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]

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

- Sample complexity:

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

Tabular Setting

Recall results for the tabular setting:

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

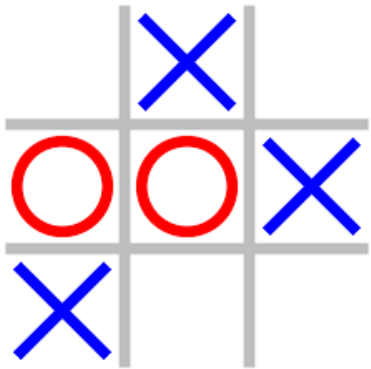
$$\text{Regret}(T) = O(\sqrt{H^3SAT})$$

- Sample complexity:

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

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

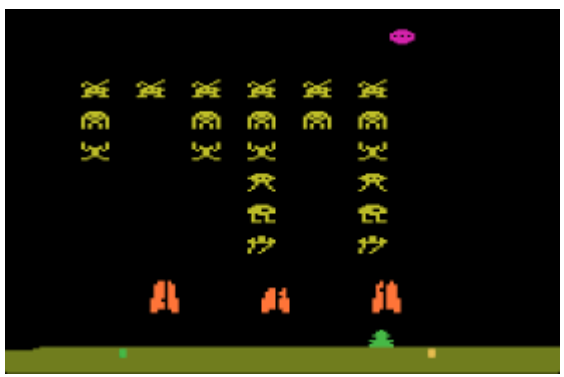
Why "Tabular"?



- 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 F.
- Hope that F captures the MDP structure appropriately and leverage the information in structure of F to learn faster if possible.
- Typical function classes: Linear, Kernel-based, NN-based

Tabular $\rightarrow$ Linear $\rightarrow$ 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

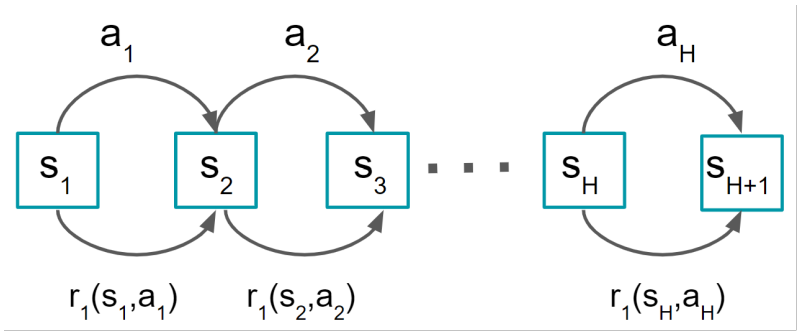
- Generative oracle, Offline, Online
- Episodic, Infinite horizon (discounted)
- Model-based, Model-free

In this part we focus on:

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

Setting

For a clear and sharp presentation we focus on episodic MDPs

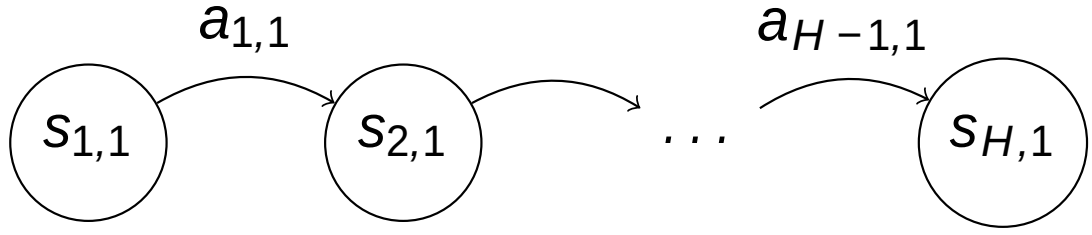


For simplicity, we assume r is known and deterministic

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

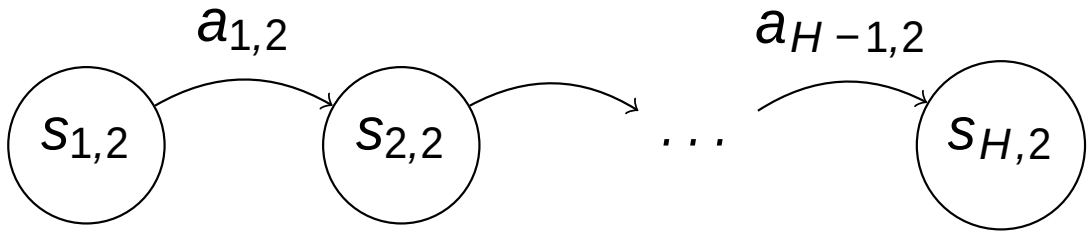
Episodic MDP

Episode 1:

