Towards a Theory of Generalization in Reinforcement Learning

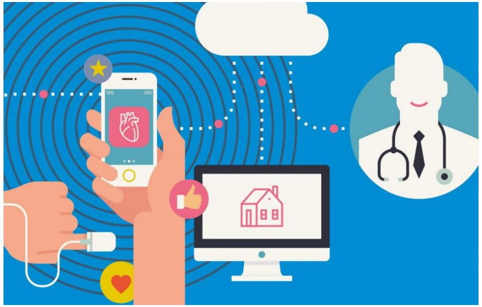
Sham M. Kakade

University of Washington & Microsoft Research

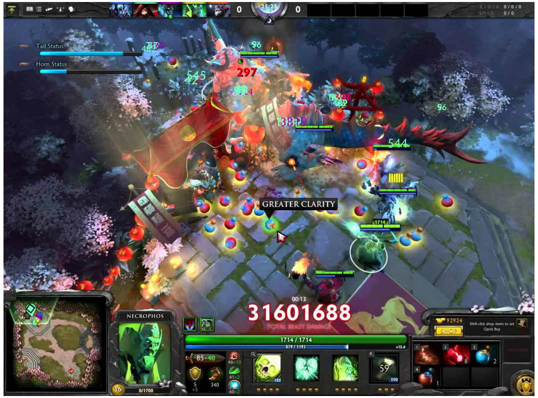
Hello Harvard

(& MIT) !

Progress of RL in Practice

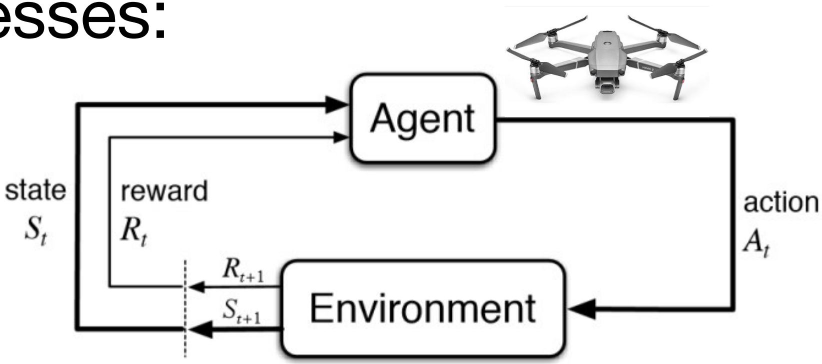


[AlphaZero, Silver et.al, 17]



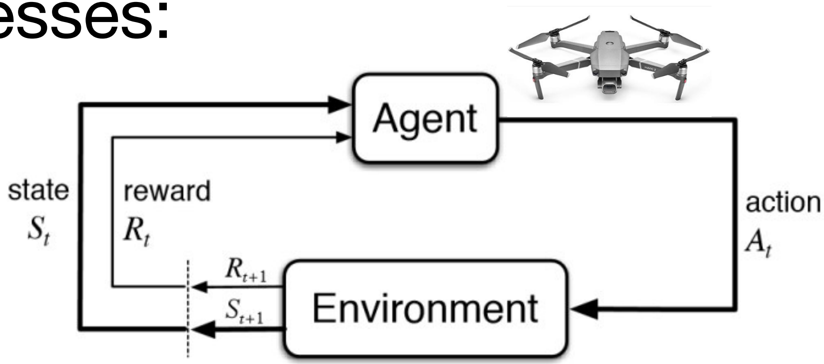
[OpenAI Five, 18]

Markov Decision Processes:
a framework for RL



Markov Decision Processes:
a framework for RL

- A policy:
  π : States $\rightarrow$ Actions



Markov Decision Processes:

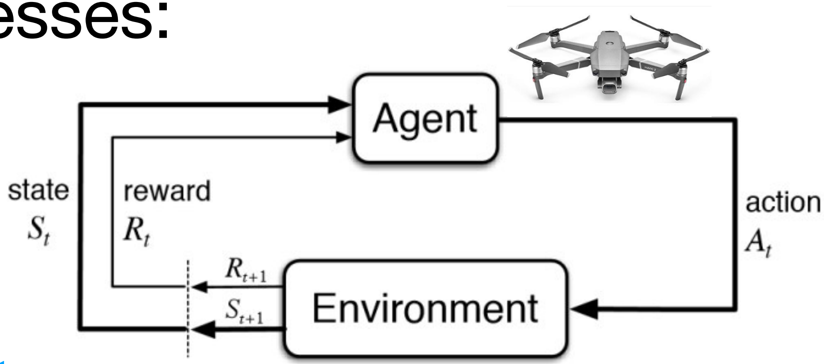
a framework for RL

- A policy:

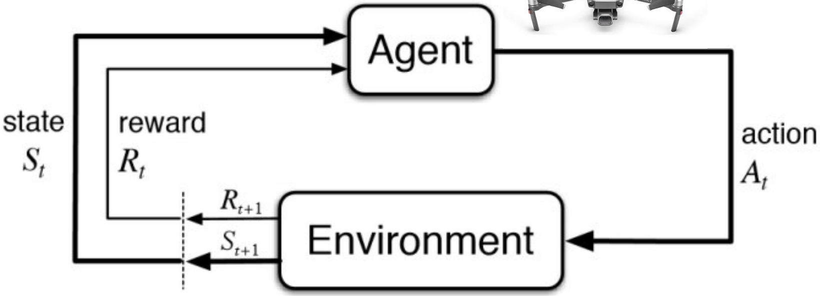
  π : States $\rightarrow$ Actions

- Execute π to obtain a trajectory:

  $s_0, a_0, r_0, s_1, a_1, r_1 \ldots s_{H-1}, a_{H-1}, r_{H-1}$



Markov Decision Processes:
a framework for RL



- A policy:

  π : States $\rightarrow$ Actions

- Execute π to obtain a trajectory:

  $s_0, a_0, r_0, s_1, a_1, r_1 \ldots s_{H-1}, a_{H-1}, r_{H-1}$

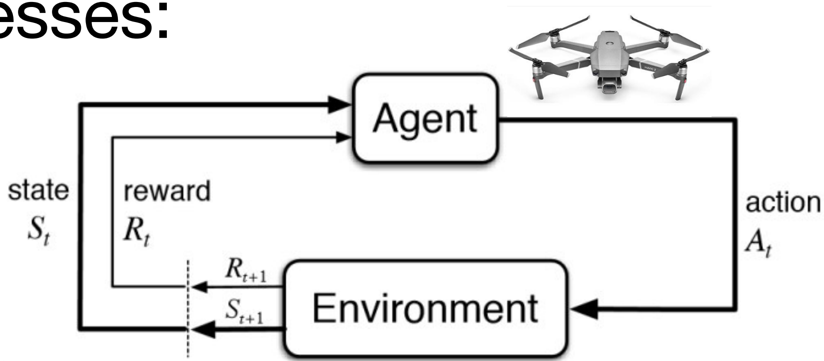
- Cumulative H-step reward:

state-action value

$$V_H^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{H-1} r_t \middle| s_0 = s \right], \quad Q_H^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{H-1} r_t \middle| s_0 = s, a_0 = a \right]$$

Markov Decision Processes:

a framework for RL



- A policy:

  π : States $\rightarrow$ Actions

- Execute π to obtain a trajectory:

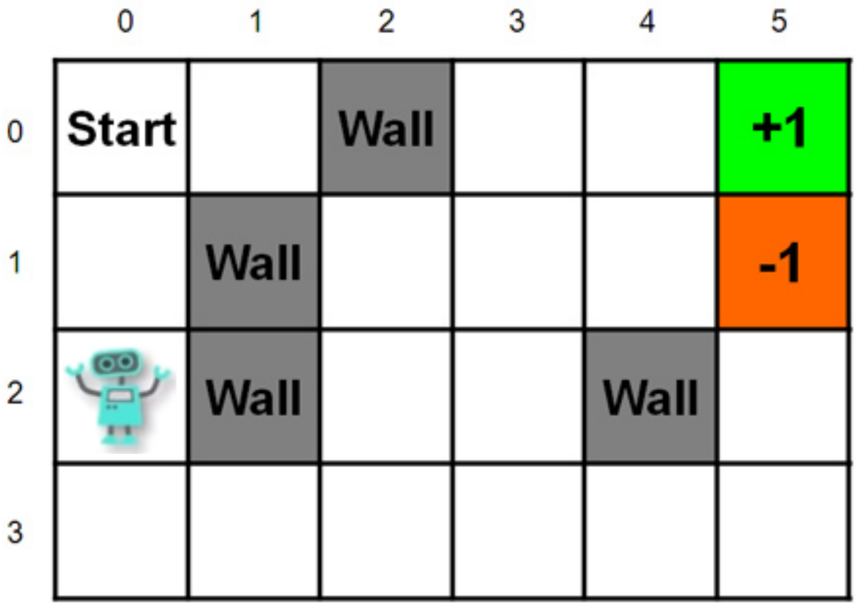
  $s_0, a_0, r_0, s_1, a_1, r_1 \ldots s_{H-1}, a_{H-1}, r_{H-1}$

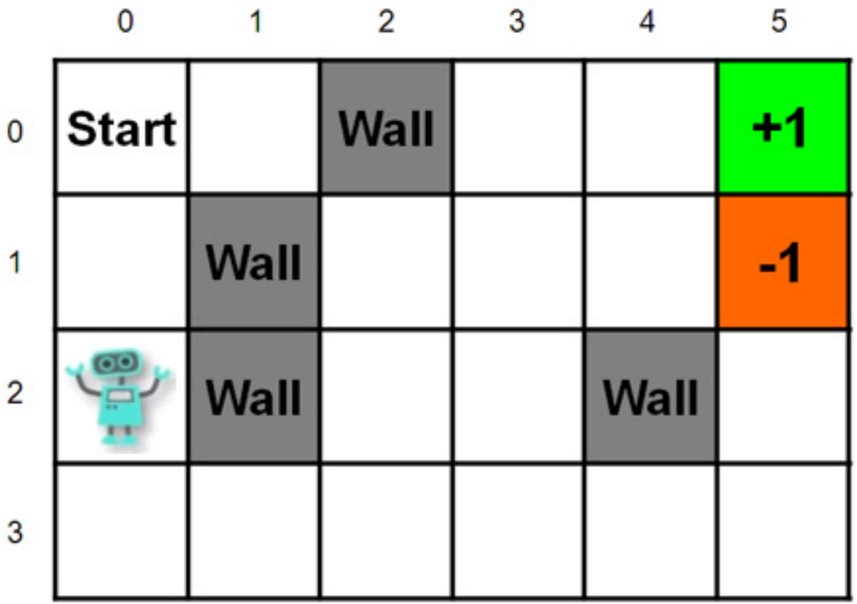
- Cumulative H-step reward:

$$V_H^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{H-1} r_t \middle| s_0 = s\right], \quad Q_H^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{H-1} r_t \middle| s_0 = s, a_0 = a\right]$$

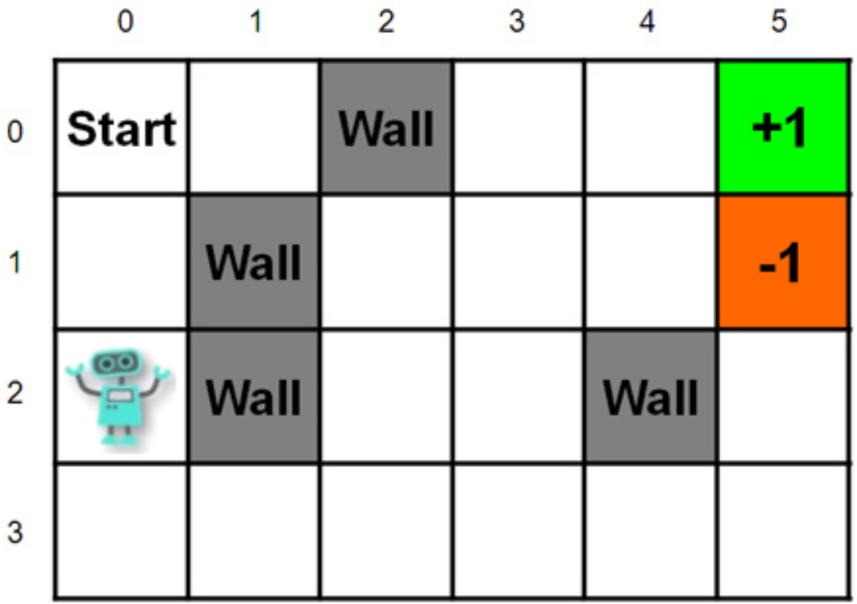
- Goal: Find a policy π that maximizes our value $V^{\pi}(s_0)$ from s_0.

