Learning with Linear Bellman Completion & Generative Model

Sham Kakade and Wen Sun

CS 6789: Foundations of Reinforcement Learning

Recap: Linear Bellman Completion

Given feature ϕ, take any linear function $w^\top \phi(s, a)$:

$$\forall h, \exists \theta \in \mathbb{R}^d, s.t., \theta^\top \phi(s, a) = r(s, a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} w^\top \phi(s', a'), \forall s, a$$

$T_h(w)$

Recap: Linear Bellman Completion

Given feature ϕ, take any linear function $w^\top \phi(s,a)$:

$$\forall h, \exists \theta \in \mathbb{R}^d, s.t., \theta^\top \phi(s,a) = r(s,a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} w^\top \phi(s', a'), \forall s, a$$

It implies that $Q_h^\star$ is linear in ϕ: $Q_h^\star = (\theta_h^\star)^\top \phi, \forall h$

Recap: Linear Bellman Completion

Given feature ϕ, take any linear function $w^\top \phi(s,a)$:

$$\forall h, \exists \theta \in \mathbb{R}^d, s.t., \theta^\top \phi(s,a) = r(s,a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} w^\top \phi(s',a'), \forall s, a$$

It implies that $Q_h^\star$ is linear in ϕ: $Q_h^\star = (\theta_h^\star)^\top \phi, \forall h$

Captures Tabular MDPs, and Linear Quadratic Regulators

Recap: Linear Bellman Completion

Given feature ϕ, take any linear function $w^\top \phi(s,a)$:

$$\forall h, \exists \theta \in \mathbb{R}^d, s.t., \theta^\top \phi(s,a) = r(s,a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} w^\top \phi(s',a'), \forall s, a$$

It implies that $Q_h^\star$ is linear in ϕ: $Q_h^\star = (\theta_h^\star)^\top \phi, \forall h$

Captures Tabular MDPs, and Linear Quadratic Regulators

But adding additional elements may just break the condition

Recap: Least-Square Value Iteration

Recap: Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Recap: Least-Square Value Iteration

Datasets $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$, w/

$\mathcal{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

Recap: Least-Square Value Iteration

Datasets $\mathcal{D}_0, \ldots, \mathcal{D}_{H-1}$, w/

$\mathcal{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$$\theta_h = \arg\min_\theta \sum_{\mathcal{D}_h} \left(\theta^T \phi(s, a) - \left(r + V_{h+1}(s')\right)\right)^2$$

Recap: Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s,a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$$\theta_h = \arg\min_\theta \sum_{\mathscr{D}_h} \left(\theta^T \phi(s,a) - \left(r + V_{h+1}(s')\right)\right)^2$$

Set $V_h(s) := \max_a \theta_h^\top \phi(s,a), \forall s$

$\approx Q_h^*$

Recap: Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$\approx V^\star_{h+1}(s')$

$$\theta_h = \arg\min_{\theta} \sum_{\mathscr{D}_h} \left(\theta^T \phi(s, a) - \left(r + V_{h+1}(s')\right)\right)^2$$

Set $V_h(s) := \max_a \theta_h^\top \phi(s, a), \forall s$

Return $\hat{\pi}_h(s) = \arg\max_a \theta_h^\top \phi(s, a), \forall h$

Recap: Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$$\theta_h = \arg\min_{\theta} \sum_{\mathscr{D}_h} \left(\theta^T \phi(s, a) - \left(r + V_{h+1}(s') \right) \right)^2$$

Set $V_h(s) := \max_a \theta_h^\top \phi(s, a), \forall s$

Return $\hat{\pi}_h(s) = \arg\max_a \theta_h^\top \phi(s, a), \forall h$

BC always ensures linear regression is realizable:

i.e., our regression target

$$r(s, a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} \theta_{h+1}^\top \phi(s', a')$$

is always linear:

Outline for Today

1. Proof Sketch of LSVI

2. LSVI in Offline RL

Theorem

Theorem: There exists a way to construct datasets $\{\mathcal{D}_h\}_{h=0}^{H-1}$, such that with probability at least $1 - \delta$, we have:

$$V^{\hat{\pi}} - V^\star \leq \epsilon$$

w/ total number of samples in these datasets scaling $\widetilde{O}\left(d^2 + H^6 d^2/\epsilon^2\right)$

Theorem

Theorem: There exists a way to construct datasets $\{\mathscr{D}_h\}_{h=0}^{H-1}$, such that with probability at least $1-\delta$, we have:

$$V^{\hat{\pi}} - V^{\star} \leq \epsilon$$

w/ total number of samples in these datasets scaling $\widetilde{O}\left(d^2 + H^6 d^2/\epsilon^2\right)$

1. How to actively design / construct datasets $\mathscr{D}_h$ via the Generative Model property

Theorem

Theorem: There exists a way to construct datasets $\{\mathcal{D}_h\}_{h=0}^{H-1}$, such that with probability at least $1-\delta$, we have:

$$V^{\hat{\pi}} - V^{\star} \leq \epsilon$$

w/ total number of samples in these datasets scaling $\widetilde{O}\left(d^2 + H^6 d^2/\epsilon^2\right)$

1. How to actively design / construct datasets $\mathcal{D}_h$ via the Generative Model property
2. Show that our estimators are near-bellman consistent: $\|\theta_h^\top \phi - \mathcal{T}_h(\theta_{h+1}^\top \phi)\|_\infty$ is small

Theorem

Theorem: There exists a way to construct datasets $\{\mathscr{D}_h\}_{h=0}^{H-1}$, such that with probability at least $1-\delta$, we have:

$$V^{\hat{\pi}} - V^\star \le \epsilon$$

w/ total number of samples in these datasets scaling $\widetilde{O}\left(d^2 + H^6 d^2/\epsilon^2\right)$

1. How to actively design / construct datasets $\mathscr{D}_h$ via the Generative Model property
2. Show that our estimators are near-bellman consistent: $\|\theta_h^\top \phi - \mathscr{T}_h(\theta_{h+1}^\top \phi)\|_\infty$ is small
3. Near-Bellman consistency implies near optimal performance (s.t. H error amplification)

Detour: Ordinary Linear Squares

Consider a dataset $\{x_i, y_i\}_{i=1}^N$, where $y_i = (\theta^\star)^\top x_i + \epsilon_i$, $\quad \mathbb{E}[\epsilon_i | x_i] = 0$, ϵ_i are independent

with $|\epsilon_i| \leq \sigma$, assume $\Lambda = \sum_{i=1}^N x_i x_i^\top / N$ is full rank;

Detour: Ordinary Linear Squares

Consider a dataset $\{x_i, y_i\}_{i=1}^{N}$, where $y_i = (\theta^\star)^\top x_i + \epsilon_i, \quad \mathbb{E}[\epsilon_i \,|\, x_i] = 0$, ϵ_i are independent

with $|\epsilon_i| \leq \sigma$, assume $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$ is full rank;

$$\text{OLS}: \hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} (\theta^\top x_i - y_i)^2$$

Detour: Ordinary Linear Squares

Consider a dataset $\{x_i, y_i\}_{i=1}^N$, where $y_i = (\theta^\star)^\top x_i + \epsilon_i, \quad \mathbb{E}[\epsilon_i | x_i] = 0$, ϵ_i are independent with $|\epsilon_i| \leq \sigma,$ assume $\Lambda = \sum_{i=1}^N x_i x_i^\top / N$ is full rank;

$$\text{OLS} : \hat{\theta} = \arg\min_\theta \sum_{i=1}^N (\theta^\top x_i - y_i)^2$$

Standard OLS guarantee: with probability at least $1 - \delta$:

$$\underbrace{(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star)}_{\Rightarrow \frac{1}{N} \sum_{i=1}^{n} \left((\hat{\theta} - \theta^\star)^\top x_i \right)^2} \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N} \right)$$

Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N}\right)$$

Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N} \right)$$

If the test point x is not covered by the training data, i.e., $x^\top \Lambda^{-1} x$ is huge, then we cannot guarantee $\hat{\theta}^\top x$ is close to $(\theta^\star)^\top x$

$x^\top \Lambda^{-1} x$

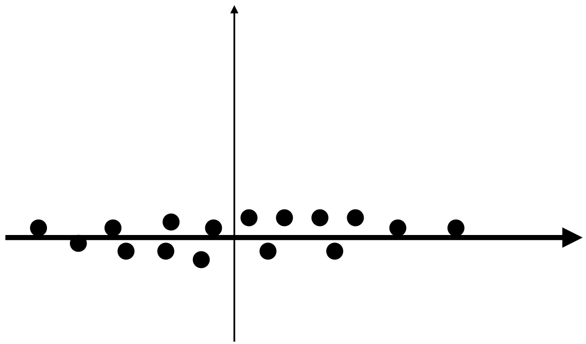
Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N}\right)$$

If the test point x is not covered by the training data, i.e., $x^\top \Lambda^{-1} x$ is huge, then we cannot guarantee $\hat{\theta}^\top x$ is close to $(\theta^\star)^\top x$



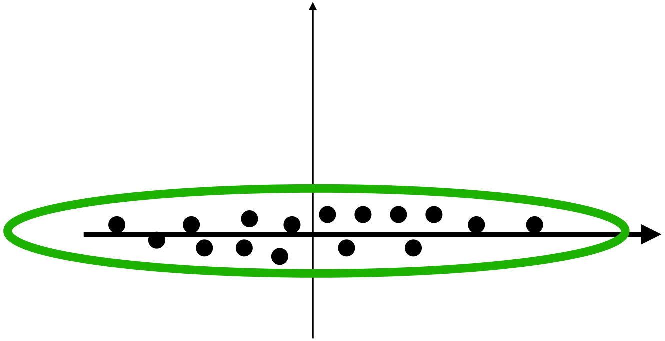
Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N}\right)$$

If the test point x is not covered by the training data, i.e., $x^\top \Lambda^{-1} x$ is huge, then we cannot guarantee $\hat{\theta}^\top x$ is close to $(\theta^\star)^\top x$



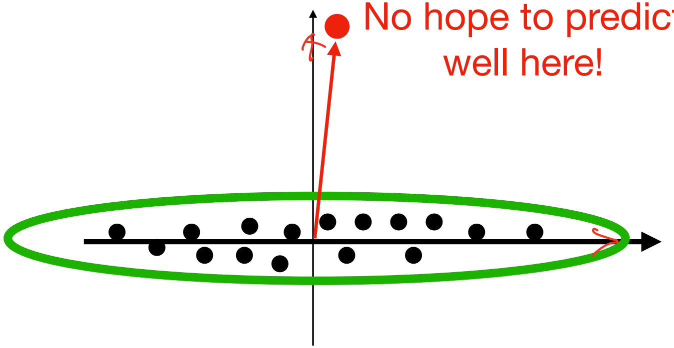
Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N}\right)$$

If the test point x is not covered by the training data, i.e., $x^\top \Lambda^{-1} x$ is huge, then we cannot guarantee $\hat{\theta}^\top x$ is close to $(\theta^\star)^\top x$



$\Lambda = U \Sigma U^T$

λ_i

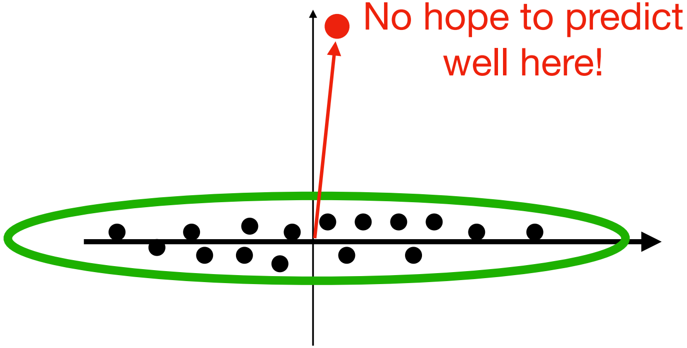
Detour: Issues in Ordinary Linear Squares