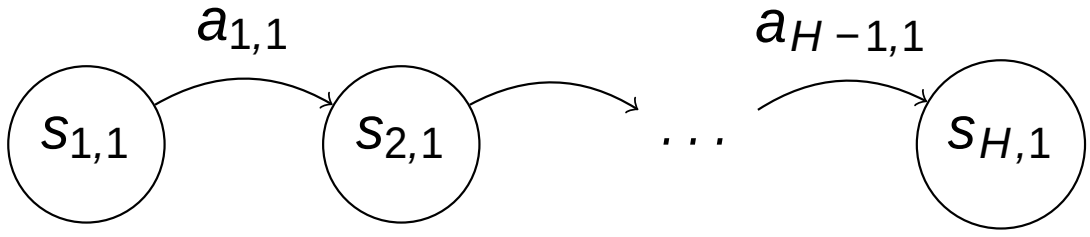


Episodic MDP

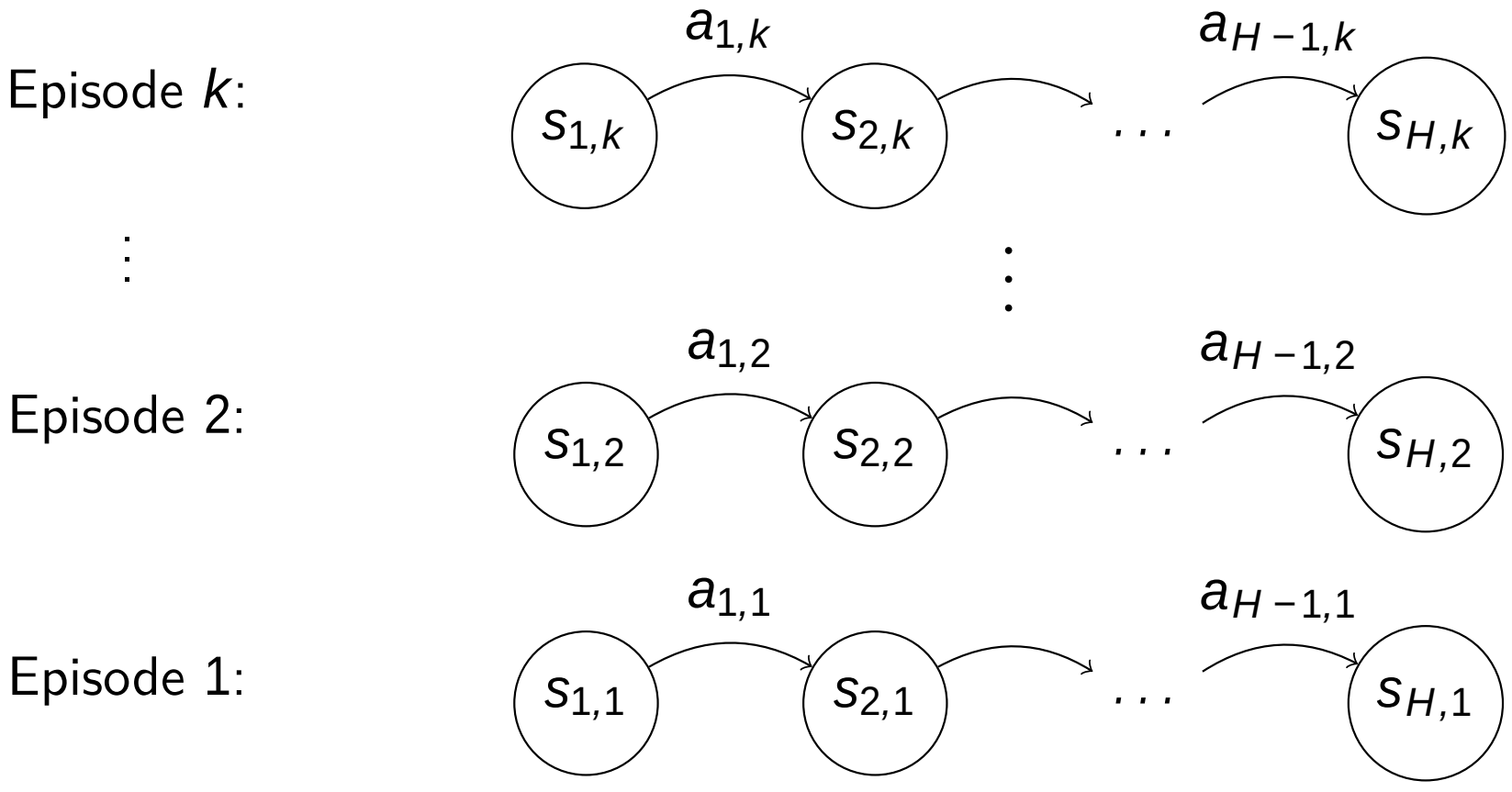
Episode 2:



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 $\quad : S \times A \to \mathbb{R}^d$.

Linear Function Approximation

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

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

We need to find a parameter vector such that $Q \approx \Phi$.
(Meaning that $Q\ (s, a) \approx$, (s, a) .)

Linear Function Approximation

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

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

We need to find a parameter vector such that $Q \approx \Phi$.
(Meaning that $Q\ (s, a) \approx$, (s, a) .)

QUESTION: When can we learn efficiently?

Linear Function Approximation

Various conditions on the feature map Φ have been studied:

- Linear Q : there exists a θ such that $Q = \Phi\theta$.
- Linear Q^π: for every policy π, there exists a θ^π such that $Q^\pi = \Phi\theta^\pi$.
- Closure under Bellman operator: for any $Q_\theta = \Phi\theta$, $TQ_\theta \in \text{span}(\Phi)$.
- 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 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 -optimal policy with a computational and sample complexity polynomial in d, $1/$ and H, independently of S and A.

What can we hope for?

WANT: nd an -optimal policy with a computational and sample complexity polynomial ind, 1= and H , independently of S and A .

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

What can we hope for?

WANT: nd an -optimal policy with a computational and sample complexity polynomial ind, 1= and H , independently of S and A .

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

Polynomial sample complexity is possible when relaxing the condition to linearQ^π -realizability, but no practical algorithms are known [Weisz et al., 2022, 2023].

What can we hope for?

WANT: nd an -optimal policy with a computational and sample complexity polynomial ind, 1= and H , independently of S and A .

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

Polynomial sample complexity is possible when relaxing the condition to linearQ -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: nd an -optimal policy with a computational and sample complexity polynomial ind, 1= and H , independently of S and A .

- This is impossible when only requiring linear Q^π-realizability! [Weisz et al., 2021]
- Polynomial sample complexity is possible when relaxing the condition to linear$Q^\star$ -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 e ciency!!! [Jin et al., 2023]

Linear MDPs

Linear transition function:

$$P_h(\,\cdot\,|s,a) = \langle \psi_h(s,a), \phi_h(\,\cdot\,)\rangle,$$

where $\phi_h(\,\cdot\,) = [\phi_h^1(\,\cdot\,), \ldots, \phi_h^d(\,\cdot\,)]$ is a d-dimensional signed measure.

Linear rewards: $r_h(s,a) = \langle \psi_h(s,a), \theta_h\rangle$.

Linear MDPs

Linear transition function:

$$P_h(\,\cdot\,|s,a) = \langle \phi_h(s,a), \mu_h(\,\cdot\,)\rangle,$$

where $\mu_h(\,\cdot\,) = [\mu_h^1(\,\cdot\,), \ldots, \mu_h^d(\,\cdot\,)]$ is a d-dimensional signed measure.

Linear rewards: $r_h(s,a) = \langle \phi_h(s,a), \theta_h\rangle$.

In matrix notation:

Transition operator $P_h \in \mathbb{R}^{(SA)\times S}$ can be written as $P_h = \Phi M_h$ for some "matrix" $M_h \in \mathbb{R}^{S\times d}$.

Reward function can be written as $r_h = \Phi\theta_h$ for some $\theta_h \in \mathbb{R}^d$.

