Recent Advances of Statistical Reinforcement Learning

Part 2

Gergely Neu (Universitat Pompeu Fabra)
Sattar Vakili (MediaTek Research)

Tutorial, UAI 2024

Part 2

1. Introduction to structural complexity
2. Linear Function Approximation
3. Non-linear Function Approximation

Tabular Setting

Recall results for the tabular setting:

- Q-learning with UCB: [Jin et al., 2018]

$$\text{Regret}(T) = \mathcal{O}(\sqrt{H^3SAT})$$

- Sample complexity:

$$\tilde{\mathcal{O}}(\frac{\text{poly}(H)SA}{\epsilon^2})$$

Tabular Setting

Recall results for the tabular setting:

- Q-learning with UCB: [Jin et al., 2018]

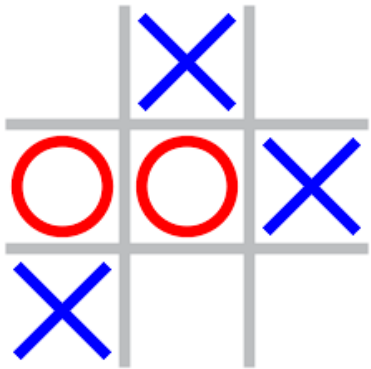
$$\text{Regret}(T) = \mathcal{O}(\sqrt{H^3SAT})$$

- Sample complexity:

$$\tilde{\mathcal{O}}(\frac{\text{poly}(H)SA}{\epsilon^2})$$

These are only meaningful if $T \ll S$ or $\epsilon \gg 1/\sqrt{S}$!

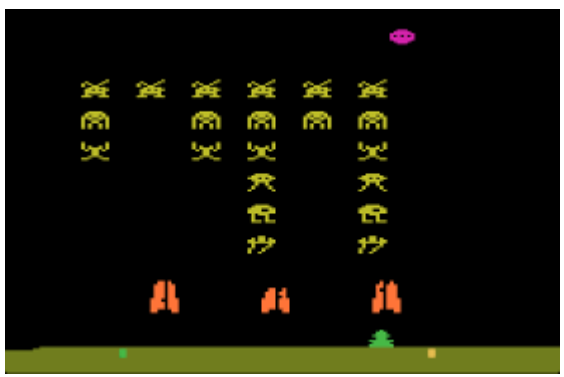
Why "Tabular"?



- Small size of state-action space
- $Q(s, a)$ can be represented as a table

Why Function Approximation?

Number of states S is enormous in real-world problems!



- Game of Go: 10^{170} states
- Atari: 10^{100} states
- Physical systems: continuum of states

Why Function Approximation?

Two types of challenges:

- Computational: Q and π cannot even be stored in memory, and Bellman equations are intractable to solve even if P and r were known
- Statistical: Most states are not visited even once! How could we expect to learn about P or r like that?

Why Function Approximation?

Two types of challenges:

- Computational: Q and π cannot even be stored in memory, and Bellman equations are intractable to solve even if P and r were known
- Statistical: Most states are not visited even once! How could we expect to learn about P or r like that?

We need to find a way to **generalize** knowledge from visited states to unvisited states by leveraging **structure**

RL with Function Approximation

- Approximate value function $Q(s, a)$ (or policy) in a class $\mathcal{F}$.
- Hope that $\mathcal{F}$ captures the MDP structure appropriately and leverage the information in structure of $\mathcal{F}$ to learn faster if possible.
- Typical function classes: Linear, Kernel-based, NN-based

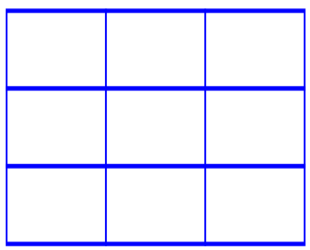
$$\text{Tabular} \rightarrow \text{Linear} \rightarrow \text{Nonlinear}$$

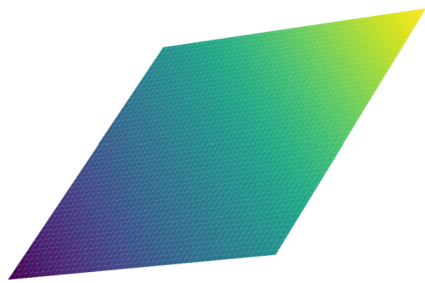
Setting

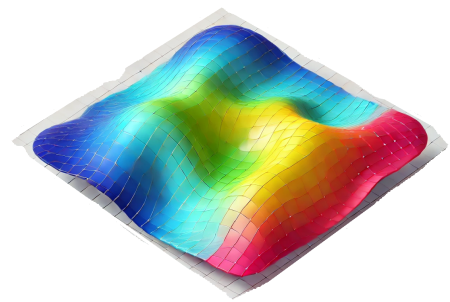
- Generative oracle, Offline, Online
- Episodic, Infinite horizon (discounted)
- Model-based, Model-free

In this part we focus on:

$$\text{Tabular} \rightarrow \text{Linear} \rightarrow \text{Nonlinear}$$





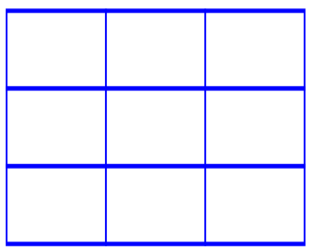


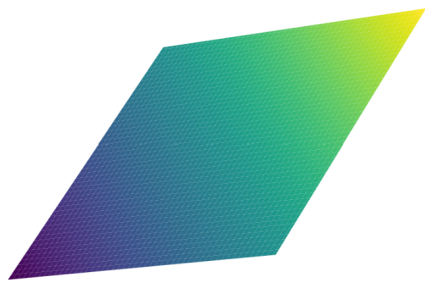
Setting

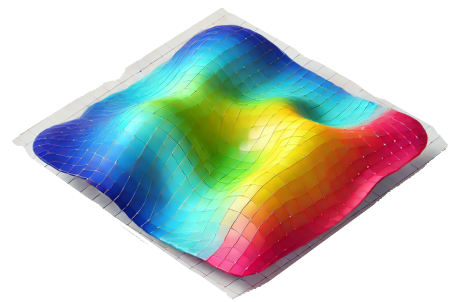
- Generative oracle, Offline, Online
- Episodic, Infinite horizon (discounted)
- Model-based, Model-free

In this part we focus on:

Tabular $\rightarrow$ Linear $\rightarrow$ Nonlinear

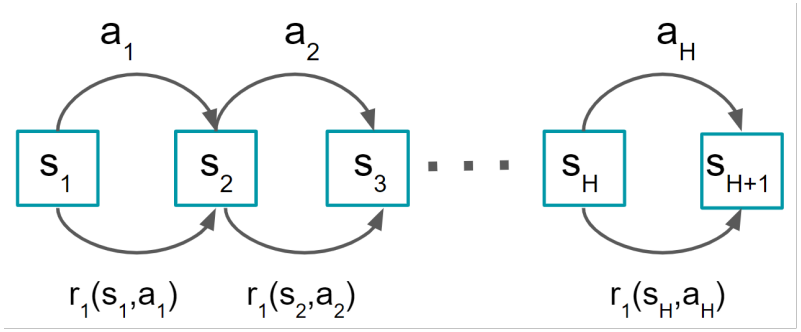






Setting

For a clear and sharp presentation we focus on episodic MDPs

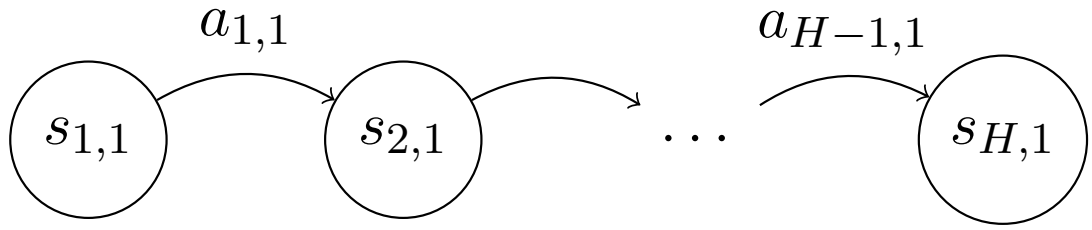


For simplicity, we assume r is known and deterministic

We focus on the structural complexity of $P(s'|s,a)$

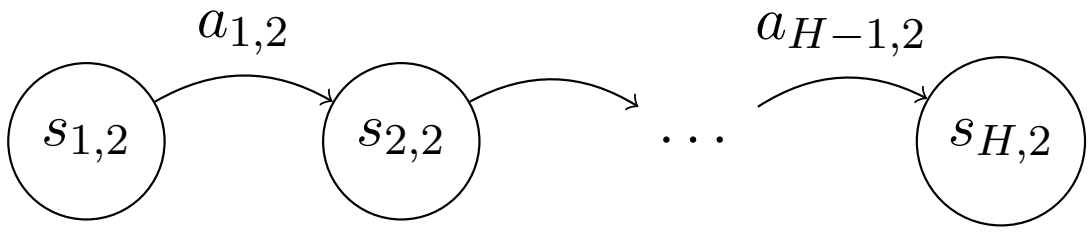
Episodic MDP

Episode 1:

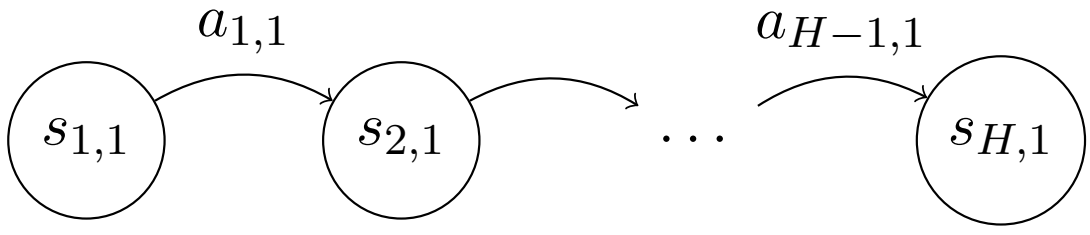


Episodic MDP

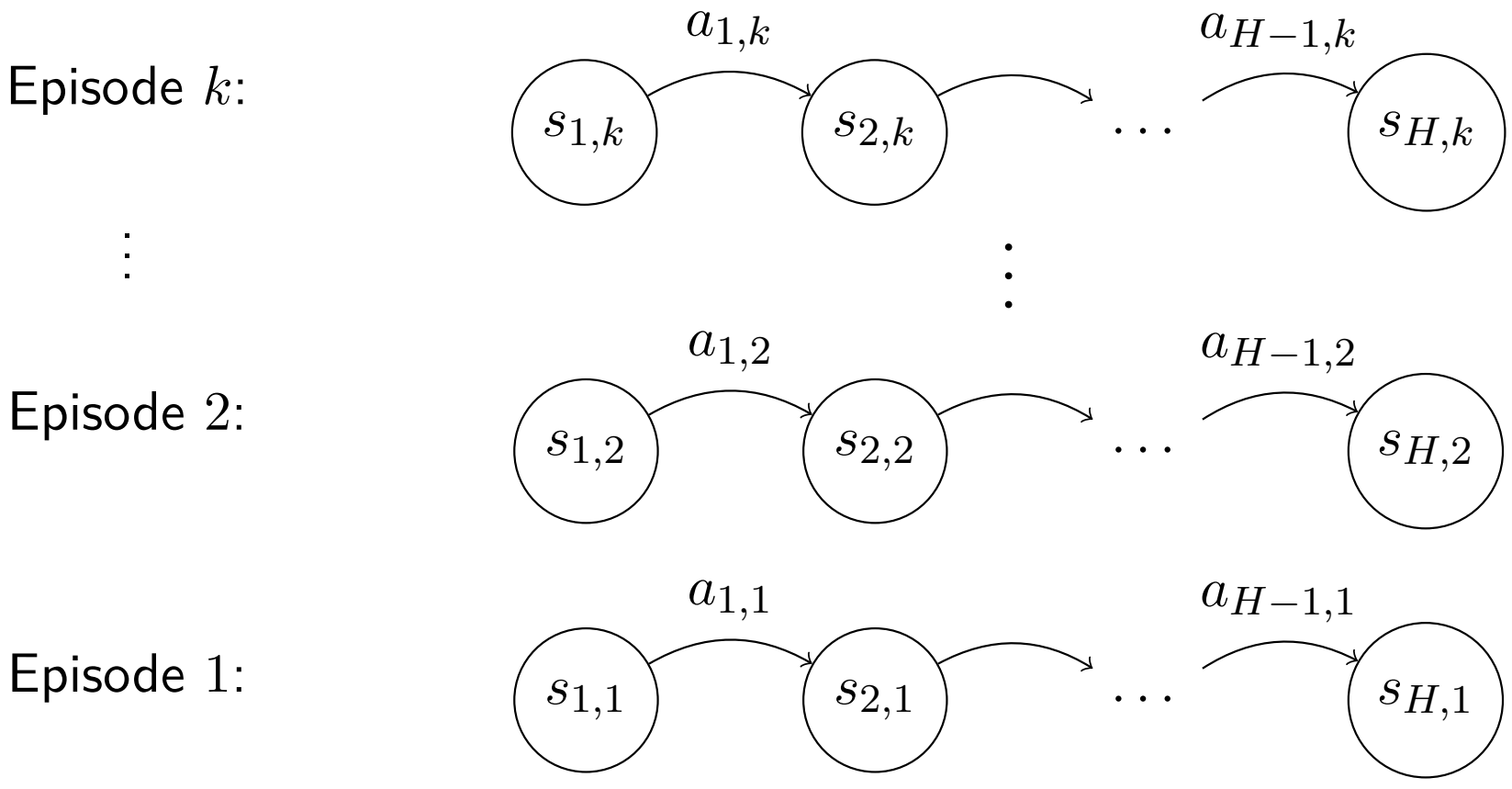
Episode 2:



Episode 1:



Episodic MDP



Part 2

1. Introduction to structural complexity
2. Linear Function Approximation
3. Non-linear Function Approximation

Linear Function Approximation

IDEA: approximate the Q-functions as linear functions of a given d-dimensional feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.

Linear Function Approximation

IDEA: approximate the Q-functions as linear functions of a given d-dimensional feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.

Let $\Phi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times d}$ be the "matrix" of stacked feature vectors $[\boldsymbol{\phi}(s_1, a_1) \dots \boldsymbol{\phi}(s_N, a_N)]^\top$ (where $N = |\mathcal{S} \times \mathcal{A}|$).

We need to find a parameter vector $\boldsymbol{\theta}^\star$ such that $Q^\star \approx \Phi\boldsymbol{\theta}^\star$.
(Meaning that $Q^\star(s, a) \approx \langle \boldsymbol{\theta}^\star, \boldsymbol{\phi}(s, a)\rangle$.)

Linear Function Approximation

IDEA: approximate the Q-functions as linear functions of a given d-dimensional feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.

Let $\Phi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times d}$ be the "matrix" of stacked feature vectors $[\boldsymbol{\phi}(s_1, a_1) \dots \boldsymbol{\phi}(s_N, a_N)]^\top$ (where $N = |\mathcal{S} \times \mathcal{A}|$).

We need to find a parameter vector $\boldsymbol{\theta}^\star$ such that $Q^\star \approx \Phi\boldsymbol{\theta}^\star$.
(Meaning that $Q^\star(s, a) \approx \langle \boldsymbol{\theta}^\star, \boldsymbol{\phi}(s, a)\rangle$.)

QUESTION: When can we learn $\boldsymbol{\theta}^\star$ efficiently?

Linear Function Approximation

Various conditions on the feature map Φ have been studied:

- Linear $Q^\star$: there exists a $\boldsymbol{\theta}^\star$ such that $Q^\star = \Phi\boldsymbol{\theta}^\star$.
- Linear Q^π: for every policy π, there exists a $\boldsymbol{\theta}^\pi$ such that $Q^\pi = \Phi\boldsymbol{\theta}^\pi$.
- Closure under Bellman operator: for any $Q_{\boldsymbol{\theta}} = \Phi\boldsymbol{\theta}$, $\mathcal{T}Q_{\boldsymbol{\theta}} \in \operatorname{span}(\Phi)$.
- Linear MDP: The transition and reward functions are linear in the features. This implies all of the above conditions.

A number of more refined conditions have been also studied, such as assuming linearity of $V^\star$ in some feature map, or other types of factorized transition models. We refer to Du et al. [2021], Jin et al. [2021] for more details on such extensions.

What can we hope for?

WANT: find an ϵ-optimal policy with a computational and sample complexity polynomial in d, $1/\epsilon$ and H, independently of S and A.

What can we hope for?

WANT: find an ϵ-optimal policy with a computational and sample complexity polynomial in d, $1/\epsilon$ and H, independently of S and A.

- This is impossible when only requiring linear $Q^\star$-realizability! [Weisz et al., 2021]

What can we hope for?

WANT: find an ϵ-optimal policy with a computational and sample complexity polynomial in d, $1/\epsilon$ and H, independently of S and A.

- This is impossible when only requiring linear $Q^\star$-realizability! [Weisz et al., 2021]
- Polynomial sample complexity is possible when relaxing the condition to linear Q^π-realizability, but no practical algorithms are known [Weisz et al., 2022, 2023].

What can we hope for?

WANT: find an ϵ-optimal policy with a computational and sample complexity polynomial in d, $1/\epsilon$ and H, independently of S and A.

- This is impossible when only requiring linear $Q^\star$-realizability! [Weisz et al., 2021]
- Polynomial sample complexity is possible when relaxing the condition to linear Q^π-realizability, but no practical algorithms are known [Weisz et al., 2022, 2023].
- Situation is similar when only assuming closure under Bellman operator / Bellman completeness [Zanette et al., 2020b, Du et al., 2021].

What can we hope for?

WANT: find an ϵ-optimal policy with a computational and sample complexity polynomial in d, $1/\epsilon$ and H, independently of S and A.

- This is impossible when only requiring linear $Q^\star$-realizability! [Weisz et al., 2021]
- Polynomial sample complexity is possible when relaxing the condition to linear Q^π-realizability, but no practical algorithms are known [Weisz et al., 2022, 2023].
- Situation is similar when only assuming closure under Bellman operator / Bellman completeness [Zanette et al., 2020b, Du et al., 2021].
- Linear MDP condition enables both statistical and computational efficiency!!! [Jin et al., 2023]

Linear MDPs

Linear transition function:

$$P_h(\cdot|s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h(\cdot) \rangle,$$

where $\boldsymbol{\mu}_h(\cdot) = [\mu_h^1(\cdot), \cdots, \mu_h^d(\cdot)]$ is a d-dimensional signed measure.

Linear rewards: $r_h(s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\vartheta}_h \rangle$.

Linear MDPs

Linear transition function:

$$P_h(\cdot|s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h(\cdot) \rangle,$$

where $\boldsymbol{\mu}_h(\cdot) = [\mu_h^1(\cdot), \cdots, \mu_h^d(\cdot)]$ is a d-dimensional signed measure.

Linear rewards: $r_h(s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\vartheta}_h \rangle$.