  Episodic setting: We start at s_0; act for H steps; repeat…


0
1
2
3
4
5
0
Start
Wall
+1
1
Wall
-1
2
Wall
Wall
3

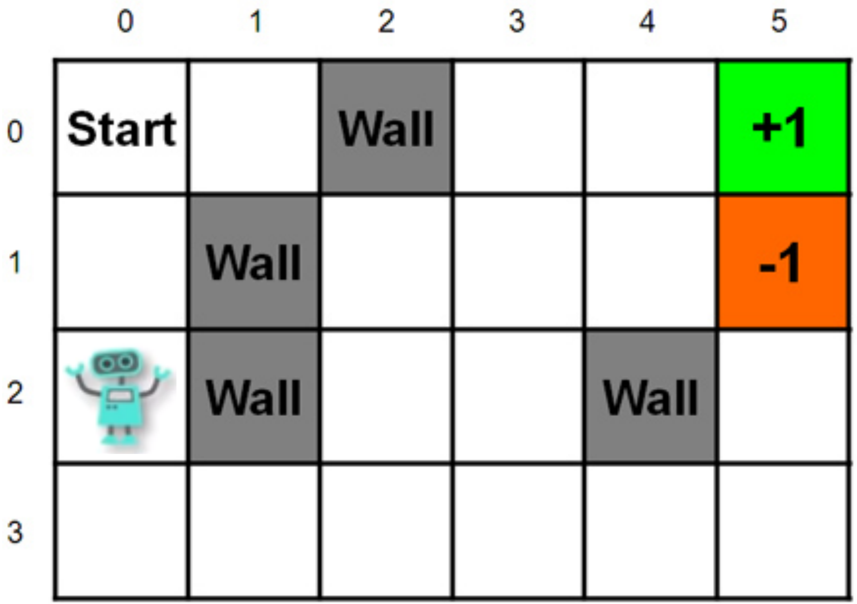


Challenges in RL



Challenges in RL

1. Exploration
   (the environment may be unknown)



Challenges in RL

1. Exploration
(the environment may be unknown)

2. Credit assignment problem
(due to delayed rewards)

Dexterous Robotic Hand Manipulation
OpenAI, ‘19

Challenges in RL

1. Exploration
   (the environment may be unknown)

2. Credit assignment problem
   (due to delayed rewards)

3. Large state/action spaces:
   hand state: joint angles/velocities
   cube state: configuration
   actions: forces applied to actuators

<u>Part-0:</u>

A Whirlwind Tour of Generalization

from Supervised Learning to RL

Provable Generalization in Supervised Learning (SL)

Generalization is possible in the IID supervised learning setting!

To get ϵ-close to best in hypothesis class $\mathscr{F}$, we need # of samples that is:

Provable Generalization in Supervised Learning (SL)

Generalization is possible in the IID supervised learning setting!

To get ϵ-close to best in hypothesis class $\mathscr{F}$, we need # of samples that is:

- "Occam's Razor" Bound (finite hypothesis class): need $O(\log|\mathscr{F}|/\epsilon^2)$

Provable Generalization in Supervised Learning (SL)

Generalization is possible in the IID supervised learning setting!

To get ϵ-close to best in hypothesis class $\mathscr{F}$, we need # of samples that is:

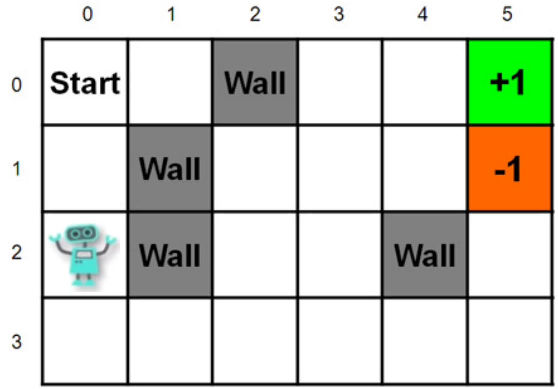
- "Occam's Razor" Bound (finite hypothesis class): need $O(\log|\mathscr{F}|/\epsilon^2)$
- Various Improvements:
  - VC dim $O(\mathrm{VC}(\mathscr{F})/\epsilon^2)$; Classification (margin bounds): $O(\mathrm{margin})/\epsilon^2)$;

    Linear regression: $O(\mathrm{dimension}/\epsilon^2)$
  - Deep Learning: the algorithm also determines the complexity control

The key idea in SL: data reuse

With a training set, we can simultaneously evaluate the loss of all hypotheses in our class!

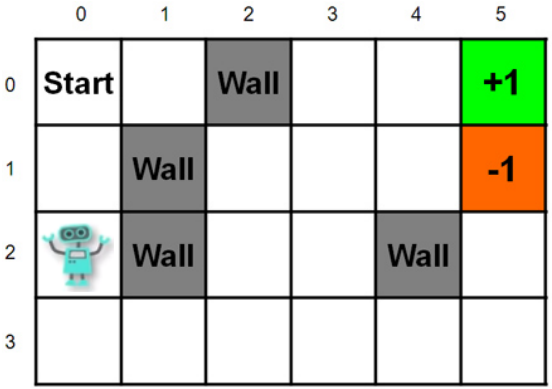
Sample Efficient RL in the Tabular Case (no generalization here)



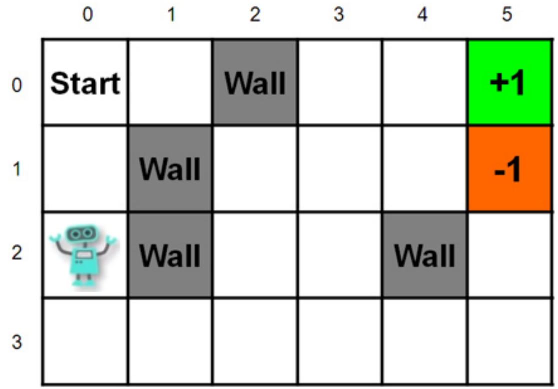
Sample Efficient RL in the Tabular Case (no generalization here)

small

- S = #states, A = #actions, H = #horizon
- We have an (unknown) MDP.

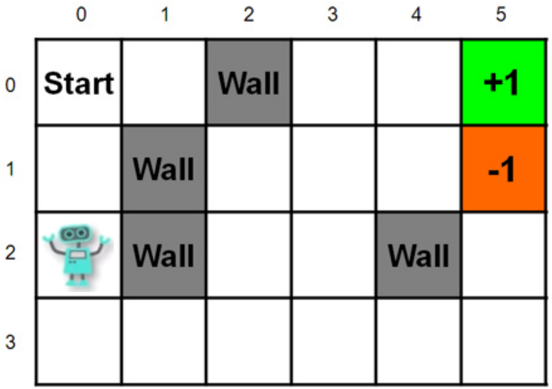


Sample Efficient RL in the Tabular Case (no generalization here)



- S = #states, A = #actions, H = #horizon
- We have an (unknown) MDP.
- Thm: [Kearns & Singh '98] In the episodic setting, E^3
  $poly(S, A, H, 1/\epsilon)$ samples suffice to find an ϵ-opt policy.
  Key idea: $\underbrace{\text{optimism}}$ + dynamic programming ← plan to explore

Sample Efficient RL in the Tabular Case (no generalization here)



goal: find π $\quad V_H^{\pi}(s_0) \geq V_H^{*}(s_0) - \varepsilon$

- S = #states, A = #actions, H = #horizon
- We have an (unknown) MDP.
- Thm: [Kearns & Singh '98] In the episodic setting, $poly(S, A, H, 1/\epsilon)$ samples suffice to find an ϵ-opt policy.

  "model based"

  Key idea: optimism + dynamic programming

  Watkins '90

  asym.

- Lots improvements on the rate:
  [Brafman& Tennenholtz '02][K. '03][Auer+ '09] [Agrawal, Jia '17]
  [Azar+ '13],[Dann & Brunskill '15] minmax optimal
- Provable Q-learning (+bonus):
  [Strehl+ (2006)], [Szita & Szepesvari '10],[Jin+ '18] "model free"

I: Provable Generalization in RL

Q1: Can we find an ϵ-opt policy with no S dependence?

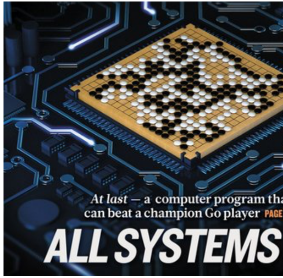
on value

- How can we reuse data to estimate the value of all policies in a policy class $\mathscr{F}$?

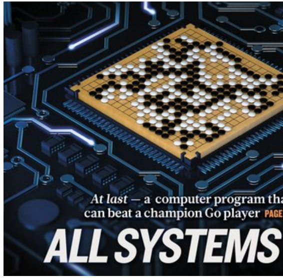
  Idea: Trajectory tree algo

  dataset collection: uniformly at random choose actions for all H steps in an episode.

  estimation: uses importance sampling to evaluate every $f \in \mathscr{F}$



I: Provable Generalization in RL



Q1: Can we find an ϵ-opt policy with no S dependence?

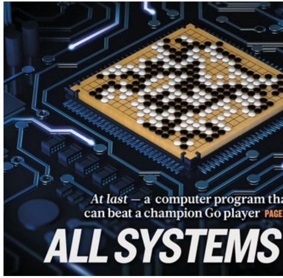
- How can we reuse data to estimate the value of all policies in a policy class $\mathscr{F}$?
  Idea: Trajectory tree algo
  dataset collection: uniformly at random choose actions for all H steps in an episode.
  estimation: uses importance sampling to evaluate every $f \in \mathscr{F}$

"classic gen" because the MDP is stoch.

- Thm:[Kearns, Mansour, & Ng '00]
  To find an ϵ-best in class policy, the trajectory tree algo uses $O(A^H \log(|\mathscr{F}|/\epsilon^2))$ samples

I: Provable Generalization in RL

Q1: Can we find an ϵ-opt policy with no S dependence?



- How can we reuse data to estimate the value of all policies in a policy class $\mathscr{F}$?
  Idea: Trajectory tree algo
  dataset collection: uniformly at random choose actions for all H steps in an episode.
  estimation: uses importance sampling to evaluate every $f \in \mathscr{F}$

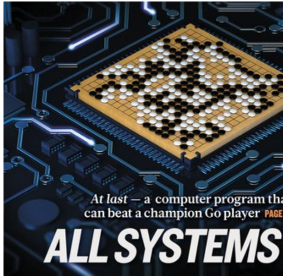
- Thm:[Kearns, Mansour, & Ng '00] "agnostic" with no assumptions on MDP.
  To find an ϵ-best in class policy, the trajectory tree algo uses $O(A^H \log(|\mathscr{F}|/\epsilon^2))$ samples
  - Only $\log(|\mathscr{F}|)$ dependence on hypothesis class size.
  - There are VC analogues as well.

  $A = 2$

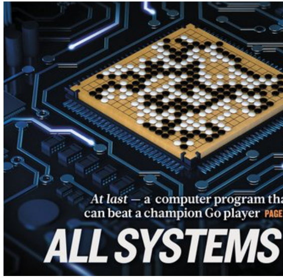
  A^H

- Can we avoid the $\cancel{2^H}$ dependence to find an an ϵ-best-in-class policy?
  Agnostically, **NO**!
  Proof: Consider a binary tree with 2^H-policies and a sparse reward at a leaf node.

II: Provable Generalization in RL

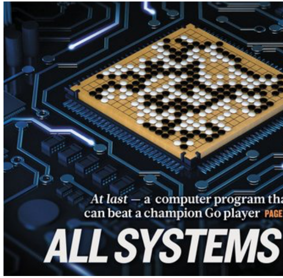


II: Provable Generalization in RL



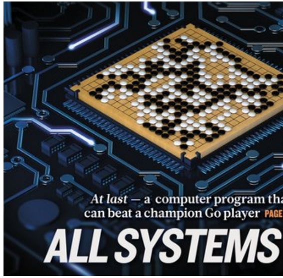
- Q2: Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?

II: Provable Generalization in RL



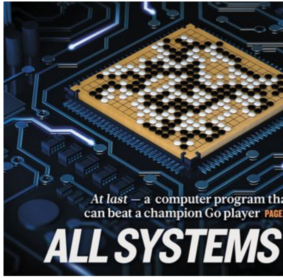
- Q2: Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?
  - Agnostically/best-in-class? NO.

II: Provable Generalization in RL



- Q2: Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?
  - Agnostically/best-in-class? NO.
  - With various stronger assumptions, of course.

II: Provable Generalization in RL



- Q2: Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon,$ "complexity measure"$)$ samples?
  - Agnostically/best-in-class? NO.
  - With various stronger assumptions, of course.

