Foundations of Reinforcement Learning

Learning and Games Bootcamp @ Simons Institute

Dylan Foster

Microsoft Research, New England

Learning and decision making

Machine learning: Predicting patterns

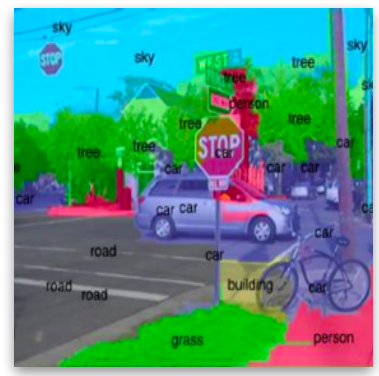
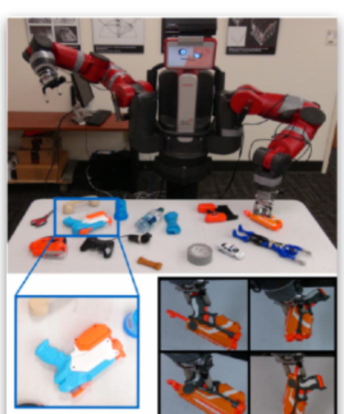
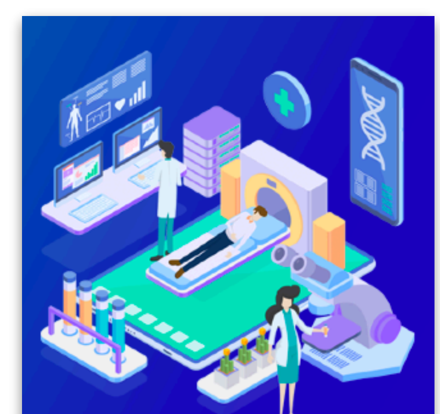


Image classification, speech recognition, machine translation

Reinforcement learning: Making *decisions*



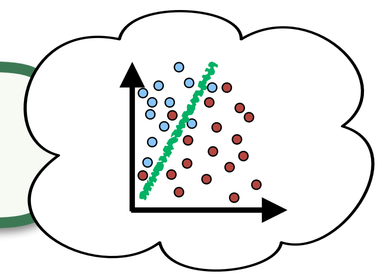


Robotics, game playing, clinical decision systems

Three problems

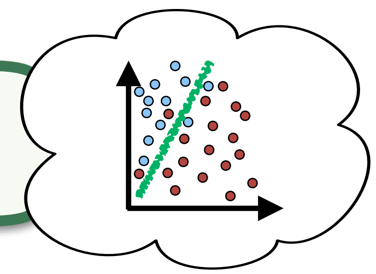
Three problems

Supervised learning

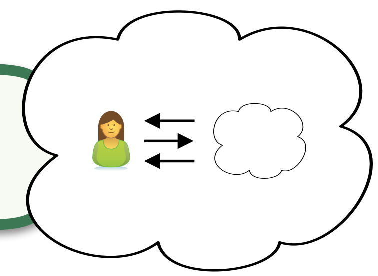


Three problems

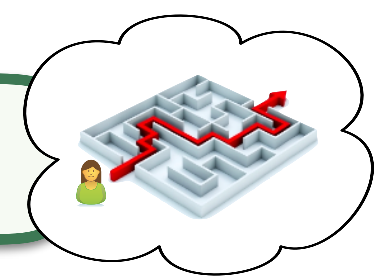
Supervised learning



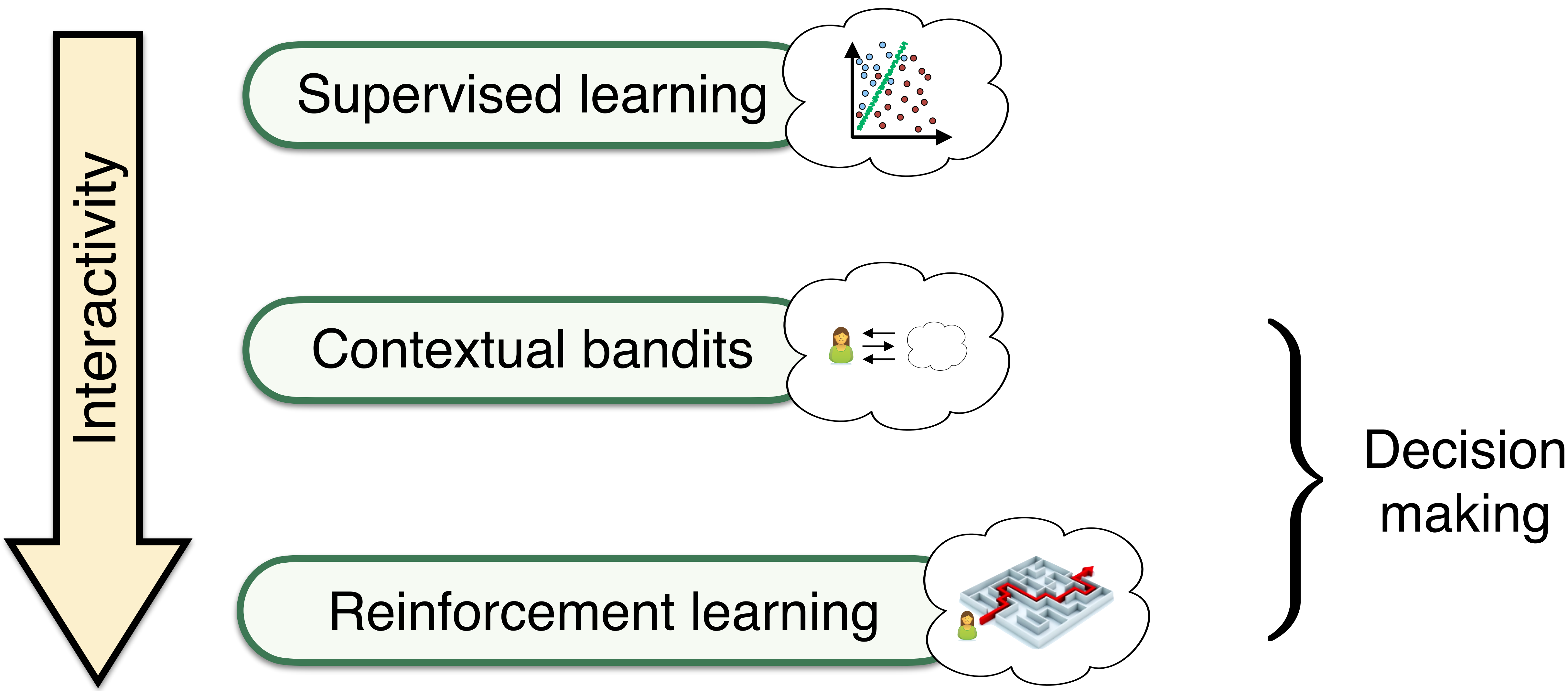
Contextual bandits



Reinforcement learning



Three problems

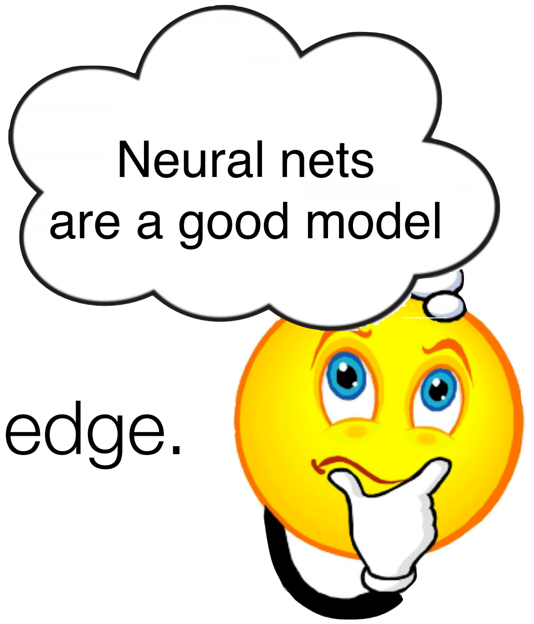


Level 1: Supervised learning

Supervised learning

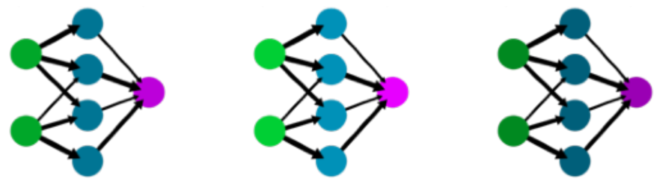
- Step 1: Pick set of models $\mathcal{F}$ that capture domain knowledge.

Level 1: Supervised learning

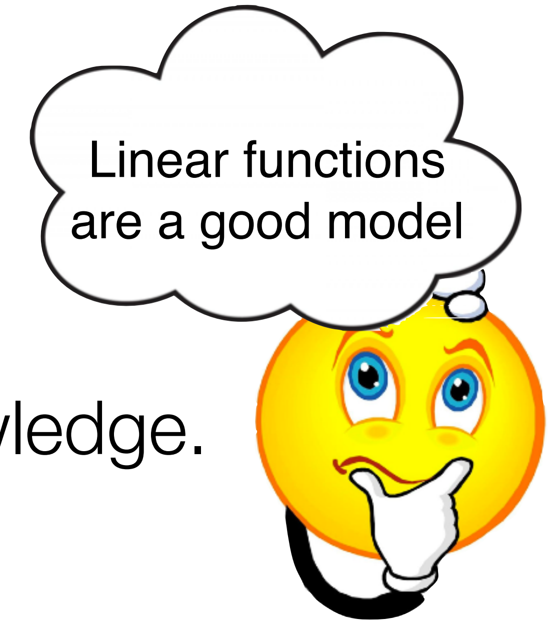


Supervised learning

- Step 1: Pick set of models $\mathcal{F}$ that capture domain knowledge.
    - Ex: Linear models, neural nets, ...

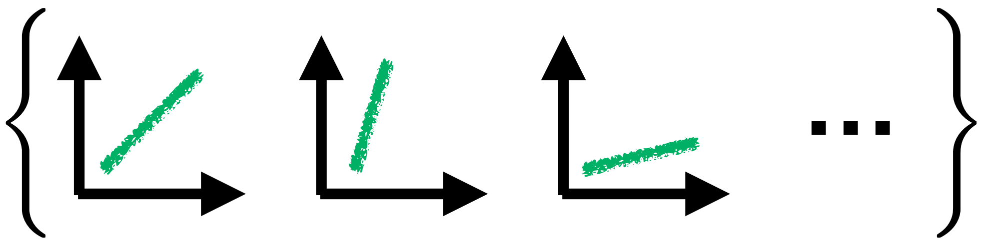
$\mathcal{F} = \{$  ... $\}$

Level 1: Supervised learning

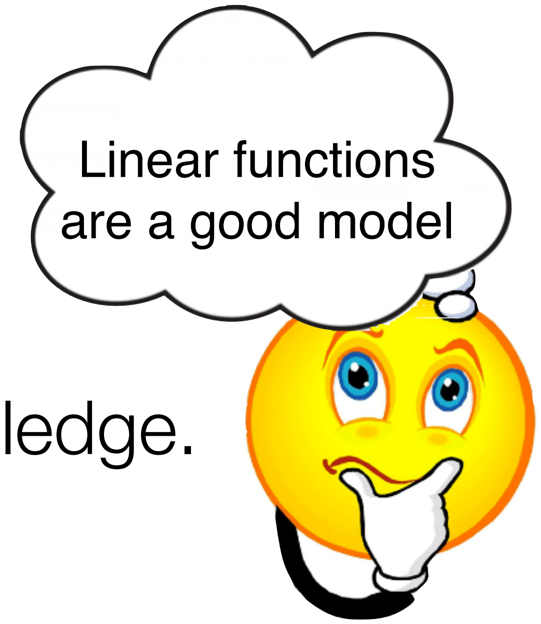


Supervised learning

- Step 1: Pick set of models $\mathcal{F}$ that capture domain knowledge.
  - Ex: Linear models, neural nets, ...

$\mathcal{F} =$ 

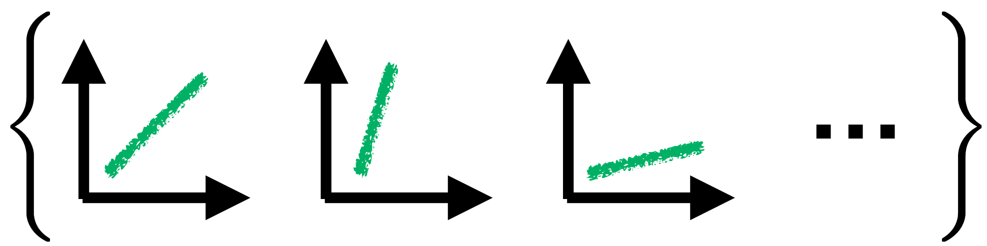
Level 1: Supervised learning



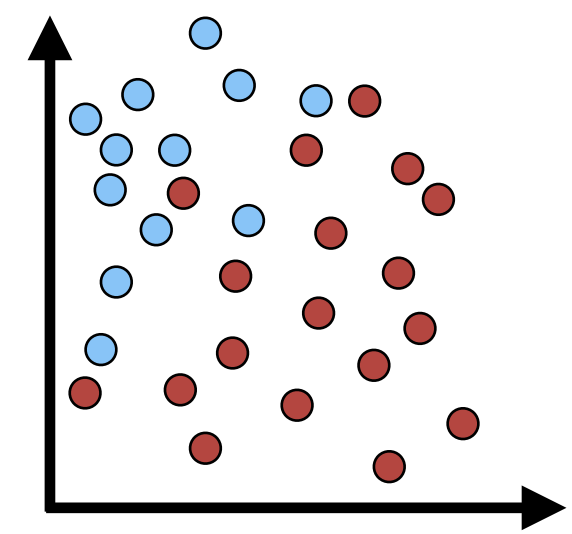
Supervised learning

- Step 1: Pick set of models $\mathcal{F}$ that capture domain knowledge.
  - Ex: Linear models, neural nets, ...

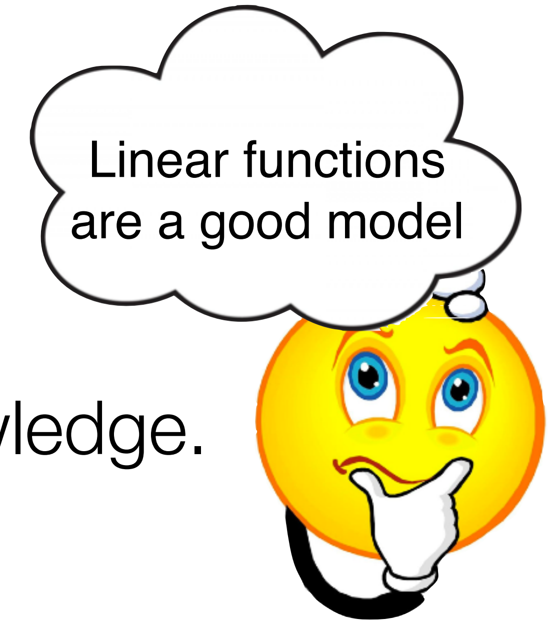
$\mathcal{F} =$



- Step 2: Gather dataset $(x_1, y_1), \ldots, (x_n, y_n)$.

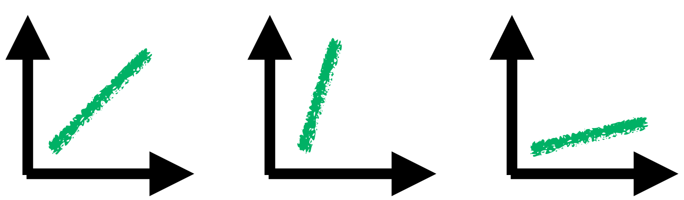


Level 1: Supervised learning



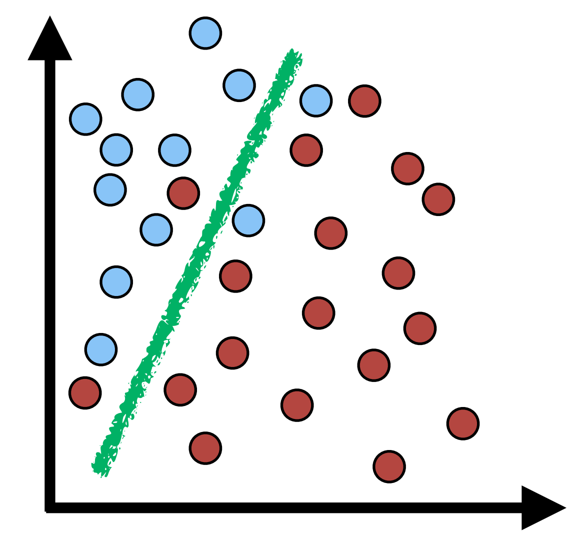
Supervised learning

- Step 1: Pick set of models $\mathcal{F}$ that capture domain knowledge.
  - Ex: Linear models, neural nets, ...

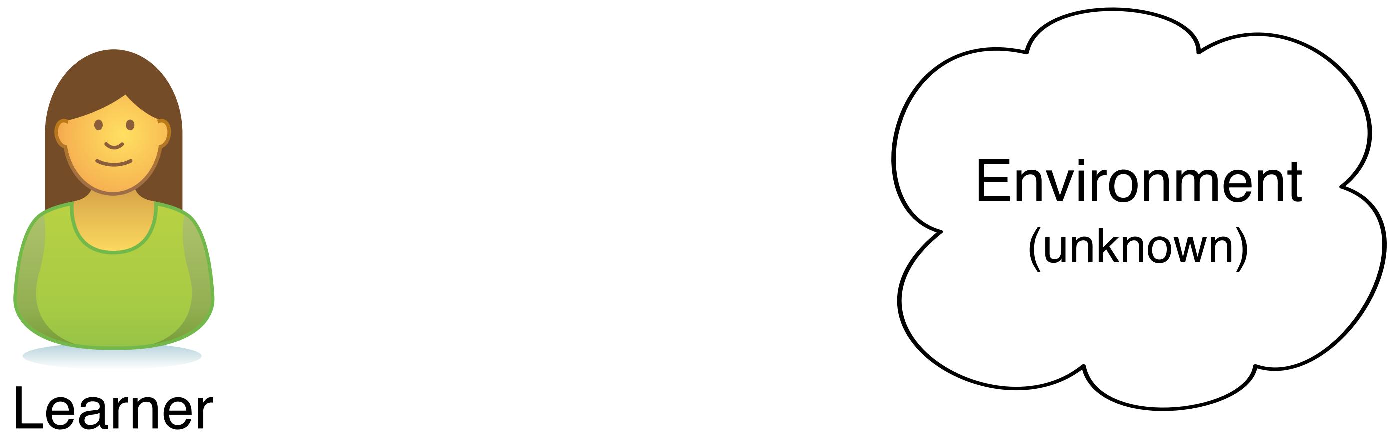
$\mathcal{F} = \{$  $\ldots \}$

- Step 2: Gather dataset $(x_1, y_1), \ldots, (x_n, y_n)$.

- Step 3: Return $\widehat{f} \in \mathcal{F}$ that fits data well.

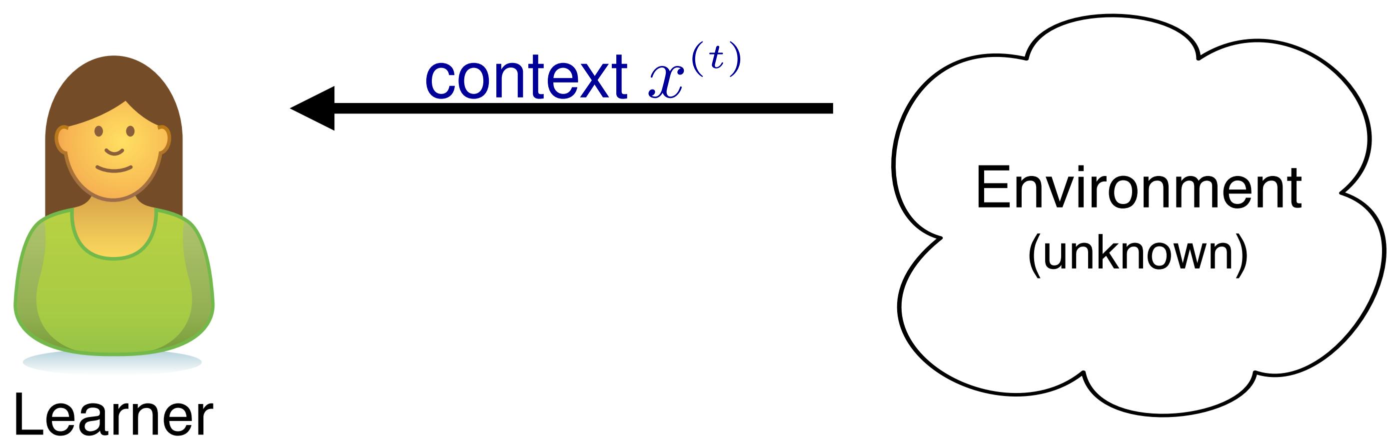


Level 2: Contextual bandits



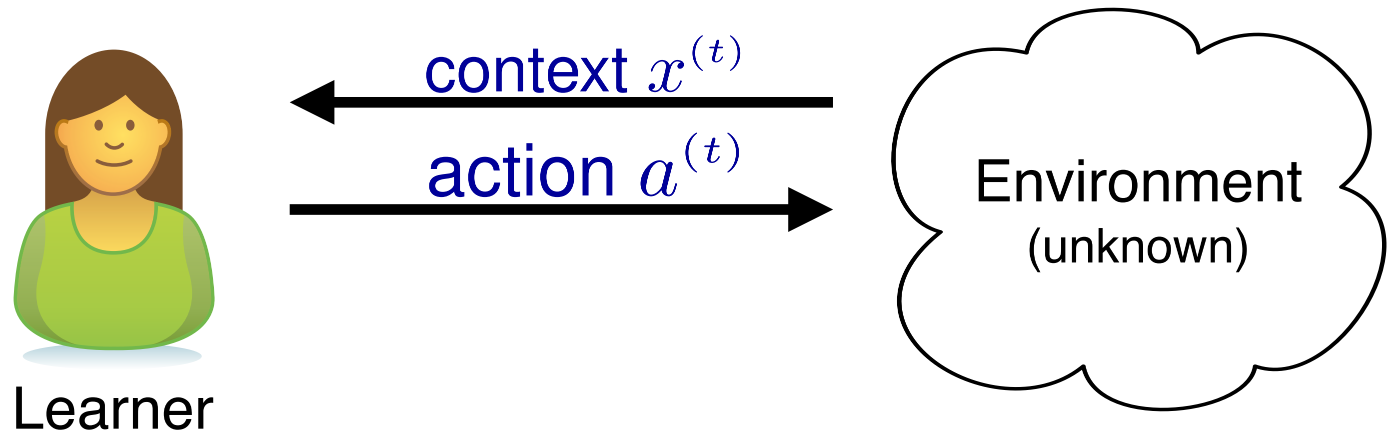
[Woodroofe '79, Clayton '89, Sarkar '91, Kaelbling '94, Abe & Long '99, Langford & Zhang '08]

Level 2: Contextual bandits



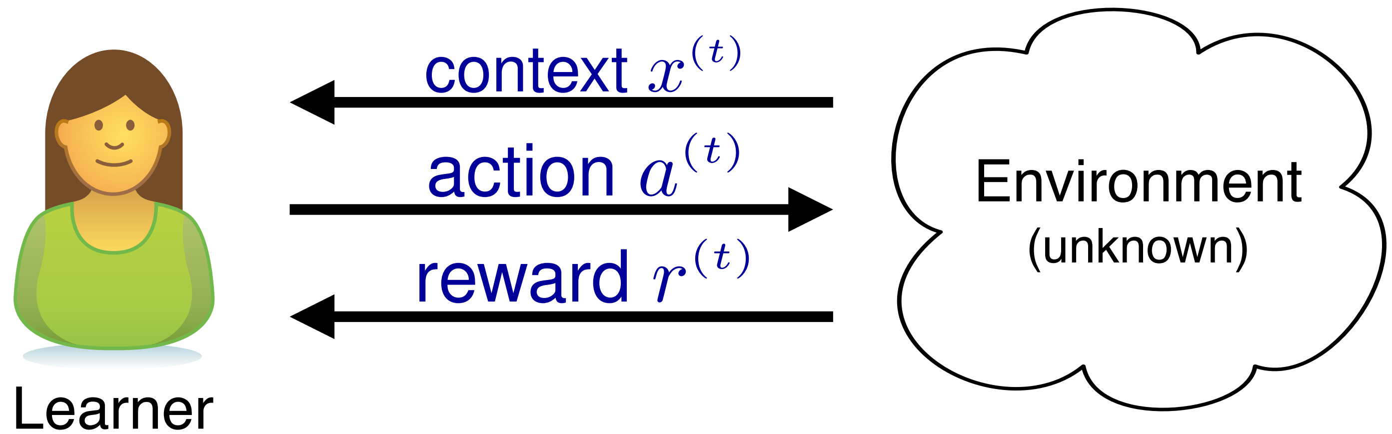
[Woodroofe ’79, Clayton ’89, Sarkar ’91, Kaelbling ’94, Abe & Long ’99, Langford & Zhang ’08]