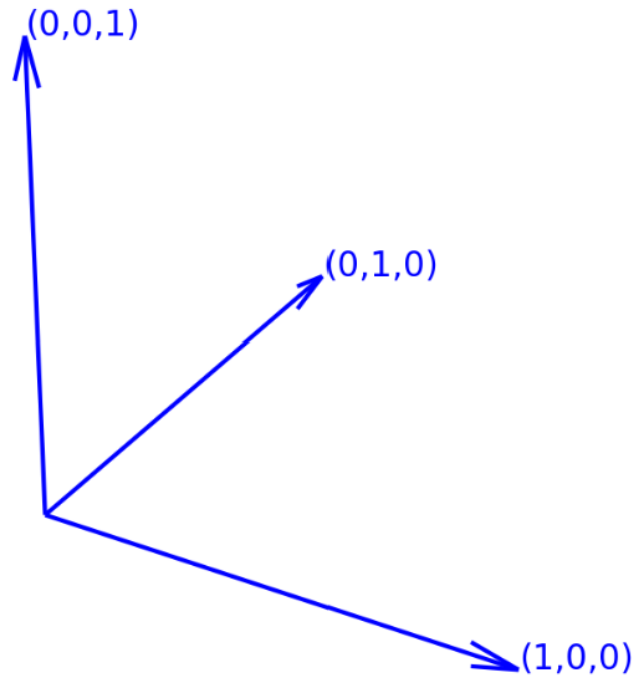
In matrix notation:

- Transition operator $P_h \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times\mathcal{S}}$ can be written as $P_h = \Phi M_h$ for some "matrix" $M_h \in \mathbb{R}^{\mathcal{S}\times d}$.
- Reward function can be written as $r_h = \Phi \boldsymbol{\vartheta}_h$ for some $\boldsymbol{\vartheta}_h \in \mathbb{R}^d$.

Tabular MDPs are linear

Tabular setting is a special case with dimension $d = SA$:

- Let $\boldsymbol{\phi}(s, a) = \boldsymbol{e}_{(s,a)}$ be the canonical basis in $\mathbb{R}^d$
- $P_h(\cdot|s, a) = \boldsymbol{e}_{s,a}^\top \boldsymbol{\mu}_h(\cdot)$



A magical property of linear MDPs

In a linear MDP, the Q-functions of all policies are linear in Φ:

$$\begin{aligned} Q_h^\pi &= r_h + P_h V_{h+1}^\pi = \Phi \vartheta_h + \Phi M_h V_{h+1}^\pi \\ &= \Phi \left(\vartheta_h + M_h V_{h+1}^\pi \right) = \Phi \boldsymbol{\theta}_h, \end{aligned}$$

with $\boldsymbol{\theta}_h = \vartheta_h + M_h V_{h+1}^\pi$.

A magical property of linear MDPs

In a linear MDP, the Q-functions of all policies are linear in Φ:

$$\begin{aligned} Q_h^\pi &= r_h + P_h V_{h+1}^\pi = \Phi \vartheta_h + \Phi M_h V_{h+1}^\pi \\ &= \Phi \left(\vartheta_h + M_h V_{h+1}^\pi \right) = \Phi \boldsymbol{\theta}_h, \end{aligned}$$

with $\boldsymbol{\theta}_h = \vartheta_h + M_h V_{h+1}^\pi$.

This implies linear $Q^\star$-realizability, linear Q^π-realizability, Bellman compleness, and many more useful properties for analysis! E.g., note that for any function $u \in \mathbb{R}^{\mathcal{S}}$, $P_h u = \Phi M_h u$ is linear in Φ.

A magical property of linear MDPs

In a linear MDP, the Q-functions of all policies are linear in Φ:

$$\begin{aligned} Q_h^\pi &= r_h + P_h V_{h+1}^\pi = \Phi \vartheta_h + \Phi M_h V_{h+1}^\pi \\ &= \Phi \left(\vartheta_h + M_h V_{h+1}^\pi \right) = \Phi \boldsymbol{\theta}_h, \end{aligned}$$

with $\boldsymbol{\theta}_h = \vartheta_h + M_h V_{h+1}^\pi$.

This implies linear $Q^\star$-realizability, linear Q^π-realizability, Bellman compleness, and many more useful properties for analysis! E.g., note that for any function $u \in \mathbb{R}^{\mathcal{S}}$, $P_h u = \Phi M_h u$ is linear in Φ.

The structure of linear MDPs allows us to import tools from linear bandit literature [Abbasi-Yadkori et al., 2011, Lattimore and Szepesvári, 2020].

Optimistic approximate dynamic programming

IDEA: Combine the techniques for tabular MDPs with exploration bonuses borrowed from the linear bandit literature!

Optimistic approximate dynamic programming

IDEA: Combine the techniques for tabular MDPs with exploration bonuses borrowed from the linear bandit literature!

UCB-VI [Azar et al., 2017]:

1. Backtrack $h = H, H-1, \ldots, 1$: run optimistic value iteration

$$Q_h = r_h + \underbrace{\widehat{P}_h}_{\text{model estimate}} V_{h+1} + \underbrace{b_h}_{\text{exploration bonus}}$$

   and set $V_h(s) = \max_a Q_h(s,a)$ for all s, a.

2. Forward $h = 1, 2, \ldots, H$: take actions according to greedy policy

$$\pi_h(s) = \arg\max_a Q_h(s,a).$$

Optimistic approximate dynamic programming

IDEA: Combine the techniques for tabular MDPs with exploration bonuses borrowed from the linear bandit literature!

UCB-VI [Azar et al., 2017]:

1. Backtrack $h = H, H-1, \ldots, 1$: run optimistic value iteration

$$Q_h = r_h + \underbrace{\widehat{P}_h}_{\text{model estimate}} V_{h+1} + \underbrace{b_h}_{\text{exploration bonus}}$$

   and set $V_h(s) = \max_a Q_h(s,a)$ for all s, a.

2. Forward $h = 1, 2, \ldots, H$: take actions according to greedy policy

$$\pi_h(s) = \arg\max_a Q_h(s,a).$$

But how do we define $\widehat{P}_h$ and b_h?

Least Squares Value Iteration (LSVI)

Transition model $\widehat{P_h}$ can be defined implicitly via least-squares:

- Solve the regularized linear regression problem

$$\widehat{\boldsymbol{w}}_{h,k} = \arg\min_{\boldsymbol{w}} \sum_{t=1}^{k} (V_{h+1,k}(s_{h+1,t}) - \langle \boldsymbol{\phi}(s_{h,t}, a_{h,t}), \boldsymbol{w} \rangle)^2 + \lambda^2 \|\boldsymbol{w}\|^2$$

- That provides a prediction

$$\widehat{[P_h V_{h+1,k}]}(s,a) = \langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_{h,k} \rangle$$

- Also, an uncertainty quantification (variance)

$$\sigma^2_{h,k}(s,a) = \|\boldsymbol{\phi}(s,a)\|^2_{(\lambda I + \Sigma_{h,k})^{-1}}$$

$$\Sigma_{h,k} = \sum_{t=1}^{k} \boldsymbol{\phi}^\top(s_{h,t}, a_{h,t}) \boldsymbol{\phi}(s_{h,t}, a_{h,t})$$

LSVI-UCB

The prediction and variance give us an upper confidence bound on $Q^\star$:

$$Q_{h,k}(s,a) = r_h(s,a) + \widehat{[P_h V_{h+1}]}(s,a) + \beta(\delta)\sigma_{h,k}(s,a)$$

This is then used to compute an UCB on $V^\star$ as

$$V_{h,k}(s) = \max_a Q_{h,k}(s,a)$$

Performance guarantees

Theorem [Jin et al., 2023] The regret of LSVI-UCB satisfies $\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{d^3K})$.

This implies a sample complexity guarantee of $\tilde{\mathcal{O}}(\frac{\text{poly}(H)d^3}{\epsilon^2})$.

Performance guarantees

Theorem [Jin et al., 2023] The regret of LSVI-UCB satisfies $\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{d^3K})$.

This implies a sample complexity guarantee of $\tilde{\mathcal{O}}(\frac{\text{poly}(H)d^3}{\epsilon^2})$.

Proof ideas:

- Prove confidence bounds

$$|[\widehat{P_h V_{h+1,k}}](s,a) - [P_h V_{h+1,k}](s,a)| \leq \beta(\delta)\sigma_{h,k}(s,a).$$

- Using standard techniques (e.g., Azar et al., 2017), show

$$\text{Regret}(K) \lesssim \textstyle\sum_{h,k} \beta(\delta)\sigma_{h,k}(s_{h,k}, a_{h,k}).$$

- Use elliptical potential lemma (e.g., Abbasi-Yadkori et al., 2011) to show

$$\textstyle\sum_{h,k} \sigma_{h,k}(s_{h,k}, a_{h,k}) \lesssim H\sqrt{Kd\log(K)}.$$

Proving the confidence bounds

By standard results on least-squares estimators (e.g., Abbasi-Yadkori et al., 2011), one can prove the following confidence bound for any fixed $u \in \mathbb{R}^{\mathcal{S}}$ that holds with probability at least $1 - \delta$:

$$\left|\widehat{[P_h u]}(s,a) - [P_h u](s,a)\right| \leq \overline{\beta}(\delta)\sigma_{h,k}(s,a)$$

for some

$$\overline{\beta}(\delta) \approx \lambda^{\frac{1}{2}}\|M_h u\| + H\sqrt{d\log(\tfrac{K}{\delta})}$$

Proving the confidence bounds

By standard results on least-squares estimators (e.g., Abbasi-Yadkori et al., 2011), one can prove the following confidence bound for any fixed $u \in \mathbb{R}^{\mathcal{S}}$ that holds with probability at least $1 - \delta$:

$$\left|\widehat{[P_h u]}(s,a) - [P_h u](s,a)\right| \leq \overline{\beta}(\delta)\sigma_{h,k}(s,a)$$

for some

$$\overline{\beta}(\delta) \approx \lambda^{\frac{1}{2}}\|M_h u\| + H\sqrt{d\log(\tfrac{K}{\delta})}$$

Challenge: $u = V_{h+1,k}$ is not fixed, but depends on all past data!