What is the nature of the assumptions under which generalization in RL is possible?
(what is necessary? what is sufficient?)

Today's Lecture

What are necessary representational and distributional conditions that permit provably sample-efficient offline reinforcement learning?

Today's Lecture

What are necessary representational and distributional conditions that permit provably sample-efficient offline reinforcement learning?

- Part I: bandits & linear bandits
  (let's start with horizon $H = 1$ case)

Today's Lecture

What are necessary representational and distributional conditions that permit provably sample-efficient offline reinforcement learning?

- Part I: bandits & linear bandits
  (let's start with horizon $H = 1$ case)

- Part II: Lower bounds:
  Linear realizability: natural conditions to impose
  Is RL possible?

Today's Lecture

What are necessary representational and distributional conditions that permit provably sample-efficient offline reinforcement learning?

- Part I: bandits & linear bandits
  (let's start with horizon $H = 1$ case)

- Part II: Lower bounds:
  Linear realizability: natural conditions to impose
  Is RL possible?

- Part III: Upper bounds:
  Are there unifying conditions that are sufficient?

<u>Part-I:</u>

Bandits (the $H = 1$ case)

(Let’s set the stage for RL!)

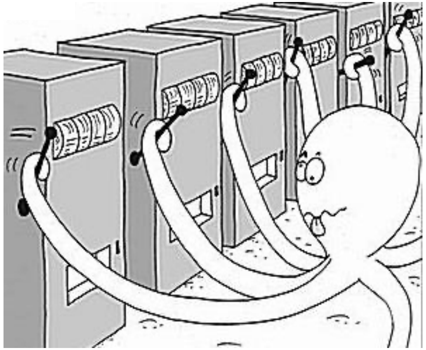
Multi-armed bandits

How should we allocate

T tokens to A "arms"

to maximize our return?

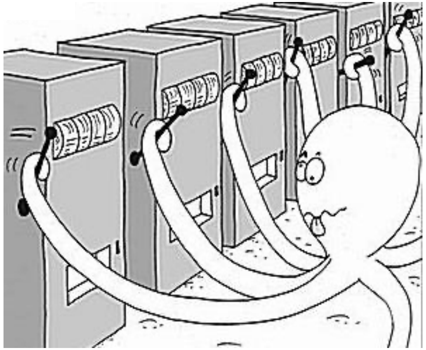
[Robins '52, Gittins'79, Lai & Robbins '85 …]



Multi-armed bandits

How should we allocate
T tokens to A "arms"
to maximize our return?

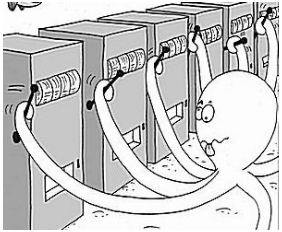
[Robins '52, Gittins'79, Lai & Robbins '85 …]



- Very successful algo when A is small.
- What can we do when the number or arms A is large?

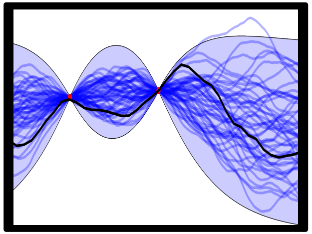
Dealing with the large action case

Bandits



- decision: pull an arm

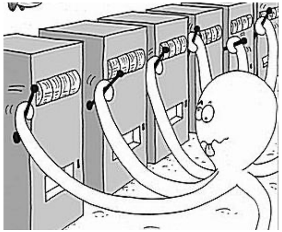
Linear (RKHS) Bandits



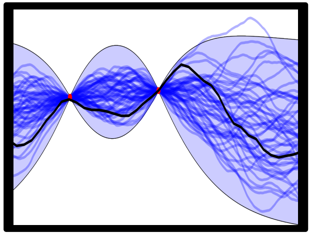
- decision: choose some $x \in \mathcal{X}$
- e.g. $x \in R$

Dealing with the large action case

Bandits



Linear (RKHS) Bandits



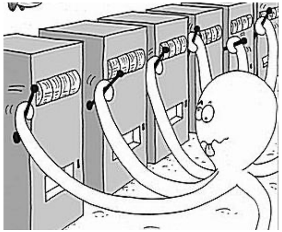
- decision: pull an arm

- decision: choose some $x \in \mathscr{X}$
- e.g. $x \in R$

- widely used generalization: The “linear bandit” model [Abe & Long+ ’99]
  successful in many applications: scheduling, ads…

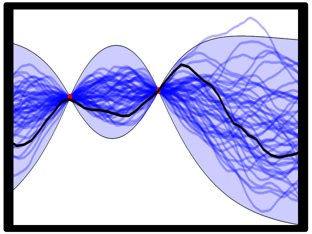
Dealing with the large action case

Bandits



- decision: pull an arm

Linear (RKHS) Bandits



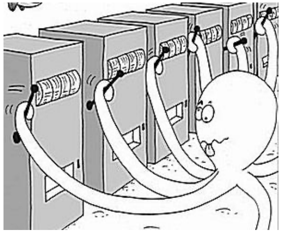
- decision: choose some $x \in \mathcal{X}$
- e.g. $x \in R$

- widely used generalization: The "linear bandit" model [Abe & Long+ '99]
  successful in many applications: scheduling, ads…
- decision: x_t, reward: r_t, reward model:

$$r_t = f(x_t) + \text{noise}, \qquad f(x) = w^\star \cdot \phi(x)$$

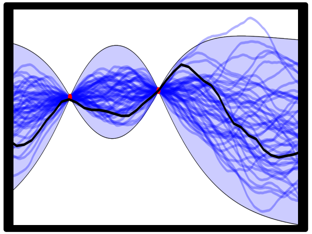
Dealing with the large action case

Bandits



• decision: pull an arm

Linear (RKHS) Bandits



• decision: choose some $x \in \mathcal{X}$

• e.g. $x \in R$

- widely used generalization: The "linear bandit" model [Abe & Long+ '99]
  successful in many applications: scheduling, ads…
- decision: x_t, reward: r_t, reward model:

$$r_t = f(x_t) + \text{noise}, \qquad f(x) = w^\star \cdot \phi(x)$$

- Hypothesis class $\mathcal{F}$ is set of linear/RKHS functions

Linear-UCB/GP-UCB:

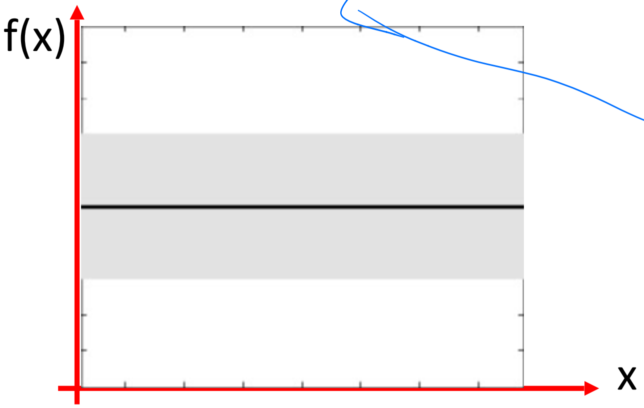
Algorithmic Principle: Optimism in the face of uncertainty

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$

uncertainty of the function

our "mean" estimate of "f"

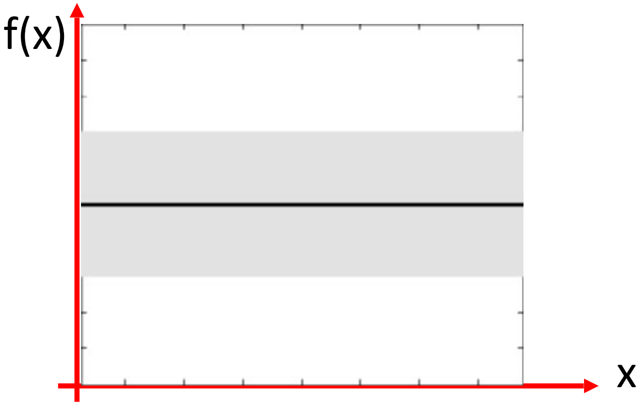


Linear-UCB/GP-UCB:

Algorithmic Principle: Optimism in the face of uncertainty

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$



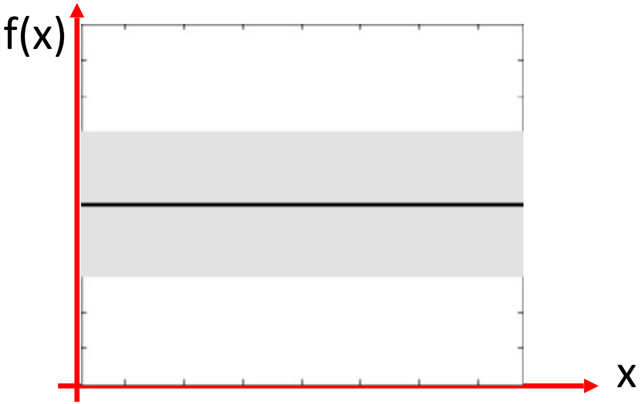
Naturally trades off exploration and exploitation

Linear-UCB/GP-UCB:

Algorithmic Principle: Optimism in the face of uncertainty

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$



Naturally trades off exploration and exploitation

Only picks plausible maximizers

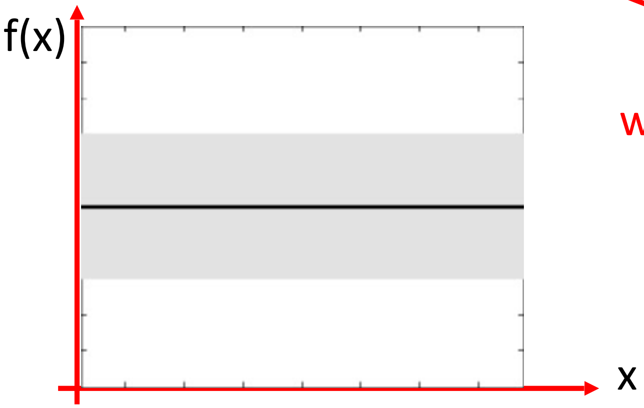
Linear-UCB/GP-UCB:

Algorithmic Principle: Optimism in the face of uncertainty

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$

How should we choose β_t?



Naturally trades off exploration and exploitation

Only picks plausible maximizers

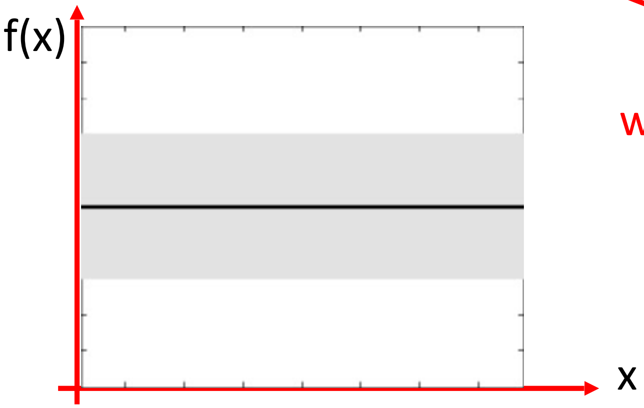
Linear-UCB/GP-UCB:

Algorithmic Principle: Optimism in the face of uncertainty

Pick input that maximizes upper confidence bound:

$$\boxed{x_t = \arg\max_{x \in D} \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}$$

How should we choose β_t?



Naturally trades off exploration and exploitation

Only picks plausible maximizers

Regret of Lin-UCB/GP-UCB

(generalization in action space)

Theorem: [Dani, Hayes, & K. ’08], [Srinivas, Krause, K. & Seeger '10]

Assuming $\mathscr{F}$ is an RKHS (with bounded norm), if we choose β_t “correctly”,

$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] = \mathcal{O}^*\left(\sqrt{\frac{\gamma_T}{T}}\right)$$

where $\gamma_T := \max_{x_0 \ldots x_{T-1} \in \mathcal{X}} \log\det\left(I + \sum_{t=0}^{T-1} \phi(x_t)\phi(x_t)^\top\right)$

$f \in$ linear in ϕ

Regret of Lin-UCB/GP-UCB

(generalization in action space)

Theorem: [Dani, Hayes, & K. '08], [Srinivas, Krause, K. & Seeger '10]

Assuming $\mathscr{F}$ is an RKHS (with bounded norm), if we choose β_t "correctly",

$$\frac{1}{T}\sum_{t=1}^{T}[f(x^*) - f(x_t)] = \mathcal{O}^*\left(\sqrt{\frac{\gamma_T}{T}}\right) \quad c\sqrt{\frac{d}{T}}$$

where $\gamma_T := \max_{x_0 \ldots x_{T-1} \in \mathscr{X}} \log\det\left(I + \sum_{t=0}^{T-1} \phi(x_t)\phi(x_t)^\top\right)$

- Key complexity concept: "*maximum information gain*" γ_T determines the regret
  - $\gamma_T \approx d \log T$ for ϕ in d-dimensions
  - Think of γ_T as the "effective dimension"
- Easy to incorporate context
- Also: [Auer+ '02; Abbasi-Yadkori+ '11]

Switch
(LinUCB analysis)

Part-2: RL

What are necessary conditions?