Level 2: Contextual bandits



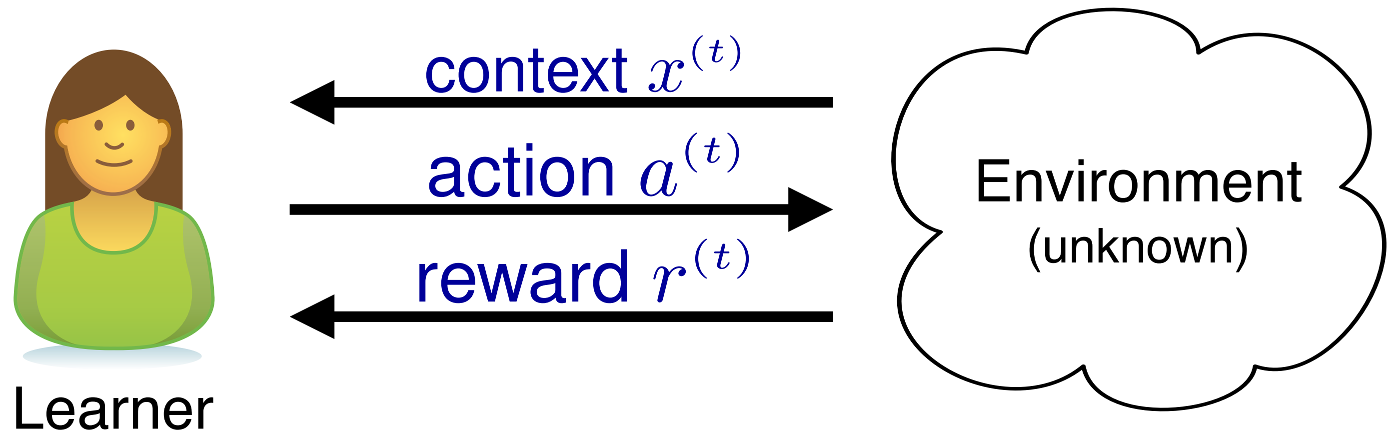
[Woodroofe ’79, Clayton ’89, Sarkar ’91, Kaelbling ’94, Abe & Long ’99, Langford & Zhang ’08]

Level 2: Contextual bandits



[Woodroofe ’79, Clayton ’89, Sarkar ’91, Kaelbling ’94, Abe & Long ’99, Langford & Zhang ’08]

Level 2: Contextual bandits

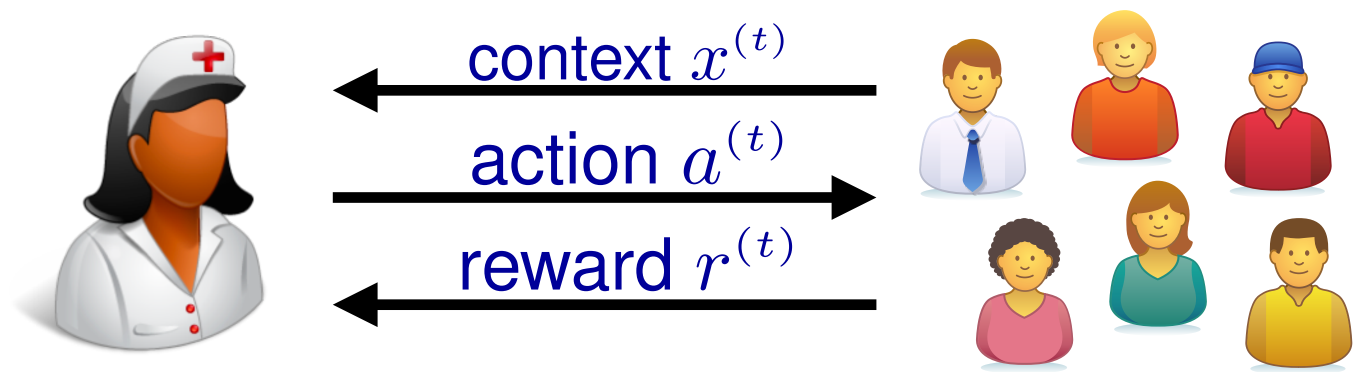


Goal: Maximize total reward

[Woodroofe '79, Clayton '89, Sarkar '91, Kaelbling '94, Abe & Long '99, Langford & Zhang '08]

Level 2: Contextual bandits

Personalized medicine

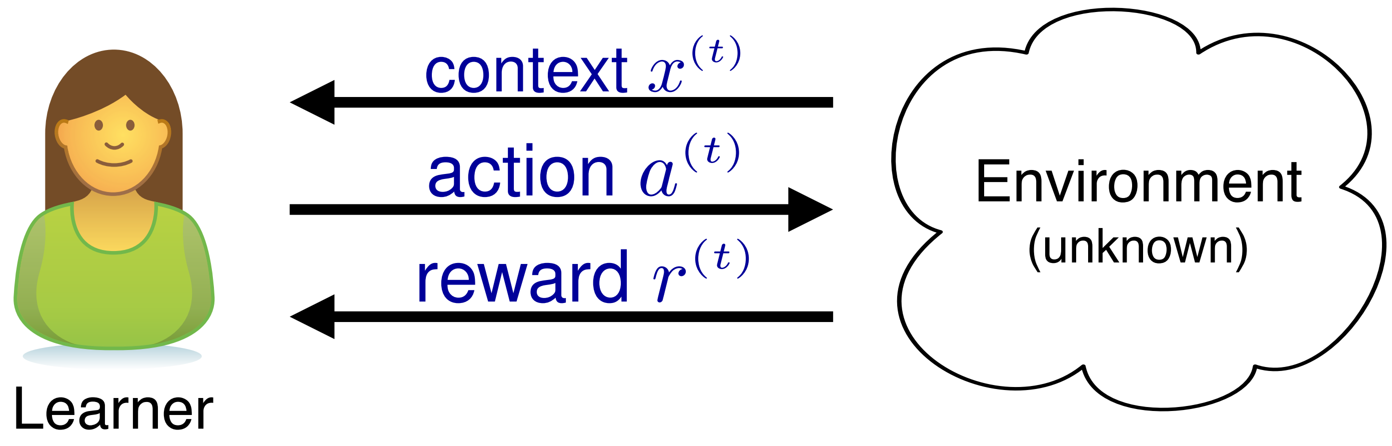


Goal: Personalize treatments to improve outcomes

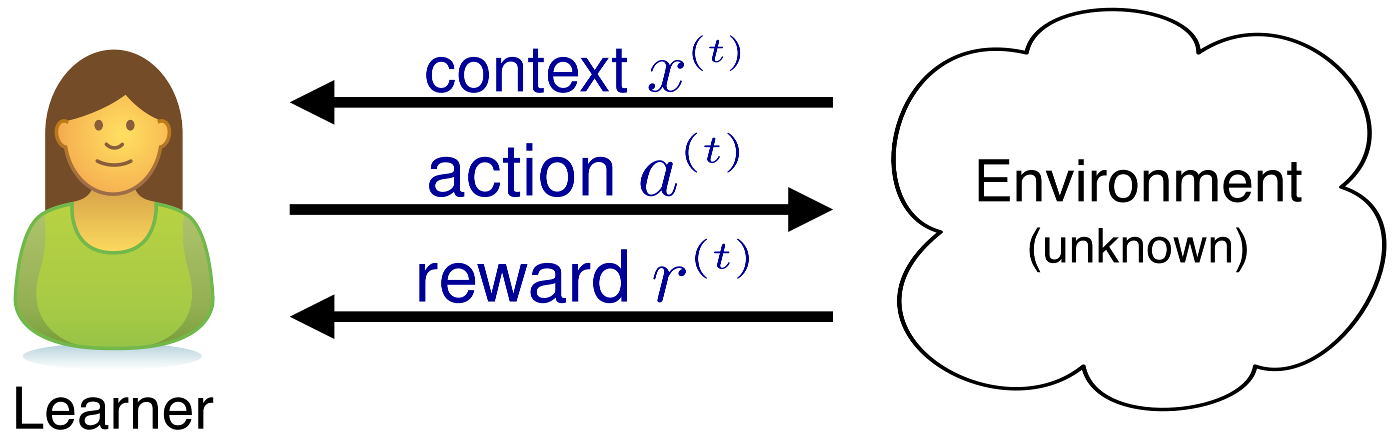
Applications:

- Personalized medicine [Mintz et al. ’17, Kallus & Zhou ’18, Bastani & Bayati ’20]
- Mobile health [Rabbi et al. ’15, Tewari & Murphy ’17, Yom-Tov et al. ’17]
- Online education [Lan & Baraniuk ’16, Segal et al. ’18, Cai et al. ‘20]
- Online recommendation [Li et al. ’10, Agarwal et al.’16]

Level 2: Contextual bandits



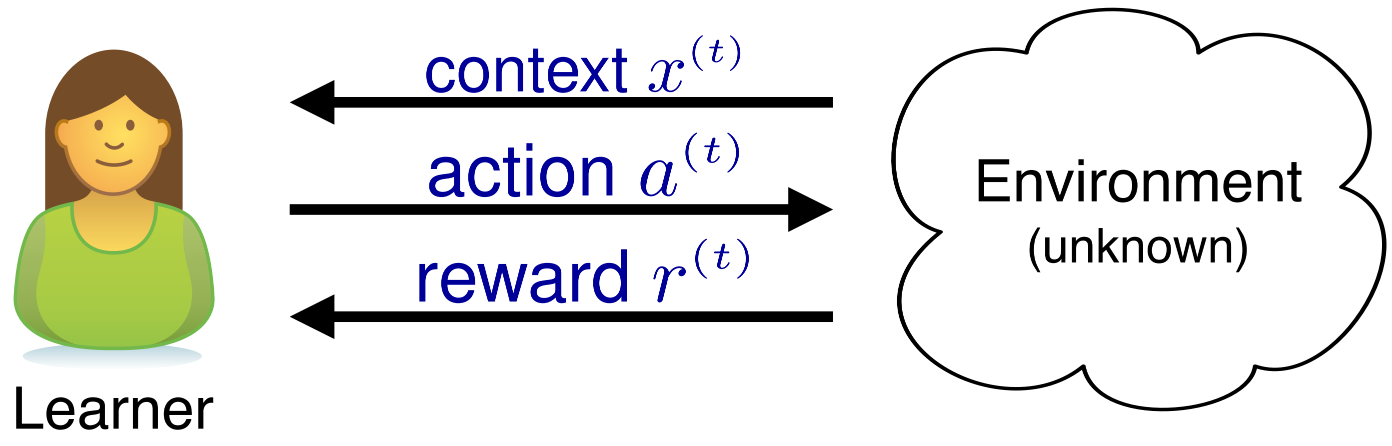
Level 2: Contextual bandits



Want to use flexible model class $\mathcal{F}$:

- Treatment effect: $(\text{context}, \text{treatment}) \mapsto \text{reward}$
- $f(x, a)$ models response of user x to treatment a

Level 2: Contextual bandits

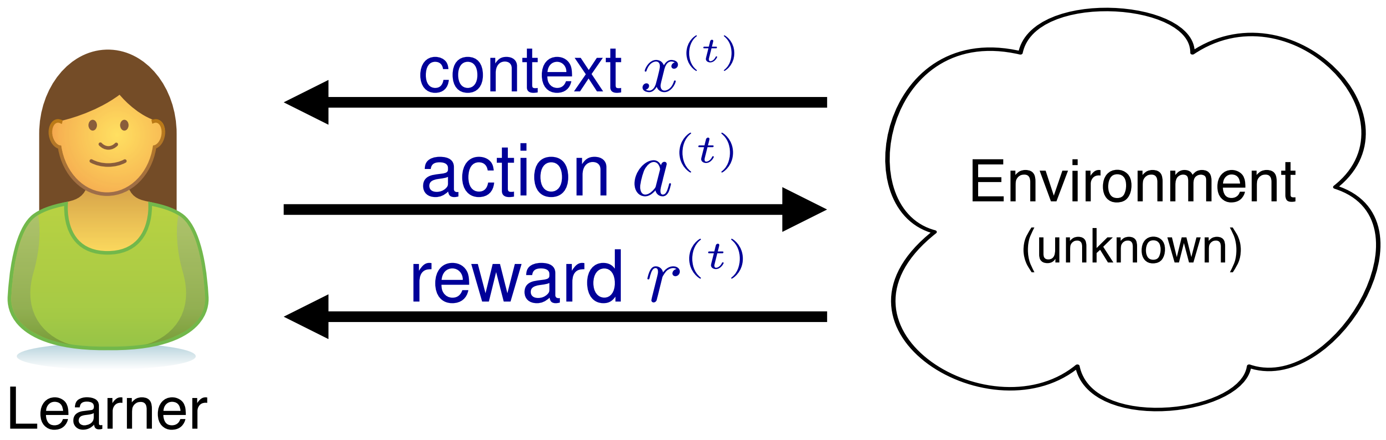


Want to use flexible model class $\mathcal{F}$:

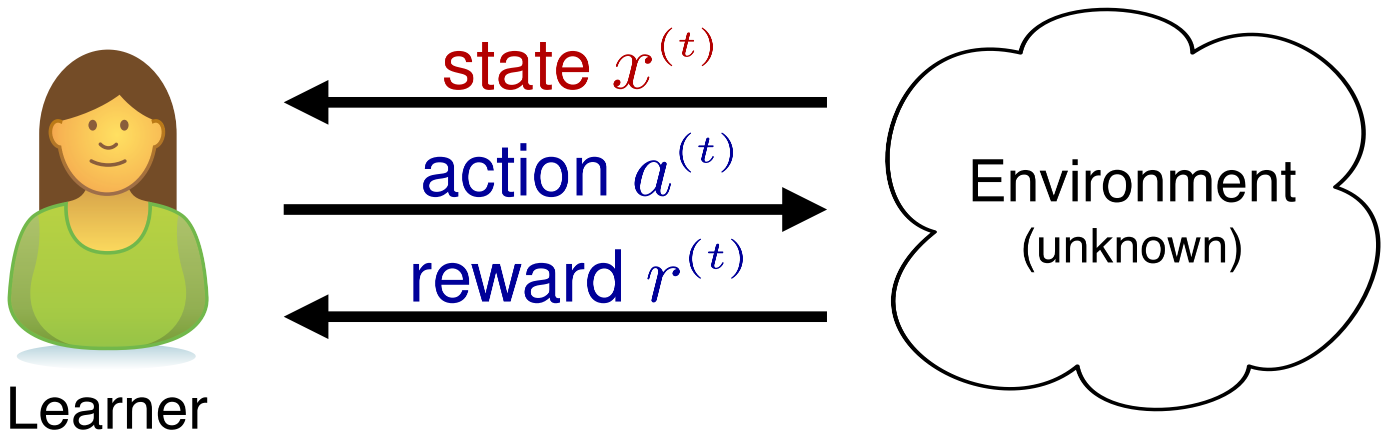
- Treatment effect: $(\text{context}, \text{treatment}) \mapsto \text{reward}$
- $f(x, a)$ models response of user x to treatment a

Need to learn a good model from data while making decisions!

Level 3: Reinforcement learning



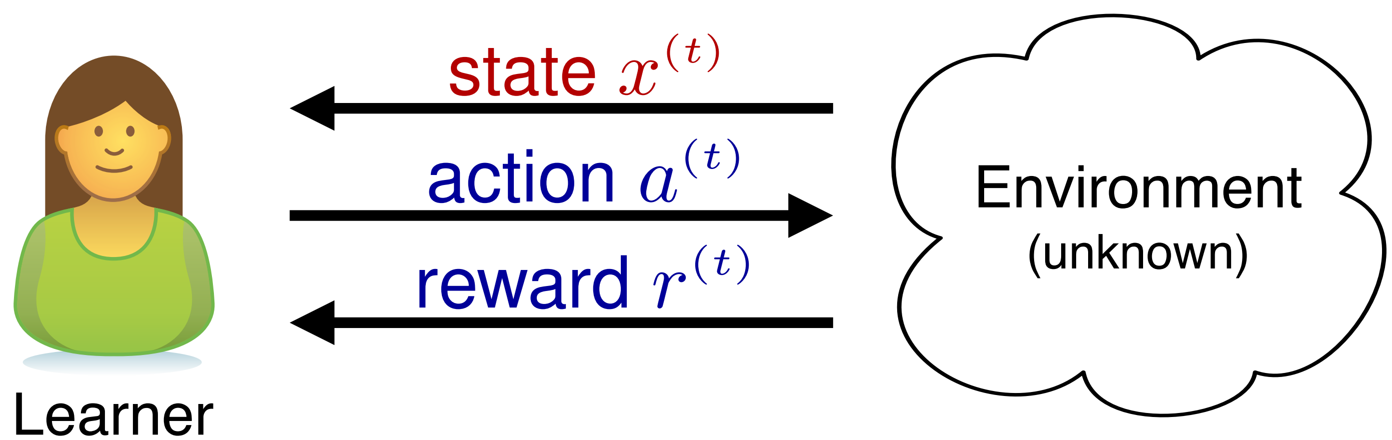
Level 3: Reinforcement learning



Contextual bandits: Actions only influence reward, not context $x^{(t)}$.

Reinforcement learning: Actions influence state $x^{(t)}$.

Level 3: Reinforcement learning



Contextual bandits: Actions only influence reward, not context $x^{(t)}$.

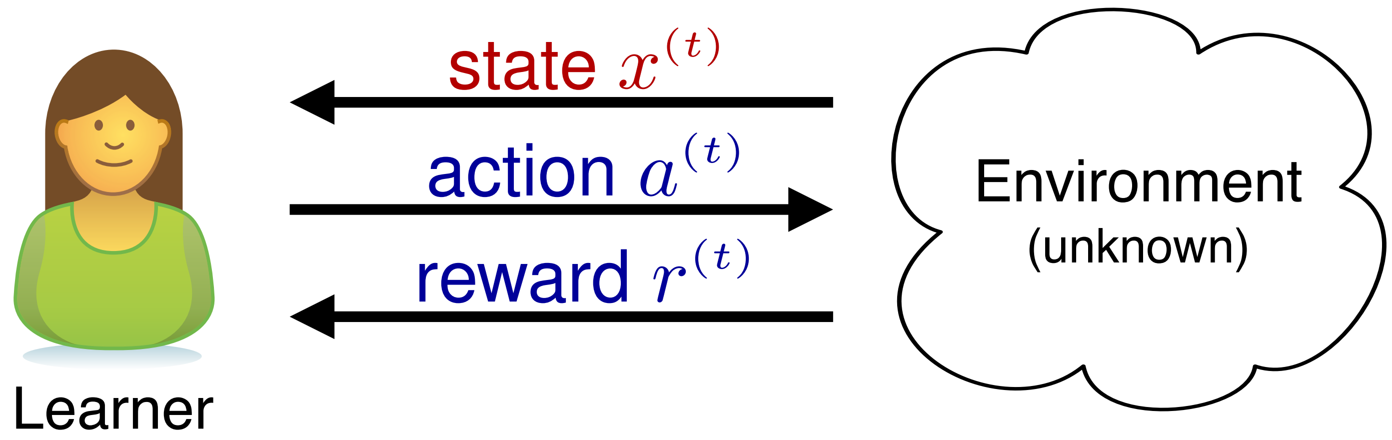
Reinforcement learning: Actions influence state $x^{(t)}$.

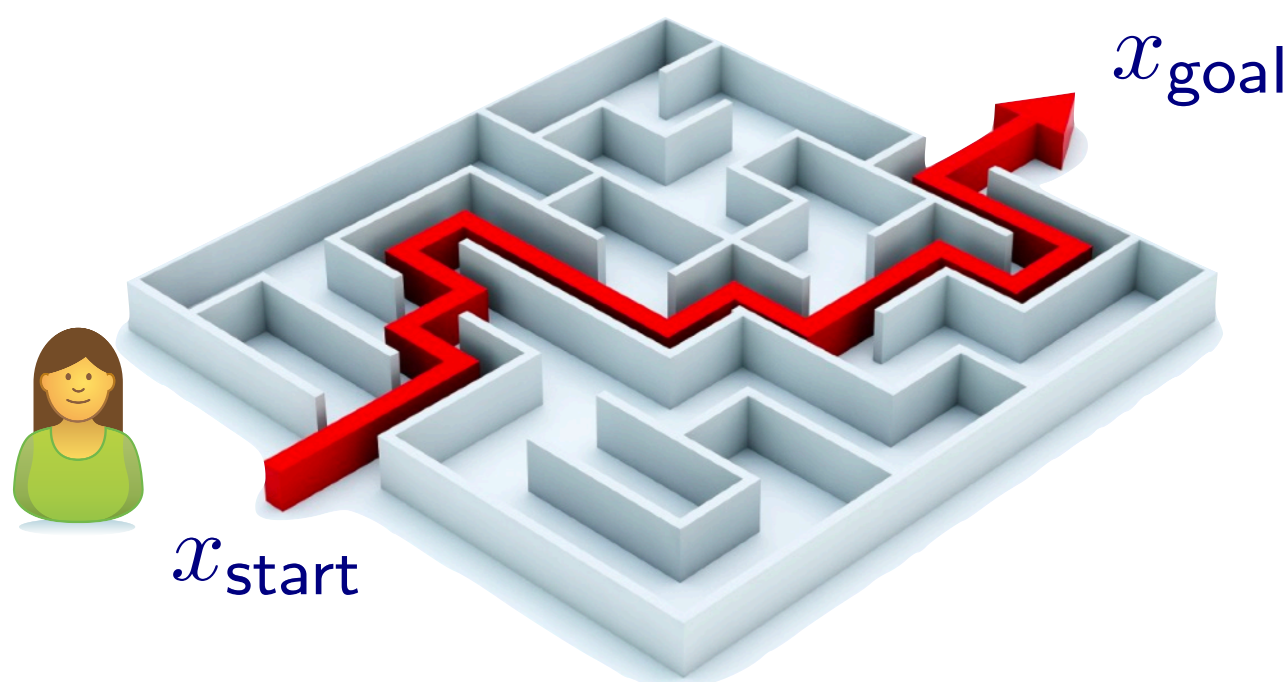
Robotics

Game playing

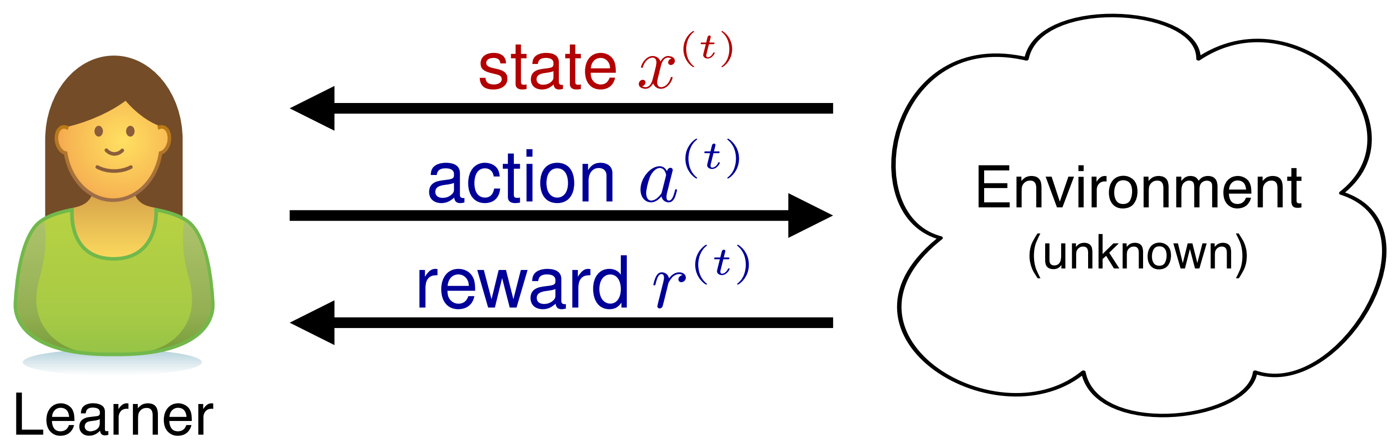
Complex treatments

Level 3: Reinforcement learning





Level 3: Reinforcement learning

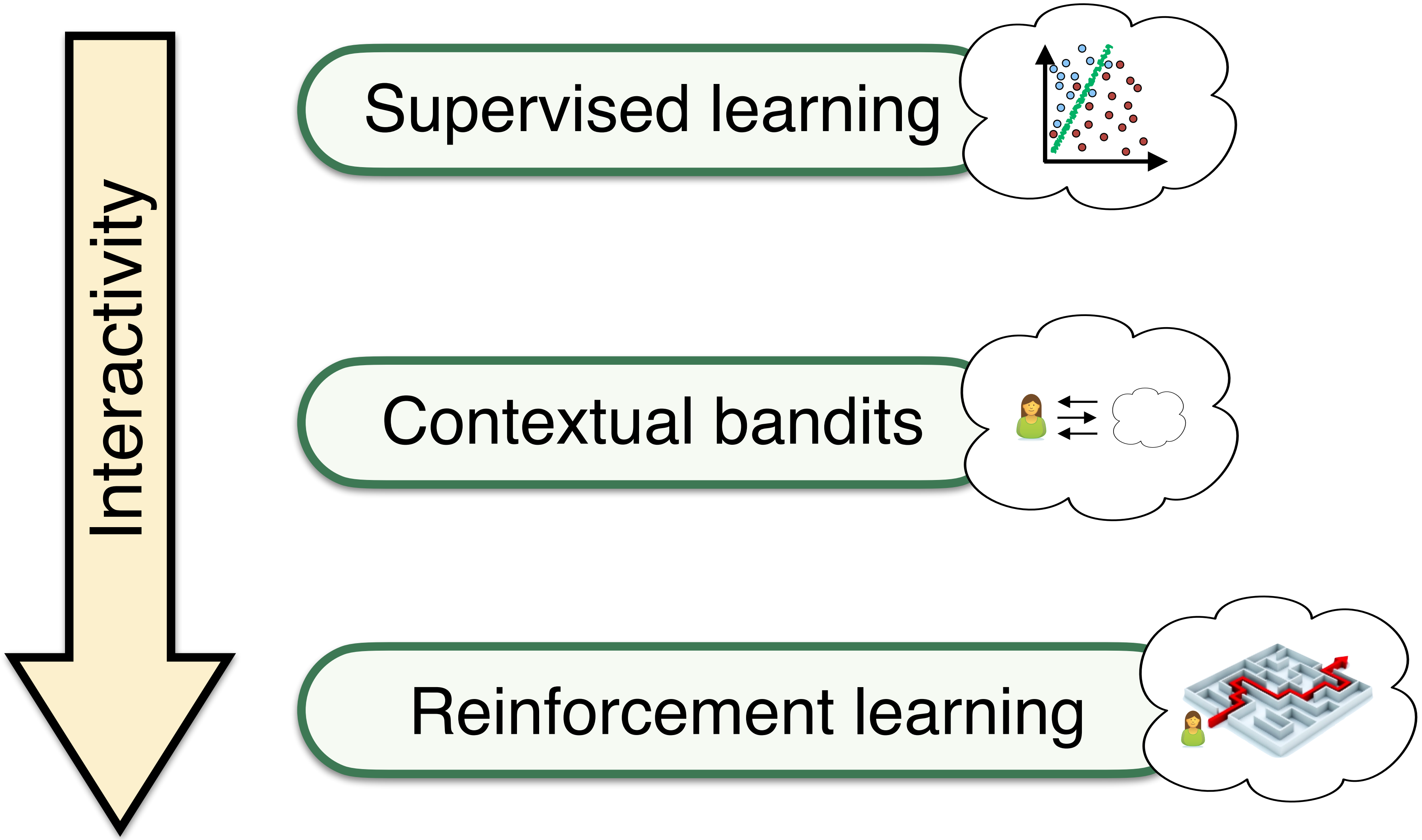


Want to use $\mathcal{F}$ to model:

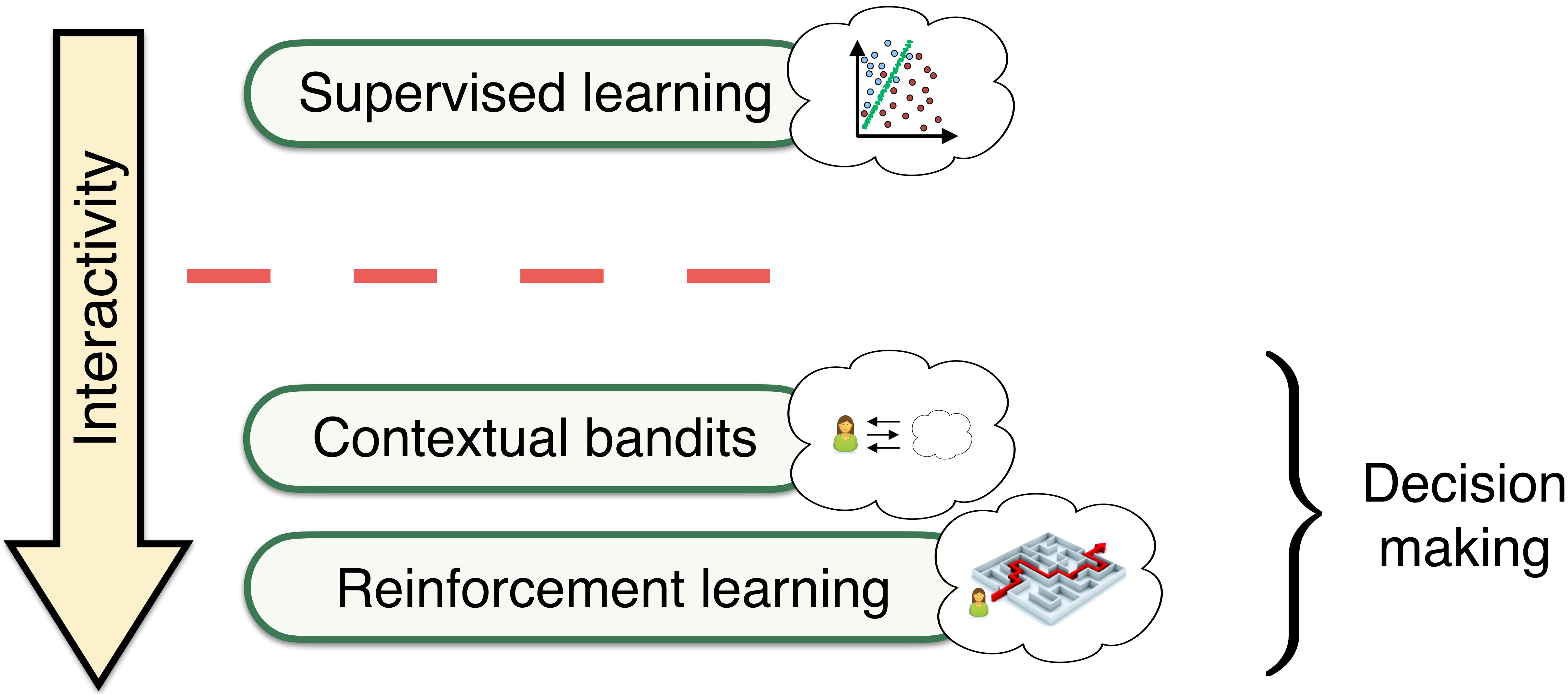
- Dynamics: $(\text{state}, \text{action}) \mapsto \text{Prob}(\text{next state})$
- Long-term rewards (value functions)

$\vdots$

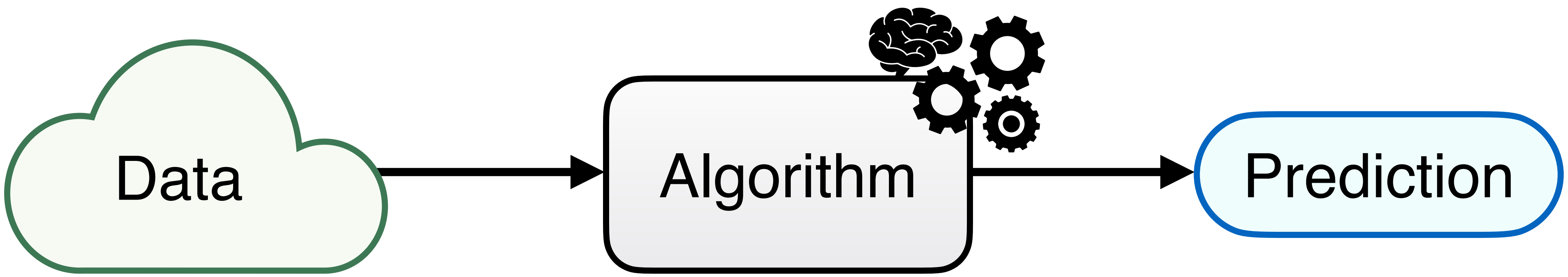
Three problems



Three problems



Gap between ML and decision making

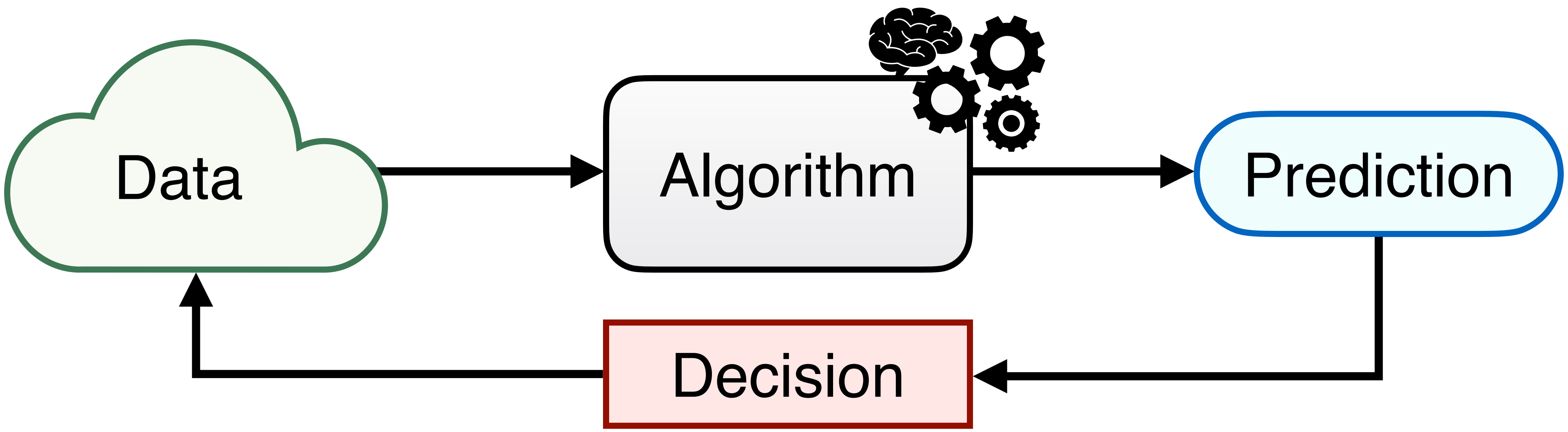


Machine learning: Good at making predictions.

("Does this image contain a cat or a dog?")

Need to know right answer for each example.

Gap between ML and decision making



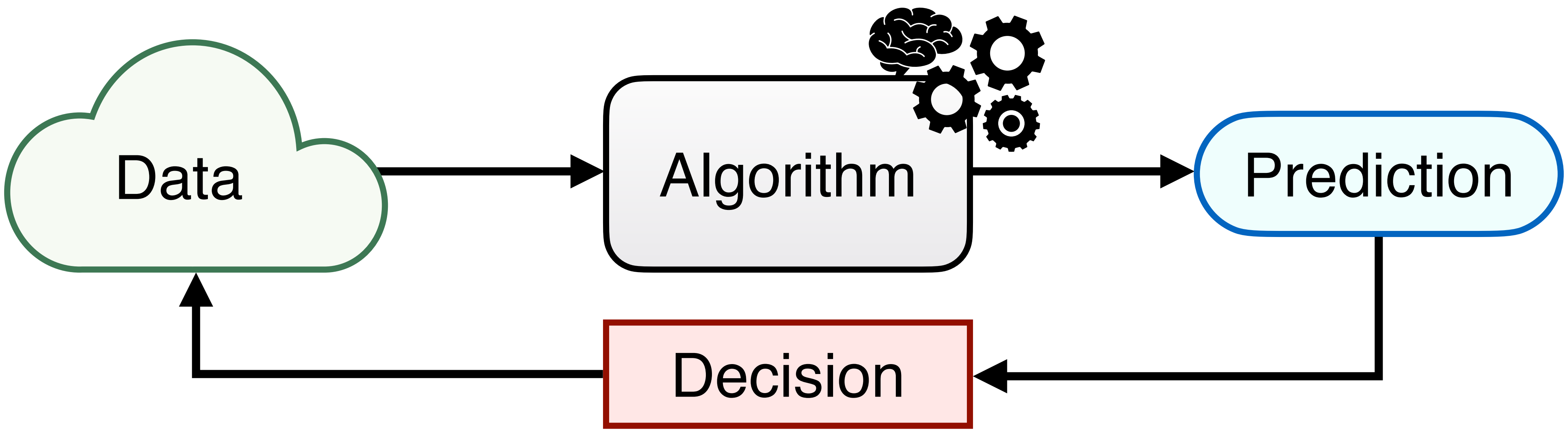
Machine learning: Good at making predictions.

(“Does this image contain a cat or a dog?”)

Need to know right answer for each example.

Decision making: Introduces feedback loops.

Gap between ML and decision making



Machine learning: Good at making predictions.

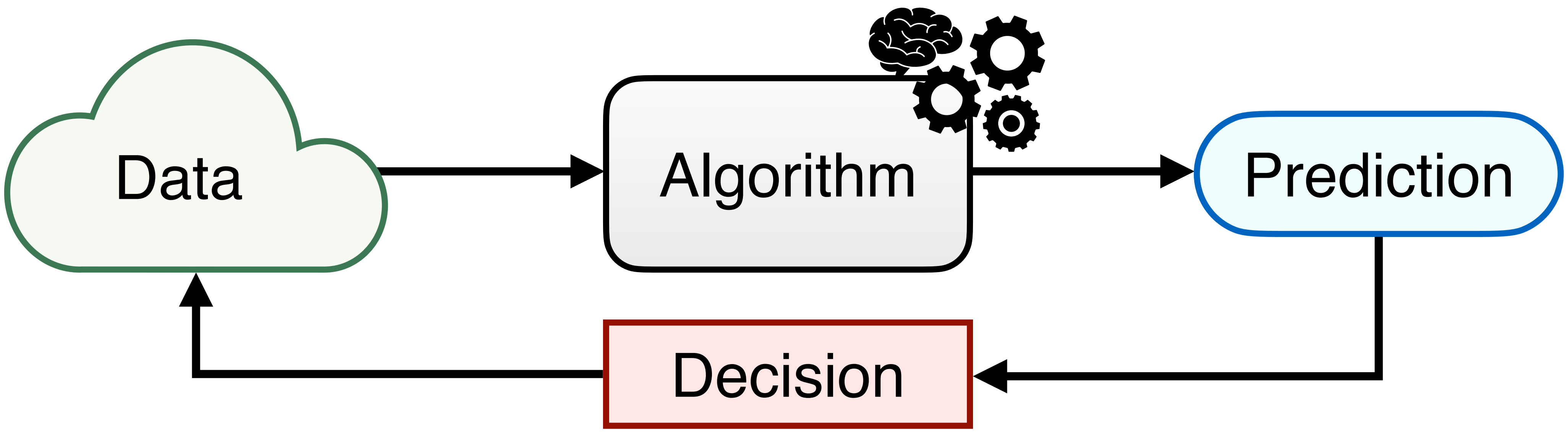
("Does this image contain a cat or a dog?")

Need to know right answer for each example.

Decision making: Introduces feedback loops.

- Need to answer counterfactuals.
  ("How would the outcome have changed if I intervened differently?")

Gap between ML and decision making



Machine learning: Good at making predictions.

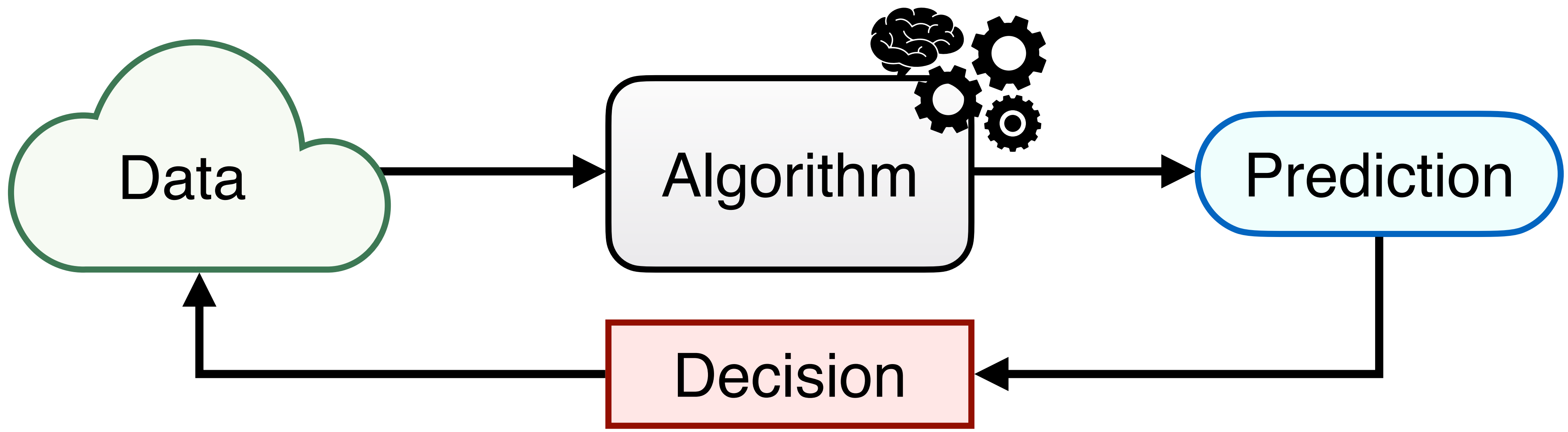
("Does this image contain a cat or a dog?")

Need to know right answer for each example.

Decision making: Introduces feedback loops.

- Need to answer counterfactuals.
  ("How would the outcome have changed if I intervened differently?")
- Need to reason about long-term impact.

Gap between ML and decision making



Naively applying ML to decision making leads to bad decisions.

Goals for this tutorial

Introduce basic concepts

Understand the statistical landscape of RL

- What assumptions on system/models lead to sample efficiency?
- Algorithmic principles and fundamental limits

Prepare for Chi's multi-agent RL tutorial

Talk outline

Statistical landscape of RL

1. **Basic concepts and solutions**

2. **The frontier**

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h \in \mathcal{X}$. **(Sensor measurement)**

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
    - Observe $x_h \in \mathcal{X}$. **(Sensor measurement)**
    - Take action $a_h \in \mathcal{A}$. **(Actuator signal)**

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h \in \mathcal{X}$. **(Sensor measurement)**
  - Take action $a_h \in \mathcal{A}$. **(Actuator signal)**
  - Observe reward $r_h \sim R(x_h, a_h)$ w/ $r_h \in [0, 1]$. **(Reached goal?)**

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h \in \mathcal{X}$. **`(Sensor measurement)`**
  - Take action $a_h \in \mathcal{A}$. **`(Actuator signal)`**
  - Observe reward $r_h \sim R(x_h, a_h)$ w/ $r_h \in [0, 1]$. **`(Reached goal?)`**
  - Transition: $x_{h+1} \sim P(\cdot \mid x_h, a_h)$. **`(System evolves)`**

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

Repeatedly:

- $x_1 \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h \in \mathcal{X}$. `(Sensor measurement)`
  - Take action $a_h \in \mathcal{A}$. `(Actuator signal)`
  - Observe reward $r_h \sim R(x_h, a_h)$ w/ $r_h \in [0, 1]$. `(Reached goal?)`
  - Transition: $x_{h+1} \sim P(\cdot \mid x_h, a_h)$. `(System evolves)`

Goal: Find policy $\widehat{\pi} : \mathcal{X} \to \mathcal{A}$ maximizing $J(\pi) := \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} r_h\right]$.

$a_h \sim \pi_h(x_h)$

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

For $t = 1, \ldots, T$:

- $x_1^{(t)} \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h^{(t)} \in \mathcal{X}$. `(Sensor measurement)`
  - Take action $a_h^{(t)} \in \mathcal{A}$. `(Actuator signal)`
  - Observe reward $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$ w/ $r_h^{(t)} \in [0, 1]$. `(Reached goal?)`
  - Transition: $x_{h+1}^{(t)} \sim P(\cdot \mid x_h^{(t)}, a_h^{(t)})$. `(System evolves)`

Goal: Find policy $\widehat{\pi} : \mathcal{X} \to \mathcal{A}$ maximizing $J(\pi) := \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} r_h\right]$.

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

For $t = 1, \ldots, T$:

- $x_1^{(t)} \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h^{(t)} \in \mathcal{X}$. `(Sensor measurement)`
  - Take action $a_h^{(t)} \in \mathcal{A}$. `(Actuator signal)`
  - Observe reward $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$ w/ $r_h^{(t)} \in [0, 1]$. `(Reached goal?)`
  - Transition: $x_{h+1}^{(t)} \sim P(\cdot \mid x_h^{(t)}, a_h^{(t)})$. `(System evolves)`

Goal: Find policy $\widehat{\pi} : \mathcal{X} \to \mathcal{A}$ maximizing $J(\pi) := \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} r_h\right]$.

PAC-RL: Find $\widehat{\pi}$ with $\max_{\pi} J(\pi) - J(\widehat{\pi}) \leq \varepsilon$ using minimal # episodes.

Reinforcement learning: Setup

This tutorial: Episodic, finite-horizon setting

For $t = 1, \ldots, T$:

- $x_1^{(t)} \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h^{(t)} \in \mathcal{X}$. **`(Sensor measurement)`**
  - Take action $a_h^{(t)} \in \mathcal{A}$. **`(Actuator signal)`**
  - Observe reward $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$ w/ $r_h^{(t)} \in [0, 1]$. **`(Reached goal?)`**
  - Transition: $x_{h+1}^{(t)} \sim P(\cdot \mid x_h^{(t)}, a_h^{(t)})$. **`(System evolves)`**

Goal: Find policy $\widehat{\pi} : \mathcal{X} \to \mathcal{A}$ maximizing $J(\pi) := \mathbb{E}^{\pi}\left[\sum_{h=1}^{H} r_h\right]$.

PAC-RL: Find $\widehat{\pi}$ with $\max_{\pi} J(\pi) - J(\widehat{\pi}) \leq \varepsilon$ using minimal # episodes.

Regret: Ensure $\mathbf{Reg}(T) := \sum_{t=1}^{T} J(\pi^{\star}) - J(\pi^{(t)}) \leq$ sublinear in T (e.g., $\sqrt{T}$)

w/ $\pi^{\star} := \arg\max_{\pi} J(\pi)$.

Reinforcement learning: Setup

Variants of the setting:

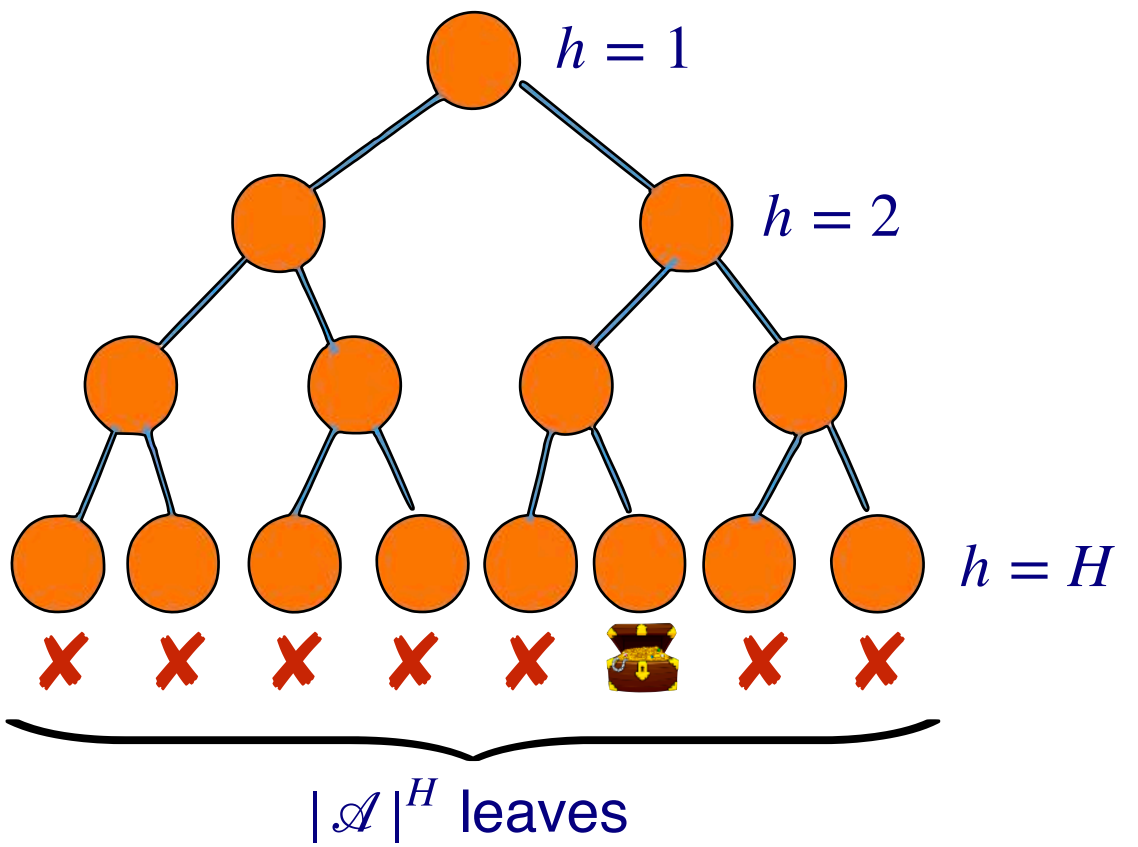
- Many episodes vs. one big trajectory
- Finite vs. infinite horizon
- Undiscounted vs. discounted rewards
  - Pick discount factor $\gamma \in (0, 1)$.
  - Instead of weighing rewards uniformly, weight r_h by γ^{h-1}.
  - Effective horizon: $1/(1 - \gamma)$.