Proving the confidence bounds

By standard results on least-squares estimators (e.g., Abbasi-Yadkori et al., 2011), one can prove the following confidence bound for any fixed $u \in \mathbb{R}^{\mathcal{S}}$ that holds with probability at least $1 - \delta$:

$$\left| \widehat{[P_h u]}(s,a) - [P_h u](s,a) \right| \leq \overline{\beta}(\delta) \sigma_{h,k}(s,a)$$

for some

$$\overline{\beta}(\delta) \approx \lambda^{\frac{1}{2}} \|M_h u\| + H\sqrt{d \log(\tfrac{K}{\delta})}$$

Challenge: $u = V_{h+1,k}$ is not fixed, but depends on all past data!

Solution: Covering number argument

Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

$$\mathcal{V} = \left\{V(s) = \min\{H,\ \max_a \boldsymbol{\phi}^\top(s,a)\widehat{\boldsymbol{w}} + \beta\|\boldsymbol{\phi}(s,a)\|_{(\lambda I+\Sigma)^{-1}}\}\right\}.$$

- **Idea:** cover the space of functions $\mathcal{V}$ such that we can rewrite

$$\left|\widehat{[P_h V_{h+1,k}]}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \epsilon + \sup_{u\in\mathcal{V}}\left|\widehat{[P_h u]}(s,a) - [P_h u](s,a)\right|.$$

- How many functions u are required to cover $\mathcal{V}$ up to ϵ error?

$$\mathcal{N}_\epsilon = \tilde{\mathcal{O}}(d^2)$$

Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

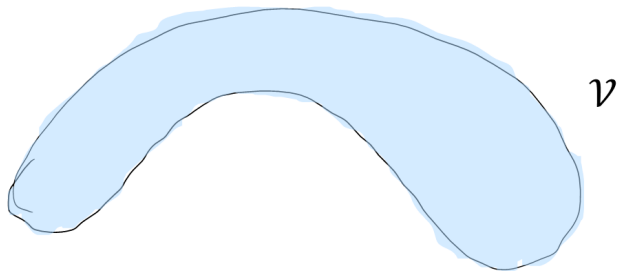
$$\mathcal{V} = \left\{V(s) = \min\{H,\ \max_a \boldsymbol{\phi}^\top(s,a)\widehat{\boldsymbol{w}} + \beta\|\boldsymbol{\phi}(s,a)\|_{(\lambda I+\Sigma)^{-1}}\}\right\}.$$

- **Idea:** cover the space of functions $\mathcal{V}$ such that we can rewrite

$$\left|\widehat{[P_h V_{h+1,k}]}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \epsilon + \sup_{u\in\mathcal{V}}\left|\widehat{[P_h u]}(s,a) - [P_h u](s,a)\right|.$$

- How many functions u are required to cover $\mathcal{V}$ up to ϵ error?

$$\mathcal{N}_\epsilon = \tilde{\mathcal{O}}(d^2)$$



Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

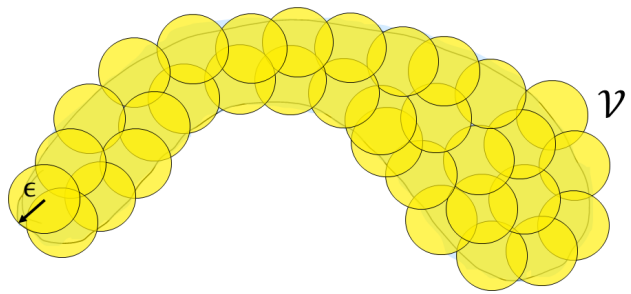
$$\mathcal{V} = \left\{V(s) = \min\{H,\ \max_a \boldsymbol{\phi}^\top(s,a)\widehat{\boldsymbol{w}} + \beta\|\boldsymbol{\phi}(s,a)\|_{(\lambda I+\Sigma)^{-1}}\}\right\}.$$

- **Idea:** cover the space of functions $\mathcal{V}$ such that we can rewrite

$$\left|\widehat{[P_h V_{h+1,k}]}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \epsilon + \sup_{u\in\mathcal{V}} \left|\widehat{[P_h u]}(s,a) - [P_h u](s,a)\right|.$$

- How many functions u are required to cover $\mathcal{V}$ up to ϵ error?

$$\mathcal{N}_\epsilon = \tilde{\mathcal{O}}(d^2)$$



Covering Number Argument

- We can now use a union-bound argument to show that

$$\sup_{u \in \mathcal{V}} \left| \widehat{[P_h u]}(s,a) - [P_h u](s,a) \right| \leq \epsilon + \overline{\beta}(\delta/\mathcal{N}_\epsilon)\sigma_{h,k}(s,a)$$

holds with probability at least $1-\delta$.

- Choosing $\epsilon \approx 1/T$, we get

$$\beta(\delta) = \overline{\beta}(\delta/\mathcal{N}_\epsilon) \approx \lambda^{\frac{1}{2}}\|MV_{h,k}\| + H\sqrt{d\log(\tfrac{K\mathcal{N}_\epsilon}{\delta})} = \tilde{\mathcal{O}}(Hd)$$

Alternative linear models

Linear MDP model factorizes $P = \Phi M$ with some known $\Phi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times d}$ and some unknown $M \in \mathbb{R}^{d\times\mathcal{S}}$.

Some alternative factorizations are:

- Linear mixture MDPs [Zhou et al., 2021]: $P = \Phi\boldsymbol{\theta}$ with some known $\Phi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A}\times\mathcal{S})\times d}$ and unknown $\boldsymbol{\theta} \in \mathbb{R}^d$. Analysis is simpler but the model doesn't allow simple and explicit Q-function approximation and leads to impractical algorithms.
- "MatrixRL" [Yang and Wang, 2020]: $P = \Phi M \Psi$ with some known $\Phi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times m}$, another known $\Psi \in \mathbb{R}^{n\times\mathcal{S}}$, and an unknown $M \in \mathbb{R}^{m\times n}$. Can be shown to be a special case of linear mixture MDPs, and suffers from the same limitations.
- Low-rank MDPs [Modi et al., 2024]: Same as linear MDPs except both Φ and M are unknown and belong to finite model class. Requires much more sophisticated techniques, but algorithms are kind of tractable.

Some References

- Linear MDPs: Jin et al. [2020, 2023], Yang and Wang [2019, 2020], Neu and Pike-Burke [2020]
- Linear Bellman complete models: Zanette et al. [2020a]
- Linear mixture MDPs: Yang and Wang [2020], Ayoub et al. [2020], Zhou et al. [2021], Moulin and Neu [2023]
- Other model classes with hidden finite-dimensional linear structure: Du et al. [2021], Jin et al. [2021]

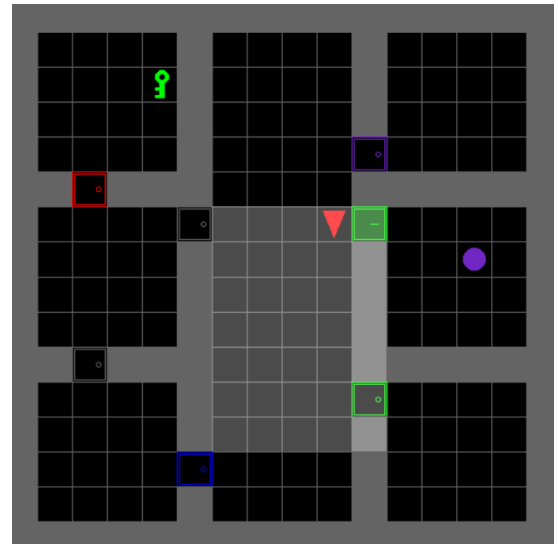
Part 2

1. Introduction to structural complexity
2. Linear Function Approximation
3. Non-linear Function Approximation

Limitations of the Linear Setting

Directly reachable states:

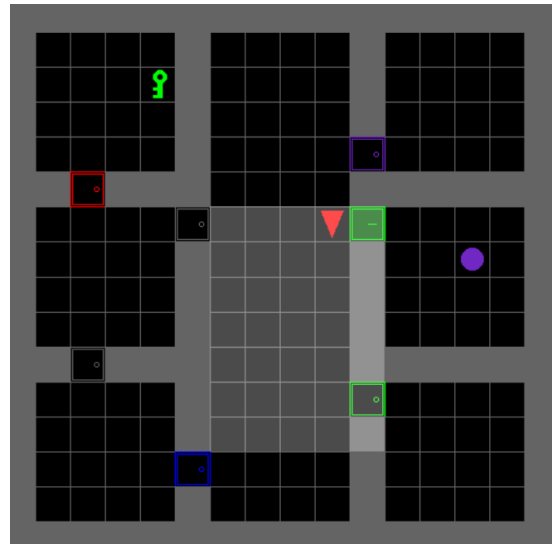
- $\mathcal{S}_{s,a} := \{s' \in \mathcal{S} : P(s'|s,a) > 0\}$
- $U := \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} |\mathcal{S}_{s,a}|$



Limitations of the Linear Setting

Directly reachable states:

- $\mathcal{S}_{s,a} := \{s' \in \mathcal{S} : P(s'|s,a) > 0\}$
- $U := \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} |\mathcal{S}_{s,a}|$

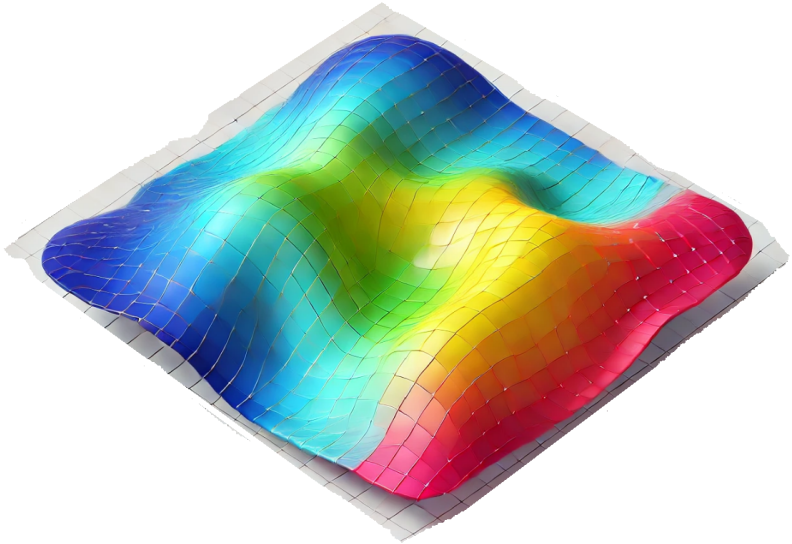


Theorem Lee and Oh [2024] For an MDP with a finite state space, the feature dimension d is lower bounded by

$$d \geq \lfloor \frac{|\mathcal{S}|}{U} \rfloor$$

Limitations of the Linear Setting

High dimensional problems



Nonlinear problems

Kernel-Based Setting

Kernel-Based Setting

- Kernel-based models are natural extensions of linear models to infinite dimensional feature maps