Recall $\Lambda = \sum_{i=1}^{N} x_i x_i^\top / N$;

With probability at least $1 - \delta$:

$$(\hat{\theta} - \theta^\star)^\top \Lambda (\hat{\theta} - \theta^\star) \leq O\left(\frac{\sigma^2 d \ln(1/\delta)}{N}\right)$$

If the test point x is not covered by the training data, i.e., $x^\top \Lambda^{-1} x$ is huge, then we cannot guarantee $\hat{\theta}^\top x$ is close to $(\theta^\star)^\top x$



Let's actively design a diverse dataset !
(D-optimal Design)

Detour: D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

Detour: D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\underline{\mathbb{E}_{x\sim\rho}\left[xx^\top\right]}\right)$

Detour: D-optimal Design

Consider a compact space $\mathcal{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathcal{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathcal{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathcal{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

Properties of the D-optimal Design:

support$(\rho^\star) \leq d(d+1)/2$

Detour: D-optimal Design

Consider a compact space $\mathcal{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathcal{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathcal{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathcal{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

Properties of the D-optimal Design:

support$(\rho^\star) \leq d(d+1)/2$

$$\max_{y\in\mathcal{X}} y^\top \left[\mathbb{E}_{x\sim\cancel{\rho}\,\rho^\star} xx^\top\right]^{-1} y \leq d$$

$$\Sigma := \mathbb{E}_{x\sim\rho^\star} xx^\top$$

Detour: OLS w/ D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

Detour: OLS w/ D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

We **actively** construct a dataset $\mathscr{D}$, which contains $\lceil \overset{*}{\rho}(x)N \rceil$ many copies of x

Detour: OLS w/ D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

We **actively** construct a dataset $\mathscr{D}$, which contains $\lceil \rho^{\star}(x) N \rceil$ many copies of x

For each $x \in \mathscr{D}$, query y (noisy measure);

$y = \theta^{\star\top} x + \varepsilon$

$\sum_x \rho^*(x) N = N$

$\Lambda = \frac{1}{N}\sum_{x\in D} xx^T$

$= \frac{1}{N}\sum_{x\in \text{support}(\rho^*)} \rho(x) N = \mathbb{E}_{x\sim\rho^*} xx^T$

$\forall x \in \mathcal{X}$

$x^T \Lambda^{-1} x = x^T \left[\mathbb{E}_{x\sim\rho^*} xx^T\right]^{-1} x \leq d$

Detour: OLS w/ D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

We **actively** construct a dataset $\mathscr{D}$, which contains $\lceil \rho(x)N \rceil$ many copies of x

For each $x \in \mathscr{D}$, query y (noisy measure);

The OLS solution $\hat{\theta}$ on $\mathscr{D}$ has the following point-wise guarantee: w/ prob $1-\delta$

$$\max_{x\in\mathscr{X}} \left| \langle \hat{\theta} - \theta^\star, x \rangle \right| \leq \frac{\sigma d \ln(1/\delta)}{\sqrt{N}}$$

Detour: OLS w/ D-optimal Design

Consider a compact space $\mathscr{X} \subset \mathbb{R}^d$ (without loss of generality, assume span$(\mathscr{X}) = \mathbb{R}^d$)

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$: $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

We **actively** construct a dataset $\mathscr{D}$, which contains $\lceil \rho(x)N \rceil$ many copies of x

For each $x \in \mathscr{D}$, query y (noisy measure);

The OLS solution $\hat\theta$ on $\mathscr{D}$ has the following point-wise guarantee: w/ prob $1-\delta$

$$\max_{x\in\mathscr{X}} \left|\langle \hat\theta - \theta^\star, x\rangle\right| \le \frac{\sigma d \ln(1/\delta)}{\sqrt{N}}$$