Let’s look at the most natural assumptions.

Approx. Dynamic Programming with Linear Function Approximation

Approx. Dynamic Programming with Linear Function Approximation

Basic idea: approximate the $Q(s, a)$ values with linear basis functions $\phi_1(s, a), \ldots \phi_d(s, a)$. (where $d \ll$ #states, #actions)

$\vec{\phi} \in \mathbb{R}^d$

Approx. Dynamic Programming with Linear Function Approximation

Basic idea: approximate the $Q(s, a)$ values with linear basis functions $\phi_1(s, a), \ldots \phi_d(s, a)$. (where $d \ll$ #states, #actions)

- C. Shannon. *Programming a digital computer for playing chess.* Philosophical Magazine, ’50.
- R.E. Bellman and S.E. Dreyfus. *Functional approximations and dynamic programming.* ’59.
- Lots of work on this approach, e.g.
  [Tesauro, ’95], [de Farias & Van Roy ’03], [Wen & Van Roy ’13]

Approx. Dynamic Programming with Linear Function Approximation

$E[r_t \mid a_t] = w^{*} \cdot \phi(a_t)$

Basic idea: approximate the $Q(s, a)$ values with linear basis functions $\phi_1(s, a), \ldots \phi_d(s, a)$. (where $d \ll$ #states, #actions)

- C. Shannon. *Programming a digital computer for playing chess.* Philosophical Magazine, '50.
- R.E. Bellman and S.E. Dreyfus. *Functional approximations and dynamic programming.* '59.
- Lots of work on this approach, e.g. [Tesauro, '95], [de Farias & Van Roy '03], [Wen & Van Roy '13]

What conditions must our basis functions (our representations) satisfy in order for his approach to work?

Approx. Dynamic Programming with Linear Function Approximation

Basic idea: approximate the $Q(s, a)$ values with linear basis functions $\phi_1(s, a), \ldots \phi_d(s, a)$. (where $d \ll$ #states, #actions)

- C. Shannon. *Programming a digital computer for playing chess.* Philosophical Magazine, ’50.
- R.E. Bellman and S.E. Dreyfus. *Functional approximations and dynamic programming.* ’59.
- Lots of work on this approach, e.g.
  [Tesauro, ’95], [de Farias & Van Roy ’03], [Wen & Van Roy ’13]

What conditions must our basis functions (our representations) satisfy in order for his approach to work?

- Let’s look at the most basic question with “linearly realizable Q*”

RL with Linearly Realizable Q*-Function Approximation

(Does there exist a sample efficient algo?)

RL with Linearly Realizable Q*-Function Approximation

(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s, a) \in R^d$.

RL with Linearly Realizable Q*-Function Approximation
(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s,a) \in R^d$.
- (A1: Linearly Realizable Q*): Assume for all $s, a, h \in [H]$, there exists $w_1^\star, \ldots w_H^\star \in R^d$ s.t.

$$E\left[\sum_{\tau=H-h}^{H} r_\tau \,\middle|\, \pi\right] \approx \quad Q_h^\star(s,a) = w_h^\star \cdot \phi(s,a)$$

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

RL with Linearly Realizable Q*-Function Approximation

(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s, a) \in R^d$.
- (A1: Linearly Realizable Q*): Assume for all $s, a, h \in [H]$, there exists $w_1^\star, \ldots w_H^\star \in R^d$ s.t.

$$Q_h^\star(s, a) = w_h^\star \cdot \phi(s, a)$$

- Aside: the linear programing viewpoint.

RL with Linearly Realizable Q*-Function Approximation

(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s, a) \in R^d$.
- (A1: Linearly Realizable Q*): Assume for all $s, a, h \in [H]$, there exists $w_1^\star, \ldots w_H^\star \in R^d$ s.t.

$$Q_h^\star(s, a) = w_h^\star \cdot \phi(s, a)$$

- Aside: the linear programing viewpoint.
  - We have an underlying LP with d variables and $O(SA)$ constraints.

RL with Linearly Realizable Q*-Function Approximation
(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s, a) \in R^d$.
- (A1: Linearly Realizable Q*): Assume for all $s, a, h \in [H]$, there exists $w_1^\star, \ldots w_H^\star \in R^d$ s.t.

$$Q_h^\star(s, a) = w_h^\star \cdot \phi(s, a)$$

- Aside: the linear programing viewpoint.
  - We have an underlying LP with d variables and $O(SA)$ constraints.
  - The LP is not general because it encodes the Bellman optimality constraints.

RL with Linearly Realizable Q*-Function Approximation

(Does there exist a sample efficient algo?)

- Suppose we have a feature map: $\overrightarrow{\phi}(s, a) \in R^d$.
- (A1: Linearly Realizable Q*): Assume for all $s, a, h \in [H]$, there exists $w_1^\star, \ldots w_H^\star \in R^d$ s.t.

$$Q_h^\star(s, a) = w_h^\star \cdot \phi(s, a)$$

- Aside: the linear programing viewpoint.
  - We have an underlying LP with d variables and $O(SA)$ constraints.
  - The LP is not general because it encodes the Bellman optimality constraints.
  - We have sampling access (in the episodic setting).

Linearly Realizability is Not Sufficient for RL

Linearly Realizability is Not Sufficient for RL

Theorem:

Linearly Realizability is Not Sufficient for RL

Theorem:

- [Weisz, Amortila, Szepesvári '21]:
  There exists an MDP and a ϕ satisfying A1 s.t any online RL algorithm (with knowledge of ϕ) requires $\Omega(\min(2^d, 2^H))$ samples to output the value $V^\star(s_0)$ up to constant additive error (with prob. ≥ 0.9).

0.1 close

Linearly Realizability is Not Sufficient for RL

Theorem:

- [Weisz, Amortila, Szepesvári '21]:
  There exists an MDP and a ϕ satisfying A1 s.t any online RL algorithm (with knowledge of ϕ) requires $\Omega(\min(2^d, 2^H))$ samples to output the value $V^\star(s_0)$ up to constant additive error (with prob. ≥ 0.9).
- [Wang, Wang, K. '21]:
  Let's make the problem even easier, where we also assume:
  A2 (Large Suboptimality Gap): for all $a \neq \pi^\star(s)$, $V_h^\star(s) - Q_h^\star(s, a) \geq 1/16.$
  The lower bound holds even with **both** A1 and A2.

orig'

min { 2^A, 2^H, 2^d }

Linearly Realizability is Not Sufficient for RL

~~xxx~~ Not possible

with sim access

Theorem:

- [Weisz, Amortila, Szepesvári '21]:
  There exists an MDP and a ϕ satisfying A1 s.t any online RL algorithm (with knowledge of ϕ) requires $\Omega(\min(2^d, 2^H))$ samples to output the value $V^\star(s_0)$ up to constant additive error (with prob. ≥ 0.9).

  ~~xxx~~ possible with sim. access.
- [Wang, Wang, K. '21]:
  Let's make the problem even easier, where we also assume:
  A2 (Large Suboptimality Gap): for all $a \neq \pi^\star(s)$, $V_h^\star(s) - Q_h^\star(s,a) \geq 1/16$.
  The lower bound holds even with **both** A1 and A2.

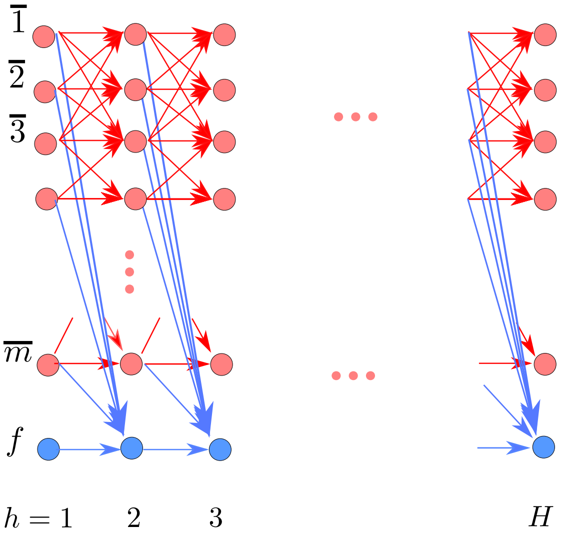
Comments: An exponential separation between online RL vs simulation access.

[Du, K., Wang, Yang '20]: A1+A2+simulator access (input: any s, a; output: $s' \sim P(\cdot \mid s, a), r(s, a)$)

$\Longrightarrow$ there is sample efficient approach to find an ϵ-opt policy.

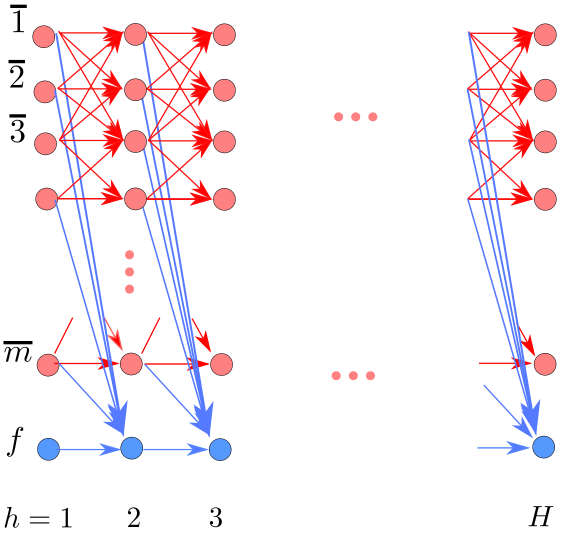
Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')

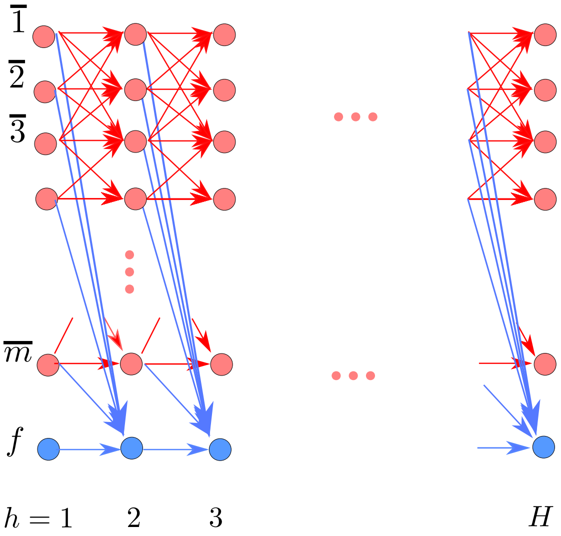


Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')



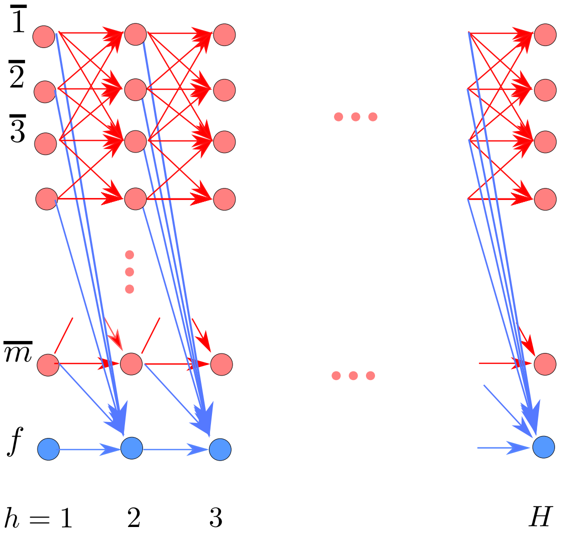
- m is an integer (we will set $m \approx 2^d$)



Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')

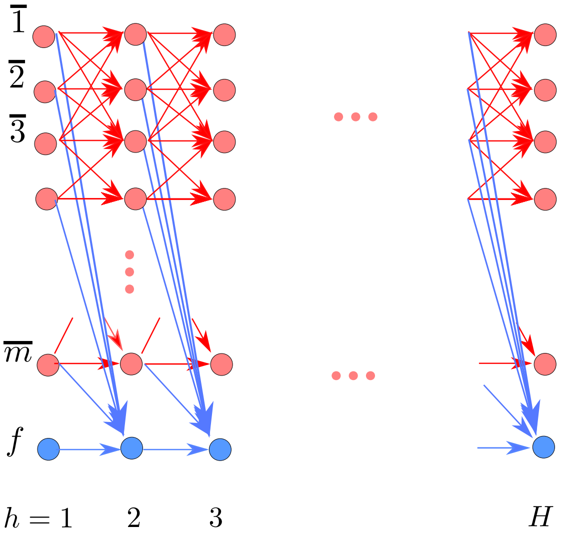
- m is an integer (we will set $m \approx 2^d$)
- the state space: $\{\bar{1}, \cdots, \bar{m}, f\}$



Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')

- m is an integer (we will set $m \approx 2^d$)
- the state space: $\{\bar{1}, \cdots, \bar{m}, f\}$
- call the special state f a "terminal state".



