal{A}\} \subset \mathbb{R}^d.$$

$= \mathrm{Tr}(\mathbb{E}_\rho[\phi\phi^\top]\, \Sigma^{-1})$

$= \mathrm{Tr}(I) = d$

D-optimal design (lemma; geometric fact)

Suppose Φ is compact. There exists a distribution ρ supported on at most $d(d + 1)/2$ state–action pairs s.t. with

$$\Sigma := \mathbb{E}_{(s,a)\sim\rho}\left[\phi(s, a)\phi(s, a)^\top\right],$$

we have $\Sigma \succ 0$ and

$$\sup_{(s,a)} \phi(s, a)^\top \Sigma^{-1} \phi(s, a) \overset{=}{\cancel{\le}} d.$$

Geometric intuition

Leverage control: the quantity $\phi^\top \Sigma^{-1} \phi$ is exactly the (population) leverage

Σ

Equivalent viewpoints (pick your favorite story):

$(1-\alpha)\Sigma + \alpha \phi\phi^\top$

- **Kiefer–Wolfowitz:** ρ maximizes $\log\det\left(\mathbb{E}_\rho[\phi\phi^\top]\right)$
- **John's ellipsoid:** the ellipsoid

$$\mathcal{E} = \{v : \; v^\top \Sigma^{-1} v \le d\}$$

is the minimum-volume centered ellipsoid containing Φ

Message: there is always a way to sample from only $O(d^2)$ points while keeping worst-case leverage $\le d$.

From Global to Pointwise Error

Sample N points from the D-optimal design ρ. Then $\Lambda = \frac{1}{N}\sum \phi\phi^\top$, and our empirical cov is $\Lambda \approx \Sigma$.

From Global to Pointwise Error

Sample N points from the D-optimal design ρ. Then $\Lambda = \frac{1}{N}\sum \phi\phi^\top$, and our empirical cov is $\Lambda \approx \Sigma$.

1. **The Leverage Score Bound (Geometry):** Since $\Lambda \approx \Sigma$, D-optimal design guarantees:

$$\sup_{\phi\in\Phi} \phi^\top \Lambda^{-1}\phi \;\approx\; \sup_{\phi\in\Phi} \phi^\top \Sigma^{-1}\phi \;\leq\; d$$

From Global to Pointwise Error

Sample N points from the D-optimal design ρ. Then $\Lambda = \frac{1}{N}\sum \phi\phi^\top$, and our empirical cov is $\Lambda \approx \Sigma$.

1. **The Leverage Score Bound (Geometry):** Since $\Lambda \approx \Sigma$, D-optimal design guarantees:

$$\sup_{\phi\in\Phi} \phi^\top \Lambda^{-1}\phi \;\approx\; \sup_{\phi\in\Phi} \phi^\top \Sigma^{-1}\phi \;\leq\; d$$

2. **The OLS Bound (Statistics):** From the first slide, we know $\|\widehat{\theta} - \theta^\star\|_\Lambda \lesssim \sigma\sqrt{\frac{d}{N}}$.

From Global to Pointwise Error

Sample N points from the D-optimal design ρ. Then $\Lambda = \frac{1}{N}\sum \phi\phi^\top$, and our empirical cov is $\Lambda \approx \Sigma$.

1. **The Leverage Score Bound (Geometry):** Since $\Lambda \approx \Sigma$, D-optimal design guarantees:

$$\sup_{\phi\in\Phi} \phi^\top \Lambda^{-1}\phi \;\approx\; \sup_{\phi\in\Phi} \phi^\top \Sigma^{-1}\phi \;\leq\; d$$

2. **The OLS Bound (Statistics):** From the first slide, we know $\|\widehat{\theta} - \theta^\star\|_\Lambda \lesssim \sigma\sqrt{\frac{d}{N}}$.

3. **Resulting Pointwise Guarantee:** Use $|\text{pointwise error}| \leq \|\widehat{\theta} - \theta^\star \cdot \|_\Lambda \sqrt{\text{Leverage}}$:

$$\sup_{(s,a)} |\widehat{Q}_h(s,a) - \mathcal{T}_h \widehat{Q}_{h+1}(s,a)| \;\lesssim\; \left(\sigma\sqrt{\frac{d}{N}}\right) \cdot \sqrt{d} \;=\; \frac{\sigma d}{\sqrt{N}}$$

This allows us to control the *max-norm* Bellman residual!

Returning to LSVI

We now have all the pieces to analyze LSVI.

1. The Data Collection (Generative Model)

- For each stage h, we don't just sample randomly.
- We compute the D-optimal design $\rho^\star$ on Φ.
- We query the simulator N times distributed according to $\rho^\star$.

2. The Rough Sketch of the Proof

$$\text{Regression Error} \xrightarrow{\text{D-Opt}} \text{Pointwise Error} \xrightarrow{\text{Completeness}}$$
$$\text{Bellman Residual} \xrightarrow{\text{Sim. Lemma}} \text{Policy Loss}$$

The Main Theorem (Informal)

Theorem (LSVI with Generative Model)
Assume Linear Bellman Completeness. If we set:

$$N \approx \frac{H^6 d^2}{\epsilon^2}$$

and collect data using D-optimal design, then LSVI returns a policy $\widehat{\pi}$ such that with high probability:

$$V^\star(s_0) - V^{\widehat{\pi}}(s_0) \le \epsilon$$

The Main Theorem (Informal)

Theorem (LSVI with Generative Model)
Assume Linear Bellman Completeness. If we set:

$$N \approx \frac{H^6 d^2}{\epsilon^2}$$

and collect data using D-optimal design, then LSVI returns a policy $\widehat{\pi}$ such that with high probability:

$$V^\star(s_0) - V^{\widehat{\pi}}(s_0) \leq \epsilon$$

Takeaway: We have achieved sample complexity polynomial in d and H, independent of $|\mathcal{S}|$!

- **Completeness** ensures realizability.
- **D-Optimal Design** ensures uniform error control.

Summary & Looking Ahead

Today: Scaling RL to large state spaces (using features)

- **The Algorithm:** Dynamic Programming as a sequence of regression problems (LSVI)
- **The Assumption Ladder:** consider different natural structural assumptions
- **Sampling:** use **D-Optimal Design** to control the uniform (ℓ_∞) error
- **Main Result: Linear BC** + **D-Optimal Design** is sufficient for $\mathrm{poly}(d, H)$ sample complexity.

Next Time (Lecture 2):

- **Rigorous Analysis:**
- **Offline RL:** adapt LSVI when we cannot choose our sampling distribution (Coverage Assumptions).