Tabular MDPs are linear

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

Let $(s; a) = e_{(s;a)}$ be the canonical basis in $\mathbb{R}^d$

$P_h(\,\cdot\,|s; a) = e_{s;a}^{\top} \quad _h(\,\cdot\,)$

A magical property of linear MDPs

In a linear MDP, theQ-functions of all policies are linear in:

$$Q_h = r_h + P_h V_{h+1} = \#_h + M_h V_{h+1}$$
$$= \#_h + M_h V_{h+1} = \_h;$$

with $\_h = \#_h + M_h V_{h+1}$.

A magical property of linear MDPs

In a linear MDP, theQ-functions of all policies are linear in:

$$Q_h = r_h + P_h V_{h+1} = \#_h + M_h V_{h+1}$$
$$= \#_h + M_h V_{h+1} = {}_h;$$

with ${}_h = \#_h + M_h V_{h+1}$.

This implies linear$Q^?$-realizability, linear$Q$ -realizability, Bellman compleness, and many more useful properties for analysis! E.g., note that for any function$u \in R^S$, $P_h u = M_h u$ is linear in .

A magical property of linear MDPs

In a linear MDP, theQ-functions of all policies are linear in:

$$Q_h = r_h + P_h V_{h+1} = \#_h + M_h V_{h+1}$$
$$= \#_h + M_h V_{h+1} = {}_h;$$

with ${}_h = \#_h + M_h V_{h+1}$.

This implies linear$Q^?$-realizability, linear$Q$ -realizability, Bellman compleness, and many more useful properties for analysis! E.g., note that for any function$u \in R^S$, $P_h u = M_h u$ is linear in .

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

Backtrack $h = H; H \quad 1; : : : ; 1$: run optimistic value iteration

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

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

Forward $h = 1; 2; : : : ; H$: take actions according to greedy policy

$$_h(s) = \arg\max_a Q_h(s; a):$$

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]:

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

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

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

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 $\hat{P}_h$ 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

$$\widehat{w}_{h;k} = \arg\min_w \sum_{t=1}^{k} (V_{h+1;k}(s_{h+1;t}) - \langle \phi_h(s_{h;t}; a_{h;t}); w \rangle)^2 + \lambda \|w\|^2$$

- That provides a prediction

$$[\widehat{P}_h V_{h+1;k}](s; a) = \langle \phi_h(s; a); \widehat{w}_{h;k} \rangle$$

- Also, an uncertainty quantification (variance)

$$\sigma^2_{h;k}(s; a) = \|\phi_h(s; a)\|^2_{(\Lambda_{h;k} + \lambda I)^{-1}}$$

$$\Lambda_{h;k} = \sum_{t=1}^{k} \phi_h(s_{h;t}; a_{h;t}) \phi_h(s_{h;t}; a_{h;t})^\top$$

LSVI-UCB

The prediction and variance give us an upper con dence bound on $Q^?$:

$$Q_{h;k}(s;a) = r_h(s;a) + [\hat{P}_h V_{h+1}](s;a) + \quad (\ )_{h;k}(s;a)$$

This is then used to compute an UCB on $V^?$ 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 $\text{Regret}(K) = \tilde{O}(H^2\sqrt{d^3K})$.

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

Performance guarantees

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

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

Proof ideas:

Prove confidence bounds

$$|[\hat{P}_h V_{h+1,k}](s,a) - [P_h V_{h+1,k}](s,a)| \leq \Gamma_{h,k}(s,a).$$

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

$$\text{Regret}(K) \lesssim \sum_{h,k} \Gamma_{h,k}(s_{h,k}, a_{h,k}).$$

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

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

Proving the con dence bounds

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

$$[\hat{P}_h u](s,a) \quad [P_h u](s,a) \quad \bar{\ }(\ )\ _{h,k}(s,a)$$

for some

$$\bar{\ }(\ ) \quad \tfrac{1}{2}\|M_h u\| + H \sqrt{\frac{d \log(\frac{K}{\quad})}{\quad}}$$

Proving the con dence bounds

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

$$[\hat{P}_h u](s,a) - [P_h u](s,a) \le \bar{\ }(\ )\ _{h,k}(s,a)$$

for some

$$\bar{\ }(\ ) \quad \tfrac{1}{2}\|M_h u\| + H\sqrt{d \log(\tfrac{K}{\ })}$$

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

Proving the con dence bounds

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

$$[\hat{P}_h u](s,a) - [P_h u](s,a) \le \bar{\beta}(\ )\ \sigma_{h,k}(s,a)$$

for some

$$\bar{\beta}(\ ) \le \tfrac{1}{2}\|M_h u\| + H\sqrt{d \log(\tfrac{K}{\ })}$$

Challenge: $u = V_{h+1,k}$ is not xed, 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

$$\mathcal{V} = \{V(s) = \min\{H, \max_a [w^\top \phi(s,a) + k\sqrt{\phi(s,a)^\top \Lambda^{-1} \phi(s,a)}]\}:$$