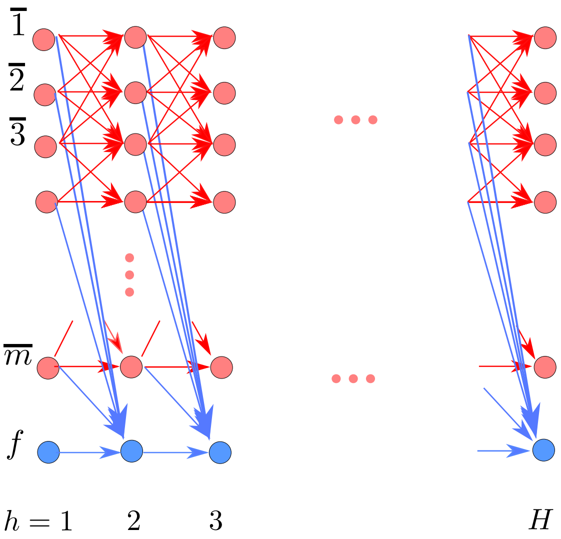
Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')

- m is an integer (we will set $m \approx 2^d$)
- the state space: $\{\bar{1}, \cdots, \bar{m}, f\}$
- call the special state f a "terminal state".
- at state $\bar{i}$, the feasible actions set is $[m]\backslash\{i\}$

  at f, the feasible action set is $[m-1]$.

  i.e. there are $m-1$ feasible actions at each state.



Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')

- m is an integer (we will set $m \approx 2^d$)
- the state space: $\{\bar{1}, \cdots, \bar{m}, f\}$
- call the special state f a "terminal state".
- at state $\bar{i}$, the feasible actions set is $[m] \backslash \{i\}$

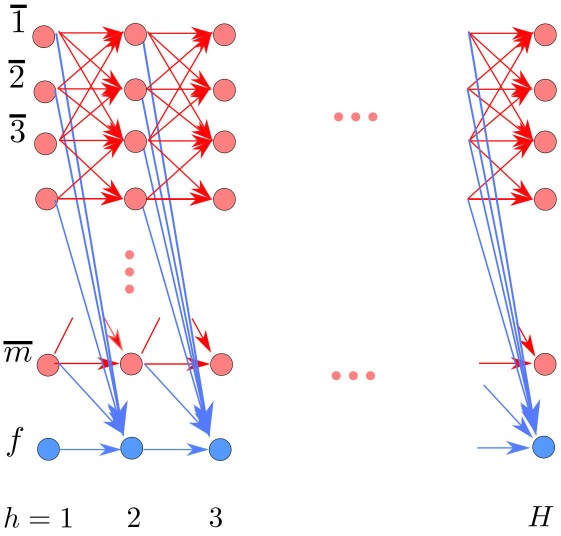
  at f, the feasible action set is $[m-1]$.

  i.e. there are $m-1$ feasible actions at each state.
- each MDP in this family is specified by an index $a^* \in [m]$ and denoted by $\mathscr{M}_{a^*}$.

  i.e. there are m MDPs in this family.

Construction Sketch: a Hard MDP Family

(A ``leaking complete graph'')



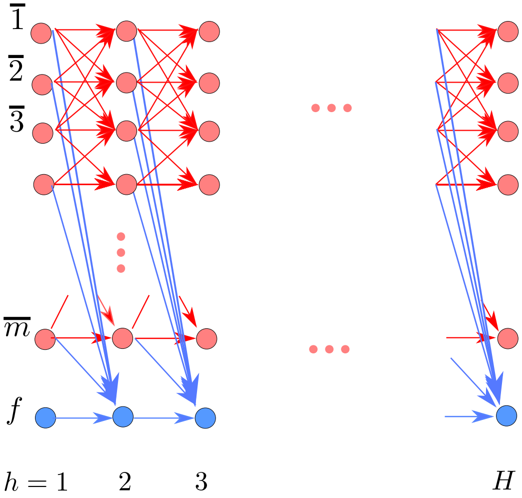
- m is an integer (we will set $m \approx 2^d$)
- the state space: $\{\bar{1}, \cdots, \bar{m}, f\}$
- call the special state f a "terminal state".
- at state $\bar{i}$, the feasible actions set is $[m]\backslash\{i\}$
  at f, the feasible action set is $[m-1]$.
  i.e. there are $m-1$ feasible actions at each state.
- each MDP in this family is specified by an index $a^* \in [m]$ and denoted by $\mathscr{M}_{a^*}$.
  i.e. there are m MDPs in this family.

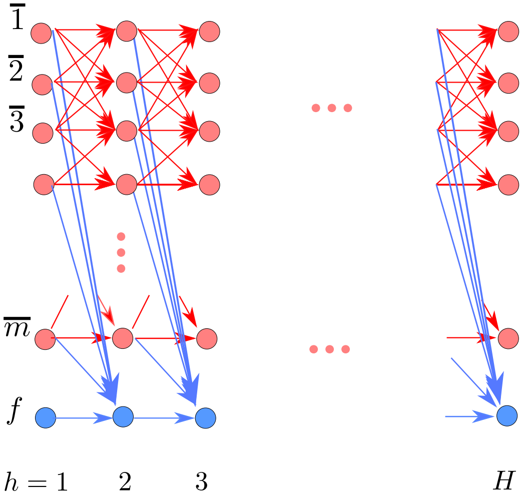
Lemma: For any $\gamma > 0$, there exist $m = \lfloor \exp(\frac{1}{8}\gamma^2 d) \rfloor$ unit vectors $\{v_1, \cdots, v_m\}$ in R^d s.t. $\forall i, j \in [m]$ and $i \neq j$, $|\langle v_i, v_j \rangle| \leq \gamma$.

We will set $\gamma = 1/4$.

(proof: Johnson-Lindenstrauss)

The construction, continued





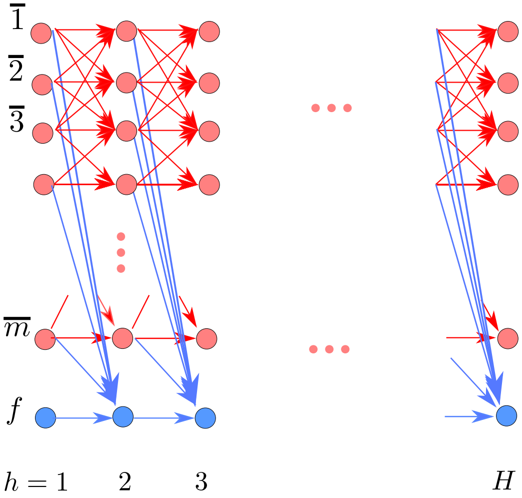
The construction, continued

Pr(next state | curr. state, false a)

- Transitions: $s_0 \sim \text{Uniform}([m])$.
  $\Pr[f \,|\, \overline{a_1}, a^*] = 1$,

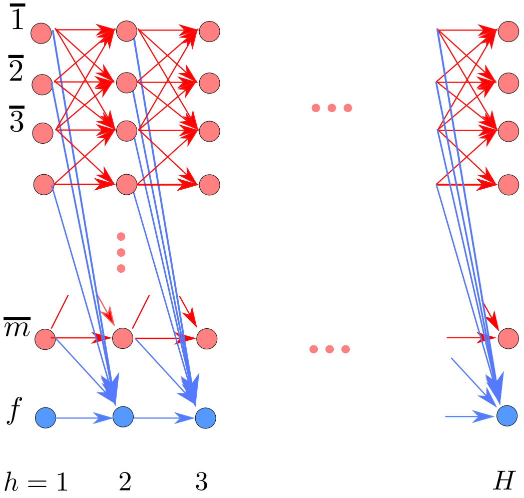
$$\Pr[\,\cdot\,|\,\overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\gamma \end{cases}, (a_2 \neq a^*, a_2 \neq a_1)$$

$$\Pr[f \,|\, f, \cdot\,] = 1.$$



The construction, continued

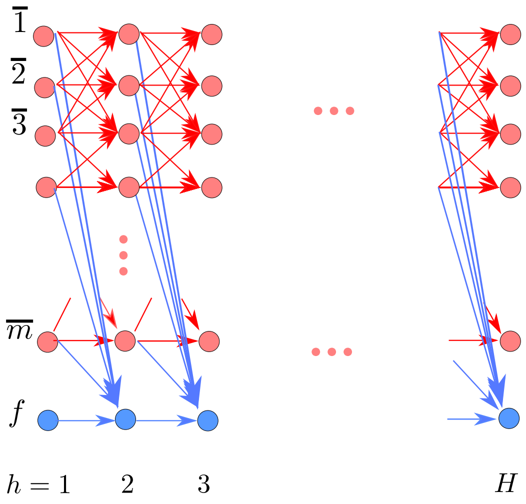
- Transitions: $s_0 \sim \text{Uniform}([m])$.
  $\Pr[f \mid \overline{a_1}, a^*] = 1$,
  $$\Pr[\,\cdot \mid \overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\gamma \end{cases}, (a_2 \neq a^*, a_2 \neq a_1)$$
  $\Pr[f \mid f, \cdot\,] = 1$.
- After taking action a_2, the next state is either $\overline{a_2}$ or f. This MDP looks like a ``leaking complete graph''



The construction, continued

- Transitions: $s_0 \sim \text{Uniform}([m])$.
  $\Pr[f \,|\, \overline{a_1}, a^*] = 1,$
  $$\Pr[\,\cdot\,|\,\overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\gamma \end{cases}, (a_2 \neq a^*, a_2 \neq a_1)$$
  $\Pr[f \,|\, f, \cdot\,] = 1.$
- After taking action a_2, the next state is either $\overline{a_2}$ or f. This MDP looks like a ``leaking complete graph''
- It is possible to visit any other state (except for $\overline{a^*}$); however, there is at least $1 - 3\gamma = 1/4$ probability of going to the terminal state f.

The construction, continued



γ $(i \neq j)$ γ

$-\frac{1}{4} \le \langle v_i, v_j \rangle \le \frac{1}{4}$

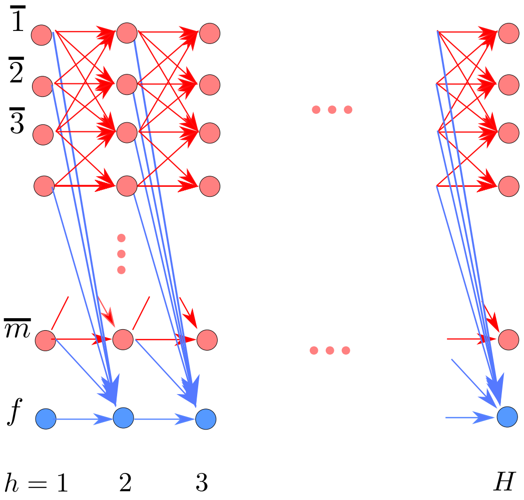
- Transitions: $s_0 \sim$ Uniform$([m])$.
  $\Pr[f \mid \overline{a_1}, a^*] = 1,$

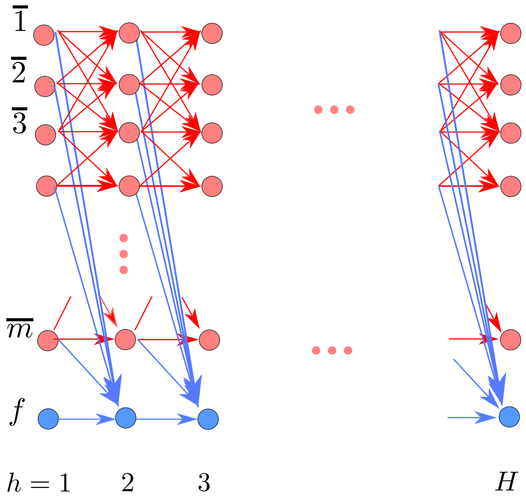
  $\frac{1}{4} \le$ $\le \frac{3}{4}$

  $$\Pr[\,\cdot \mid \overline{a_1}, a_2] = \begin{cases} \overline{a_2} : \left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \\ f : 1 - \left\langle v(a_1), v(a_2) \right\rangle - 2\gamma \end{cases}, (a_2 \neq a^*, a_2 \neq a_1)$$

  $$\Pr[f \mid f, \cdot\,] = 1.$$
- After taking action a_2, the next state is either $\overline{a_2}$ or f. This MDP looks like a ``leaking complete graph''
- It is possible to visit any other state (except for $\overline{a^*}$); however, there is at least $1 - 3\gamma = 1/4$ probability of going to the terminal state f.
- The transition probabilities are indeed valid, because
  $$0 < \gamma \le \left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \le 3\gamma < 1.$$