Kernel-Based Setting

- Kernel-based models are natural extensions of linear models to infinite dimensional feature maps
- Allow for versatile and powerful non-linear function approximation

Kernel-Based Setting

- Kernel-based models are natural extensions of linear models to infinite dimensional feature maps
- Allow for versatile and powerful non-linear function approximation
- Lend themselves to analysis

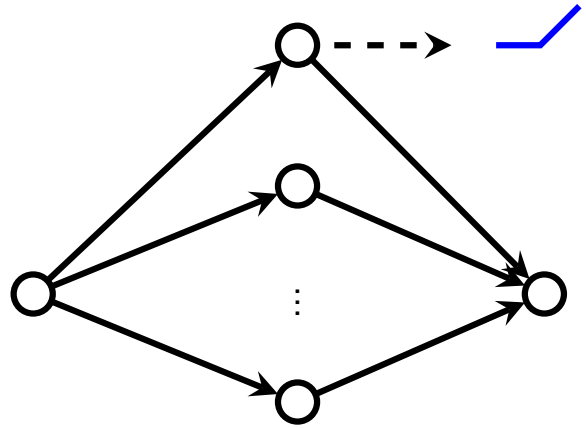
Kernel-Based Setting

- Kernel-based models are natural extensions of linear models to infinite dimensional feature maps
- Allow for versatile and powerful non-linear function approximation
- Lend themselves to analysis
- Serve as an intermediate step towards analysis of NN-based models

Kernel-Based Setting

- Kernel-based models are natural extensions of linear models to infinite dimensional feature maps
- Allow for versatile and powerful non-linear function approximation
- Lend themselves to analysis
- Serve as an intermediate step towards analysis of NN-based models

Tabular $\rightarrow$ Linear $\rightarrow$ Kernel-Based $\rightarrow$ NN-Based



Kernel-Based Setting

Function class:

$$\mathcal{F} = \left\{ f : \mathbb{R}^d \to \mathbb{R},\ f(\cdot) = \sum_{m=1}^{\infty} w_m \phi_m(\cdot) \right\}$$

Kernel-Based Setting

Function class:

$$\mathcal{F} = \left\{ f : \mathbb{R}^d \to \mathbb{R},\ f(\cdot) = \sum_{m=1}^{\infty} w_m \phi_m(\cdot) \right\}$$

An extension of linear models to infinite dimensions in the feature space ϕ

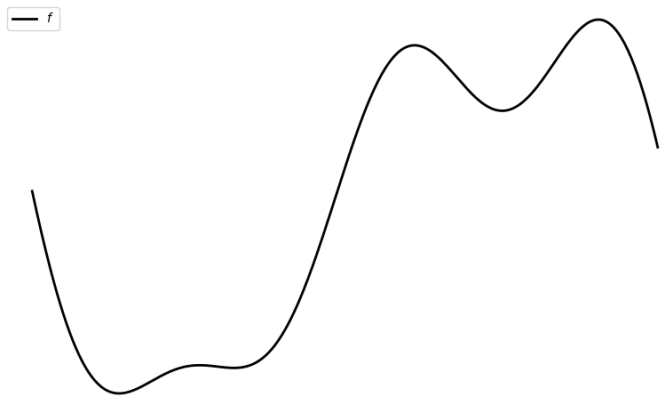
Kernel-Based Setting

Function class:

$$\mathcal{F} = \left\{ f : \mathbb{R}^d \to \mathbb{R},\ f(\cdot) = \sum_{m=1}^{\infty} w_m \phi_m(\cdot) \right\}$$

An extension of linear models to infinite dimensions in the feature space ϕ

Nonlinear functions in $\mathbb{R}^d$



Mercer Theorem

A positive definite kernel $\kappa : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$

Mercer Theorem

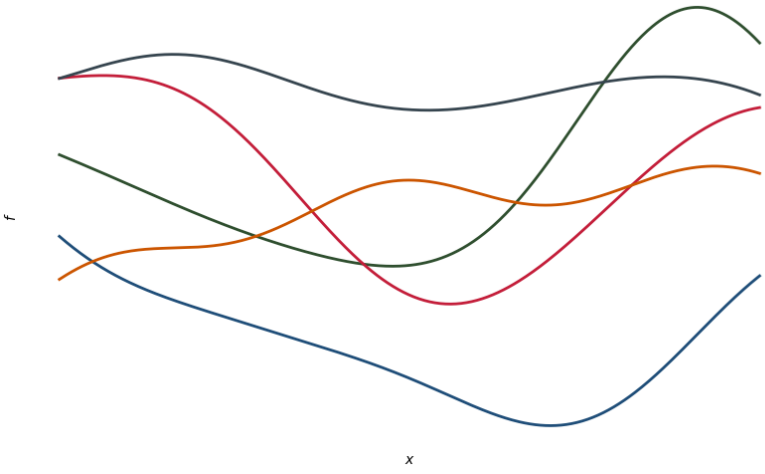
A positive definite kernel $\kappa : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$

Theorem Any positive definite kernel can be written as

$$\kappa(z, z') = \sum_{m=1}^{\infty} \lambda_m \varphi_m(z) \varphi_m(z')$$

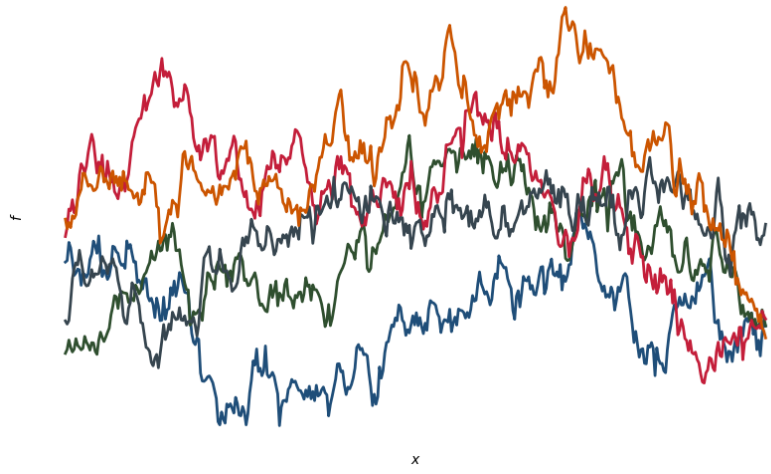
- The feature map $\phi_m(\cdot) = \lambda_m^{\frac{1}{2}} \varphi_m(\cdot)$ corresponding to κ
- λ_m are referred to as eigenvalues
- φ_m are referred to as eigenfunctions

Kernels



Squared Exponential kernel

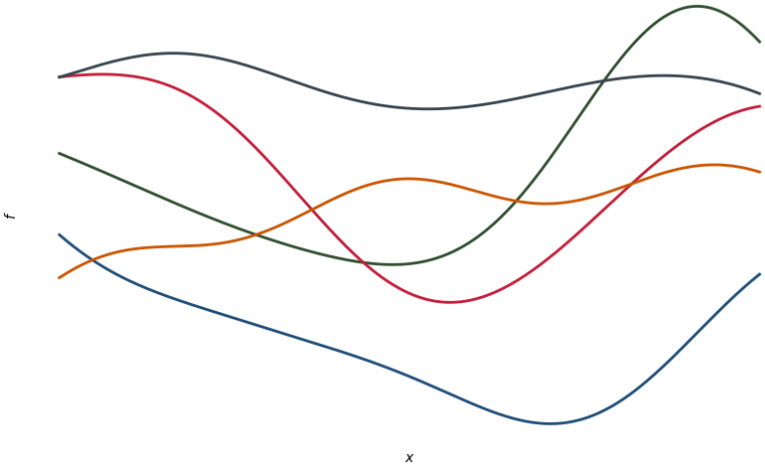
$$\kappa(z, z') = \exp\left(-\frac{\|z-z'\|^2}{2\ell^2}\right)$$