$\Lambda^{\frac{1}{2}} \Lambda^{-\frac{1}{2}} := I$

$$\left|(\hat\theta - \theta^\star)^\top x\right| \le \underbrace{\left\| \Lambda^{1/2}(\hat\theta - \theta^\star)\right\|_2}_{①} \underbrace{\left\|\Lambda^{-1/2} x\right\|_2}_{②}$$

$|a^\top b| \le \|a\|_2 \|b\|_2$

①: $(\hat\theta - \theta^\star)\Lambda(\hat\theta - \theta^\star)$

②: $x^\top \Lambda^{-1} x \le d$

Summary so far on OLS & D-optimal Design

D-optimal Design $\rho^\star \in \Delta(\mathcal{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathcal{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

Summary so far on OLS & D-optimal Design

D-optimal Design $\rho^\star \in \Delta(\mathscr{X})$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\mathscr{X})} \ln\det\left(\mathbb{E}_{x\sim\rho}\left[xx^\top\right]\right)$

D-optimal design allows us to **actively** construct a dataset $\mathscr{D} = \{x, y\}$, such that OLS solution is **POINT-WISE** accurate:

$$\max_{x\in\mathscr{X}} \left|\langle \hat{\theta} - \theta^\star, x\rangle\right| \leq \frac{\sigma d \ln(1/\delta)}{\sqrt{N}}$$

Using D-optimal design to construct $\mathscr{D}_h$ in LSVI

Consider the space $\Phi = \{\phi(s, a) : s, a \in S \times A\}$

Using D-optimal design to construct $\mathscr{D}_h$ in LSVI

Consider the space $\Phi = \{\phi(s,a) : s,a \in S \times A\}$

D-optimal Design $\rho^\star \in \Delta(\Phi)$: $\rho^\star = \arg\max_{\rho \in \Delta(\Phi)} \ln\det\left(\mathbb{E}_{s,a\sim\rho}\left[\phi(s,a)\phi(s,a)^\top\right]\right)$

Using D-optimal design to construct $\mathscr{D}_h$ in LSVI

Consider the space $\Phi = \{\phi(s,a) : s, a \in S \times A\}$

D-optimal Design $\rho^\star \in \Delta(\Phi)$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\Phi)} \ln\det\left(\mathbb{E}_{s,a\sim\rho}\left[\phi(s,a)\phi(s,a)^\top\right]\right)$

Construct $\mathscr{D}_h$ that contains $\lceil \rho^{*}(s,a)N \rceil$ many copies of $\phi(s,a)$,

for each $\phi(s,a)$, **query** $y := r(s,a) + V_{h+1}(s'), s' \sim \underbrace{P_h(\,.\,|\,s,a)}_{\text{Generative model}}$

Using D-optimal design to construct $\mathscr{D}_h$ in LSVI

Consider the space $\Phi = \{\phi(s,a) : s, a \in S \times A\}$

D-optimal Design $\rho^\star \in \Delta(\Phi)$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\Phi)} \ln\det\left(\mathbb{E}_{s,a\sim\rho}\left[\phi(s,a)\phi(s,a)^\top\right]\right)$

Construct $\mathscr{D}_h$ that contains $\lceil \rho(s,a)N \rceil$ many copies of $\phi(s,a)$,

for each $\phi(s,a)$, **query** $y := r(s,a) + V_{h+1}(s'), s' \sim P_h(\,.\,|\,s,a)$

What's the Bayes optimal $\mathbb{E}[y\,|\,s,a]$?