The construction, continued





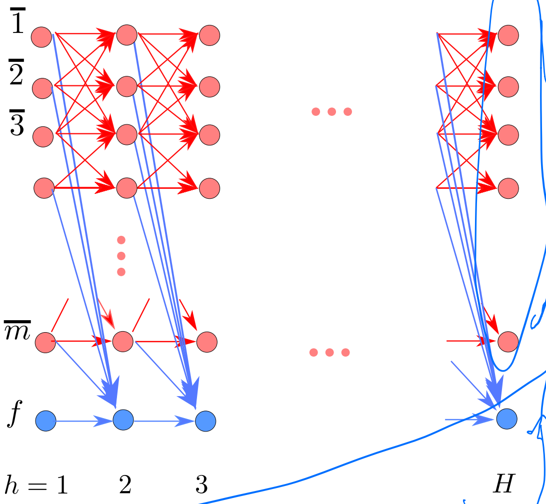
The construction, continued

- Features: of dimension d defined as:

$$\phi(\overline{a_1}, a_2) := \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right) \cdot v(a_2), \quad \forall a_1 \neq a_2$$

$$\phi(f, \cdot\ ) := \mathbf{0}$$

note: the feature map does not depend of a^*.



The construction, continued

- Features: of dimension d defined as:

$$\phi(\overline{a_1}, a_2) := \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right) \cdot v(a_2), \quad \forall a_1 \neq a_2$$

$$\phi(f, \cdot) := \mathbf{0}$$

note: the feature map does not depend of a^*.

- Rewards: $\frac{1}{4} = \gamma \leq$

for $1 \leq h < H$,

$$R_h(\overline{a_1}, a^*) := \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma,$$

$\leq 3\gamma = 3/4$

$$R_h(\overline{a_1}, a_2) := -2\gamma \left[\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right], \quad a_2 \neq a^*, a_2 \neq a$$

$\leq -2\gamma^2 \approx -\frac{1}{8}$

$$R_h(f, \cdot) := 0.$$

for $h = H$,

$$r_H(s, a) := \langle \phi(s, a), v(a^*) \rangle$$

critical

we don't get to quickly

depends on a^*

Verifying the Assumptions: Realizability and the Large Gap

Verifying the Assumptions: Realizability and the Large Gap

Lemma: For all (s, a), we have $Q_h^*(s, a) = \langle \phi(s, a), v(a^*) \rangle$ and the "gap" is $\geq \gamma/4$.

Verifying the Assumptions: Realizability and the Large Gap

Lemma: For all (s, a), we have $Q_h^*(s, a) = \langle \phi(s, a), v(a^*) \rangle$ and the "gap" is $\geq \gamma/4$.

Proof: throughout $a_2 \neq a^*$

Verifying the Assumptions: Realizability and the Large Gap

Lemma: For all (s, a), we have $Q_h^*(s, a) = \langle \phi(s, a), v(a^*) \rangle$ and the "gap" is $\geq \gamma/4$.

Proof: throughout $a_2 \neq a^*$

- First, let's verify $Q^\pi(s, a) = \langle \phi(s, a), v(a^*) \rangle$ is the value of the policy $\pi(\overline{a}) = a^\star$. By induction, we can show:

$\phi(s, a)$

$$Q_h^\pi(\overline{a_1}, a_2) = \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right) \cdot \left\langle v(a_2), v(a^*) \right\rangle,$$

$$Q_h^\pi(\overline{a_1}, a^*) = \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma$$

$$\phi(\overline{a}_1, a_2) = \left(\langle v(a_1), v(a_2) \rangle + 2\gamma \right) \vec{v}(a_2)$$

Verifying the Assumptions: Realizability and the Large Gap

Lemma: For all (s, a), we have $Q_h^*(s, a) = \langle \phi(s, a), v(a^*) \rangle$ and the "gap" is $\geq \gamma/4$.

Proof: throughout $a_2 \neq a^*$

- First, let's verify $Q^\pi(s, a) = \langle \phi(s, a), v(a^*) \rangle$ is the value of the policy $\pi(\bar{a}) = a^\star$. By induction, we can show:

$$Q_h^\pi(\overline{a_1}, a_2) = \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right) \cdot \left\langle v(a_2), v(a^*) \right\rangle,$$

$$Q_h^\pi(\overline{a_1}, a^*) = \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma$$

- Proving optimality: for $a_2 \neq a^*, a_1$

$$Q_h^\pi(\overline{a_1}, a_2) \leq 3\gamma^2, \quad Q_h^\pi(\overline{a_1}, a^*) = \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma \geq \gamma > 3\gamma^2$$

$\implies$ π is optimal

Verifying the Assumptions: Realizability and the Large Gap

Lemma: For all (s, a), we have $Q_h^*(s, a) = \langle \phi(s, a), v(a^*) \rangle$ and the "gap" is $\geq \gamma/4$.

Proof: throughout $a_2 \neq a^*$

- First, let's verify $Q^\pi(s, a) = \langle \phi(s, a), v(a^*) \rangle$ is the value of the policy $\pi(\overline{a}) = a^\star$.
  By induction, we can show:

$$Q_h^\pi(\overline{a_1}, a_2) = \left(\left\langle v(a_1), v(a_2) \right\rangle + 2\gamma \right) \cdot \left\langle v(a_2), v(a^*) \right\rangle,$$

$$Q_h^\pi(\overline{a_1}, a^*) = \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma$$

- Proving optimality: for $a_2 \neq a^*, a_1$

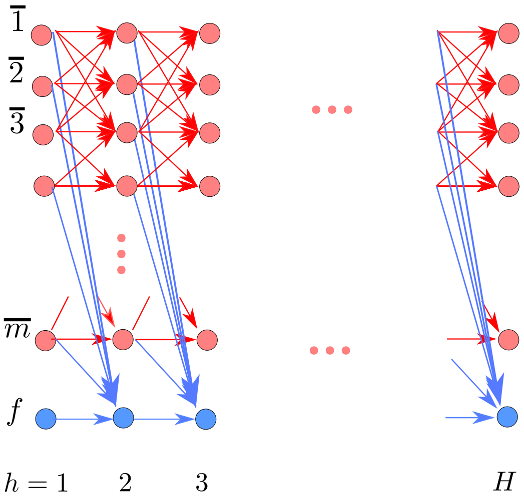
$$Q_h^\pi(\overline{a_1}, a_2) \leq 3\gamma^2, \quad Q_h^\pi(\overline{a_1}, a^*) = \left\langle v(a_1), v(a^*) \right\rangle + 2\gamma \geq \gamma > 3\gamma^2$$

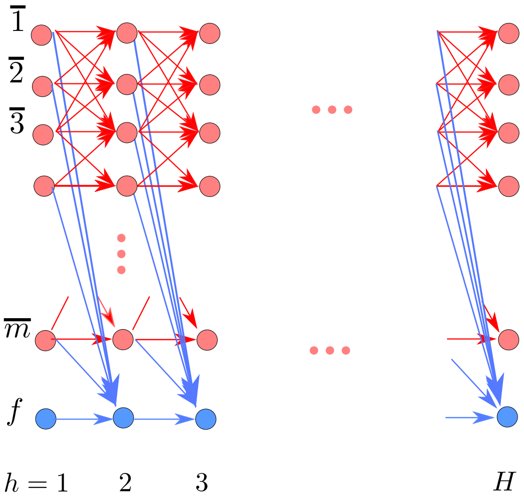
$\implies$ π is optimal

- Proving the large gap: for $a_2 \neq a^*$

$$V_h^*(\overline{a_1}) - Q_h^*(\overline{a_1}, a_2) = Q_h^\pi(\overline{a_1}, a^*) - Q_h^\pi(\overline{a_1}, a_2) > \gamma - 3\gamma^2 \geq \frac{1}{4}\gamma\,.$$

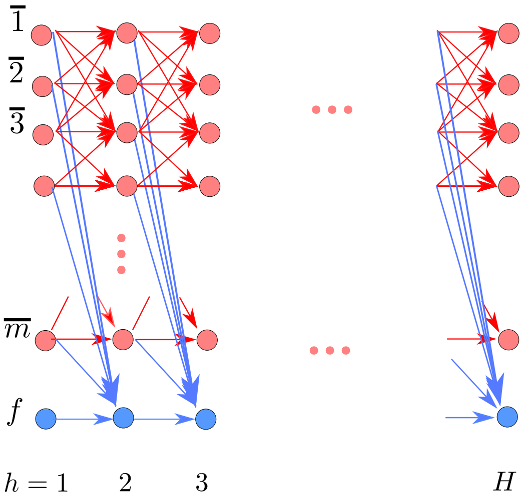
The information theoretic proof:





The information theoretic proof:

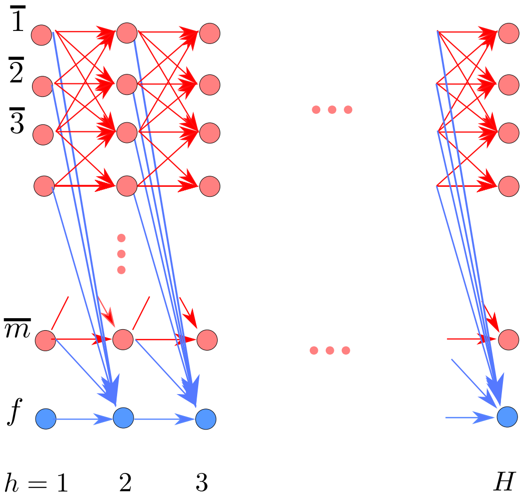
Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

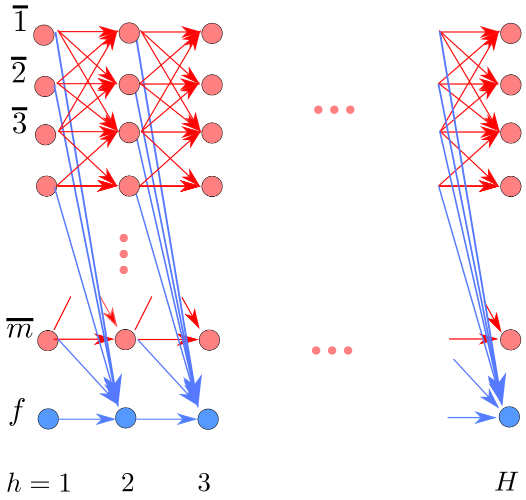
- Features: The construction of ϕ does not depend on $a^\star$.