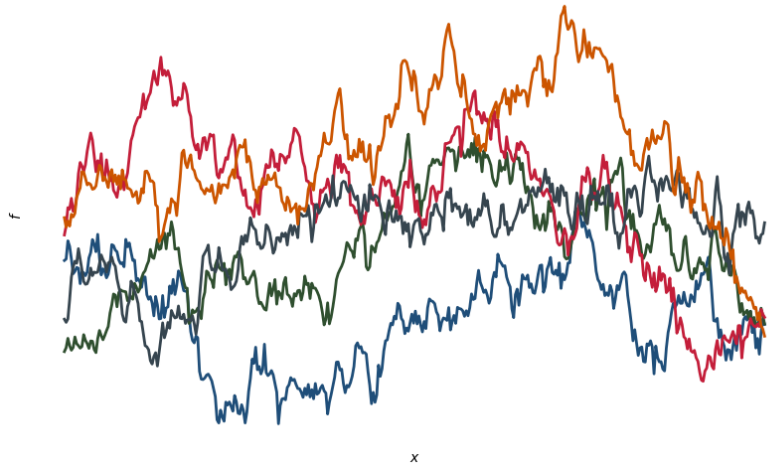


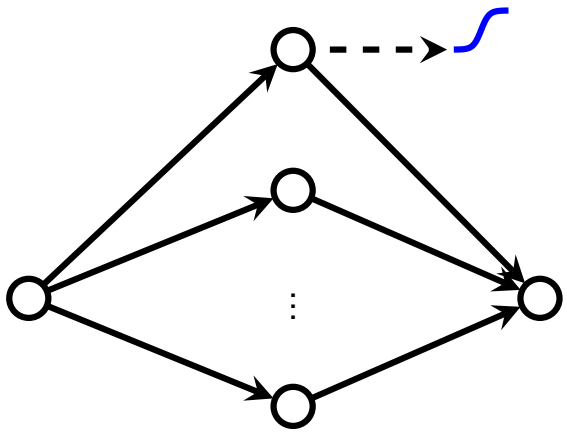
Matérn-ν kernel

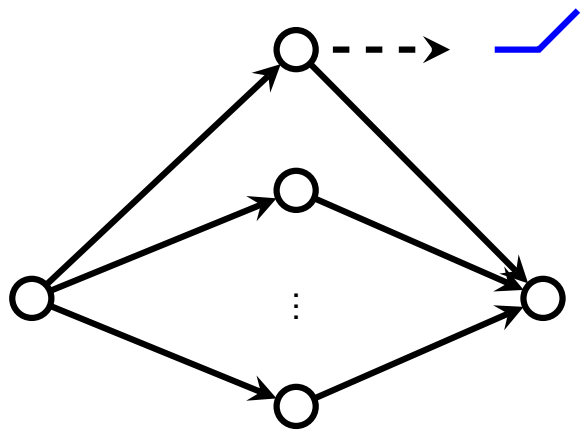
$$\kappa(z, z') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{\ell}\|z - z'\|\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}\|z - z'\|\right)$$

Kernels









Reproducing Kernel Hilbert Space

RKHS:

$$\mathcal{H}_\kappa = \{f(\cdot) = \sum_{m=1}^{\infty} w_m \phi_m(\cdot)\}$$

- Inner product $\langle f, g \rangle_k = \boldsymbol{w}_f^\top \boldsymbol{w}_g$
- $\|f\|_{\mathcal{H}_\kappa} = \|\mathbf{w}\|$
- $\phi_m = \sqrt{\lambda_m}\varphi_m$ form an orthonormal basis

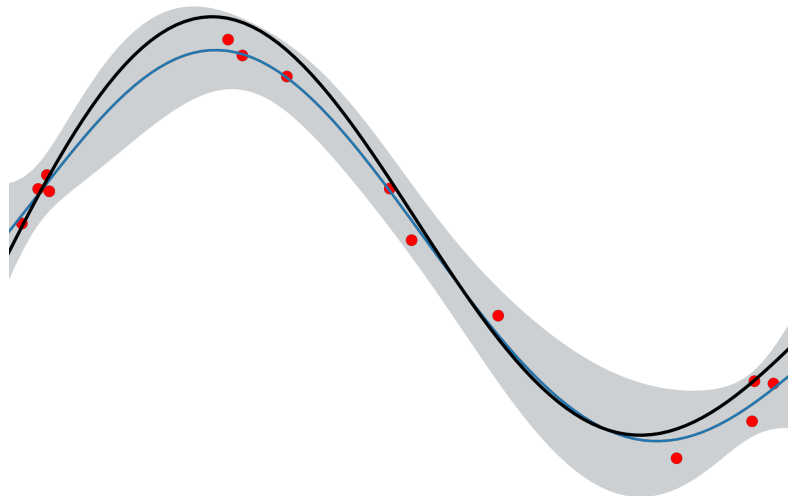
Kernel Based Regression

Provided a dataset of t observation:

$$\left\{(z_j, Y(z_j))\right\}_{j=1}^{t}, \; Y(z_j) = f(z_j) + \varepsilon_j$$

Regularized Least Squares Error:

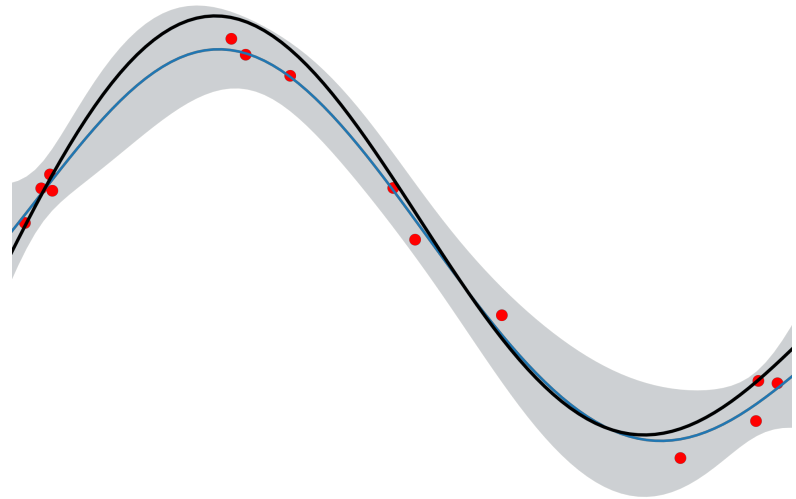
$$\hat{f} = \; \arg\min_{g \in \mathcal{H}_\kappa} \sum_{j=1}^{t} (Y(z_j) - g(z_j)) + \lambda \|g\|^2_{\mathcal{H}_\kappa}$$



Kernel-Based Regression

Predictor:

$$\hat{f}(z) = \boldsymbol{\kappa}_t^\top(z)(\mathbf{K}_t + \lambda I)^{-1}\mathbf{y}_t$$

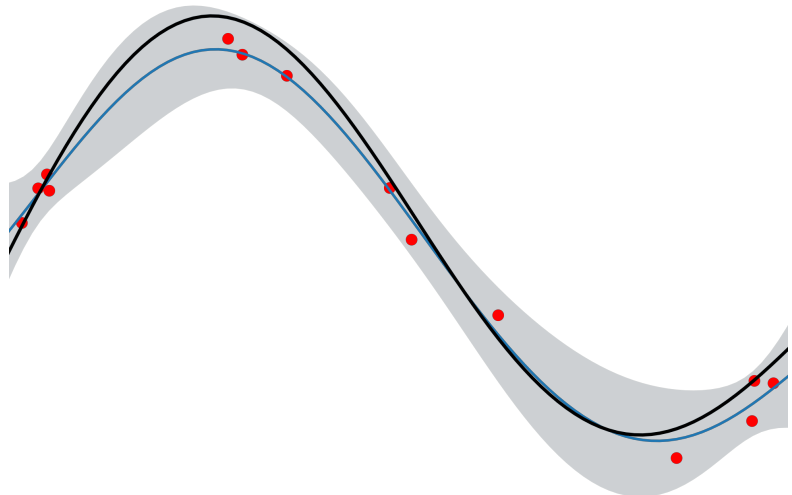


- $\boldsymbol{\kappa}_t(z) = [k(z_1, z), k(z_2, z), \cdots, k(z_t, z)]$
- $\mathbf{K}_t = [k(z_i, z_j)]_{i,j=1}^t$
- $\mathbf{y}_t = [Y(z_1), Y(z_2), \cdots, Y(z_t)]$

Kernel-Based Regression

Uncertainty estimator:

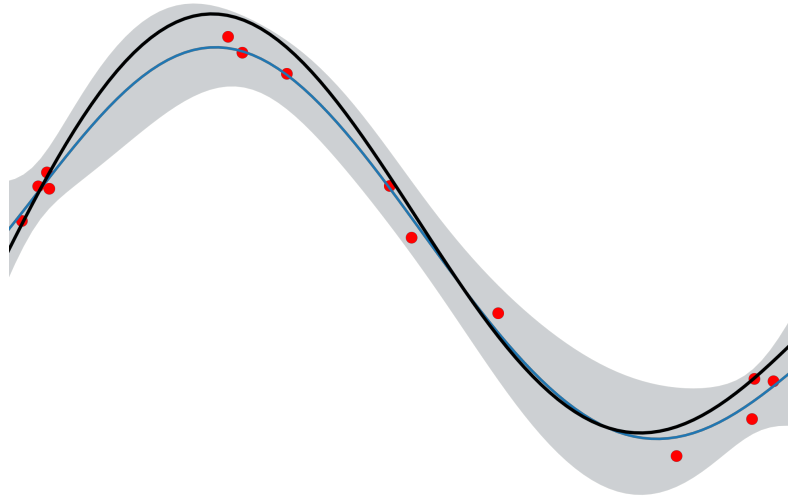
$$(\sigma_t(z))^2 = \kappa(z,z) - \boldsymbol{\kappa}_t^\top(z)(\mathbf{K}_t + \lambda I)^{-1}\boldsymbol{\kappa}_t(z)$$



Kernel-Based Regression

Uncertainty estimator:

$$(\sigma_t(z))^2 = \kappa(z,z) - \boldsymbol{\kappa}_t^\top(z)(\mathbf{K}_t + \lambda I)^{-1}\boldsymbol{\kappa}_t(z)$$



Closed from expressions for prediction and uncertainty quantification!

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

Want: Find an ϵ-optimal policy with a computational and sample complexity polynomial in $1/\epsilon$ and H

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

Want: Find an ϵ-optimal policy with a computational and sample complexity polynomial in $1/\epsilon$ and H

—possibly some kernel parameters—

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

Want: Find an ϵ-optimal policy with a computational and sample complexity polynomial in $1/\epsilon$ and H

—possibly some kernel parameters—

independently of S and A.