$\vdots$

We will focus on episodic, finite-horizon, and undiscounted.

What does it mean to be sample-efficient?

Consider an exponentially large binary tree with reward at a single leaf.



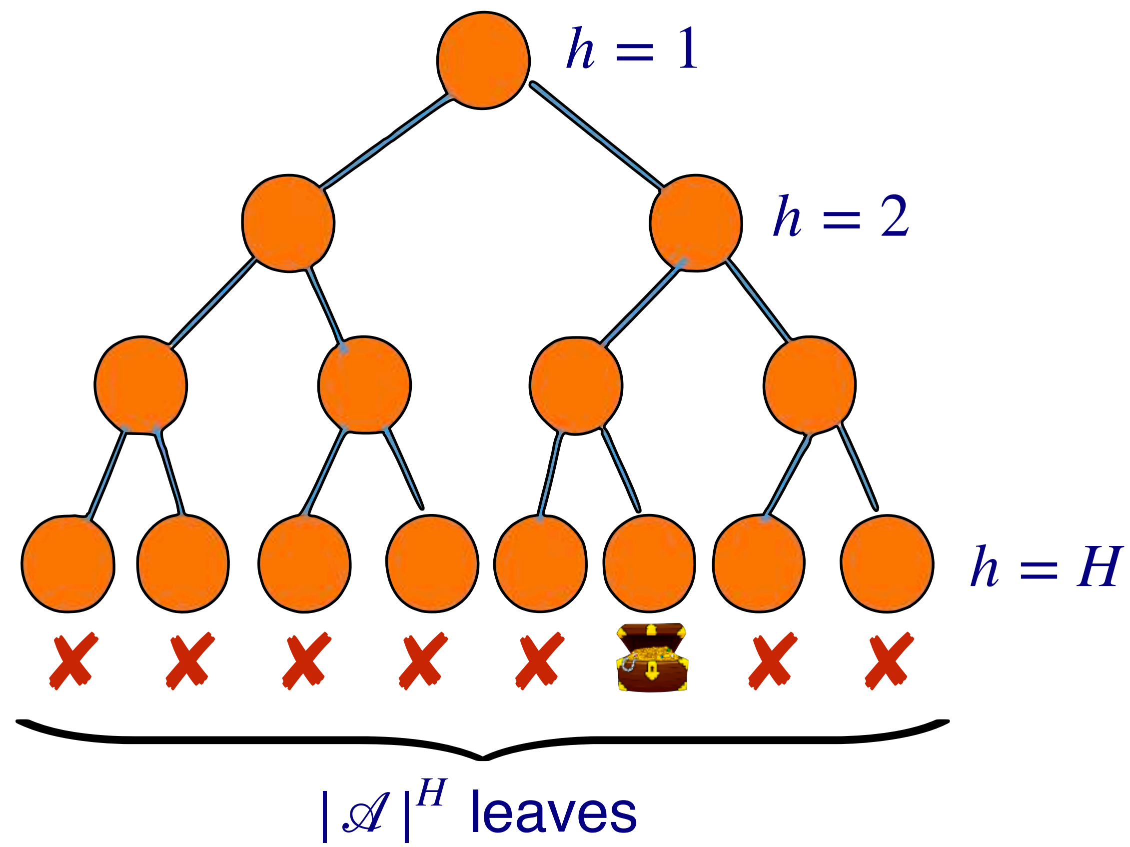
What does it mean to be sample-efficient?

Consider an exponentially large binary tree with reward at a single leaf.

Need to try all leaves to get reward.

$\Longrightarrow$ $|\mathcal{A}|^H$ **episodes required!**

[e.g., Kearns et al. '02, Krishnamurthy et al.'16].



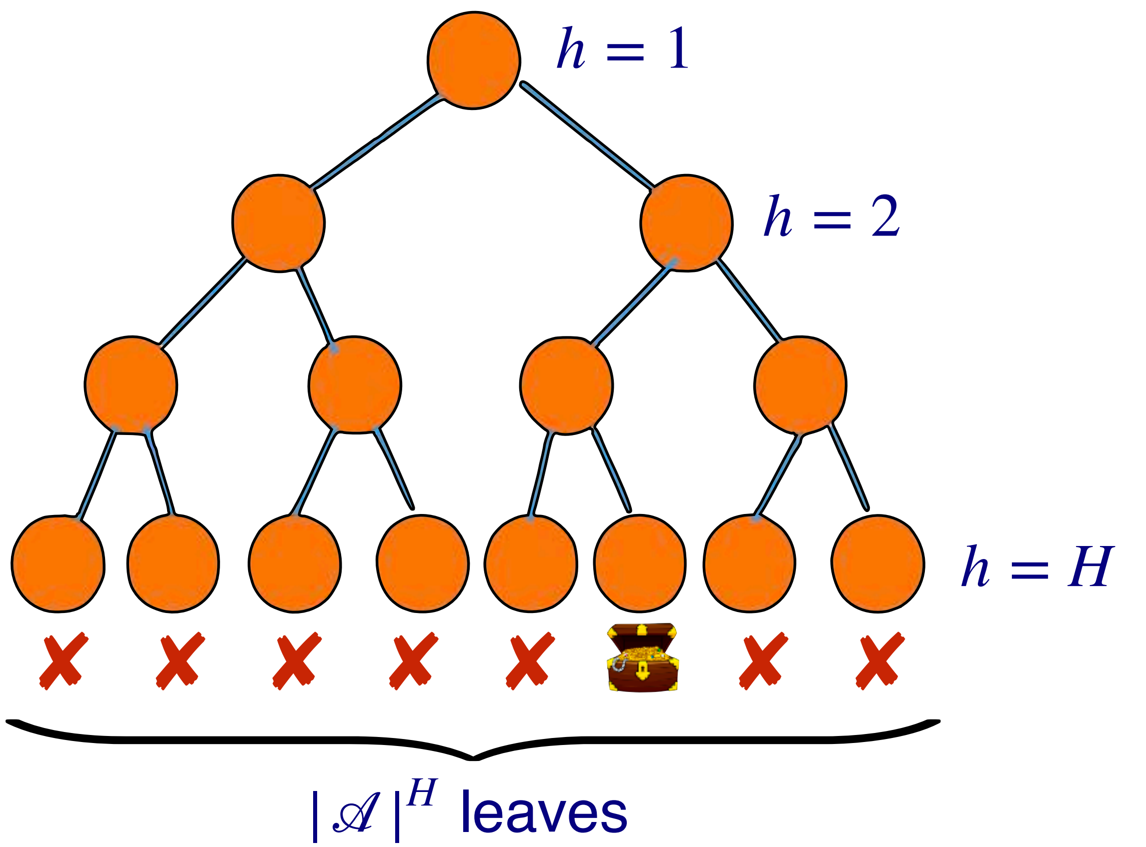
What does it mean to be sample-efficient?

Consider an exponentially large binary tree with reward at a single leaf.

Need to try all leaves to get reward.

$\Longrightarrow \ |\mathcal{A}|^H$ **episodes required!**

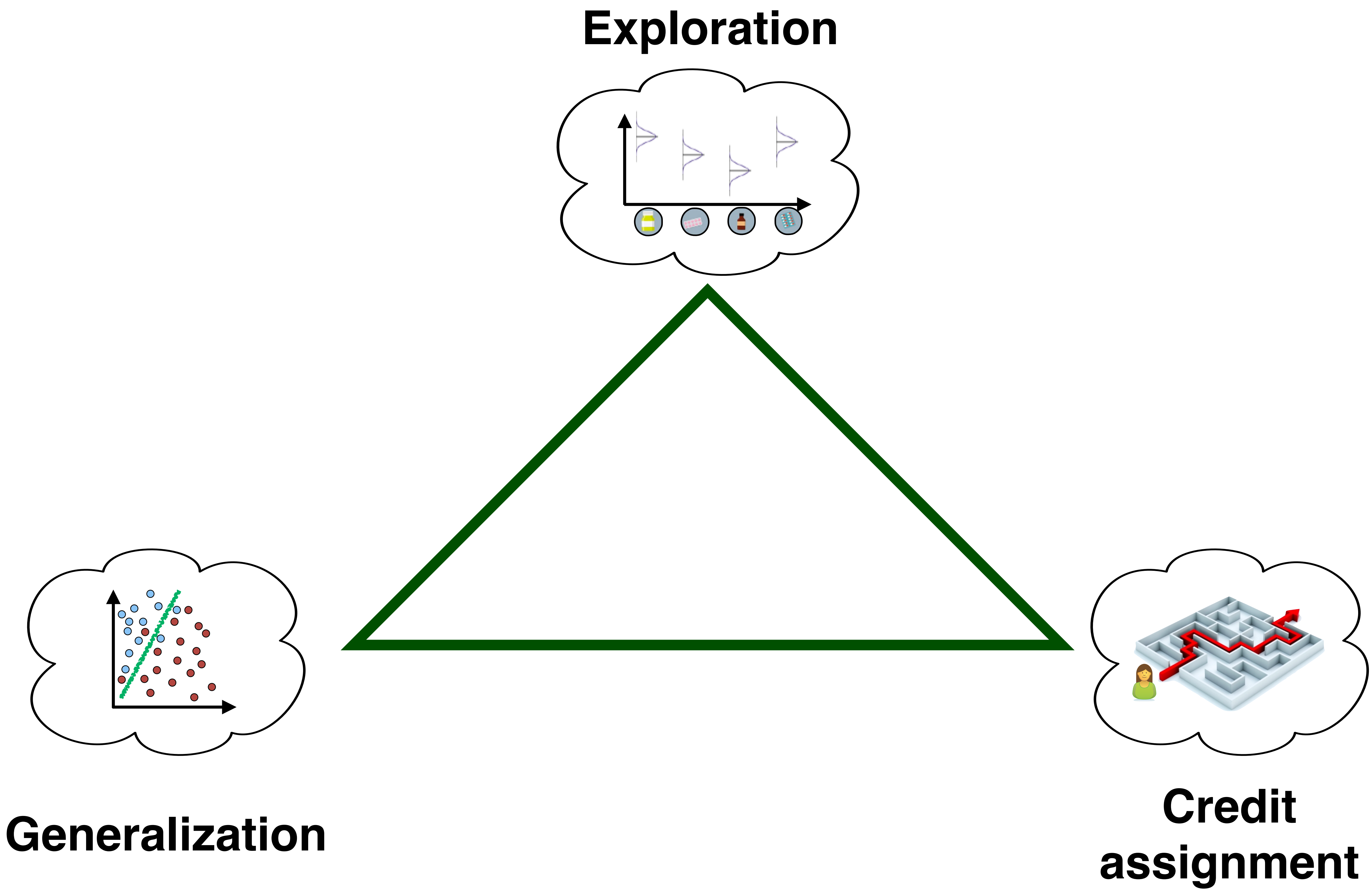
[e.g., Kearns et al. '02, Krishnamurthy et al.'16].



Conclusions:

- Further modeling assumptions required to avoid exponential sample comp.

Challenges of RL



[Credit: John Langford]

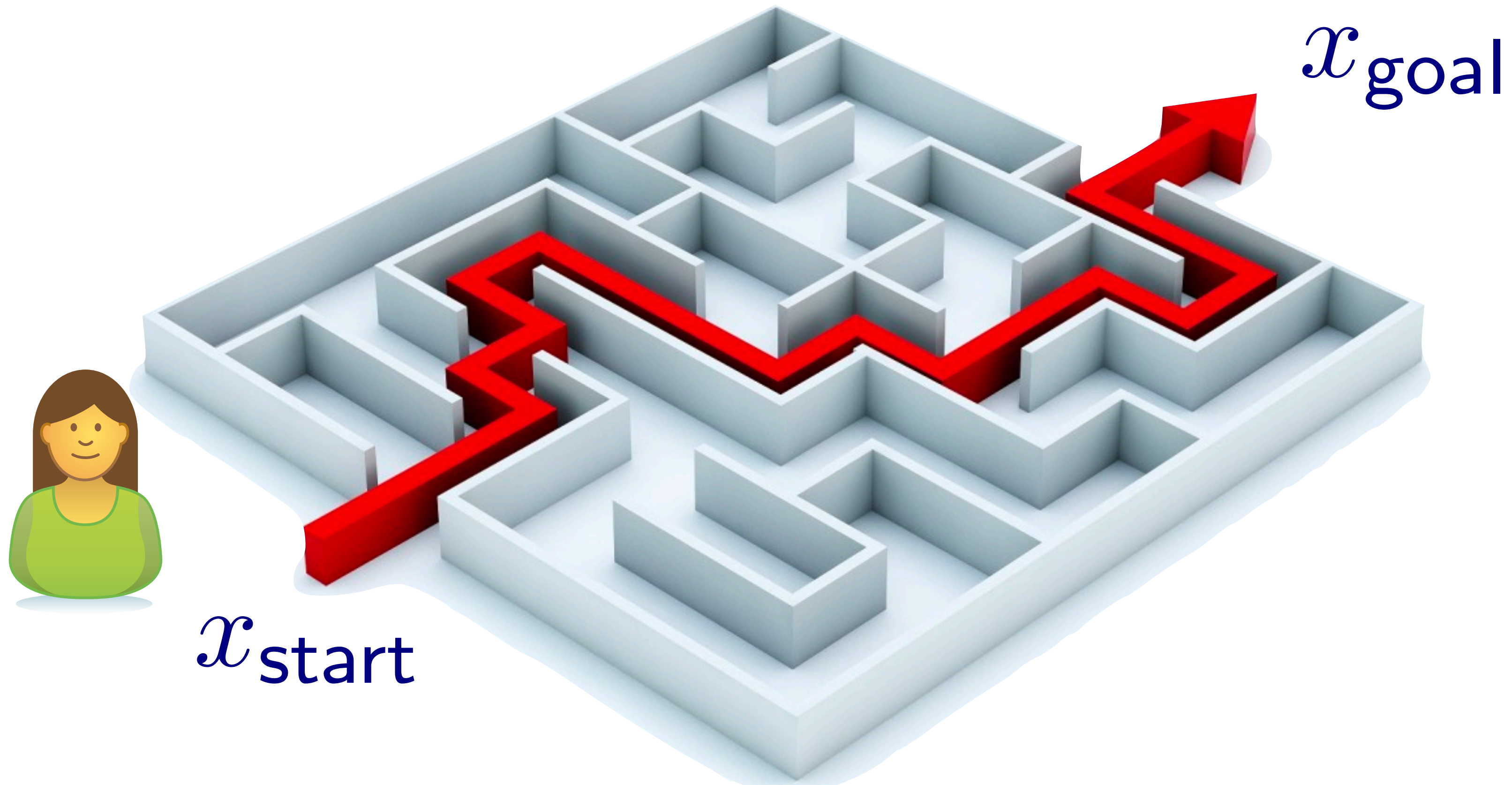
Roadmap

Basic challenges and solutions

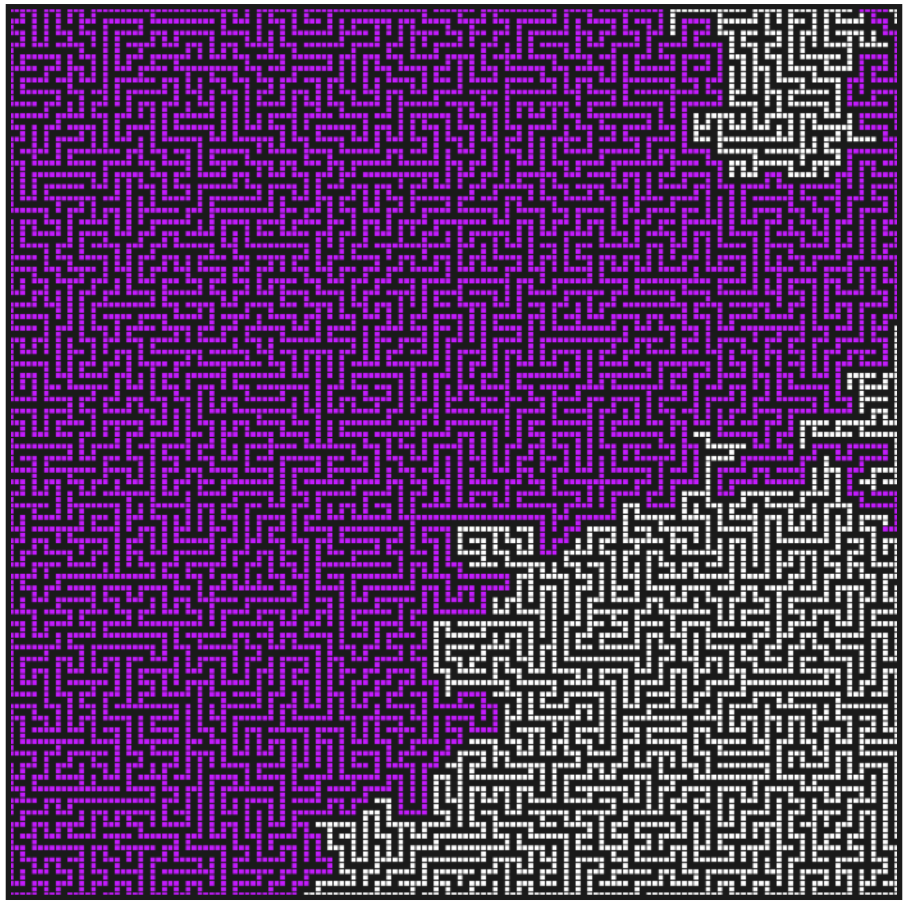
- Credit assignment
- Exploration
- Generalization

Challenge #1: Credit assignment

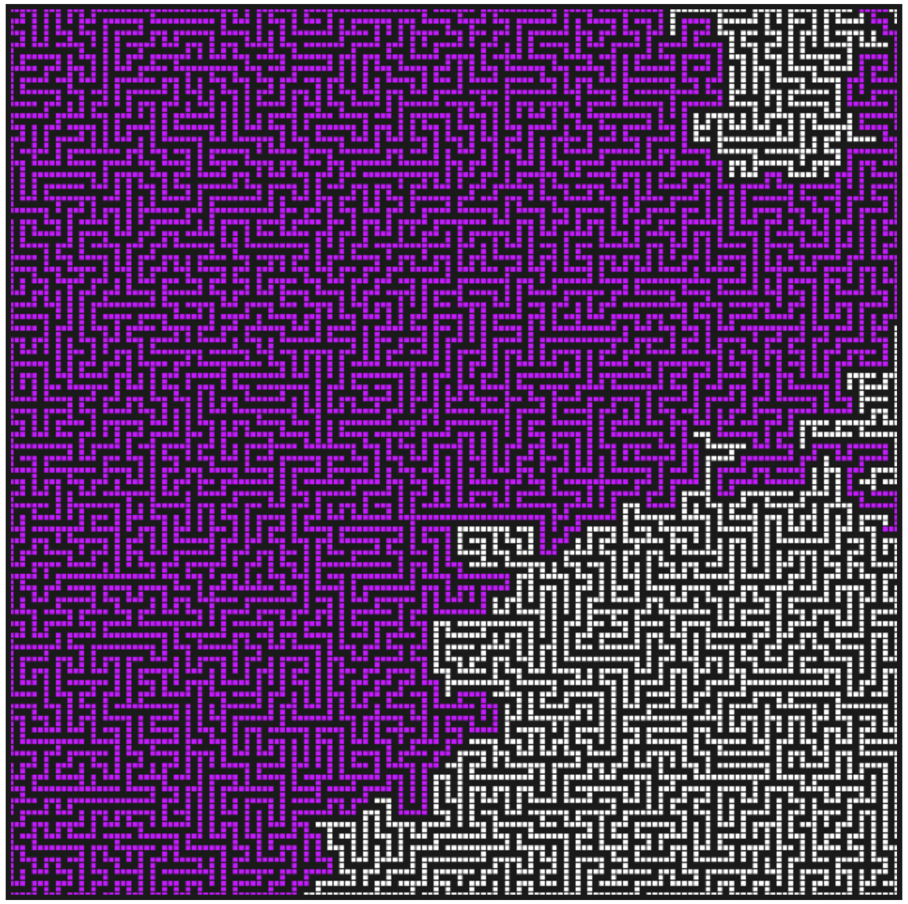
Challenge #1: Credit assignment



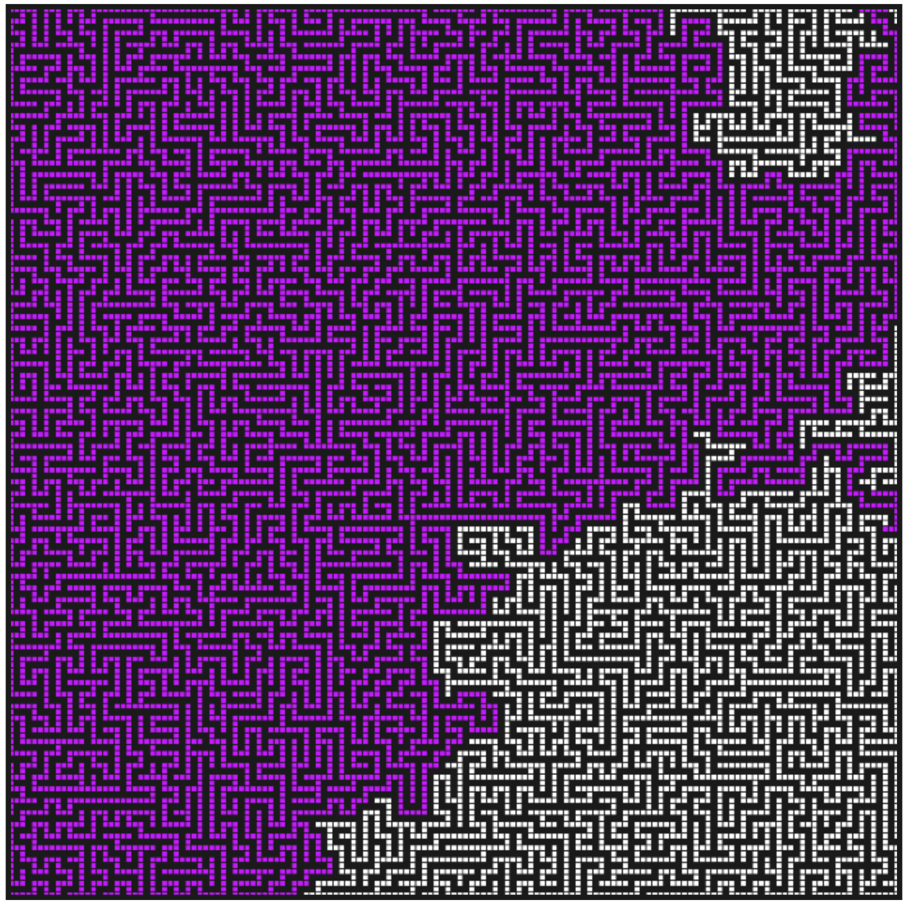
Approach: Dynamic programming



Approach: Dynamic programming



Approach: Dynamic programming

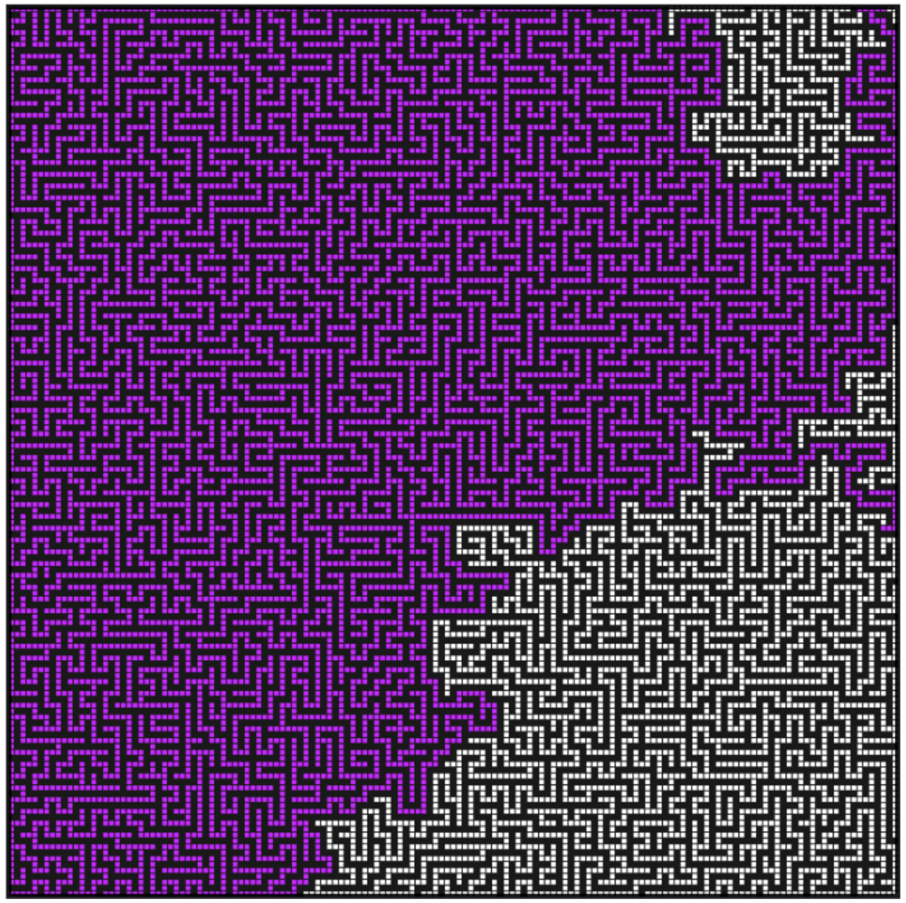


Value functions:

- $V_h^\star(x) = \mathbb{E}^{\pi^\star}\left[\sum_{h'=h}^{H} r_{h'} \mid x_h = x\right]$ (state value function)
- $Q_h^\star(x,a) = \mathbb{E}^{\pi^\star}\left[\sum_{h'=h}^{H} r_{h'} \mid x_h = x, a_h = a\right]$ (state-action value function)

Can define $Q_h^\pi(x,a)$, $V_h^\pi(x)$ analogously for any π.