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

- Features: The construction of ϕ does not depend on $a^\star$.
- Transitions: if we take a^*, only then does the dynamics leak info about a^* (but there $O(2^d)$ actions)



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

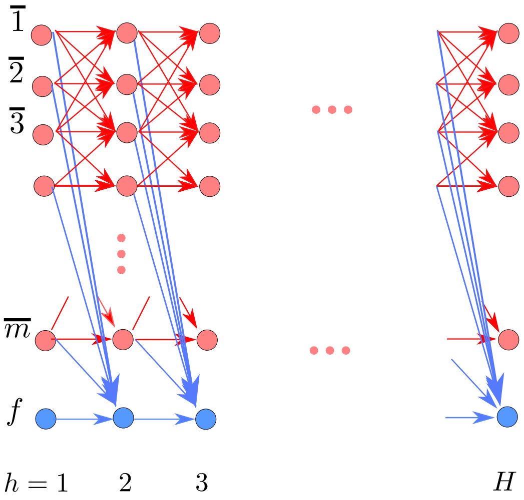
- Features: The construction of ϕ does not depend on $a^\star$.
- Transitions: if we take a^*, only then does the dynamics leak info about a^* (but there $O(2^d)$ actions)
- Rewards: two cases which leak info about $a^\star$

  (1) if we take a^* at any h, then reward leaks info about a^* (but there $\Omega(2^d)$ actions) $\left(\frac{4}{3}\right)^d$

  (2) also, if we terminate at $s_H \neq f$, then the reward r_H leaks info about on a^*

  - But there is always at least $1/4$ chance of moving to f
  - So need at least $\Omega(2^H)$ trajectories to hit $s_H \neq f$

    $\left(\frac{4}{3}\right)^H$



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

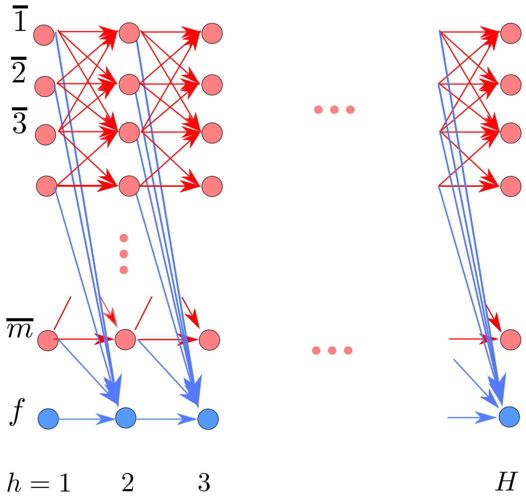
- Features: The construction of ϕ does not depend on $a^\star$.
- Transitions: if we take a^*, only then does the dynamics leak info about a^* (but there $O(2^d)$ actions)
- Rewards: two cases which leak info about $a^\star$

  (1) if we take a^* at any h, then reward leaks info about a^* (but there $O(2^d)$ actions)

  (2) also, if we terminate at $s_H \neq f$, then the reward r_H leaks info about on a^*

  - But there is always at least $1/4$ chance of moving to f
  - So need at least $O(2^H)$ trajectories to hit $s_H \neq f$

$\Longrightarrow$ need $\Omega(\min(2^d, 2^H))$ samples to discover $\mathscr{M}_{a^*}$.



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

- Features: The construction of ϕ does not depend on $a^\star$.
- Transitions: if we take a^*, only then does the dynamics leak info about a^* (but there $O(2^d)$ actions)
- Rewards: two cases which leak info about $a^\star$

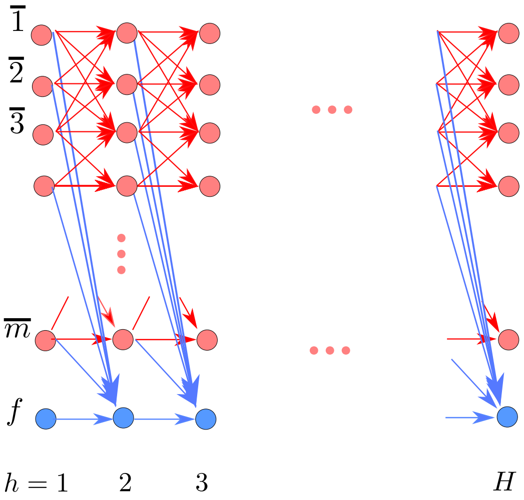
  (1) if we take a^* at any h, then reward leaks info about a^* (but there $O(2^d)$ actions)

  (2) also, if we terminate at $s_H \neq f$, then the reward r_H leaks info about on a^*

  - But there is always at least $1/4$ chance of moving to f
  - So need at least $O(2^H)$ trajectories to hit $s_H \neq f$

$\Longrightarrow$ need $\Omega(\min(2^d, 2^H))$ samples to discover $\mathscr{M}_{a^*}$.

Caveats: Haven't handled the state $\bar{a}^*$ cafefully.



The information theoretic proof:

Proof: When is info revealed about $\mathscr{M}_{a^*}$, indexed by a^*?

- Features: The construction of ϕ does not depend on $a^\star$.
- Transitions: if we take a^*, only then does the dynamics leak info about a^* (but there $O(2^d)$ actions)
- Rewards: two cases which leak info about $a^\star$
  (1) if we take a^* at any h, then reward leaks info about a^* (but there $O(2^d)$ actions)
  (2) also, if we terminate at $s_H \neq f$, then the reward r_H leaks info about on a^*
  - But there is always at least $1/4$ chance of moving to f
  - So need at least $O(2^H)$ trajectories to hit $s_H \neq f$

$\Longrightarrow$ need $\Omega(\min(2^d, 2^H))$ samples to discover $\mathscr{M}_{a^*}$.

Caveats: Haven't handled the state $\overline{a}^*$ cafefully.

Open Problem: Can we prove a lower bound with $A = 2$ actions?

Interlude:

Are these issues relevant in practice?

These Representational Issues are Relevant for Practice!

(related concepts: distribution shift, “the deadly triad”, offline RL)

These Representational Issues are Relevant for Practice!

(related concepts: distribution shift, "the deadly triad", offline RL)

Theorem [Wang, Foster, K., '20]:

Analogue for "offline" RL: linearly realizability is also not sufficient.

These Representational Issues are Relevant for Practice!

(related concepts: distribution shift, "the deadly triad", offline RL)

Theorem [Wang, Foster, K., '20]:

Analogue for "offline" RL: linearly realizability is also not sufficient.

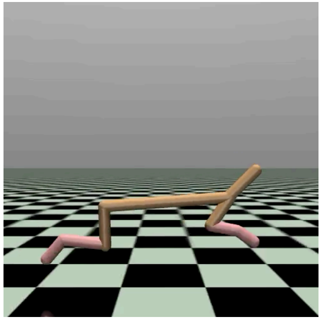
Practice: [Wang, Wu, Salakhutdinov, K., 2021]:

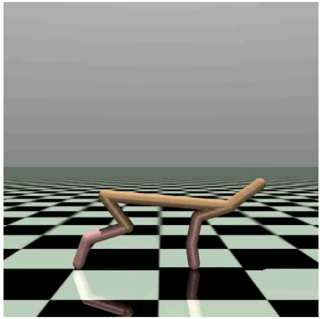
Does it matter in practice? Say given good ""deep-pre-trained- features"? YES!

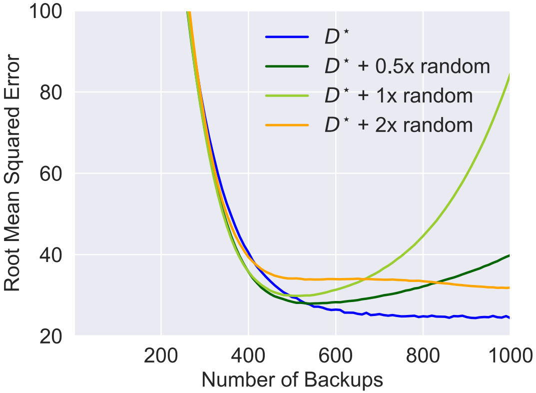
Offline dataset is a mix of two sources:
running & random

Use SL to evaluate the running policy with "deep-pre-trained- features"

Massive error amplification even with 50/50% mixed offline data





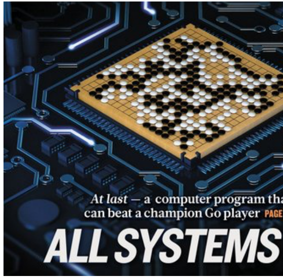


Part-3:
What are sufficient conditions?

Is there a common theme to positive results?

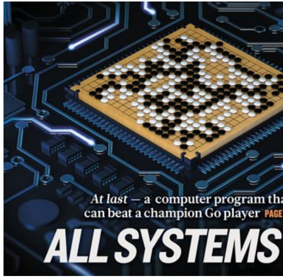
Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?



Provable Generalization in RL

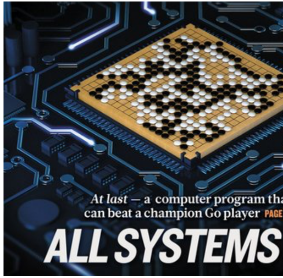
Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?



Agnostically/best-in-class? **NO.**

Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?

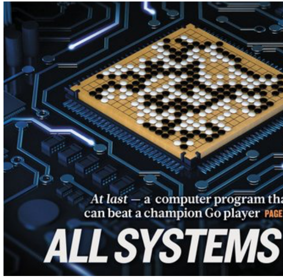


Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?



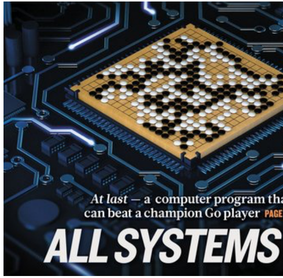
Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

- With various stronger assumptions, YES! Many special cases:
  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun ’20]
  - Linear Mixture MDPs: [Modi+’20, Ayoub+ ’20]
  - Block MDPs [Du+ ’19]
  - Factored MDPs [Sun+ ’19]

Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?



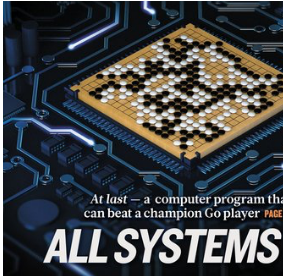
Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

- With various stronger assumptions, YES! Many special cases:
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]

Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?



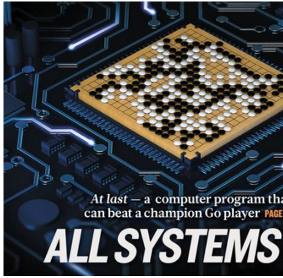
Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

- With various stronger assumptions, YES! Many special cases:
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..

Provable Generalization in RL

Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon, \text{"complexity measure"})$ samples?

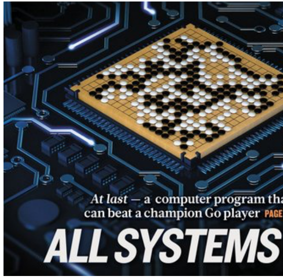


Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

- With various stronger assumptions, YES! Many special cases:
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..
- Are there structural commonalities between these underlying assumptions/models?

Provable Generalization in RL



Can we find an ϵ-opt policy with no S, A dependence and $poly(H, 1/\epsilon,$ "complexity measure"$)$ samples?

Agnostically/best-in-class? **NO.**

With linearly realizable Q^*? **Also NO.**

- With various stronger assumptions, YES! Many special cases:
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..
- Are there structural commonalities between these underlying assumptions/models?
  - almost: **Bellman rank [Jiang+ '17]; Witness rank [Wen+ '19]**

Intuition: properties of linear bandits

(back to $H = 1$ RL problem)

Intuition: properties of linear bandits

(back to $H = 1$ RL problem)

- Linear (contextual) bandits:
  context: s action: a
  observed reward: $r = w^\star \cdot \phi(s, a) + \epsilon$

Intuition: properties of linear bandits

(back to $H = 1$ RL problem)

- Linear (contextual) bandits:
  context: s action: a
  observed reward: $r = w^\star \cdot \phi(s, a) + \epsilon$
- Hypothesis class: $\{f(s, a) = w(f) \cdot \phi(s, a),\ w \in \mathscr{W}\}$
  Let π_f be the greedy policy for f

Intuition: properties of linear bandits
(back to $H = 1$ RL problem)

- Linear (contextual) bandits:
  context: s action: a
  observed reward: $r = w^\star \cdot \phi(s, a) + \epsilon$
- Hypothesis class: $\{f(s, a) = w(f) \cdot \phi(s, a),\ w \in \mathscr{W}\}$
  Let π_f be the greedy policy for f

An important structural property:

Intuition: properties of linear bandits

(back to $H = 1$ RL problem)

- Linear (contextual) bandits:
  context: s action: a
  observed reward: $r = w^\star \cdot \phi(s, a) + \epsilon$
- Hypothesis class: $\{f(s, a) = w(f) \cdot \phi(s, a),\ w \in \mathscr{W}\}$
  Let π_f be the greedy policy for f

An important structural property:

- Data reuse: difference between f and r is estimable when playing π_g

$$E_{a\sim\pi_g}[f(s, a) - r] = \left\langle w(f) - w^\star, E_{\pi_g}[\phi(s, a)] \right\rangle$$

Special case: linear Bellman complete classes

(stronger conditions over linear realizability)

Special case: linear Bellman complete classes

(stronger conditions over linear realizability)

- Linear hypothesis class: $\mathscr{F} = \{Q_f : Q_f(s, a) = w(f) \cdot \phi(s, a)\}$

  with associated (greedy) value $V_f(s)$ and (greedy) policy: π_f

Special case: linear Bellman complete classes
(stronger conditions over linear realizability)

- Linear hypothesis class: $\mathscr{F} = \{Q_f : Q_f(s,a) = w(f) \cdot \phi(s,a)\}$
  with associated (greedy) value $V_f(s)$ and (greedy) policy: π_f
- Completeness: suppose $\mathscr{T}(Q_f) \in \mathscr{F}$

$\mathcal{T}(Q)$ Bellman backup operator.

$$\mathcal{T}(Q)_{s,a} = \max_a \left(r(s,a) + \mathbb{E}_{s' \sim P_{s,a}} \max Q_{s',a'} \right)$$

Special case: linear Bellman complete classes
(stronger conditions over linear realizability)

- Linear hypothesis class: $\mathscr{F} = \{Q_f : Q_f(s,a) = w(f) \cdot \phi(s,a)\}$

  with associated (greedy) value $V_f(s)$ and (greedy) policy: π_f
- Completeness: suppose $\mathscr{T}(Q_f) \in \mathscr{F}$
- Completes is very strong condition!