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

$$[P_h V_{h+1;k}](s,a) - [P_h V_{h+1;k}](s,a) \leq \sup_{u \in \mathcal{V}} [P_h u](s,a) - [P_h u](s,a):$$

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

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

Covering Number Argument

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

$$\mathcal{V} = \{V(s) = \min\{H, \max_a [w^\top \phi(s,a) + k\sqrt{\phi(s,a)^\top \Lambda^{-1} \phi(s,a)}]\}:$$

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

$$[P_h V_{h+1;k}](s,a) - [P_h V_{h+1;k}](s,a) \leq \sup_{u \in \mathcal{V}} [P_h u](s,a) - [P_h u](s,a):$$

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

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

Covering Number Argument

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

$$\mathcal{V} = \{V(s) = \min\{H, \max_a [w^\top \phi(s,a) + k\sqrt{\phi(s,a)^\top \Lambda^{-1} \phi(s,a)}]\}:$$

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

$$[P_h V_{h+1;k}](s,a) - [P_h V_{h+1;k}](s,a) \leq \sup_{u \in \mathcal{V}} [P_h u](s,a) - [P_h u](s,a):$$

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

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

Covering Number Argument

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

$$\sup_{u \in V} \left[\hat{P}_h u\right](s,a) - \left[P_h u\right](s,a) \leq \bar{\epsilon}(\alpha/N)\, \sigma_{h,k}(s,a)$$

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

- Choosing $\alpha = 1/T$, we get

$$\epsilon(\alpha) = \bar{\epsilon}(\alpha/N) \leq \tfrac{1}{2} k M V_{h,k} k + H \sqrt{d \log\left(\frac{K N}{\alpha}\right)} = \tilde{O}(H d)$$

Alternative linear models

Linear MDP model factorizes $P = \quad M$ with some known $2 R^{(S A) d}$ and some unknown $M \ 2 R^{d S}$.

Some alternative factorizations are:

Linear mixture MDPs [Zhou et al., 2021]: $P = \quad$ with some known $2 R^{(S A S) d}$ and unknown $2 R^{d}$. 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]: $P = \quad M \quad$ with some known $2 R^{(S A) m}$, another known $2 R^{n S}$, and an unknown $M \ 2 R^{m \ n}$. Can be shown to be a special case of linear mixture MDPs, and su ers from the same limitations.

Low-rank MDPs [Modi et al., 2024]: Same as linear MDPs except both and M are unknown and belong to nite 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 nite-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:

$$S_{s;a} := \{ s' \in S : P(s'|s;a) > 0 \}$$

$$U := \max_{(s;a) \in S \times A} |S_{s;a}|$$

Limitations of the Linear Setting

Directly reachable states:

$$S_{s;a} := \{ s^0 \in S : P(s^0 | s; a) > 0 \}$$

$$U := \max_{(s;a) \in S \times A} |S_{s;a}|$$

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

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

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 in nite dimensional feature maps

Kernel-Based Setting

Kernel-based models are natural extensions of linear models to in nite 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 in nite 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 in nite 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

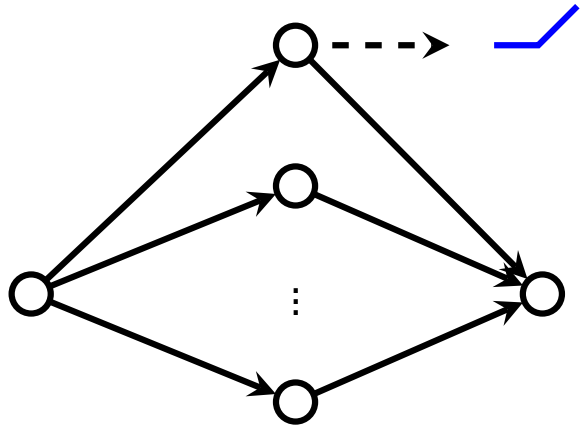
Kernel-based models are natural extensions of linear models to in nite dimensional feature maps