$V_{h+1}(s') = \max_{a'} \theta_{h+1}^\top \phi(s',a')$

Bayes optimal: $\mathcal{T}_h(\theta_{h+1})^\top \phi(s,a)$

Using D-optimal design to construct $\mathcal{D}_h$ in LSVI

Consider the space $\Phi = \{\phi(s,a) : s, a \in S \times A\}$

D-optimal Design $\rho^\star \in \Delta(\Phi)$**:** $\rho^\star = \arg\max_{\rho \in \Delta(\Phi)} \ln\det\left(\mathbb{E}_{s,a\sim\rho}\left[\phi(s,a)\phi(s,a)^\top\right]\right)$

Construct $\mathcal{D}_h$ that contains $\lceil \rho(s,a)N \rceil$ many copies of $\phi(s,a)$,
for each $\phi(s,a)$, **query** $y := r(s,a) + V_{h+1}(s'), s' \sim P_h(\,.\,|\,s,a)$

What's the Bayes optimal $\mathbb{E}[y\,|\,s,a]$?

OLS /w D-optimal design implies that θ_h is point-wise accurate:

$$\max_{s,a} \left| \underbrace{\theta_h^\top}_{\theta_h} \phi(s,a) - \underbrace{\mathcal{T}_h(\theta_{h+1})^\top}_{\text{Bayes optimal}\ \hookrightarrow\ T_h(\theta_{h+1})} \phi(s,a) \right| \le \widetilde{O}\left(Hd/\sqrt{N}\right).$$

Concluding the proof of LSVI

1. OLS /w D-optimal design implies that θ_h is point-wise accurate:

$$\max_{s,a} \left| \theta_h^\top \phi(s,a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s,a) \right| \le O\left(Hd/\sqrt{N} \right).$$

Concluding the proof of LSVI

1. OLS /w D-optimal design implies that θ_h is point-wise accurate:

$$\max_{s,a} \left| \theta_h^\top \phi(s,a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s,a) \right| \le O\left(Hd/\sqrt{N}\right).$$

2. This implies that our estimator $Q_h := \theta_h^\top \phi$ is nearly **Bellman-consistent**, i.e.,

$$\| Q_h - \mathcal{T}_h Q_{h+1} \|_\infty \le O\left(Hd/\sqrt{N}\right)$$

Concluding the proof of LSVI

1. OLS /w D-optimal design implies that θ_h is point-wise accurate:

$$\max_{s,a} \left| \theta_h^\top \phi(s,a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s,a) \right| \le O\left(Hd/\sqrt{N}\right).$$

2. This implies that our estimator $Q_h := \theta_h^\top \phi$ is nearly **Bellman-consistent**, i.e.,

$$\left\| Q_h - \mathcal{T}_h Q_{h+1} \right\|_\infty \le O\left(Hd/\sqrt{N}\right)$$

3. Nearly-Bellman consistency implies Q_h is close to $Q_h^\star$ (this holds in general)

$$\|Q_h - Q_h^\star\|_\infty \le O(H^2 d/\sqrt{N})$$

Concluding the proof of LSVI

1. OLS /w D-optimal design implies that θ_h is point-wise accurate:

$$\max_{s,a} \left| \theta_h^\top \phi(s,a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s,a) \right| \le O\left(Hd/\sqrt{N}\right).$$

2. This implies that our estimator $Q_h := \theta_h^\top \phi$ is nearly **Bellman-consistent**, i.e.,

$$\left\| Q_h - \mathcal{T}_h Q_{h+1} \right\|_\infty \le O\left(Hd/\sqrt{N}\right)$$

3. Nearly-Bellman consistency implies Q_h is close to $Q_h^\star$ (this holds in general)

$$\|Q_h - Q_h^\star\|_\infty \le O(H^2 d/\sqrt{N})$$

$$\Rightarrow V^\star - V^{\hat{\pi}} \le \widetilde{O}(H^3 d/\sqrt{N}) \Rightarrow \varepsilon$$

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

$\Rightarrow$ VI

$\widetilde{O}$: ignore $(\ln 1/\delta)$

Outline for Today

✓ 1. Proof Sketch of LSVI

2. LSVI in **Offline RL**

Offline Reinforcement Learning

Offline Reinforcement Learning

Learner **cannot interact** with the environment, instead, learner is given **static** datasets:

$$\mathcal{D}_h = \{s, a, r, s'\}, \quad s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$$