Approach: Dynamic programming



Dynamic programming (“value iteration”): [Bellman ’54]

Starting with $V^\star_{H+1}(x) := 0$, iterate

$$Q^\star_h(x,a) = \mathbb{E}[r_h + V^\star_{h+1}(x_{h+1}) \mid x_h = x, a_h = a], \quad V^\star_h(x) = \max_{a \in \mathcal{A}} Q^\star_h(x,a).$$

Optimal policy is $\pi^\star_h(x) := \arg\max_{a \in \mathcal{A}} Q^\star_h(x,a)$.

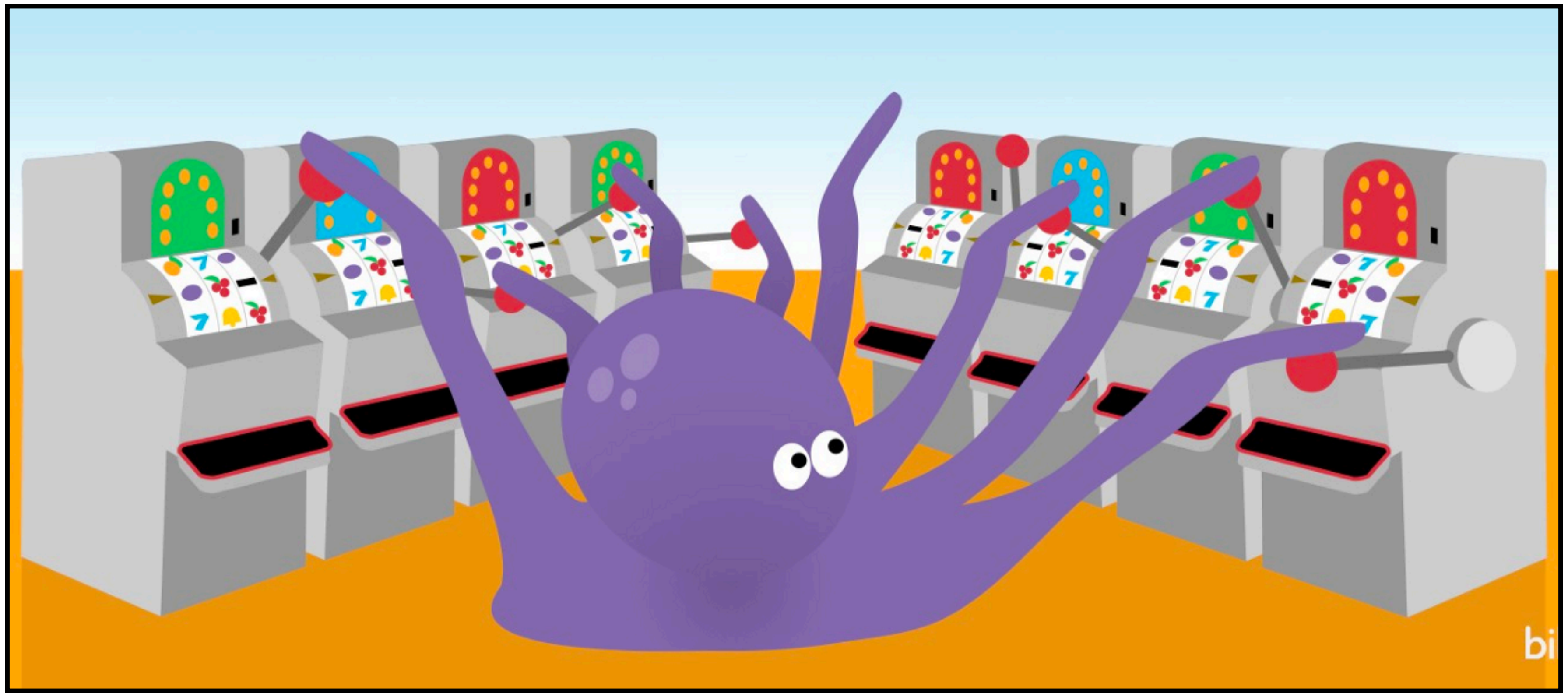
See also: [Puterman ’94, Sutton & Barto ’98]

Roadmap

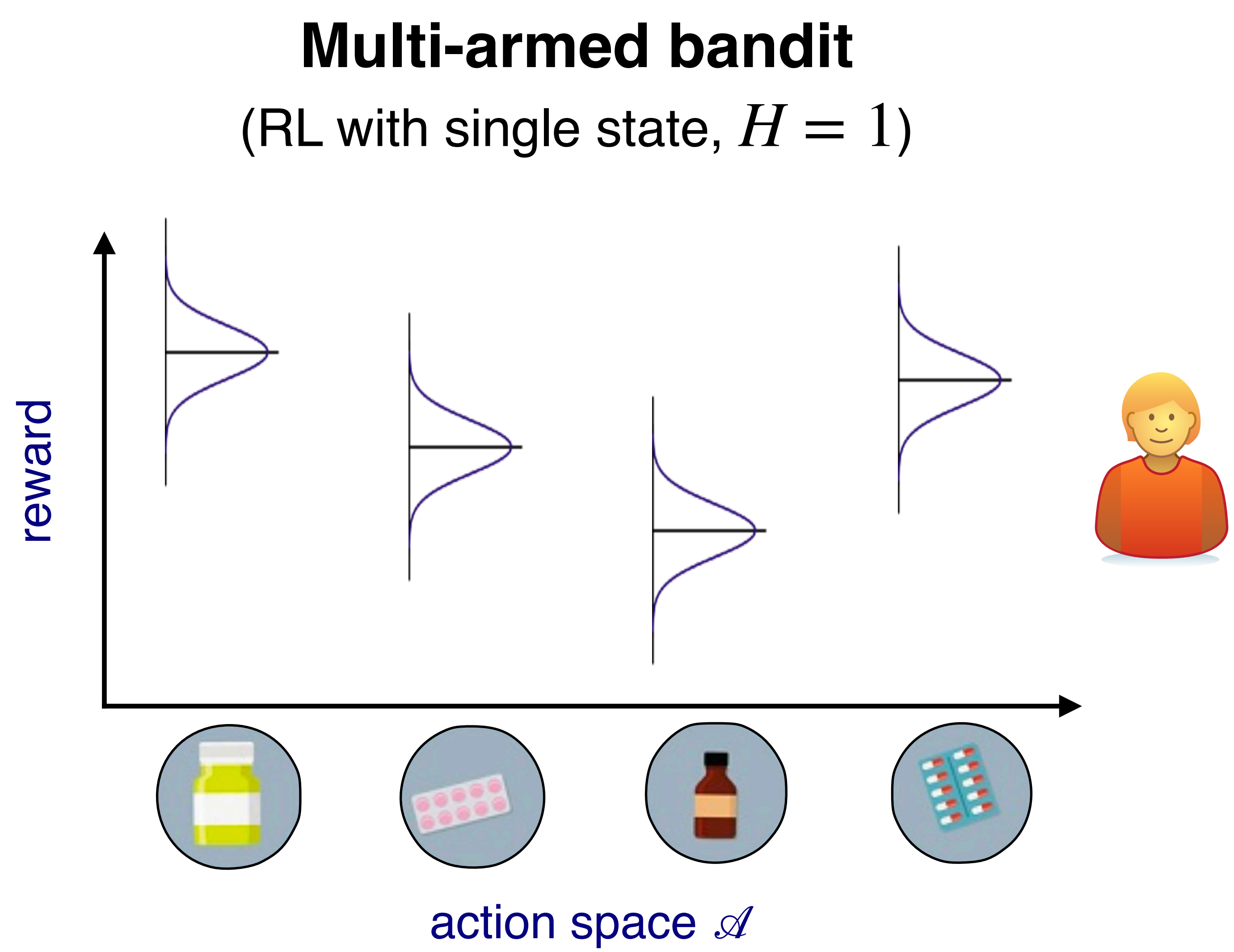
Basic challenges and solutions

- Credit assignment ✔
- Exploration
- Generalization

Challenge #2: Exploration



Exploration: Multi-armed bandit

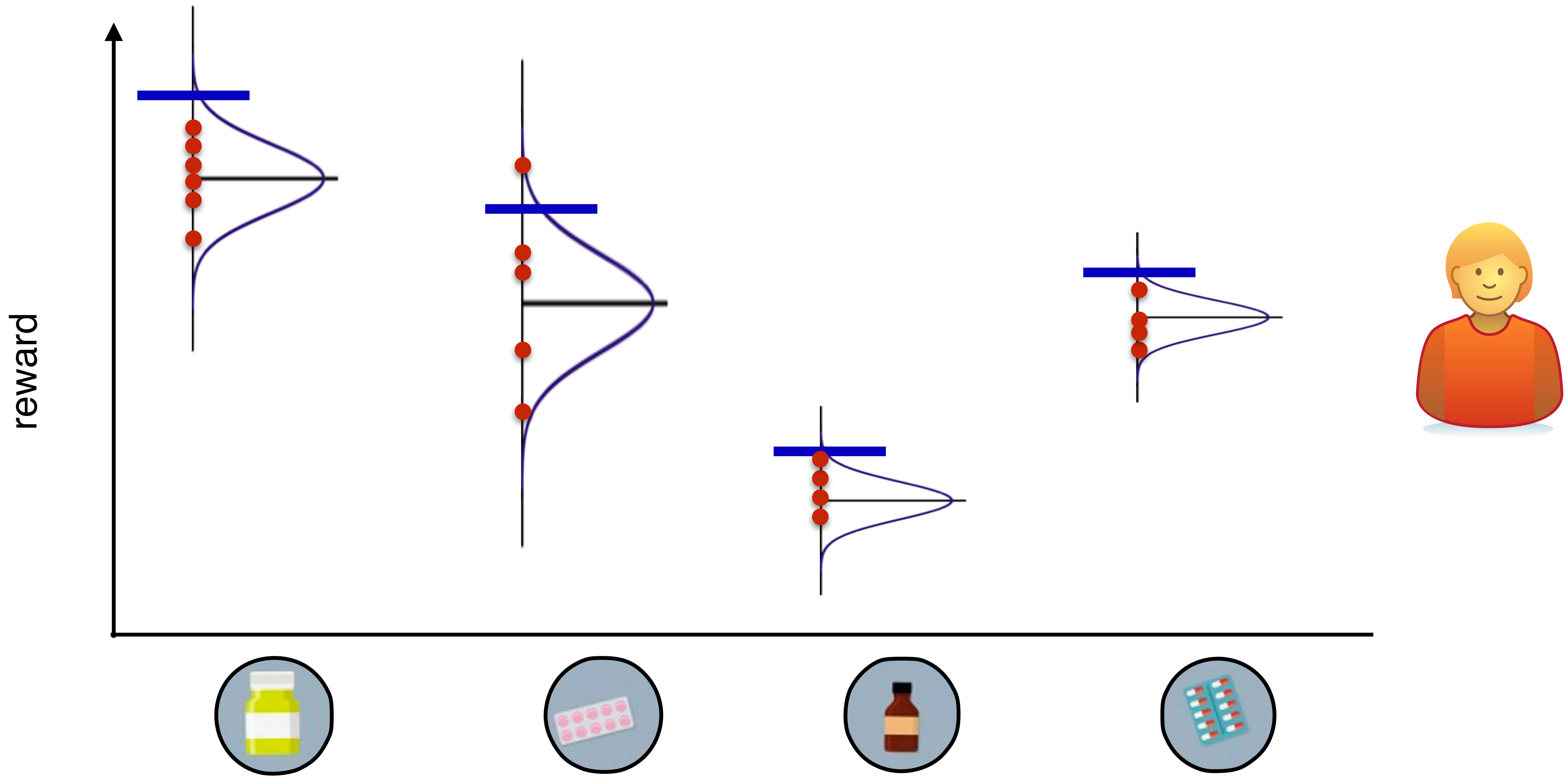


Basic issue: Only see response for actions we take.

Tension between:

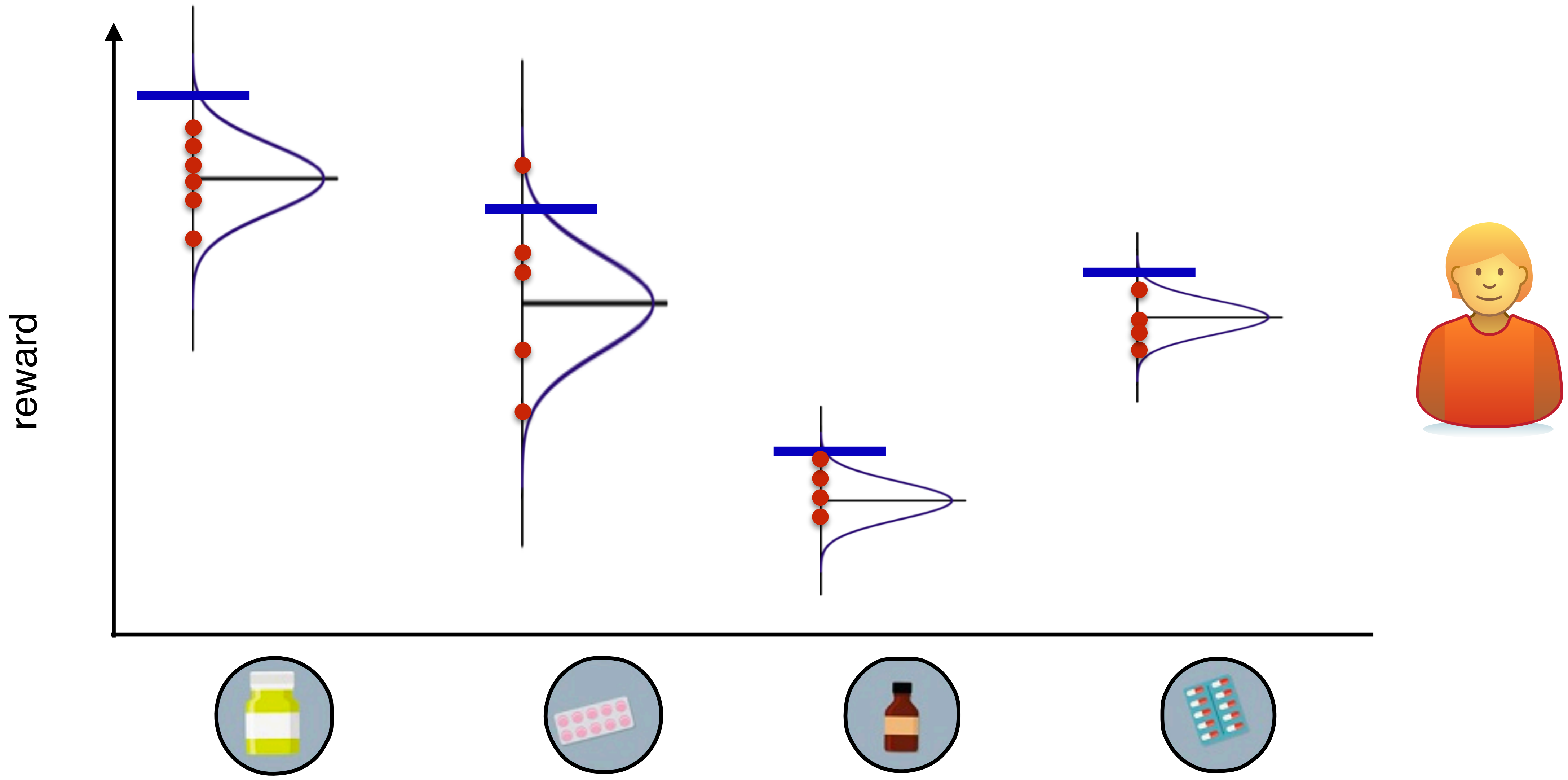
- Exploiting actions we already think are good.
- Exploring new actions to get more information.

Approach: Upper Confidence Bound



[Lai & Robbins ’85, Agrawal ’95, Auer et al. ’02]

Approach: Upper Confidence Bound



Sample complexity: $\frac{|\mathcal{A}|}{\varepsilon^2}$, Regret: $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}| \cdot T}$.

[Lai & Robbins ’85, Agrawal ’95, Auer et al. ’02]

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \textsf{bon}^{(t)}(a)$, w/ $\textsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^{\star}(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^{\star}(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^{\star}(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.

Azuma-Hoeffding

$$\left| \frac{1}{n} \sum_{t=1}^{n} Z_t - \mathbb{E}[Z_t \mid Z_1, \ldots, Z_{t-1}] \right| \leq \sqrt{\frac{\log(\delta^{-1})}{n}} \quad \text{w.p.} \quad 1 - \delta$$

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Round t: By optimism,

$$\max_a f^\star(a) - f^\star(a^{(t)})$$

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Round t: By optimism,

$$\max_a f^\star(a) - f^\star(a^{(t)}) \leq \max_a \bar{f}^{(t)}(a) - f^\star(a^{(t)})$$

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^{\star}(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^{\star}(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^{\star}(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Round t: By optimism,

$$\max_a f^{\star}(a) - f^{\star}(a^{(t)}) \leq \max_a \bar{f}^{(t)}(a) - f^{\star}(a^{(t)}) = \bar{f}^{(t)}(a^{(t)}) - f^{\star}(a^{(t)}),$$

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a) :=$ # arm pulls for a and $\widehat{f}^{(t)}(a) :=$ sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \textsf{bon}^{(t)}(a)$, w/ $\textsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Round t: By optimism,

$$\max_a f^\star(a) - f^\star(a^{(t)}) \leq \max_a \bar{f}^{(t)}(a) - f^\star(a^{(t)}) = \bar{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}),$$

and $\bar{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}) = \widehat{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}) + \textsf{bon}^{(t)}(a^{(t)}) \leq 2\sqrt{\frac{1}{n^{(t)}(a^{(t)})}}$.

Approach: Upper Confidence Bound

UCB algorithm: For each time t:

- Let $n^{(t)}(a)$:= # arm pulls for a and $\widehat{f}^{(t)}(a)$:= sample mean.
- Upper confidence bound: $\bar{f}^{(t)}(a) := \widehat{f}^{(t)}(a) + \mathsf{bon}^{(t)}(a)$, w/ $\mathsf{bon}^{(t)}(a) \propto \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Play $a^{(t)} = \arg\max_{a \in \mathcal{A}} \bar{f}^{(t)}(a)$.

Proof sketch: Let $f^\star(a) = \mathbb{E}[r \mid a]$.

- **Optimism:** $\bar{f}^{(t)}(a) \geq f^\star(a) \;\; \forall a, t$, since $|\widehat{f}^{(t)}(a) - f^\star(a)| \lesssim \sqrt{\frac{1}{n^{(t)}(a)}}$.
- Round t: By optimism,

$$\max_a f^\star(a) - f^\star(a^{(t)}) \leq \max_a \bar{f}^{(t)}(a) - f^\star(a^{(t)}) = \bar{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}),$$

and $\bar{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}) = \widehat{f}^{(t)}(a^{(t)}) - f^\star(a^{(t)}) + \mathsf{bon}^{(t)}(a^{(t)}) \leq 2\sqrt{\frac{1}{n^{(t)}(a^{(t)})}}$.

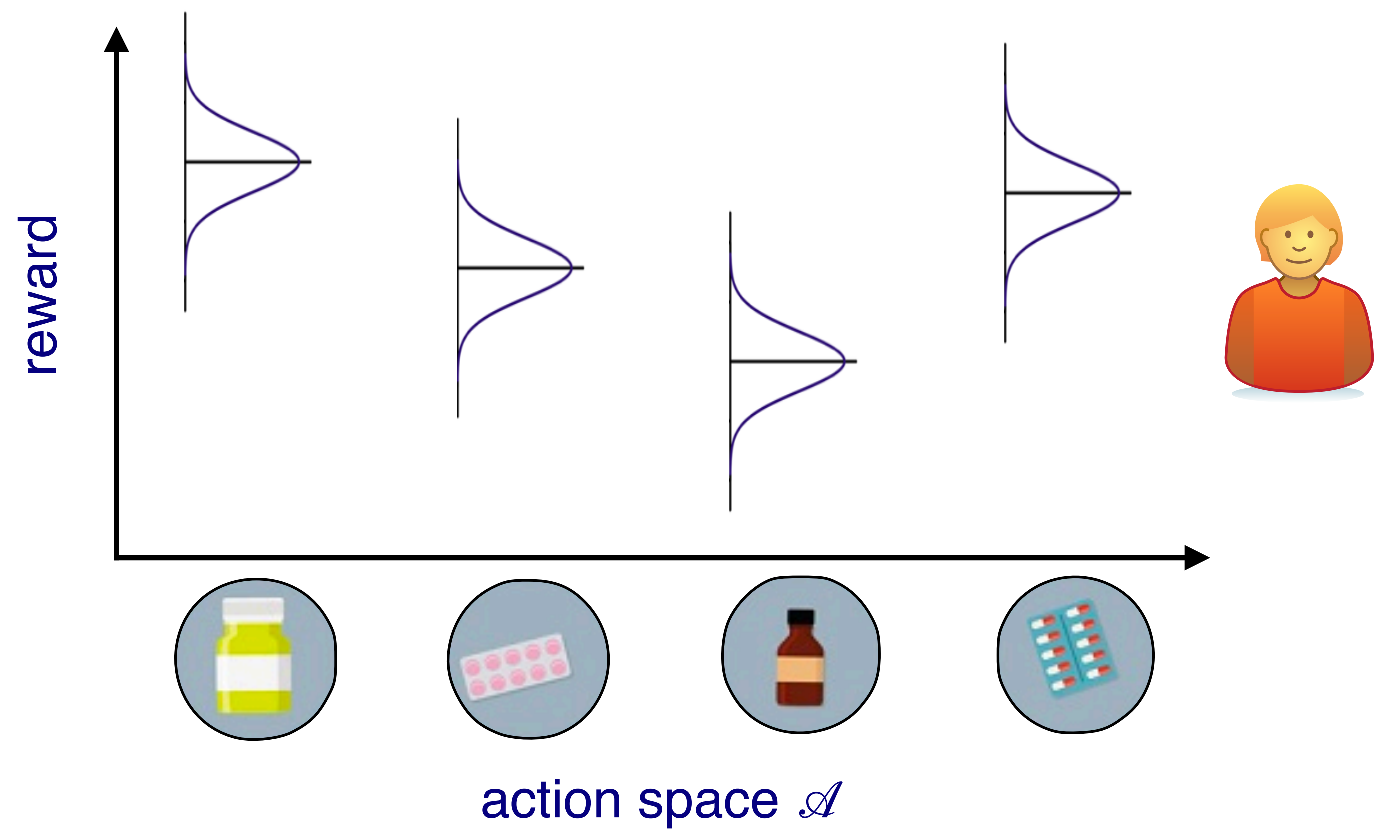
- Regret bound: By pigeonhole,

$$\mathbf{Reg}(T) = \sum_{t=1}^{T} \max_a f^\star(a) - f^\star(a^{(t)}) \lesssim \sum_{t=1}^{T} \sqrt{\frac{1}{n^{(t)}(a^{(t)})}} \leq \sqrt{|\mathcal{A}|T}.$$

Approach: ε-Greedy

Multi-armed bandit

(RL with single state, $H = 1$)



ε**-Greedy**: For each time t:

- Get reward estimate $\widehat{f}^{(t)}(a)$ for each action.
- Play $a^{(t)} = \widehat{a}^{(t)} := \arg\max_a \widehat{f}^{(t)}(a)$ w/ prob. $1 - \varepsilon$, else sample $a^{(t)} \sim \mathcal{A}$ uniformly.