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

Want: Find an ϵ-optimal policy with a computational and sample complexity polynomial in $1/\epsilon$ and H

—possibly some kernel parameters—

independently of S and A.

$$\text{Sample complexity: } \tilde{\mathcal{O}}\left((\tfrac{1}{\epsilon})^2\right)$$

RL with kernel-based function approximation

IDEA: Approximate the Q-functions as a function in RKHS

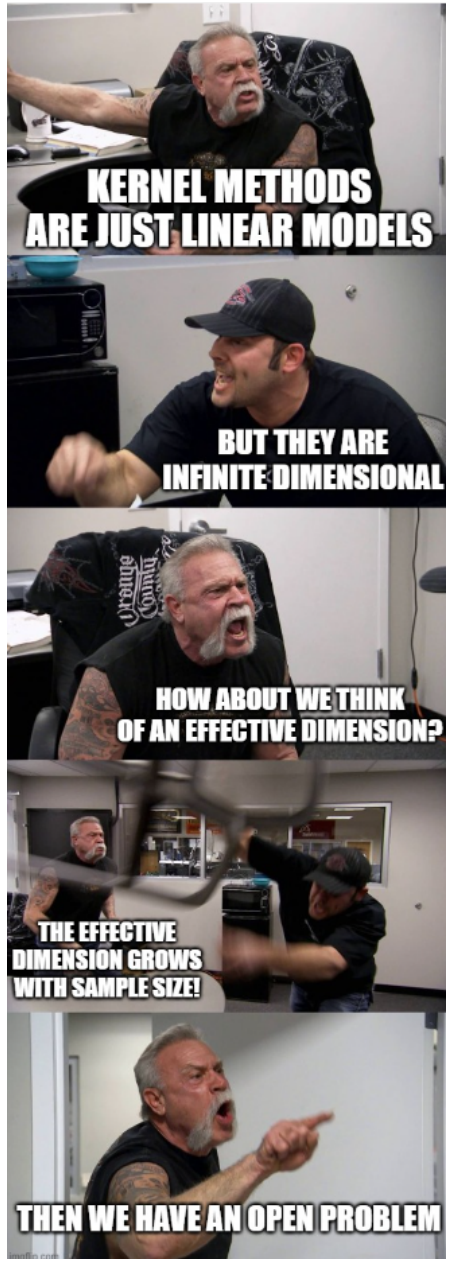
Want: Find an ϵ-optimal policy with a computational and sample complexity polynomial in $1/\epsilon$ and H

—possibly some kernel parameters—

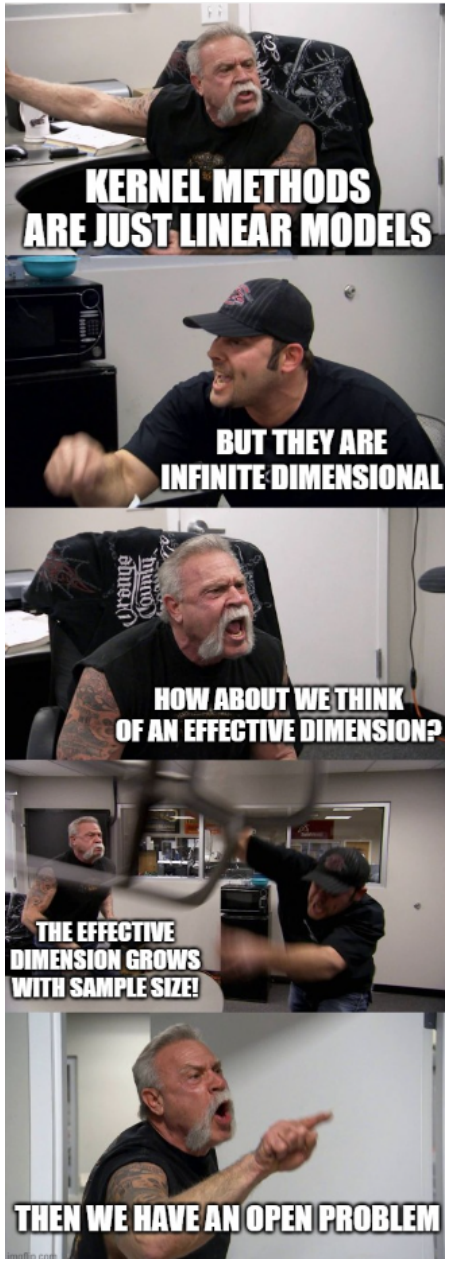
independently of S and A.

$$\text{Sample complexity: } \tilde{\mathcal{O}}\left(\left(\frac{1}{\epsilon}\right)^{?}\right)$$

RL with kernel-based function approximation



RL with kernel-based function approximation



Effective Dimension:

$$\Big[\underbrace{\phi_1, \phi_2, \cdots, \phi_{\mathfrak{D}}}_{\mathfrak{D}\text{dimension}}, \ \phi_{\mathfrak{D}+1}, \cdots \Big]$$

$$\mathfrak{D} \approx \tfrac{1}{2} \log\det(I + \tfrac{1}{\lambda}\mathbf{K}_t)$$

- In the linear setting: $\mathfrak{D} \approx d$
- For Squared Exponential kernel: $\mathfrak{D} \approx \text{poly}\log(T)$
- For Matérn kernel: $\mathfrak{D} \approx T^{\frac{d}{d+\nu}}$ [Vakili et al., 2021]

RL with Kernel-based FA

Kernel-based transition assumption

For all s': $P(s'|s,a) \in \mathcal{H}_\kappa$

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s' \text{: } P(s'|s,a) \in \mathcal{H}_\kappa$$

- A significant generalization of linear models

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s' \text{: } P(s'|s,a) \in \mathcal{H}_\kappa$$

- A significant generalization of linear models
- Linear model is a special case with linear kernel: $\kappa(s,a,s',a') = \boldsymbol{\phi}^\top(s,a)\boldsymbol{\phi}(s',a')$

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s' \text{: } P(s'|s,a) \in \mathcal{H}_\kappa$$

- A significant generalization of linear models
- Linear model is a special case with linear kernel:
  $\kappa(s,a,s',a') = \boldsymbol{\phi}^\top(s,a)\boldsymbol{\phi}(s',a')$
- RKHS of common kernels can approximate almost all continuous functions

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s' \text{: } P(s'|s,a) \in \mathcal{H}_\kappa$$

- A significant generalization of linear models
- Linear model is a special case with linear kernel:
  $\kappa(s,a,s',a') = \boldsymbol{\phi}^\top(s,a)\boldsymbol{\phi}(s',a')$
- RKHS of common kernels can approximate almost all continuous functions

  For integrable $V : \mathcal{S} \to \mathbb{R}$, $[PV] = \int_{s'} P(s'|s,a)V(s') \in \mathcal{H}_k$

Optimistic approximate DP goes kernelized

IDEA: Combine the techniques for tabular MDPs with exploration bonuses borrowed from the ~~linear~~ kernel bandit literature!

UCB-VI [Azar et al., 2017]:

1. Backtrack $h = H, H-1, \ldots, 1$: run optimistic value iteration

$$Q_h = r_h + \underbrace{\widehat{P}_h}_{\text{model estimate}} V_{h+1} + \underbrace{b_h}_{\text{exploration bonus}}$$

   and set $V_h(s) = \max_a Q_h(s,a)$ for all s, a.

2. Forward $h = 1, 2, \ldots, H$: take actions according to greedy policy

$$\pi_h(s) = \arg\max_a Q_h(s,a).$$

But how do we define $\widehat{P}_h$ and b_h?

Kernel-based optimistic value iteration (KOVI)

Transition model $\widehat{P_h}$ can be defined implicitly via least-squares:

- Solve the regularized linear regression problem

$$\hat{f}_h = \arg\min_{f \in \mathcal{H}_\kappa} \sum_{t=1}^{k} (V_{h+1}(s_{h+1,t}) - f(s_{h,t}, a_{h,t})^2 + \lambda \|f\|^2_{\mathcal{H}_\kappa}$$

- That provides a prediction

$$[\widehat{P_h V_{h+1,k}}](s,a) = \widehat{f}_h(s,a) = \boldsymbol{\kappa}_{h,k}^\top(s,a)(\mathbf{K}_{h,k} + \lambda I)^{-1}\mathbf{v}_{h,k}$$
$$\mathbf{v}_{h,k} = [V_{h+1}(s_{h+1,1}), V_{h+1}(s_{h+1,2}), \cdots, V_{h+1}(s_{h+1,k})]$$

- Also, an uncertainty quantification (variance)

$$\sigma^2_{h,k}(s,a) = \kappa\big((s,a),(s,a)\big) - \boldsymbol{\kappa}_{h,k}^\top(s,a)(\mathbf{K}_{h,k} + \lambda I)^{-1}\boldsymbol{\kappa}_{h,k}(s,a)$$

Kernel-based optimistic value iteration (KOVI)

The prediction and variance give us an upper confidence bound on $Q^\star$:

$$Q_{h,k}(s,a) = r_h(s,a) + \widehat{[P_h V_{h+1}]}(s,a) + \beta(\delta)\sigma_{h,k}(s,a)$$

This is then used to compute an UCB on $V^\star$ as

$$V_{h,k}(s) = \max_a Q_{h,k}(s,a)$$

Performance guarantees

Theorem [Yang et al., 2020] The regret of KOVI satisfies $\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{(\mathfrak{D}^2 + \mathfrak{D}\log\mathcal{N}_\epsilon)K})$.

Performance guarantees

Theorem [Yang et al., 2020] The regret of KOVI satisfies
$\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{(\mathfrak{D}^2 + \mathfrak{D}\log\mathcal{N}_\epsilon)K})$.

Proof ideas:

- Prove confidence bounds

$$|\widehat{[P_h V_{h+1,k}]}(s,a) - [P_h V_{h+1,k}](s,a)| \le \beta(\delta)\sigma_{h,k}(s,a).$$

- Using standard techniques, show

$$\text{Regret}(K) \lesssim \textstyle\sum_{h,k} \beta(\delta)\sigma_{h,k}(s_{h,k}, a_{h,k}).$$

- Kernelized elliptical potential lemma (e.g., Srinivas et al., 2010)

$$\textstyle\sum_{h,k} \sigma_{h,k}(s_{h,k}, a_{h,k}) \lesssim H\sqrt{K\mathfrak{D}\log(K)}.$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

- For a fixed $f \in \mathcal{H}_\kappa$ with non-adaptive inputs $z_1, \ldots, z_k$,

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{d\log(\tfrac{T}{\delta})}$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

- For a fixed $f \in \mathcal{H}_\kappa$ with non-adaptive inputs $z_1, \ldots, z_k$,

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{d\log(\tfrac{T}{\delta})}$$

Challenge 1: Inputs $(s_1, a_1), \ldots, (s_k, a_k)$ are adaptive!