Offline Reinforcement Learning

Learner **cannot interact** with the environment, instead, learner is given **static** datasets:

$$\mathscr{D}_h = \{s, a, r, s'\}, \quad s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$$

Offline Distribution (e.g., maybe is d^{π_b} for some behavior policy π_b)

Offline Reinforcement Learning

Learner **cannot interact** with the environment, instead, learner is given **static** datasets:

$$\mathcal{D}_h = \{s, a, r, s'\}, \quad s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$$

Offline Distribution (e.g., maybe is d^{π_b} for some behavior policy π_b)

Offline RL is promising for safety critical applications
(i.e., learning from logged data for health applications…)

Recall Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s,a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$$\theta_h = \arg\min_{\theta} \sum_{\mathscr{D}_h} \left(\theta^T \phi(s,a) - \left(r + V_{h+1}(s')\right)\right)^2$$

$\Rightarrow r + \max_{a'} \theta_{h+1}^T \phi(s', a')$

Set $V_h(s) := \max_a \theta_h^\top \phi(s,a), \forall s$

$Q^*_h(s,a) = \theta^{*T} \phi(s,a)$

Return $\hat{\pi}_h(s) = \arg\max_a \theta_h^\top \phi(s,a), \forall h$

Recall Least-Square Value Iteration

Datasets $\mathscr{D}_0, \ldots, \mathscr{D}_{H-1}$, w/

$\mathscr{D}_h = \{s, a, r, s'\}, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Set $V_H(s) = 0, \forall s$

For h = H-1 to 0:

$$\theta_h = \arg\min_{\theta} \sum_{\mathscr{D}_h} \left(\theta^T \phi(s, a) - \left(r + V_{h+1}(s')\right)\right)^2$$

Set $V_h(s) := \max_a \theta_h^\top \phi(s, a), \forall s$

Return $\hat{\pi}_h(s) = \arg\max_a \theta_h^\top \phi(s, a), \forall h$

LSVI directly can directly operate in offline model!

Least-Square Value Iteration Guarantee

Recall $\mathscr{D}_h = \{s, a, r, s'\}, s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Least-Square Value Iteration Guarantee

Recall $\mathscr{D}_h = \{s, a, r, s'\}, s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Assumptions

1. Full offline data coverage: $\sigma_{\min}\left(\mathbb{E}_{s,a\sim\nu}\phi(s,a)\phi(s,a)^{\top}\right) \geq \kappa$
2. Linear Bellman completion

Least-Square Value Iteration Guarantee

Recall $\mathscr{D}_h = \{s, a, r, s'\}, s, a \sim \nu, r = r(s, a), s' \sim P_h(\cdot \mid s, a)$

Assumptions

1. Full offline data coverage: $\sigma_{\min}\left(\mathbb{E}_{s,a\sim\nu}\phi(s, a)\phi(s, a)^\top\right) \geq \kappa$
2. Linear Bellman completion

Then, with probability at least $1 - \delta$, LSVI return $\hat{\pi}$ with $V^\star - V^{\hat{\pi}} \leq \epsilon$, using at most poly $\left(H, 1/\epsilon, 1/\kappa, d, \ln(1/\delta)\right)$

4

The proof for the offline set is almost identical

Key step:

Linear Bellman completion + Linear Regression w/ full data coverage

=> Near-Bellman consistency, **i.e.,** $\|Q_h - \mathcal{T}_h Q_{h+1}\|_\infty$ **is small**

The proof for the offline set is almost identical

Key step:

Linear Bellman completion + Linear Regression w/ full data coverage

=> Near-Bellman consistency, **i.e.,** $\|Q_h - \mathcal{T}_h Q_{h+1}\|_\infty$ **is small**

e.g., with N training examples where $(s, a) \sim \nu$, and $r = r(s, a), s' \sim P_h(\cdot \mid s, a)$, we have

$$\mathbb{E}_{s,a\sim\nu}\left(\theta_h^\top \phi(s, a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s, a)\right)^2 \leq \text{poly}(H, d, 1/N)$$