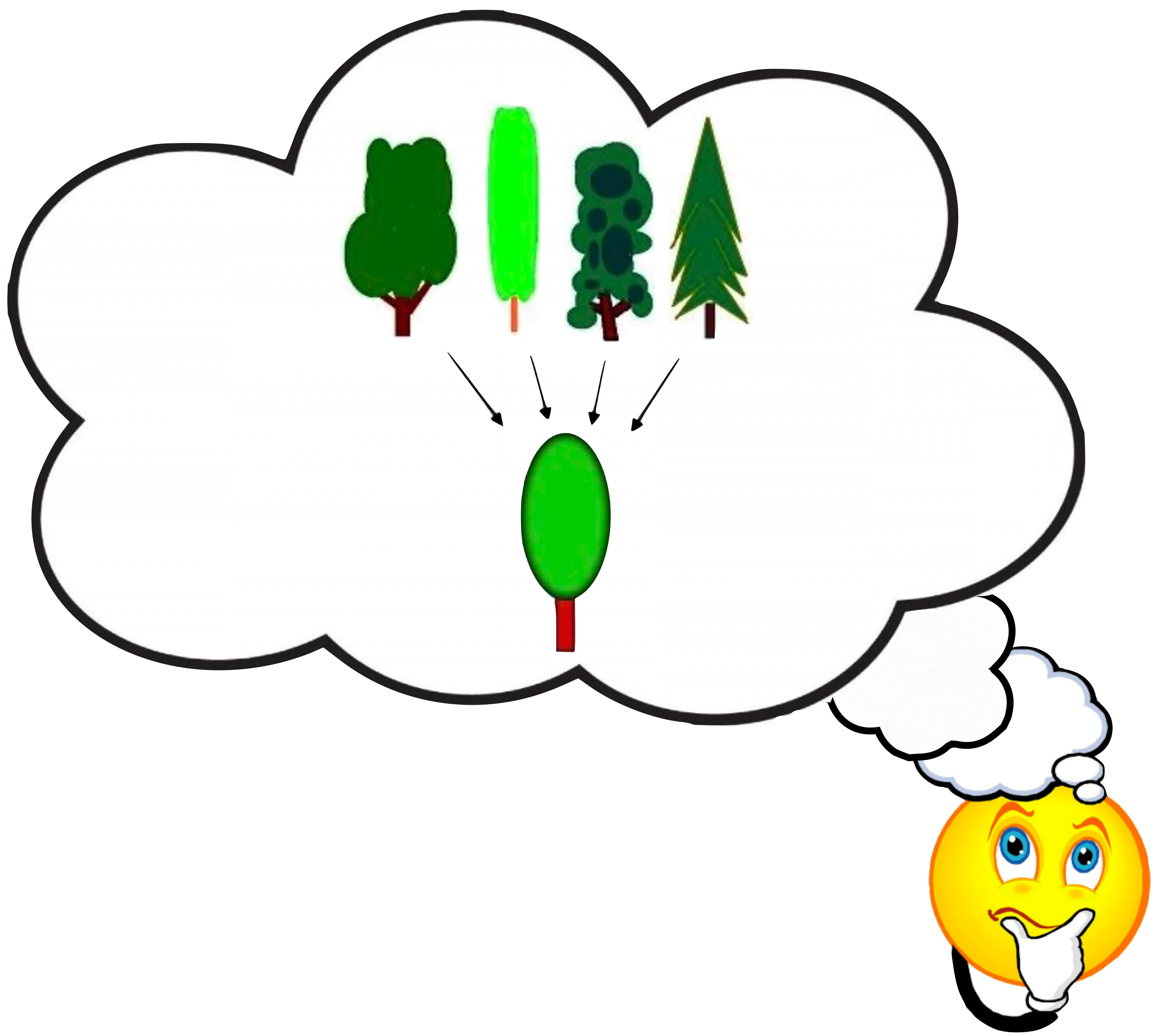
Sample complexity: $\dfrac{|\mathcal{A}|}{\varepsilon^2}$, Regret: $\mathbf{Reg}(T) \leq |\mathcal{A}|^{2/3} T^{2/3}$.

Roadmap

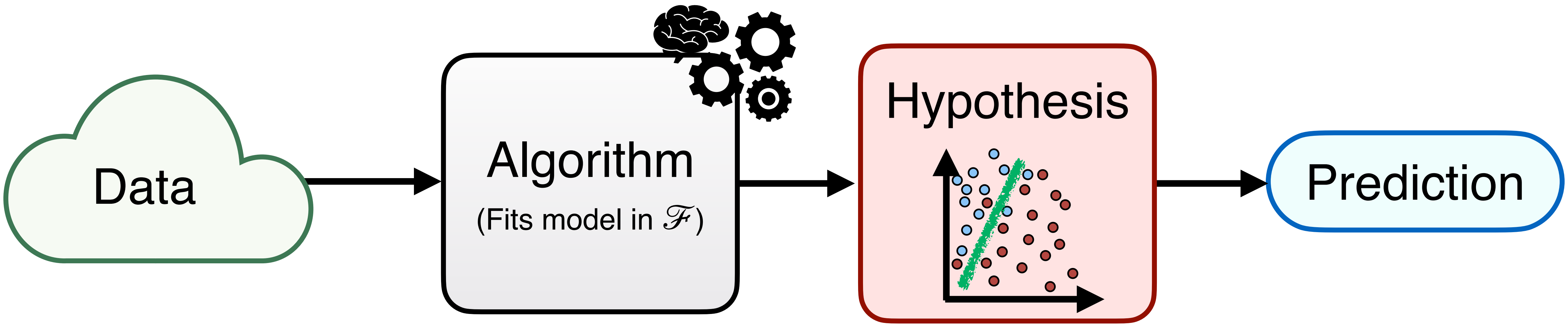
Basic challenges and solutions

- Credit assignment ✔
- Exploration ✔
- Generalization

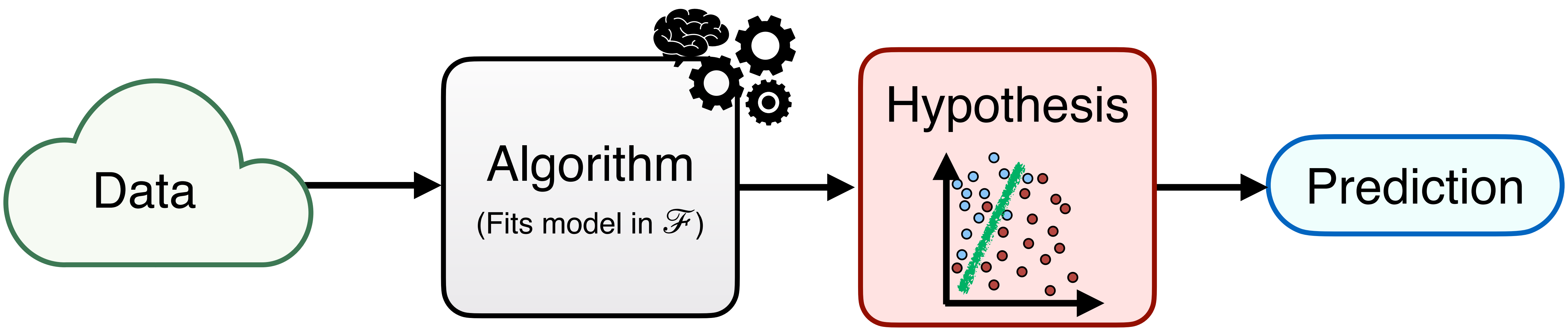
Challenge #3: Generalization



Approach: Statistical learning

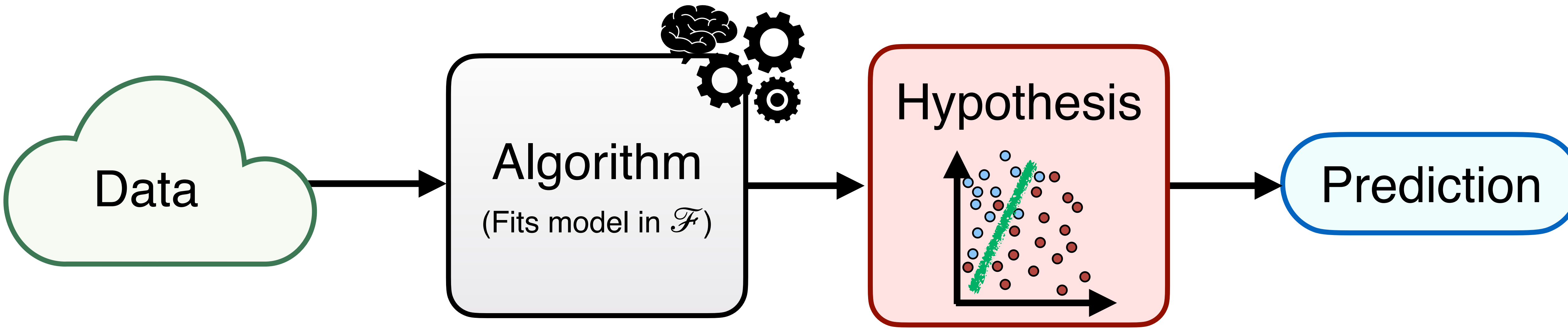


Approach: Statistical learning



Statistical learning: If data is independent/identically distributed, generalize to future examples [Vapnik & Chervonenkis '71].

Approach: Statistical learning



Statistical learning: If data is independent/identically distributed, generalize to future examples [Vapnik & Chervonenkis '71].

Empirical risk minimization ($\widehat{f} = \arg\min_{f \in \mathcal{F}} \mathsf{Error}_{\mathsf{dataset}}(f)$):

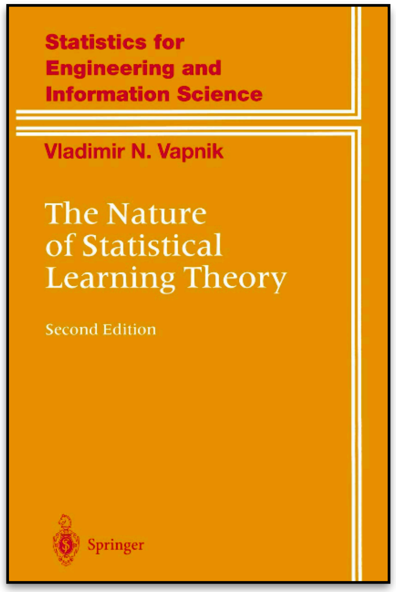
$$\mathsf{Error}_{\mathsf{future}}(\widehat{f}) \leq \min_{f \in \mathcal{F}} \mathsf{Error}_{\mathsf{future}}(f) + \sqrt{\frac{\mathsf{comp}(\mathcal{F})}{n}}.$$

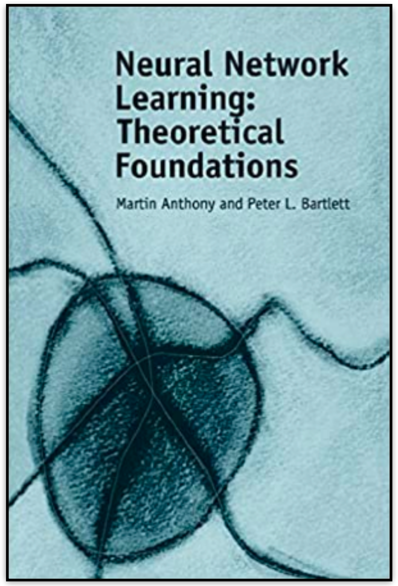
Complexity $\mathsf{comp}(\mathcal{F})$ reflects statistical capacity of $\mathcal{F}$.

Statistical learning: Complexity measures

Complexity measures:

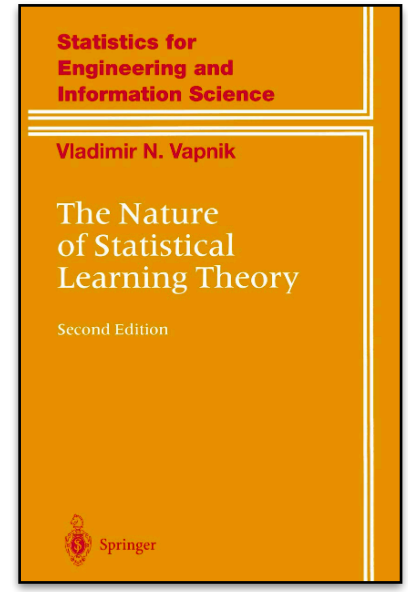
- VC Dimension (classification)
- Fat-shattering dimension (regression)
- Rademacher complexity (both)
- Covering numbers (both)

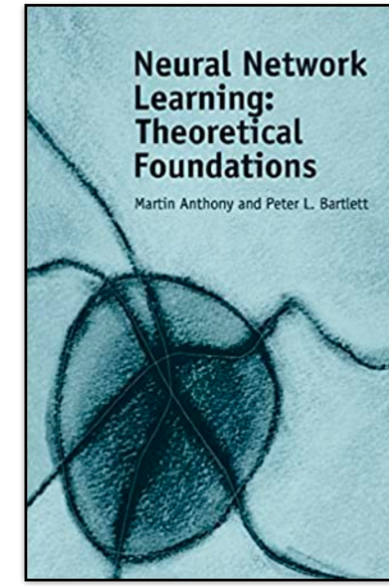




[e.g., Vapnik '95, Anthony & Bartlett '99, Bousquet-Boucheron-Lugosi '03]

Statistical learning: Complexity measures





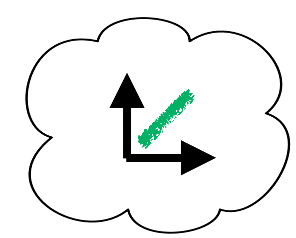
Complexity measures:

- VC Dimension (classification)
- Fat-shattering dimension (regression)
- Rademacher complexity (both)
- Covering numbers (both)

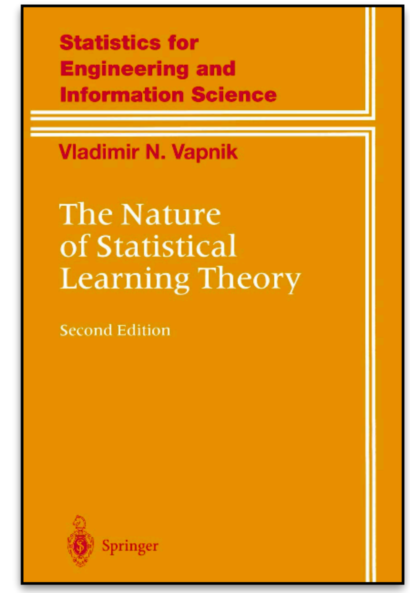
[e.g., Vapnik '95, Anthony & Bartlett '99, Bousquet-Boucheron-Lugosi '03]

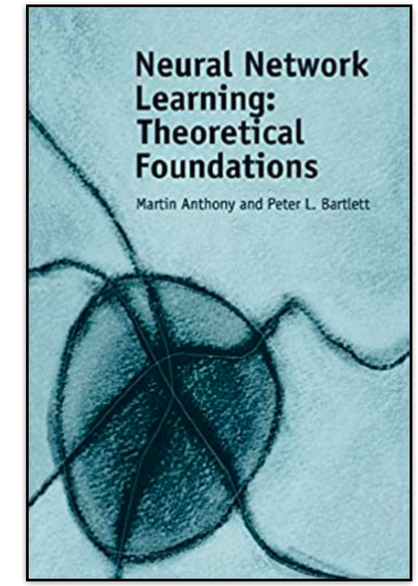
Examples:

- Finite class: $\mathrm{comp}(\mathcal{F}) \leq \log|\mathcal{F}|$
- Linear classification: $\mathrm{comp}(\mathcal{F}) \leq \textsf{dimension}$ (VC dim)
- Linear regression: $\mathrm{comp}(\mathcal{F}) \leq (\textsf{weight norm})^2$ (fat-shattering)
- Similar bounds for neural nets, kernels, ...



Statistical learning: Complexity measures





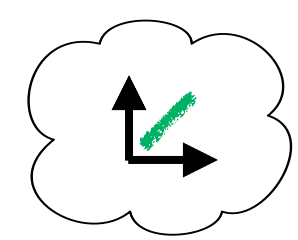
Complexity measures:

- VC Dimension (classification)
- Fat-shattering dimension (regression)
- Rademacher complexity (both)
- Covering numbers (both)

[e.g., Vapnik '95, Anthony & Bartlett '99, Bousquet-Boucheron-Lugosi '03]

Examples:

- Finite class: $\mathrm{comp}(\mathcal{F}) \leq \log|\mathcal{F}|$
- Linear classification: $\mathrm{comp}(\mathcal{F}) \leq \textsf{dimension}$ (VC dim)
- Linear regression: $\mathrm{comp}(\mathcal{F}) \leq (\textsf{weight norm})^2$ (fat-shattering)
- Similar bounds for neural nets, kernels, ...



No explicit dependence on $|\mathcal{X}|$!

RL: The need for modeling and generalization

Challenge: States/observations are typically rich/complex/high-dimensional.

- Ex: robotics: x_h = camera image, $\mathcal{X}$ = all possible images

  $\implies |\mathcal{X}|$ = intractably large

Approach: Use hypothesis class $\mathcal{F}$ to model:

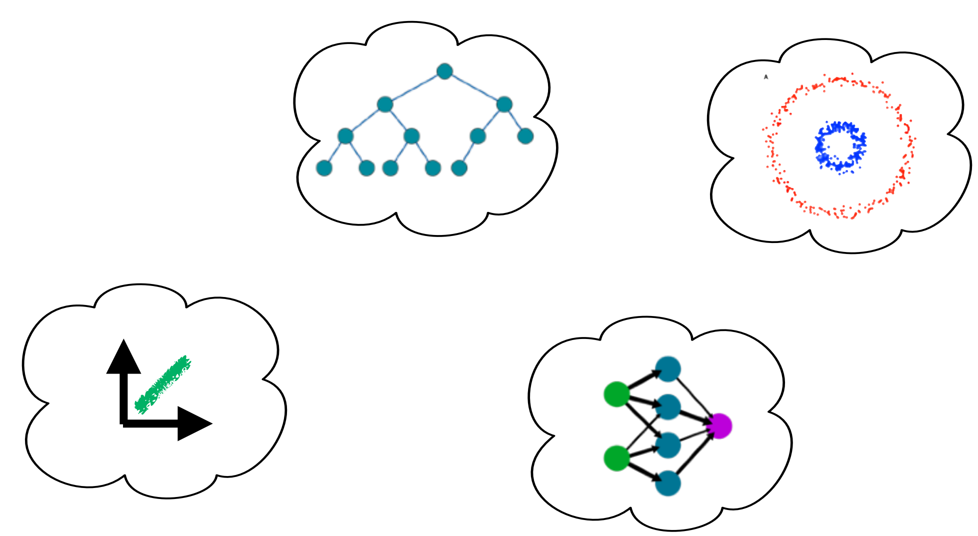
- Rewards/responses/treatment effects
- Dynamics
- Long-term rewards

  $\vdots$

In general, model class $\mathcal{F}$ might consist of:

- Deep neural networks
- Generalized linear models
- Kernels

  $\vdots$



Research questions: Supervised learning vs. RL

Algorithm design

General-purpose algorithmic principles that work for any $\mathcal{F}$?

Research questions: Supervised learning vs. RL

Algorithm design

General-purpose algorithmic principles that work for any $\mathcal{F}$?

- Supervised learning: Minimize empirical risk (take best fitting model)

Research questions: Supervised learning vs. RL

Algorithm design

General-purpose algorithmic principles that work for any $\mathcal{F}$?

- Supervised learning: Minimize empirical risk (take best fitting model)
- Decision making (contextual bandits, RL, ...): ???

Research questions: Supervised learning vs. RL

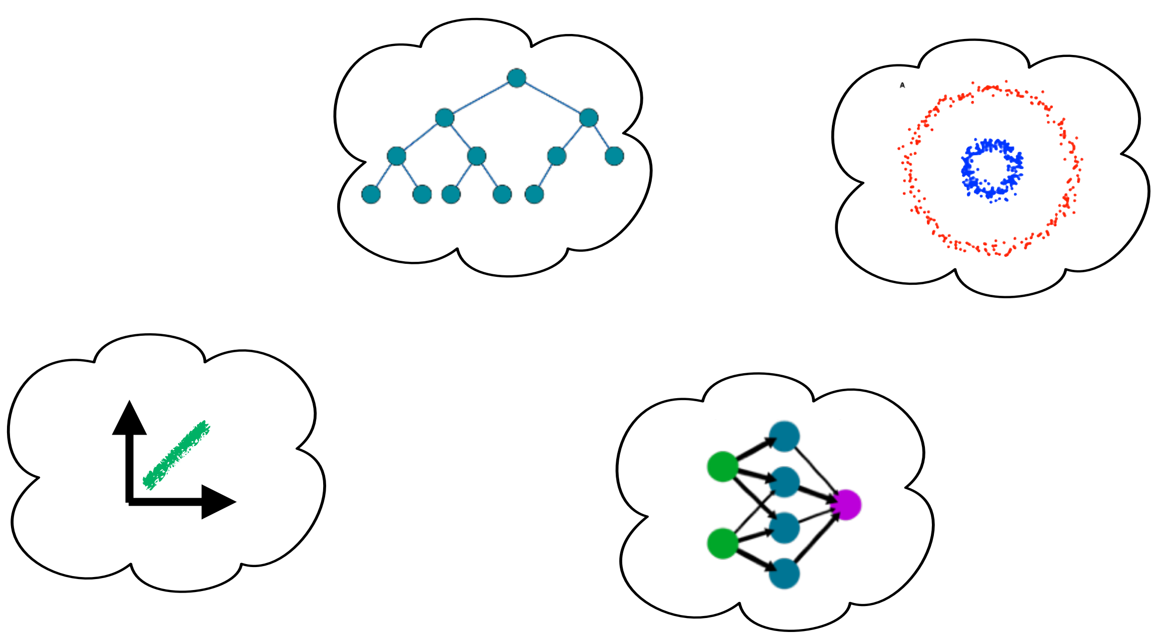
Algorithm design

General-purpose algorithmic principles that work for any $\mathcal{F}$?

- Supervised learning: Minimize empirical risk (take best fitting model)
- Decision making (contextual bandits, RL, ...): ???

What we want:

Algorithm makes accurate decisions out of the box for any $\mathcal{F}$.



Research questions: Supervised learning vs. RL

Sample complexity

How many samples are necessary / sufficient to learn with $\mathcal{F}$?

Research questions: Supervised learning vs. RL

Sample complexity

How many samples are necessary / sufficient to learn with $\mathcal{F}$?