  Adding a feature to ϕ can break the completeness property.

Analogous structural property holds for $\mathscr{F}$:

Special case: linear Bellman complete classes

(stronger conditions over linear realizability)

- Linear hypothesis class: $\mathscr{F} = \{Q_f : Q_f(s,a) = w(f) \cdot \phi(s,a)\}$

  with associated (greedy) value $V_f(s)$ and (greedy) policy: π_f
- Completeness: suppose $\mathscr{T}(Q_f) \in \mathscr{F}$
- Completes is very strong condition!

  Adding a feature to ϕ can break the completeness property.

Analogous structural property holds for $\mathscr{F}$:

- Data reuse: Bellman error of any f is estimable when playing π_g:

  $E_{\pi_g}\left[Q_f(s_h, a_h) - r(s_h, a_h) - V_f(s_{h+1})\right] \leq \left\langle w_h(f) - \mathscr{T}\left(w_h(f)\right), E_{\pi_g}\left[\phi(s_h, a_h)\right]\right\rangle$

  (where expectation is with respect to trajectories under π_g)

Special case: linear Bellman complete classes

(stronger conditions over linear realizability)

- Linear hypothesis class: $\mathcal{F} = \{Q_f : Q_f(s,a) = w(f) \cdot \phi(s,a)\}$

  with associated (greedy) value $V_f(s)$ and (greedy) policy: π_f
- Completeness: suppose $\mathcal{T}(Q_f) \in \mathcal{F}$
- Completes is very strong condition!

  Adding a feature to ϕ can break the completeness property.

Analogous structural property holds for $\mathcal{F}$:

- Data reuse: Bellman error of any f is estimable when playing π_g:

  $E_{\pi_g}\left[Q_f(s_h, a_h) - r(s_h, a_h) - V_f(s_{h+1})\right] \leq \left\langle w_h(f) - \mathcal{T}\left(w_h(f)\right), E_{\pi_g}\left[\phi(s_h, a_h)\right]\right\rangle$

  (where expectation is with respect to trajectories under π_g)

- (recall) Bellman optimality: suppose $Q^\star - \mathcal{T}(Q^\star) = 0$

BiLinear Regret Classes: structural properties to enable generalization in RL

BiLinear Regret Classes: structural properties to enable generalization in RL

- Hypothesis class: $\{f \in \mathcal{F}\}$,
  with associated state-action value, (greedy) value and policy: $Q_f(s, a), V_f(s), \pi_f$
  - can be model based or model-free class.

BiLinear Regret Classes: structural properties to enable generalization in RL

- Hypothesis class: $\{f \in \mathcal{F}\}$,
  with associated state-action value, (greedy) value and policy: $Q_f(s, a), V_f(s), \pi_f$
  - can be model based or model-free class.

Def: A $(\mathcal{F}, \ell)$ forms an (implicit) Bilinear class class if:

BiLinear Regret Classes: structural properties to enable generalization in RL

- Hypothesis class: $\{f \in \mathscr{F}\}$,
  with associated state-action value, (greedy) value and policy: $Q_f(s,a), V_f(s), \pi_f$
  - can be model based or model-free class.

Def: A $(\mathscr{F}, \ell)$ forms an (implicit) Bilinear class class if:

- Bilinear regret: on-policy difference between claimed reward and true reward

$$\left| E_{\pi_f}\left[Q_f(s_h,a_h) - r(s_h,a_h) - V_f(s_{h+1})\right] \right| \leq \langle w_h(f) - w_h^\star, \Phi_h(f) \rangle$$

BiLinear Regret Classes: structural properties to enable generalization in RL

- Hypothesis class: $\{f \in \mathcal{F}\}$,
  with associated state-action value, (greedy) value and policy: $Q_f(s,a), V_f(s), \pi_f$
  - can be model based or model-free class.

Def: A $(\mathcal{F}, \ell)$ forms an (implicit) Bilinear class class if:

- Bilinear regret: on-policy difference between claimed reward and true reward

$$\left| E_{\pi_f}\left[Q_f(s_h, a_h) - r(s_h, a_h) - V_f(s_{h+1})\right] \right| \leq \langle w_h(f) - w_h^\star, \Phi_h(f) \rangle$$

- Data reuse: there is function $\ell_f(s, a, s', g)$ s.t.

$$E_{\pi_f}\left[\ell_f(s_h, a_h, s_{h+1}, g)\right] = \langle w_h(g) - w_h^\star, \Phi_h(f) \rangle$$

Theorem: Structural Commonalities and Bilinear Classes

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun ’20]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun ’20]
  - Linear Mixture MDPs: [Modi+’20, Ayoub+ ’20]
  - Block MDPs [Du+ ’19]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun ’20]
  - Linear Mixture MDPs: [Modi+’20, Ayoub+ ’20]
  - Block MDPs [Du+ ’19]
  - Factored MDPs [Sun+ ’19]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang ’19]

  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$

  - Linear Bellman Completion: [Munos, ’05, Zanette+ ‘19]
    - Linear MDPs: [Wang & Yang’18]; [Jin+ ’19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun ’20]
  - Linear Mixture MDPs: [Modi+’20, Ayoub+ ’20]
  - Block MDPs [Du+ ’19]
  - Factored MDPs [Sun+ ’19]
  - Kernelized Nonlinear Regulator [K.+ ’20]
  - And more…..

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..

- (almost) all "named" models (with provable generalization) are bilinear classes
  two exceptions: deterministic linear $Q^\star$; $Q^\star$-state aggregation

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..

- (almost) all "named" models (with provable generalization) are bilinear classes
  two exceptions: deterministic linear $Q^\star$; $Q^\star$-state aggregation
- Bilinear classes generalize the: **Bellman rank [Jiang+ '17]; Witness rank [Wen+ '19]**

Theorem: Structural Commonalities and Bilinear Classes

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  The following models are bilinear classes for some discrepancy function $\ell(\cdot)$
  - Linear Bellman Completion: [Munos, '05, Zanette+ '19]
    - Linear MDPs: [Wang & Yang'18]; [Jin+ '19] (the transition matrix is low rank)
    - Linear Quadratic Regulators (LQR): standard control theory model
  - FLAMBE / Feature Selection: [Agarwal, K., Krishnamurthy, Sun '20]
  - Linear Mixture MDPs: [Modi+'20, Ayoub+ '20]
  - Block MDPs [Du+ '19]
  - Factored MDPs [Sun+ '19]
  - Kernelized Nonlinear Regulator [K.+ '20]
  - And more…..

- (almost) all "named" models (with provable generalization) are bilinear classes
  two exceptions: deterministic linear $Q^\star$; $Q^\star$-state aggregation
- Bilinear classes generalize the: **Bellman rank [Jiang+ '17]; Witness rank [Wen+ '19]**
- The framework easily leads to new models (see paper).

The Algorithm: BiLin-UCB

(specialized to the Linear Bellman Complete case)

The Algorithm: BiLin-UCB

(specialized to the Linear Bellman Complete case)

- Find the "optimistic" $f \in \mathcal{F}$:

$$\arg\max_{f} V_f(s_0) + \beta\sigma(f)$$

The Algorithm: BiLin-UCB

(specialized to the Linear Bellman Complete case)

- Find the "optimistic" $f \in \mathscr{F}$:

$$\arg\max_{f} V_f(s_0) + \beta\sigma(f)$$

- Sample m trajectories π_f and create a batch dataset:

$$D = \{(s_h, a_h, s_{h+1}) \in \text{trajectories}\}$$

The Algorithm: BiLin-UCB

(specialized to the Linear Bellman Complete case)

- Find the "optimistic" $f \in \mathscr{F}$:

$$\arg\max_{f} V_f(s_0) + \beta\sigma(f)$$

- Sample m trajectories π_f and create a batch dataset:

$$D = \{(s_h, a_h, s_{h+1}) \in \text{trajectories}\}$$

- Update the cumulative discrepancy function function $\sigma(\cdot)$

$$\sigma^2(f) \leftarrow \sigma^2(f) + \left(\sum_{(s_h, a_h, s_{h+1}) \in D} Q_f(s_h, a_h) - r(s_h, a_h) - V_f(s_{h+1}) \right)^2$$

The Algorithm: BiLin-UCB
(specialized to the Linear Bellman Complete case)

- Find the "optimistic" $f \in \mathscr{F}$:

$$\arg\max_f V_f(s_0) + \beta\sigma(f)$$

- Sample m trajectories π_f and create a batch dataset:

$$D = \{(s_h, a_h, s_{h+1}) \in \text{trajectories}\}$$

- Update the cumulative discrepancy function function $\sigma(\cdot)$

$$\sigma^2(f) \leftarrow \sigma^2(f) + \left(\sum_{(s_h,a_h,s_{h+1}) \in D} Q_f(s_h, a_h) - r(s_h, a_h) - V_f(s_{h+1}) \right)^2$$

- return: the best policy π_f found

Theorem 2: Generalization in RL

Theorem 2: Generalization in RL

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  Assume $\mathscr{F}$ is a bilinear class and the class is realizable, i.e. $Q^\star \in \mathscr{F}$.
  Using $\gamma_T^3 \cdot poly(H) \cdot \log(1/\delta)/\epsilon^2$ trajectories, the BiLin-UCB algorithm returns an ϵ-opt policy (with prob. $\geq 1 - \delta$).

Theorem 2: Generalization in RL

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  Assume $\mathscr{F}$ is a bilinear class and the class is realizable, i.e. $Q^\star \in \mathscr{F}$.
  Using $\gamma_T^3 \cdot poly(H) \cdot \log(1/\delta)/\epsilon^2$ trajectories, the BiLin-UCB algorithm returns an ϵ-opt policy (with prob. $\geq 1 - \delta$).
- again, γ_T is the max. info. gain $\gamma_T := \max_{f_0 \ldots f_{T-1} \in \mathscr{F}} \ln\det\left(I + \frac{1}{\lambda}\sum_{t=0}^{T-1} \Phi(f_t)\Phi(f_t)^\top\right)$

Theorem 2: Generalization in RL

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  Assume $\mathscr{F}$ is a bilinear class and the class is realizable, i.e. $Q^\star \in \mathscr{F}$.
  Using $\gamma_T^3 \cdot poly(H) \cdot \log(1/\delta)/\epsilon^2$ trajectories, the BiLin-UCB algorithm returns an ϵ-opt policy (with prob. $\geq 1 - \delta$).
  - again, γ_T is the max. info. gain $\gamma_T := \max_{f_0 \ldots f_{T-1} \in \mathscr{F}} \ln \det \left(I + \frac{1}{\lambda} \sum_{t=0}^{T-1} \Phi(f_t) \Phi(f_t)^\top \right)$
  - $\gamma_T \approx d \log T$ for Φ in d-dimensions

Theorem 2: Generalization in RL

- Theorem: [Du, K., Lee, Lovett, Mahajan, Sun, Wang '19]
  Assume $\mathscr{F}$ is a bilinear class and the class is realizable, i.e. $Q^\star \in \mathscr{F}$.
  Using $\gamma_T^3 \cdot poly(H) \cdot \log(1/\delta)/\epsilon^2$ trajectories, the BiLin-UCB algorithm returns an ϵ-opt policy (with prob. $\geq 1 - \delta$).
  - again, γ_T is the max. info. gain $\gamma_T := \max_{f_0 \ldots f_{T-1} \in \mathscr{F}} \ln \det \left(I + \frac{1}{\lambda} \sum_{t=0}^{T-1} \Phi(f_t)\Phi(f_t)^\top \right)$
  - $\gamma_T \approx d \log T$ for Φ in d-dimensions

- The proof is "elementary" using the elliptical potential function.
  [Dani, Hayes, K. '08]

Thanks!

- A generalization theory in RL is possible and different from SL!
  - necessary: linear realizability insufficient. need much stronger assumptions.
  - sufficient: lin. bandit theory → RL theory (bilinear classes) is rich.
    - covers known cases and new cases
    - FLAMBE: [Agarwal+ '20] feature learning possible in this framework.
  - practice: these issues are relevant ("deadly triad"/RL can be unstable)

See https://rltheorybook.github.io/ for forthcoming book!