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 ! Linear ! Kernel-Based ! NN-Based



Kernel-Based Setting

Function class:

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

Kernel-Based Setting

Function class:

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

An extension of linear models to in nite dimensions in the feature space

Kernel-Based Setting

Function class:

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

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

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

Mercer Theorem

A positive de nite kernel : Z Z ! R

Mercer Theorem

A positive de nite kernel $: Z \quad Z ! \quad R$

Theorem Any positive de nite kernel can be written as

$$(z; z^0) = \sum_{m=1}^{1} \quad _m \, ' \, _m(z) \, ' \, _m(z^0)$$

The feature map $_m( ) = \quad \frac{1}{2}_m \, ' \, _m( )$ corresponding to

$_m$ are referred to as eigenvalues

$' \, _m$ are referred to as eigenfunctions

Kernels

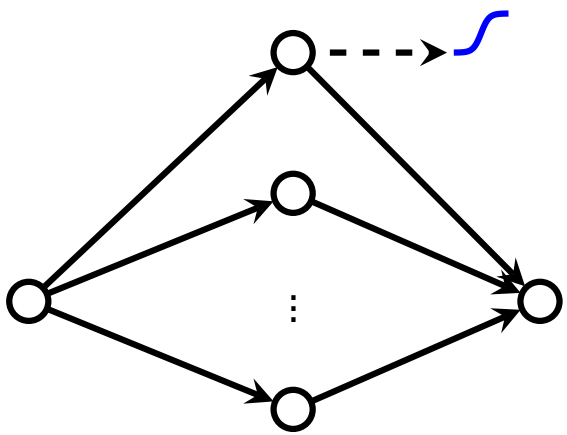
Squared Exponential kernel

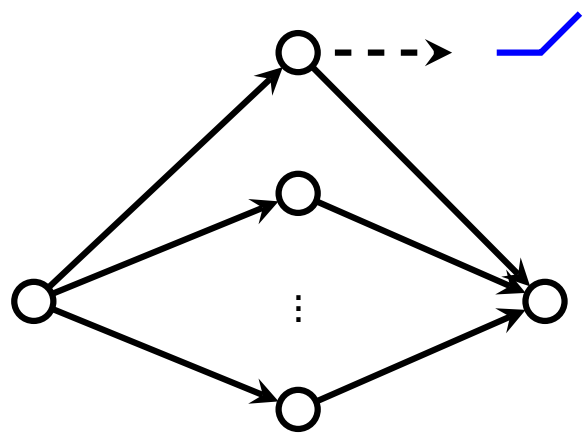
$$(z; z') = \exp \quad \frac{kz \quad z'k^2}{2`^2}$$

Matern- kernel

$$(z; z') = \frac{2^1}{(\quad)} \quad \frac{p\overline{2}}{`} kz \quad z'k \quad K \quad \frac{p\overline{2}}{l} kz \quad z'k$$

Kernels





Reproducing Kernel Hilbert Space

RKHS:

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

Inner product $\langle f, g \rangle_k = w_f^{\top} w_g$

$\|f\|_H = \|w\|$

$\phi_m = \sqrt{\lambda_m}\, e_m$, e_m form an orthonormal basis

Kernel Based Regression

Provided a dataset of observation:

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

Regularized Least Squares Error:

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

Kernel-Based Regression

Predictor:

$$\hat{f}(z) = {}_t^{>}(z)(K_t + I)^{-1} y_t$$

$${}_t(z) = [k(z_1; z); k(z_2; z); \quad ; k(z_t; z)]$$

$$K_t = [k(z_i; z_j)]_{i;j=1}^{t}$$

$$y_t = [Y(z_1); Y(z_2); \quad ; Y(z_t)]$$

Kernel-Based Regression

Uncertainty estimator:

$$(\ _t(z))^2 = \quad (z;z) \qquad {}^{>}_t(z)(K_t + I\ )^{\ 1}\ _t(z)$$

Kernel-Based Regression

Uncertainty estimator:

$$(\ _t(z))^2 = \quad (z;z) \qquad \overset{>}{_t}(z)(K_t + \ I\ )^{\ 1}\ \ _t(z)$$

Closed from expressions for prediction and uncertainty quanti cation!