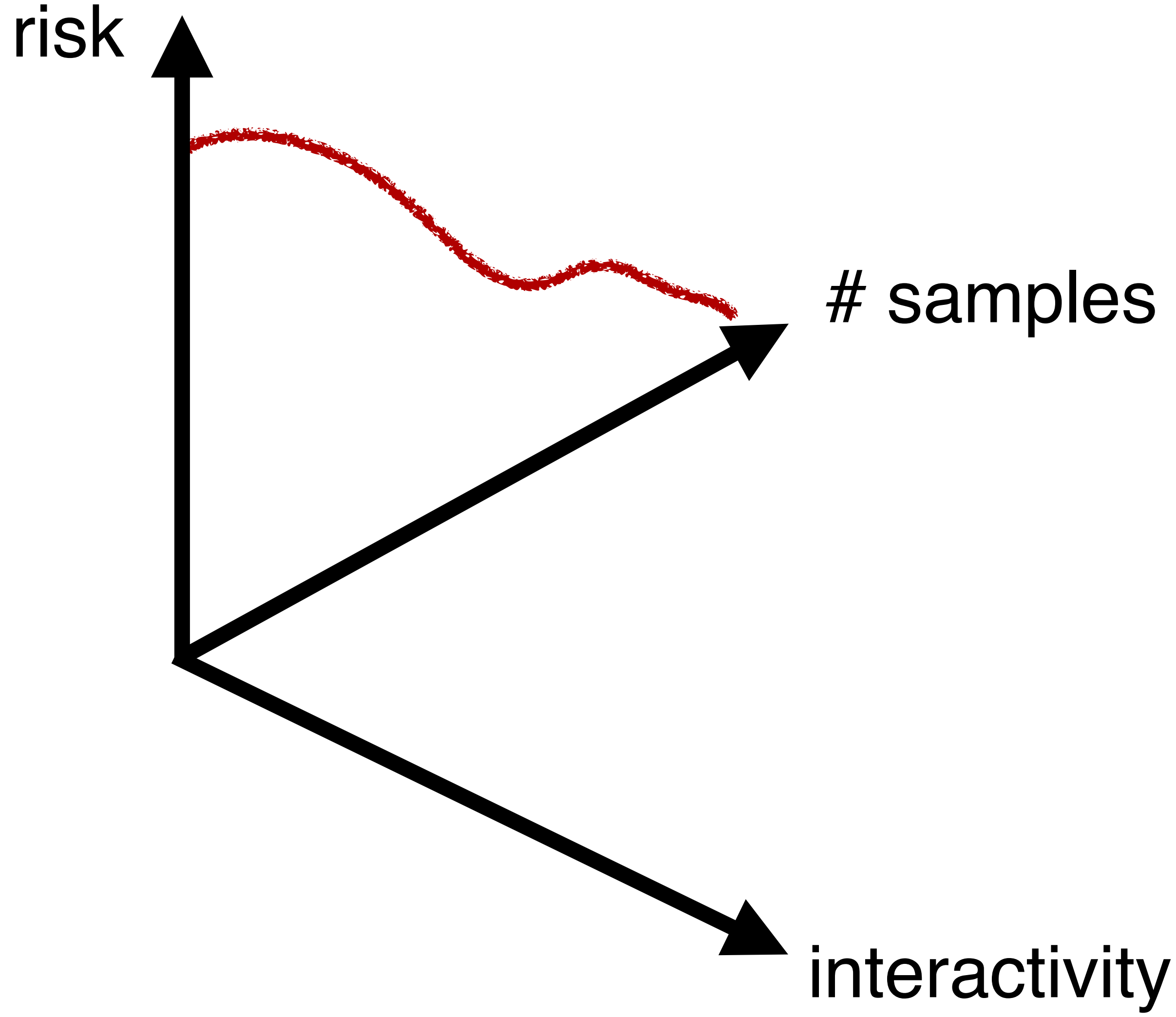
- Supervised learning: Vapnik-Chervonenkis (VC) theory, PAC learning

Research questions: Supervised learning vs. RL

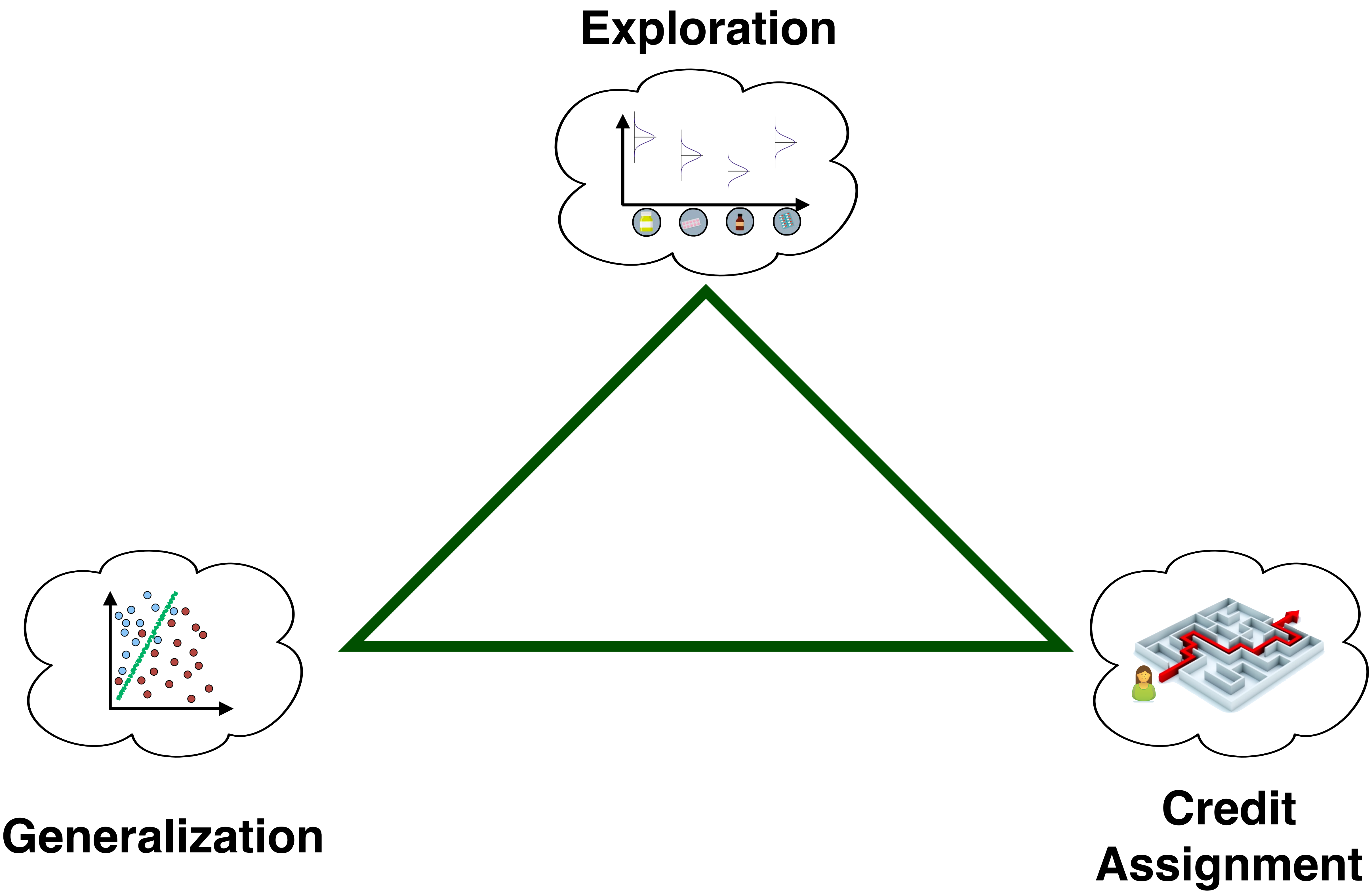
Sample complexity

How many samples are necessary / sufficient to learn with $\mathcal{F}$?

- Supervised learning: Vapnik-Chervonenkis (VC) theory, PAC learning
- Decision making (contextual bandits, RL, ...): ???

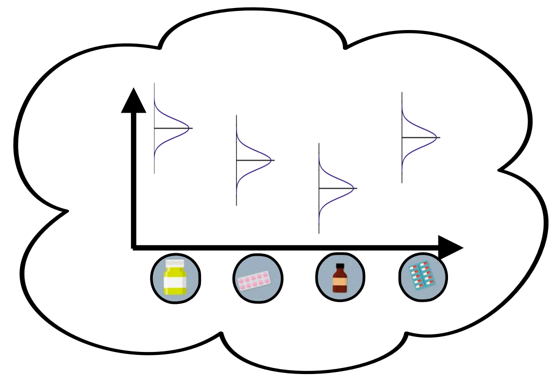


Challenges of RL



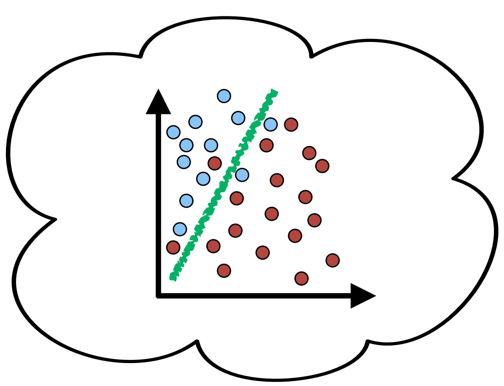
Challenges of RL

Exploration

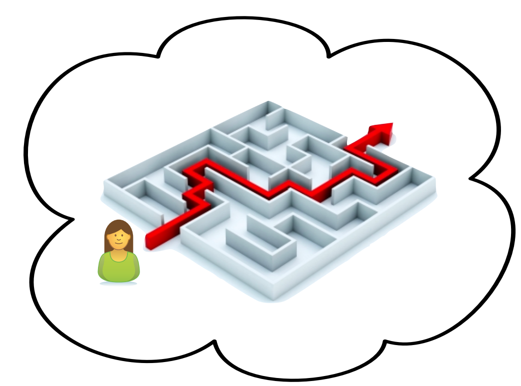


Generalization + Exploration:
Contextual Bandits

Exploration + Credit:
Tabular PAC-RL



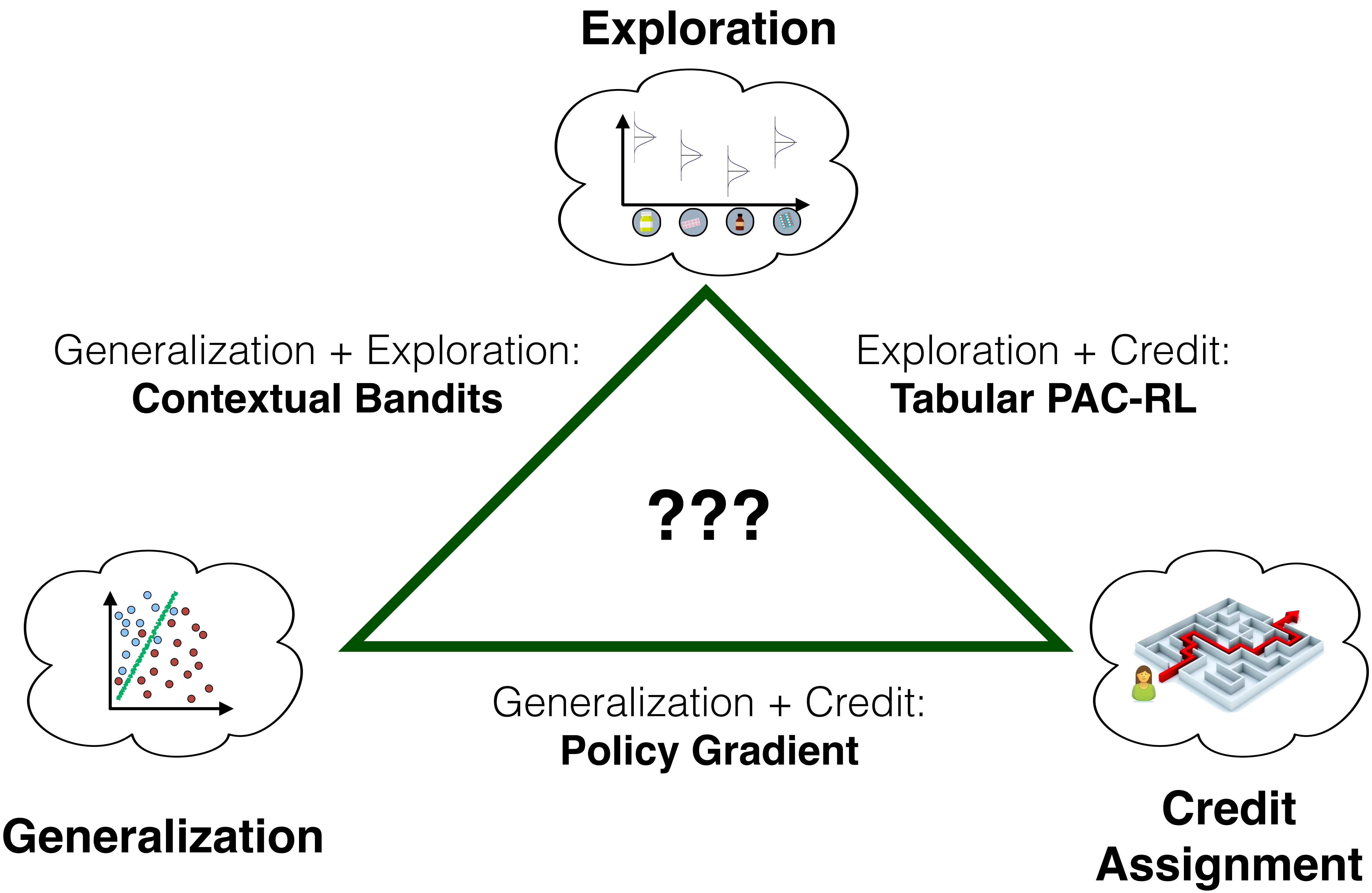
Generalization + Credit:
Policy Gradient



Generalization

Credit Assignment

Challenges of RL



Roadmap

Basic challenges and solutions

- Credit assignment ✔
- Exploration ✔
- Generalization ✔

Roadmap

Basic challenges and solutions

- Credit assignment ✔
- Exploration ✔
- Generalization ✔

Intermediate level

- Exploration + credit assignment: Tabular RL
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

Roadmap

Basic challenges and solutions

- Credit assignment ✔
- Exploration ✔
- Generalization ✔

Intermediate level

- Exploration + credit assignment: Tabular RL
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

Roadmap

Basic challenges and solutions

- Credit assignment ✔
- Exploration ✔
- Generalization ✔

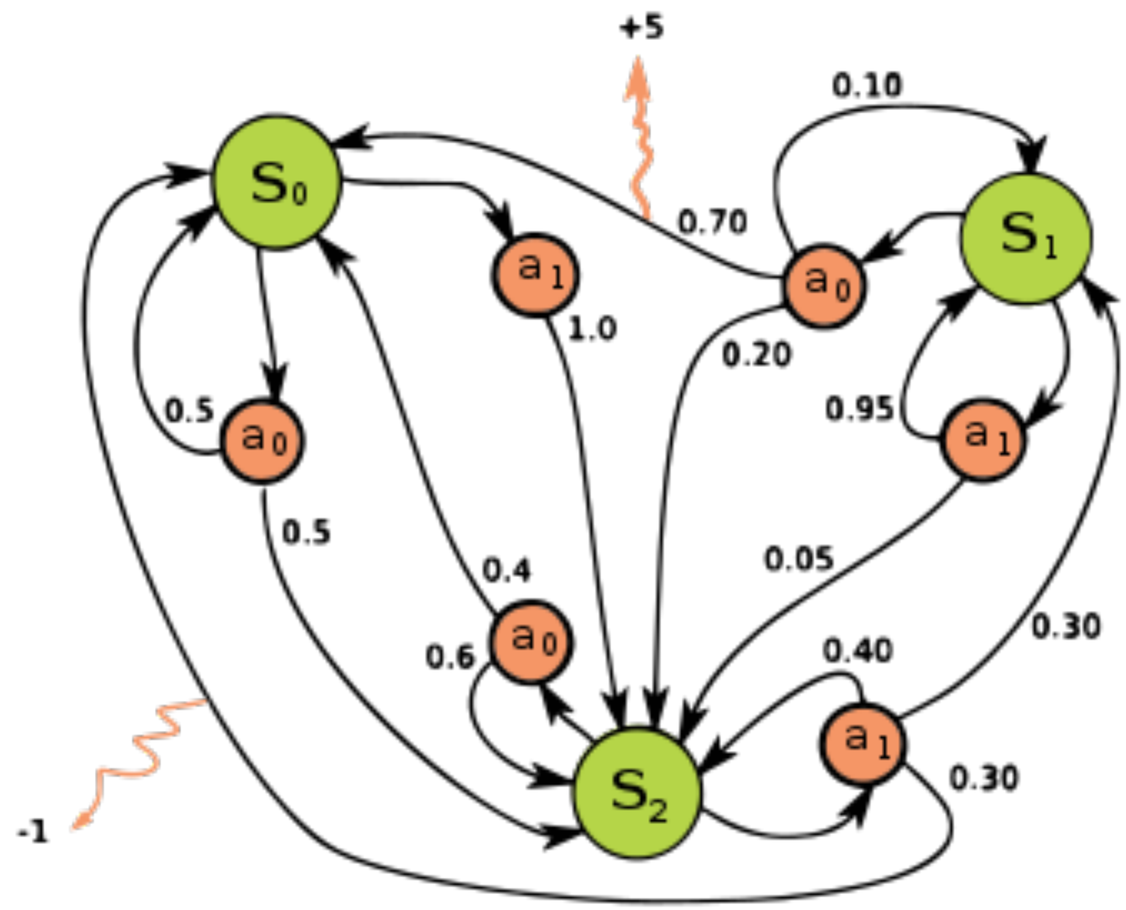
Intermediate level

- Exploration + credit assignment: Tabular RL
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

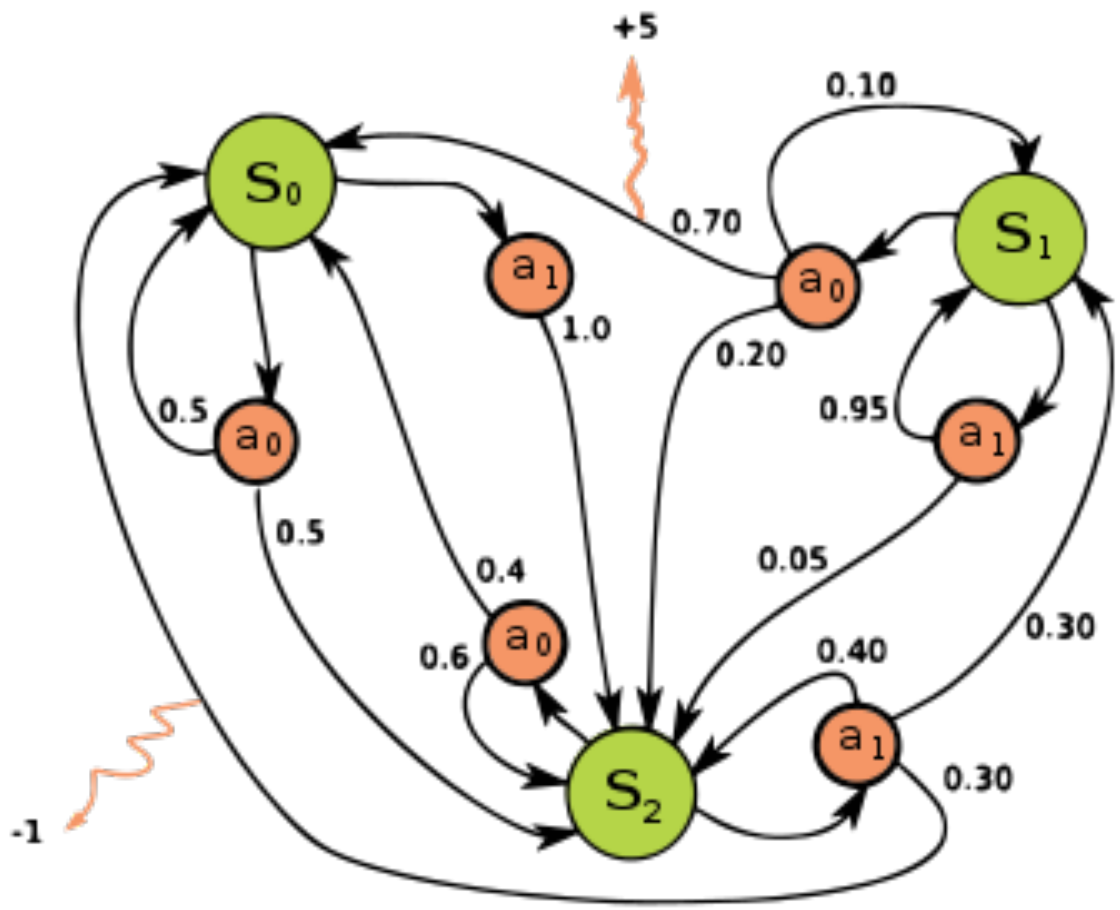
Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.



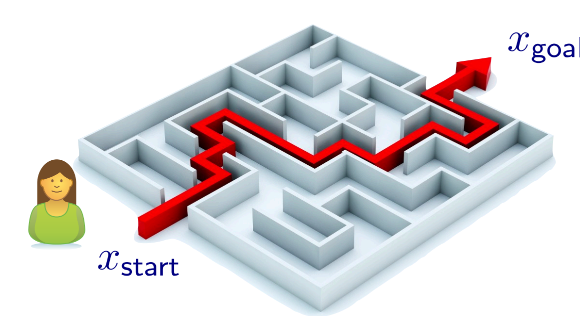
Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.



Non-trivial problem:

- Naive (uniform) exploration has sample complexity $|\mathscr{A}|^H$



Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x, a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \ldots, T$:

Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \ldots, T$:

- State-action frequencies:

$$n^{(t)}(x,a,x') := \sum_{i<t,h} \mathbb{I}\{(x_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)}) = (x,a,x')\}, \quad n^{(t)}(x,a) := \sum_{x'} n^{(t)}(x,a,x').$$

Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \ldots, T$:

- State-action frequencies:

$$n^{(t)}(x,a,x') := \sum_{i<t,h} \mathbb{I}\{(x_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)}) = (x,a,x')\}, \quad n^{(t)}(x,a) := \sum_{x'} n^{(t)}(x,a,x').$$

- Estimate transitions/rewards:

$$\widehat{P}^{(t)}(x' \mid x,a) := \frac{n^{(t)}(x,a,x')}{n^{(t)}(x,a)}, \quad \text{and} \quad \widehat{f}^{(t)}(x,a) := \textsf{sample mean for } (x,a).$$

Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \ldots, T$:

- State-action frequencies:

$$n^{(t)}(x,a,x') := \sum_{i<t,h} \mathbb{I}\{(x_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)}) = (x,a,x')\}, \quad n^{(t)}(x,a) := \sum_{x'} n^{(t)}(x,a,x').$$

- Estimate transitions/rewards:

$$\widehat{P}^{(t)}(x' \mid x,a) := \frac{n^{(t)}(x,a,x')}{n^{(t)}(x,a)}, \quad \text{and} \quad \widehat{f}^{(t)}(x,a) := \textsf{sample mean for } (x,a).$$

- Exploration bonus: $\textsf{bon}^{(t)}(x,a) \propto H \cdot \sqrt{\frac{1}{n^{(t)}(x,a)}}$.

Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \dots, T$:

- State-action frequencies:

$$n^{(t)}(x,a,x') := \sum_{i<t,h} \mathbb{I}\{(x_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)}) = (x,a,x')\}, \quad n^{(t)}(x,a) := \sum_{x'} n^{(t)}(x,a,x').$$

- Estimate transitions/rewards:

$$\widehat{P}^{(t)}(x' \mid x,a) := \frac{n^{(t)}(x,a,x')}{n^{(t)}(x,a)}, \quad \text{and} \quad \widehat{f}^{(t)}(x,a) := \text{sample mean for } (x,a).$$

- Exploration bonus: $\mathsf{bon}^{(t)}(x,a) \propto H \cdot \sqrt{\frac{1}{n^{(t)}(x,a)}}$.

Value iteration with

$$\left\{ \widehat{f}^{(t)} + \mathsf{bon}^{(t)}, \widehat{P}^{(t)} \right\}$$

Exploration + Credit Assignment: Tabular RL

Tabular MDP: $|\mathcal{X}| < \infty$, $|\mathcal{A}| < \infty$. Trans. $P(x' \mid x, a)$, rewards $f^\star(x,a) := \mathbb{E}_{r \sim R(x,a)}[r]$.

UCB-VI Algorithm [Azar et al. '17]: For $t = 1, \ldots, T$:

- State-action frequencies:

$$n^{(t)}(x,a,x') := \sum_{i<t,h} \mathbb{I}\{(x_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)}) = (x,a,x')\}, \quad n^{(t)}(x,a) := \sum_{x'} n^{(t)}(x,a,x').$$

- Estimate transitions/rewards:

$$\widehat{P}^{(t)}(x' \mid x,a) := \frac{n^{(t)}(x,a,x')}{n^{(t)}(x,a)}, \quad \text{and} \quad \widehat{f}^{(t)}(x,a) := \textsf{sample mean for } (x,a).$$

- Exploration bonus: $\textsf{bon}^{(t)}(x,a) \propto H \cdot \sqrt{\frac{1}{n^{(t)}(x,a)}}$.

- **Optimistic value iteration:** Starting with $\overline{V}_{H+1}^{(t)}(x) := 0$, iterate

$$\overline{Q}_h^{(t)}(x,a) := \widehat{f}^{(t)}(x,a) + \textsf{bon}^{(t)}(x,a) + \mathbb{E}_{x' \sim \widehat{P}^{(t)}(x,a)}[\overline{V}_{h+1}^{(t)}(x')],$$

  and $\overline{V}_h^{(t)}(x) := \max_a \overline{Q}_h^{(t)}(x,a)$.

- Final policy: $\pi_h^{(t)}(x) = \arg\max_a \overline{Q}_h^{(t)}(x,a)$, so $a_h^{(t)} = \pi_h^{(t)}(x_h^{(t)})$.

Tabular RL: UCB-VI

Regret bound for UCB-VI [Azar et al. '17]:*

$$\mathbf{Reg}(T) \leq H\sqrt{|\mathcal{X}||\mathcal{A}|\,T}.$$

$\Longrightarrow\ \mathrm{poly}\big(|\mathcal{X}|\,, |\mathcal{A}|\,, H\big)$ sample complexity and computation.

Tabular RL: UCB-VI

Regret bound for UCB-VI [Azar et al. '17]:*

$$\mathbf{Reg}(T) \leq H\sqrt{|\mathcal{X}||\mathcal{A}|T}.$$

$\implies$ $\mathrm{poly}\big(|\mathcal{X}|, |\mathcal{A}|, H\big)$ sample complexity and computation.

Tabular RL history:

- E^3 [Kearns & Singh '02], $R_{\max}$ [Brafman & Tennenholtz '02]: Polynomial sample complexity
- Delayed-Q learning [Strehl et al. '06]: Sample comp. linear in $|\mathcal{X}|$.
- UCRL [Jaksch, Ortner, & Auer '10]: Optimal regret/sample comp w.r.t. T (resp. ε).
- UCB-VI [Azar, Osban, & Munos '17]: Minimax optimal.

Tabular RL: UCB-VI

Regret bound for UCB-VI [Azar et al. '17]:*

$$\mathbf{Reg}(T) \leq H\sqrt{|\mathcal{X}||\mathcal{A}|T}.$$

$\implies \mathrm{poly}\left(|\mathcal{X}|, |\mathcal{A}|, H\right)$ sample complexity and computation.