The proof for the offline set is almost identical

Key step:

Linear Bellman completion + Linear Regression w/ full data coverage

=> Near-Bellman consistency, **i.e.,** $\| Q_h - \mathcal{T}_h Q_{h+1} \|_\infty$ **is small**

e.g., with N training examples where $(s, a) \sim \nu$, and $r = r(s, a), s' \sim P_h(\cdot \mid s, a)$, we have

$$\mathbb{E}_{s,a\sim\nu} \left(\theta_h^\top \phi(s, a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s, a)\right)^2 \leq \text{poly}(H, d, 1/N)$$

Then with Cauchy-Schwartz, we get

$$\forall s, a, \left| (\theta_h - \mathcal{T}_h(\theta_{h+1}))^\top \phi(s, a) \right| \leq \|\theta_h - \mathcal{T}_h(\theta_{h+1})\|_\Sigma \|\phi(s, a)\|_{\Sigma^{-1}}$$

The proof for the offline set is almost identical

Key step:

Linear Bellman completion + Linear Regression w/ full data coverage

=> Near-Bellman consistency, **i.e.,** $\|Q_h - \mathcal{T}_h Q_{h+1}\|_\infty$ **is small**

e.g., with N training examples where $(s, a) \sim \nu$, and $r = r(s, a), s' \sim P_h(\cdot \mid s, a)$, we have

$$\mathbb{E}_{s,a\sim\nu} \left(\theta_h^\top \phi(s, a) - \mathcal{T}_h(\theta_{h+1})^\top \phi(s, a)\right)^2 \leq \text{poly}(H, d, 1/N)$$

Then with Cauchy-Schwartz, we get

$$\forall s, a, \left|(\theta_h - \mathcal{T}_h(\theta_{h+1}))^\top \phi(s, a)\right| \leq \|\theta_h - \mathcal{T}_h(\theta_{h+1})\|_\Sigma \|\phi(s, a)\|_{\Sigma^{-1}}$$

(we will give a HW question on a related topic)

Summary

1. Linear Bellman Completion definition (a strong assumption, though captures some models)

Summary

1. Linear Bellman Completion definition (a strong assumption, though captures some models)

2. Least square value iteration: integrate Linear regression into DP, i.e., $Q_h := \theta_h^\top \phi \approx Q_h^\star$ via

$$\phi(s,a) \mapsto r(s,a) + \max_{a'} \theta_{h+1}^\top \phi(s',a')$$

Summary

1. Linear Bellman Completion definition (a strong assumption, though captures some models)

2. Least square value iteration: integrate Linear regression into DP, i.e., $Q_h := \theta_h^\top \phi \approx Q_h^\star$ via

$$\phi(s,a) \mapsto r(s,a) + \max_{a'} \theta_{h+1}^\top \phi(s', a')$$

3. Leverage D-optimal design, we make sure that θ_h is point-wise accurate, which ensures near Bellman consistent, i.e., $\left\| Q_h - \mathcal{T}_h Q_{h+1} \right\|_\infty$ is small

Summary

1. Linear Bellman Completion definition (a strong assumption, though captures some models)

2. Least square value iteration: integrate Linear regression into DP, i.e., $Q_h := \theta_h^\top \phi \approx Q_h^\star$ via

$$\phi(s, a) \mapsto r(s, a) + \max_{a'} \theta_{h+1}^\top \phi(s', a')$$

3. Leverage D-optimal design, we make sure that θ_h is point-wise accurate, which ensures near Bellman consistent, i.e., $\left\| Q_h - \mathcal{T}_h Q_{h+1} \right\|_\infty$ is small

4. Near-Bellman consistency implies small approximation error of Q_h (holds in general)

Next week

Fitted Dynamic Programming — can we extend linear function approx to general function approx (e.g., neural network, decision tree, etc)?

Exploration: Multi-armed Bandits and Stochastic Linear Bandits (Bandits = MDP w/ $H = 1$)