RL with kernel-based function approximation

IDEA: Approximate theQ-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 -optimal policy with a computational and sample complexity polynomial in 1= and H

RL with kernel-based function approximation

IDEA: Approximate theQ-functions as a function in RKHS

Want: Find an -optimal policy with a computational and sample complexity polynomial in1= 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 $1/\varepsilon$ and H

|possibly some kernel parameters|

independently of S and A.

$$\text{Sample complexity: } \tilde{O}\left(\left(\frac{1}{\varepsilon}\right)^2\right)$$

RL with kernel-based function approximation

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

Want: Find an -optimal policy with a computational and sample complexity polynomial in 1= and H

|possibly some kernel parameters|

independently of S and A.

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

RL with kernel-based function approximation

RL with kernel-based function approximation

E ective Dimension:

$$\underbrace{\lambda_1, \lambda_2, \ldots, \lambda_D}_{D \text{ dimension}}, \lambda_{D+1}, \ldots$$

$$D \quad \frac{1}{2} \log \det(I + \frac{1}{\lambda} K_t)$$

In the linear setting: $D \quad d$

For Squared Exponential kernel:
$D \quad \text{polylog}(T)$

For Matern kernel:
$D \quad T^{\frac{d}{d+}}$ [Vakili et al., 2021]

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s': \; P(s' \mid s, a) \in H$$

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s', s, a: \; P(s' \mid s, a) \in H$$

A significant generalization of linear models

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s', s, a: \; P(s' \mid s, a) \in H$$

A significant generalization of linear models

Linear model is a special case with linear kernel:
$k((s, a), (s', a')) = \phi^\top(s, a)\, \phi(s', a')$

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s', P(s' \mid s, a) \in H$$

A significant generalization of linear models

Linear model is a special case with linear kernel:
$k((s, a), (s', a')) = \phi^\top(s, a)\, \phi(s', a')$

RKHS of common kernels can approximate almost all continuous functions

RL with Kernel-based FA

Kernel-based transition assumption

$$\text{For all } s', s, a: \; P(s' \mid s, a) \in H$$

A significant generalization of linear models

Linear model is a special case with linear kernel:
$k((s, a), (s', a')) = \phi^\top(s, a)\, \phi(s', a')$

RKHS of common kernels can approximate almost all continuous functions

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

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]:

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

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

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

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 $\hat{P}_h$ and b_h?

Kernel-based optimistic value iteration (KOVI)

Transition model P_h can be defined implicitly via least-squares:

- Solve the regularized linear regression problem

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

- That provides a prediction

$$[\hat{P}_h V_{h+1,k}](s,a) = \hat{f}_h(s,a) = k_{h,k}(s,a)(K_{h,k} + I)^{-1} v_{h,k}$$

$$v_{h,k} = [V_{h+1}(s_{h+1,1}), V_{h+1}(s_{h+1,2}), \ldots, V_{h+1}(s_{h+1,k})]$$

- Also, an uncertainty quantification (variance)

$$\sigma^2_{h,k}(s,a) = k(s,a), k(s,a) - k_{h,k}(s,a)(K_{h,k} + I)^{-1} k_{h,k}(s,a)$$

Kernel-based optimistic value iteration (KOVI)

The prediction and variance give us an upper con dence bound on $Q^?$:

$$Q_{h;k}(s;a) = r_h(s;a) + [\hat{P}_h V_{h+1}](s;a) + \quad (\ )\ _{h;k}(s;a)$$

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

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

Performance guarantees

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

Performance guarantees

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

Proof ideas:

Prove confidence bounds

$$|[\hat{P}_h V_{h+1,k}](s,a) - [P_h V_{h+1,k}](s,a)| \le \beta(\cdot)\, \Gamma_{h,k}(s,a).$$

Using standard techniques, show

$$\mathrm{Regret}(K) \lesssim \sum_{h,k} \beta(\cdot)\, \Gamma_{h,k}(s_{h,k}, a_{h,k}).$$

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

$$\sum_{h,k} \Gamma_{h,k}(s_{h,k}, a_{h,k}) \lesssim H \sqrt{KD \log(K)}.$$

Kernel-based Setting - Analysis

- We need a con dence bound of the form

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\ )\ _h(s,a):$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\quad)\ _h(s,a):$$

- For a fixed $f \ 2 \ H$ with non-adaptive inputs $z_1; : : : ; z_k$,

$$(\quad) \quad k \ f \ k_H \ + \ p \frac{H}{=} \ q \ \overline{d \log(\frac{T}{-})}$$

Kernel-based Setting - Analysis

- We need a con dence bound of the form

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\quad)\ _h(s,a):$$

- For a xed $f \ 2 \ H$ with non-adaptive inputs $z_1; : : : ; z_k$,

$$(\quad) \quad k \ f \ k_H \ + \ p^{\frac{H}{=}} \ q \ \overline{d \log(\frac{T}{-})}$$

Challenge 1: Inputs $(s_1; a_1); : : : ; (s_k; a_k)$ are adaptive!