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

- For a fixed $f \in \mathcal{H}_\kappa$ with non-adaptive inputs $z_1, \ldots, z_k$,

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{d\log(\tfrac{T}{\delta})}$$

Challenge 1: Inputs $(s_1, a_1), \ldots, (s_k, a_k)$ are adaptive!

Solution: Self-normalized concentration inequalities for vector-valued martingales extended to kernel setting [Abbasi-Yadkori, 2013, Whitehouse et al., 2023]:

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log(\tfrac{1}{\delta})}$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

- For a fixed $f \in \mathcal{H}_\kappa$ with non-adaptive inputs $z_1, \ldots, z_k$,

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{d\log(\tfrac{T}{\delta})}$$

Challenge 1: Inputs $(s_1,a_1),\ldots,(s_k,a_k)$ are adaptive!

Solution: Self-normalized concentration inequalities for vector-valued martingales extended to kernel setting [Abbasi-Yadkori, 2013, Whitehouse et al., 2023]:

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log(\tfrac{1}{\delta})}$$

Challenge 2: $f = P_h V_{h+1,k}$ is not fixed, but depends on past data!

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\left|\hat{f}(s,a) - [P_h V_{h+1,k}](s,a)\right| \leq \beta(\delta)\sigma_h(s,a).$$

- For a fixed $f \in \mathcal{H}_\kappa$ with non-adaptive inputs $z_1, \ldots, z_k$,

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{d\log(\tfrac{T}{\delta})}$$

Challenge 1: Inputs $(s_1,a_1),\ldots,(s_k,a_k)$ are adaptive!

Solution: Self-normalized concentration inequalities for vector-valued martingales extended to kernel setting [Abbasi-Yadkori, 2013, Whitehouse et al., 2023]:

$$\beta(\delta) \approx \|f\|_{\mathcal{H}_\kappa} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log(\tfrac{1}{\delta})}$$

Challenge 2: $f = P_h V_{h+1,k}$ is not fixed, but depends on past data!

Solution: Covering number argument

Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

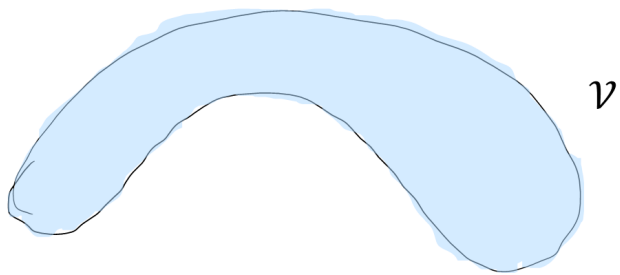
- How many functions V are required to cover $\mathcal{V}$ up to ϵ error?

Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

- How many functions V are required to cover $\mathcal{V}$ up to ϵ error?

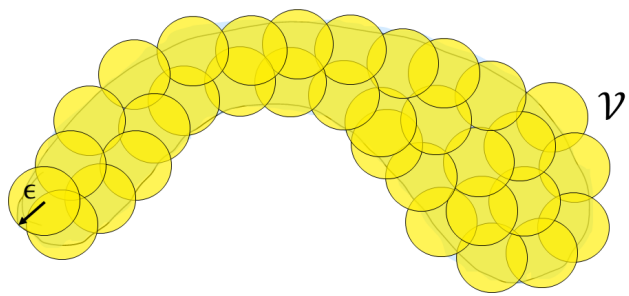


Covering Number Argument

- Notice that all value functions $V_{h,k}$ belong to the function class

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

- How many functions V are required to cover $\mathcal{V}$ up to ϵ error?



Covering Number Argument

- We can now use a union-bound argument

$$\beta(\delta) = \overline{\beta}(\delta/\mathcal{N}_\epsilon) \approx \|f\|_{\mathcal{H}_k} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log \mathcal{N}_\epsilon + \frac{1}{\delta}}$$

Covering Number Argument

- We can now use a union-bound argument

$$\beta(\delta) = \overline{\beta}(\delta/\mathcal{N}_\epsilon) \approx \|f\|_{\mathcal{H}_k} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log \mathcal{N}_\epsilon + \frac{1}{\delta}}$$

- Regret ($\epsilon \approx \frac{1}{K}$) [Yang et al., 2020]

$$\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{\mathfrak{D}^2 K + \mathfrak{D}\log \mathcal{N}_\epsilon K})$$

Covering Number Argument

- We can now use a union-bound argument

$$\beta(\delta) = \overline{\beta}(\delta/\mathcal{N}_\epsilon) \approx \|f\|_{\mathcal{H}_k} + \frac{H}{\sqrt{\lambda}}\sqrt{\mathfrak{D} + \log \mathcal{N}_\epsilon + \frac{1}{\delta}}$$

- Regret ($\epsilon \approx \frac{1}{K}$) [Yang et al., 2020]

$$\text{Regret}(K) = \tilde{\mathcal{O}}(H^2\sqrt{\mathfrak{D}^2 K + \mathfrak{D}\log \mathcal{N}_\epsilon K})$$

- Sample Complexity
  - Very smooth kernels $\mathfrak{D}$ and $\log(\mathcal{N}_\epsilon) \approx \text{poly}\log(K)$

$$\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right)$$

  - In general could be vacuous!

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$$\mathcal{V} \in \mathcal{H}_{\kappa'}$$

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$$\mathcal{V} \in \mathcal{H}_{\kappa'}$$

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

Idea: Leverage kernel mean embedding

$$\text{Regret}(K) = \tilde{\mathcal{O}}(H^2 \mathfrak{D} \sqrt{K})$$

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$$\mathcal{V} \in \mathcal{H}_{\kappa'}$$

$$\mathcal{V} = \left\{ V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \right\}$$

Idea: Leverage kernel mean embedding

$$\text{Regret}(K) = \tilde{\mathcal{O}}(H^2 \mathfrak{D} \sqrt{K})$$

Doen not hold in the linear setting!

Open Problem Vakili [2024]

(a) Can a no-regret learning algorithm be designed?

(b) What is the minimum regret growth rate with K (and also H)? And, can a learning algorithm be designed to achieve order optimal (or near-optimal) regret performance, closely aligning with the established lower bound?

Some References

- Chowdhury and Gopalan [2019]
- Yang et al. [2020]
- Domingues et al. [2021]
- Vakili and Olkhovskaya [2023]
- ...

References I

Y. Abbasi-Yadkori. Online learning for linearly parametrized control problems. *PhD Thesis, University of Alberta*, 2013.

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24, 2011.

A. Ayoub, Z. Jia, C. Szepesvari, M. Wang, and L. Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pages 263–272. PMLR, 2017.

S. R. Chowdhury and A. Gopalan. Online learning in kernelized Markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3197–3205. PMLR, 2019.

S. R. Chowdhury and R. Oliveira. Value function approximations via kernel embeddings for no-regret reinforcement learning. In *Asian Conference on Machine Learning*, pages 249–264. PMLR, 2023.

O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko. Kernel-based reinforcement learning: A finite-time analysis. In *International Conference on Machine Learning*, pages 2783–2792. PMLR, 2021.

References II

S. Du, S. Kakade, J. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in RL. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

C. Jin, Q. Liu, and S. Miryoosefi. Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. *Mathematics of Operations Research*, 48(3):1496–1521, 2023.

T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

J. Lee and M.-h. Oh. Demystifying linear mdps and novel dynamics aggregation framework. In *The Twelfth International Conference on Learning Representations*, 2024.

References III

A. Modi, J. Chen, A. Krishnamurthy, N. Jiang, and A. Agarwal. Model-free representation learning and exploration in low-rank mdps. *Journal of Machine Learning Research*, 25 (6):1–76, 2024.

A. Moulin and G. Neu. Optimistic planning by regularized dynamic programming. In *International Conference on Machine Learning*, pages 25337–25357. PMLR, 2023.

G. Neu and C. Pike-Burke. A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1392–1403, 2020.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

S. Vakili. Open problem: Order optimal regret bounds for kernel-based reinforcement learning. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 5340–5344. PMLR, 2024.

S. Vakili and J. Olkhovskaya. Kernelized reinforcement learning with order optimal regret bounds. *Advances in Neural Information Processing Systems*, 36, 2023.

S. Vakili, K. Khezeli, and V. Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.

References IV

G. Weisz, P. Amortila, and C. Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR, 2021.

G. Weisz, A. György, T. Kozuno, and C. Szepesvári. Confident approximate policy iteration for efficient local planning in q^π-realizable mdps. *Advances in Neural Information Processing Systems*, 35:25547–25559, 2022.

G. Weisz, A. György, and C. Szepesvári. Online RL in linearly q^π-realizable MDPs is as easy as in linear MDPs if you learn what to ignore. *Advances in Neural Information Processing Systems*, 36, 2023.

J. Whitehouse, A. Ramdas, and S. Z. Wu. On the sublinear regret of gp-ucb. *Advances in Neural Information Processing Systems*, 36, 2023.

L. Yang and M. Wang. Sample-optimal parametric q-learning using linearly additive features. In *International conference on machine learning*, pages 6995–7004. PMLR, 2019.

L. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

References V

Z. Yang, C. Jin, Z. Wang, M. Wang, and M. Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. *Advances in Neural Information Processing Systems*, 33:13903–13916, 2020.

A. Zanette, D. Brandfonbrener, E. Brunskill, M. Pirotta, and A. Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964. PMLR, 2020a.

A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill. Learning near optimal policies with low inherent Bellman error. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10978–10989. PMLR, 13–18 Jul 2020b.

D. Zhou, J. He, and Q. Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802. PMLR, 2021.