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\ ) \ _h(s,a):$$

- For a fixed $f \ 2 \ H$ with non-adaptive inputs $z_1, : : : , z_k$,

$$(\ ) \quad k \ f \ k_H \quad + \ p \frac{H}{} \sqrt{d \log(\frac{T}{})}$$

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

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

$$(\ ) \quad k \ f \ k_H \quad + \ p \frac{H}{} \sqrt{D + \log(\frac{1}{})}$$

Kernel-based Setting - Analysis

- We need a confidence bound of the form

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\ ) \ _h(s,a):$$

- For a fixed $f \ 2 \ H$ with non-adaptive inputs $z_1, : : : , z_k$,

$$(\ ) \quad k \ f \ k_H \quad + \ p \frac{H}{=} \quad \sqrt{d \log(\frac{T}{})}$$

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

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

$$(\ ) \quad k \ f \ k_H \quad + \ p \frac{H}{=} \quad \sqrt{D + \log(\frac{1}{})}$$

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

$$\hat{f}(s,a) \quad [P_h V_{h+1,k}](s,a) \qquad (\ ) \ _h(s,a):$$

- For a fixed $f \ 2 \ H$ with non-adaptive inputs $z_1; : : : ; z_k$,

$$(\ ) \quad k \ f \ k_H \quad + \ p \overline{H} \sqrt{d \log(\tfrac{T}{\ })}$$

Challenge 1: Inputs $(s_1; a_1); : : : ; (s_k; a_k)$ are adaptive!

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

$$(\ ) \quad k \ f \ k_H \quad + \ p \overline{H} \sqrt{D + \log(\tfrac{1}{\ })}$$

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

$$V = \{ V(s) = \min\{H, \max_a [\hat{r}(s,a) + \beta(s,a)]\} \}$$

- How many functions V are required to cover V up to ε error?

Covering Number Argument

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

$$V = \{ V(s) = \min\{H, \max_a [\hat{r}(s,a) + \beta(s,a)]\} \}$$

- How many functions V are required to cover V up to ε error?

Covering Number Argument

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

$$V = \{ V(s) = \min\{H, \max_a [\hat{r}(s,a) + \beta(s,a)]\} \}$$

- How many functions V are required to cover V up to ε error?

Covering Number Argument

- We can now use a union-bound argument

$$( ) = \bar{}(=N) \quad k \; f \; k_{H\,k} + \; p \overline{=}^{H} \quad q \; \frac{}{D + \log N \; + \; \overline{=}^{1}}$$

Covering Number Argument

- We can now use a union-bound argument

$$( ) = \bar{}(=N) \quad k \; f \; k_{H\,k} + \; p\frac{H}{=} \quad \sqrt{\frac{}{D + \log N \; + \; \frac{1}{=}}}$$

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

$$\text{Regret}(K) = \tilde{O}(H^{2}\sqrt{p\,D^{2}K \; + \; D \log N \; K})$$

Covering Number Argument

- We can now use a union-bound argument

$$\beta(\delta) = \overline{\beta}(\delta/N) \approx f_{H_k} + \frac{H}{\sqrt{\lambda}} \sqrt{\frac{\mathsf{D} + \log N + \frac{1}{\delta}}{}}$$

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

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

- Sample Complexity

  - Very smooth kernels D and $\log(N) \approx \text{poly}\log(K)$

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

  - In general could be vacuous!

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$V \quad H_K$

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

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$$V \quad H_K$$

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

Idea: Leverage kernel mean embedding

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

Optimistic Closure

Chowdhury and Oliveira [2023] Optimistic Closure Assumption:

$$V \quad H_K$$

$$V = \square \; V(s) = \min\{H,\ \max_a \hat{f}(s,a) + \beta\sigma(s,a)\} \; \square$$

Idea: Leverage kernel mean embedding

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

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.