Tabular RL history:

- E^3 [Kearns & Singh '02], $R_{\max}$ [Brafman & Tennenholtz '02]: Polynomial sample complexity
- Delayed-Q learning [Strehl et al. '06]: Sample comp. linear in $|\mathcal{X}|$.
- UCRL [Jaksch, Ortner, & Auer '10]: Optimal regret/sample comp w.r.t. T (resp. ε).
- UCB-VI [Azar, Osban, & Munos '17]: Minimax optimal.
- UCB-Q [Jin et al. '18]: Near-optimal regret for **model-free**.

(first four items:) **"model-based"**

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall\, (x,a,h)$.

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall \, (x,a,h)$.

Proof: Assume $\overline{Q}_{h+1}^{(t)}(x,a) \geq Q_{h+1}^\star(x,a)$.

$$Q_h^\star(x,a) - \overline{Q}_h^{(t)}(x,a)$$

Bellman Equation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + V_{h+1}^\star(x_{h+1}) \mid x_h = x, a_h = a\right]$$

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall\, (x,a,h)$.

Bellman Equation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + V_{h+1}^\star(x_{h+1}) \mid x_h = x, a_h = a\right]$$

Proof: Assume $\overline{Q}_{h+1}^{(t)}(x,a) \geq Q_{h+1}^\star(x,a)$.

$$Q_h^\star(x,a) - \overline{Q}_h^{(t)}(x,a)$$

$$\leq \mathsf{err}^{(t)}(x,a) - \mathsf{bon}^{(t)}(x,a) + \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right],$$

w/ $\mathsf{err}^{(t)}(x,a) := |f^\star(x,a) - \widehat{f}^{(t)}(x,a)| + \|P(x,a) - \widehat{P}^{(t)}(x,a)\|_1 \lesssim \mathsf{bon}^{(t)}(x,a)$

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall\, (x,a,h)$.

Bellman Equation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + V_{h+1}^\star(x_{h+1}) \mid x_h = x, a_h = a\right]$$

Proof: Assume $\overline{Q}_{h+1}^{(t)}(x,a) \geq Q_{h+1}^\star(x,a)$.

$$Q_h^\star(x,a) - \overline{Q}_h^{(t)}(x,a)$$

$$\leq \mathsf{err}^{(t)}(x,a) - \mathsf{bon}^{(t)}(x,a) + \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right],$$

w/ $\mathsf{err}^{(t)}(x,a) := |f^\star(x,a) - \widehat{f}^{(t)}(x,a)| + \|P(x,a) - \widehat{P}^{(t)}(x,a)\|_1 \lesssim \mathsf{bon}^{(t)}(x,a)$

$$\leq \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right] \leq 0.$$

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall\, (x,a,h)$.

Proof: Assume $\overline{Q}_{h+1}^{(t)}(x,a) \geq Q_{h+1}^\star(x,a)$.

Bellman Equation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + V_{h+1}^\star(x_{h+1}) \mid x_h = x, a_h = a\right]$$

$$Q_h^\star(x,a) - \overline{Q}_h^{(t)}(x,a)$$

$$\leq \mathsf{err}^{(t)}(x,a) - \mathsf{bon}^{(t)}(x,a) + \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right],$$

w/ $\mathsf{err}^{(t)}(x,a) := |f^\star(x,a) - \widehat{f}^{(t)}(x,a)| + \|P(x,a) - \widehat{P}^{(t)}(x,a)\|_1 \lesssim \mathsf{bon}^{(t)}(x,a)$

$$\leq \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right] \leq 0.$$

Regret bound for optimistic algorithms (“performance difference lemma” [Kakade ’03]):

$$J(\pi^\star) - J(\pi^{(t)}) = \sum_{h=1}^{H} \mathbb{E}^{\pi^{(t)}}\left[Q_h^\star(x,\pi_h^\star(x_h)) - Q_h^\star(x,\pi_h^{(t)}(x_h))\right] \lesssim \mathbb{E}^{\pi^{(t)}}\left[\sum_{h=1}^{H} \mathsf{bon}^{(t)}(x_h,a_h)\right]$$

Tabular RL: UCB-VI

Proof sketch: Claim: Optimism. With high prob., $\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \;\; \forall\, (x,a,h)$.

Bellman Equation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + V_{h+1}^\star(x_{h+1}) \mid x_h = x, a_h = a\right]$$

Proof: Assume $\overline{Q}_{h+1}^{(t)}(x,a) \geq Q_{h+1}^\star(x,a)$.

$$Q_h^\star(x,a) - \overline{Q}_h^{(t)}(x,a)$$

$$\leq \mathsf{err}^{(t)}(x,a) - \mathsf{bon}^{(t)}(x,a) + \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right],$$

w/ $\mathsf{err}^{(t)}(x,a) := |f^\star(x,a) - \widehat{f}^{(t)}(x,a)| + \|P(x,a) - \widehat{P}^{(t)}(x,a)\|_1 \lesssim \mathsf{bon}^{(t)}(x,a)$

$$\leq \mathbb{E}\left[V_{h+1}^\star(x_{h+1}) - \overline{V}_{h+1}^{(t)}(x_{h+1}) \mid x,a\right] \leq 0.$$

Regret bound for optimistic algorithms ("performance difference lemma" [Kakade '03]):

$$J(\pi^\star) - J(\pi^{(t)}) = \sum_{h=1}^H \mathbb{E}^{\pi^{(t)}}\left[Q_h^\star(x,\pi_h^\star(x_h)) - Q_h^\star(x,\pi_h^{(t)}(x_h))\right] \lesssim \mathbb{E}^{\pi^{(t)}}\left[\sum_{h=1}^H \mathsf{bon}^{(t)}(x_h,a_h)\right]$$

so that by pigeonhole,

$$\mathbf{Reg}(T) \lesssim \sum_{t=1}^T \sum_{h=1}^H \mathsf{bon}^{(t)}(x_h^{(t)}, a_h^{(t)}) \approx \sum_{t=1}^T \sum_{h=1}^H \sqrt{\frac{1}{n^{(t)}(x_h^{(t)}, a_h^{(t)})}} \leq \mathrm{poly}(H) \cdot \sqrt{|\mathcal{X}||\mathcal{A}|T}.$$

Roadmap

Basic challenges and solutions ✔

- Credit assignment
- Exploration
- Generalization

Intermediate level

- Exploration + credit assignment: Tabular RL ✔
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

Roadmap

Basic challenges and solutions ✔

- Credit assignment
- Exploration
- Generalization

Intermediate level

- Exploration + credit assignment: Tabular RL ✔
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

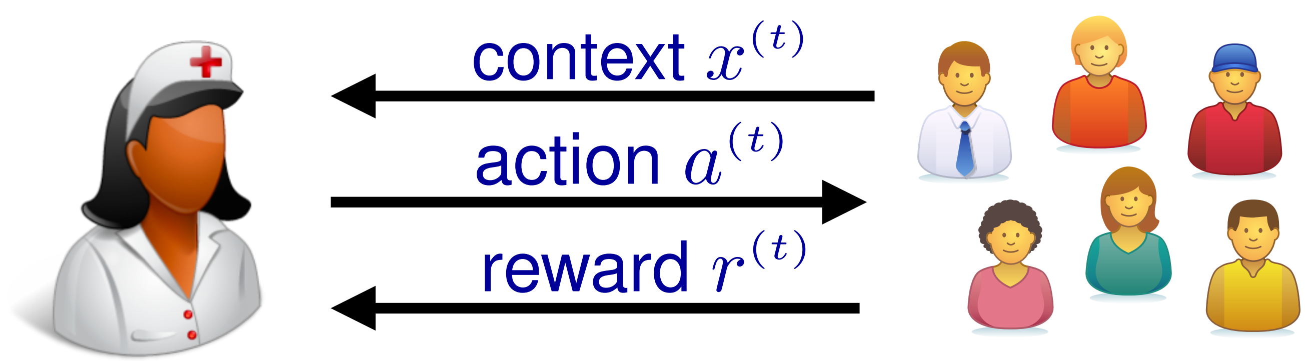
The frontier: Exploration + generalization + credit assignment

Exploration + Generalization: Contextual Bandits

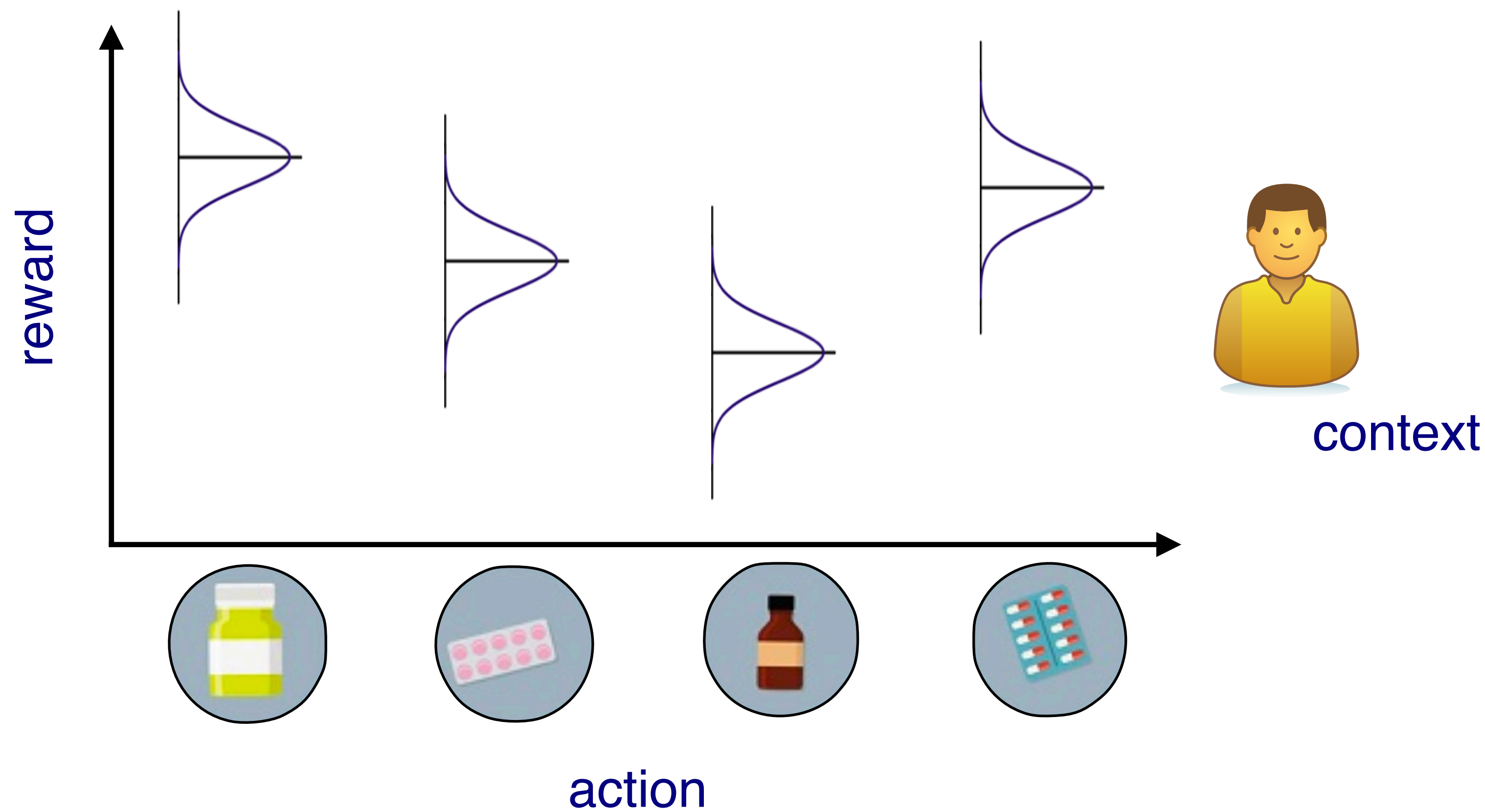
Contextual bandits:

- Reinforcement learning with $H = 1$
- Need to generalize across contexts (states)

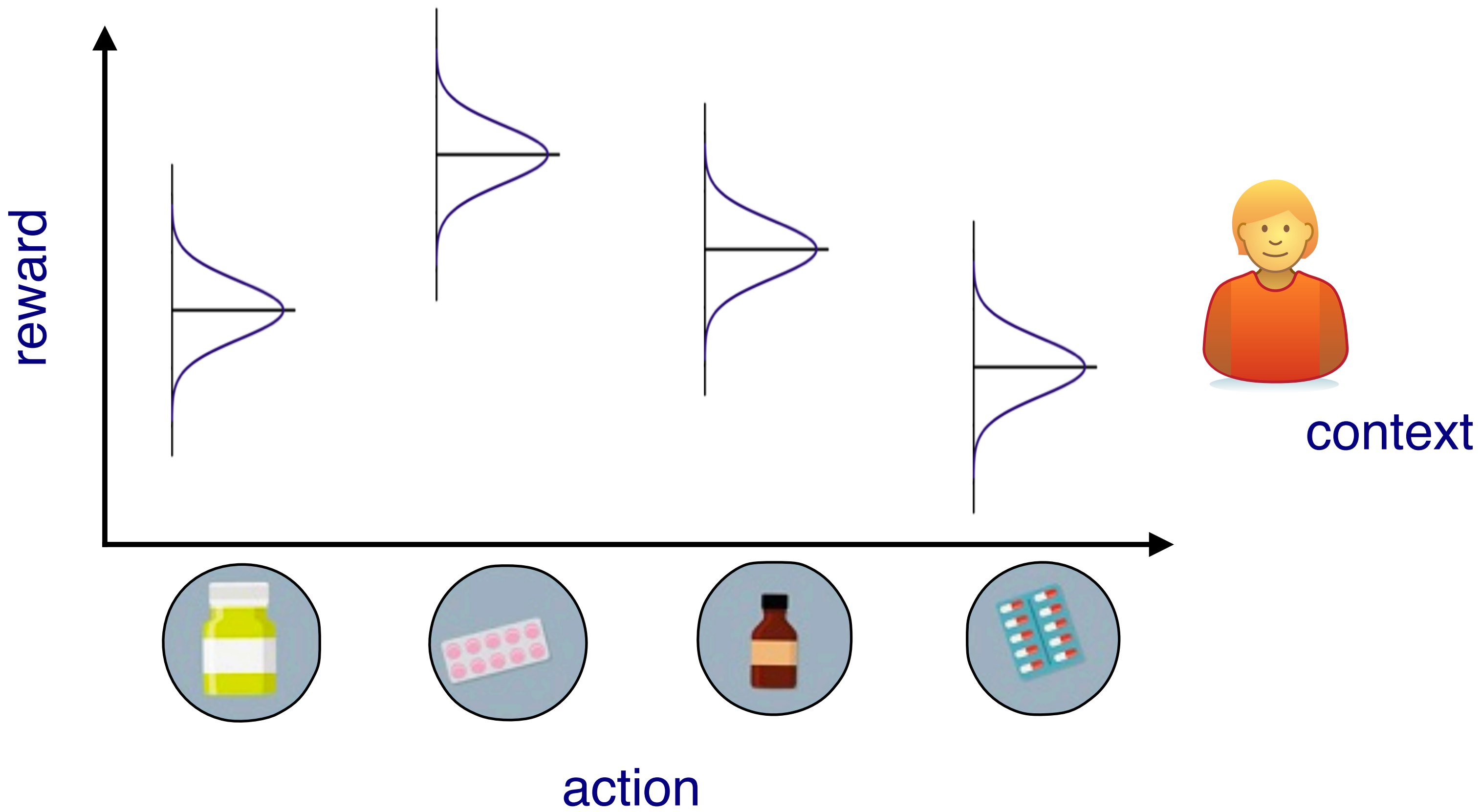
Ex: Personalized medicine



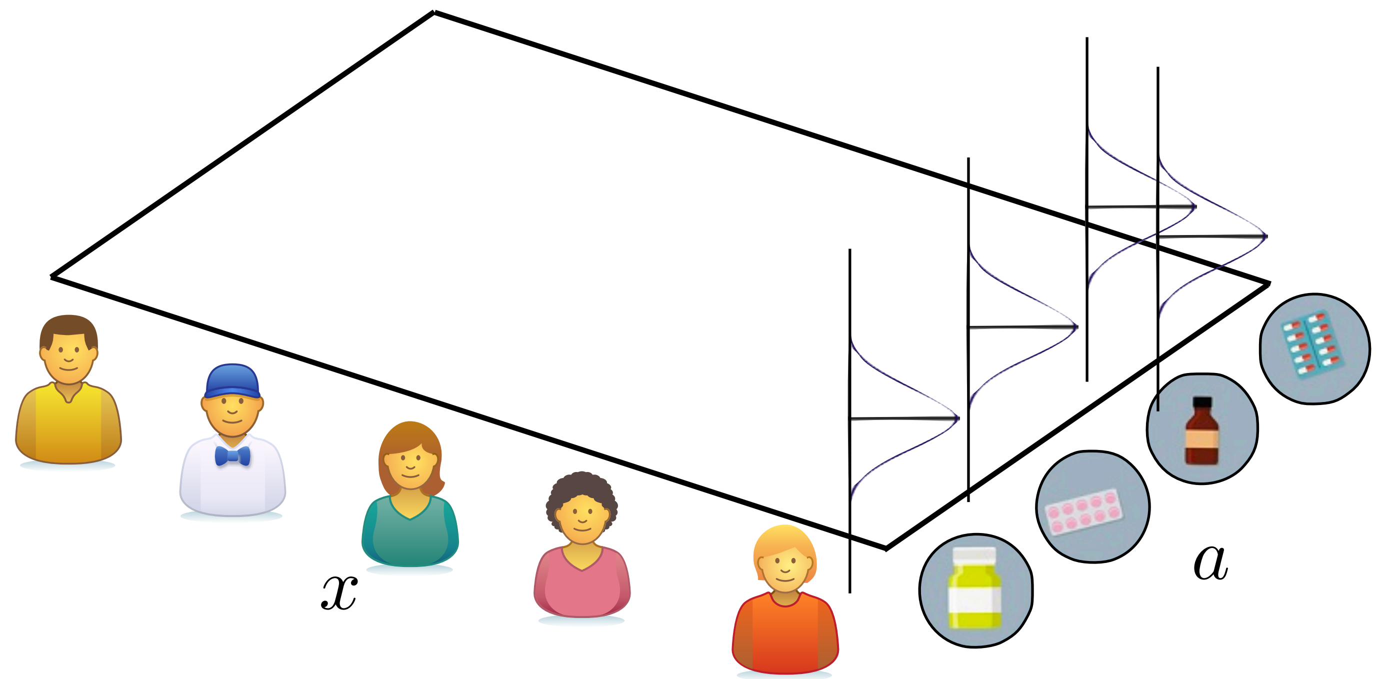
Exploration + Generalization: Contextual Bandits



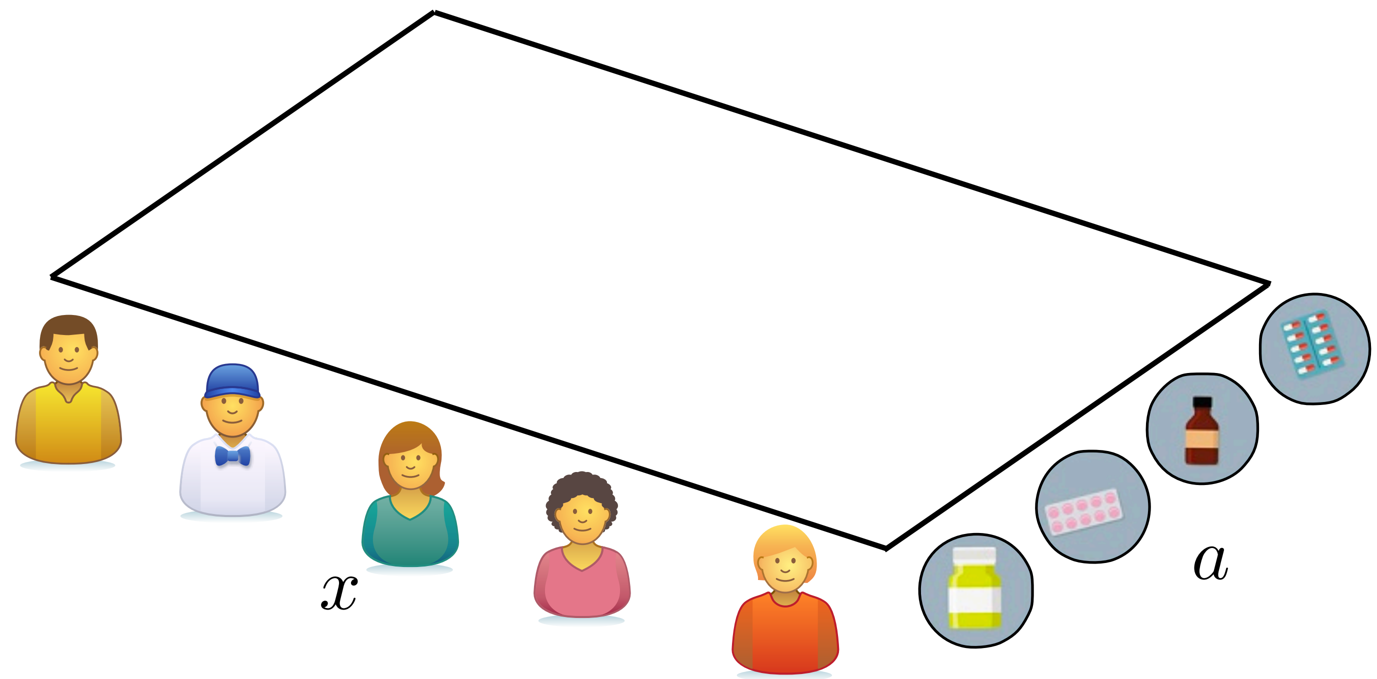
Exploration + Generalization: Contextual Bandits



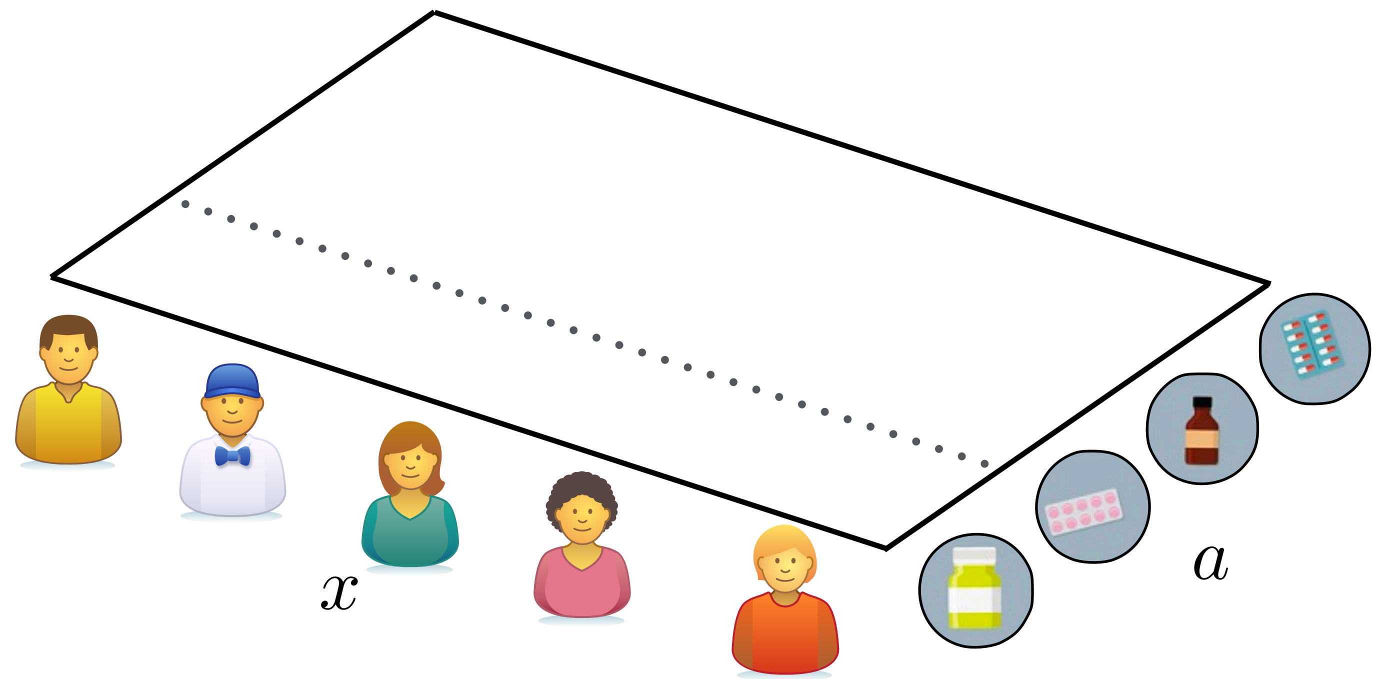
Exploration + Generalization: Contextual Bandits



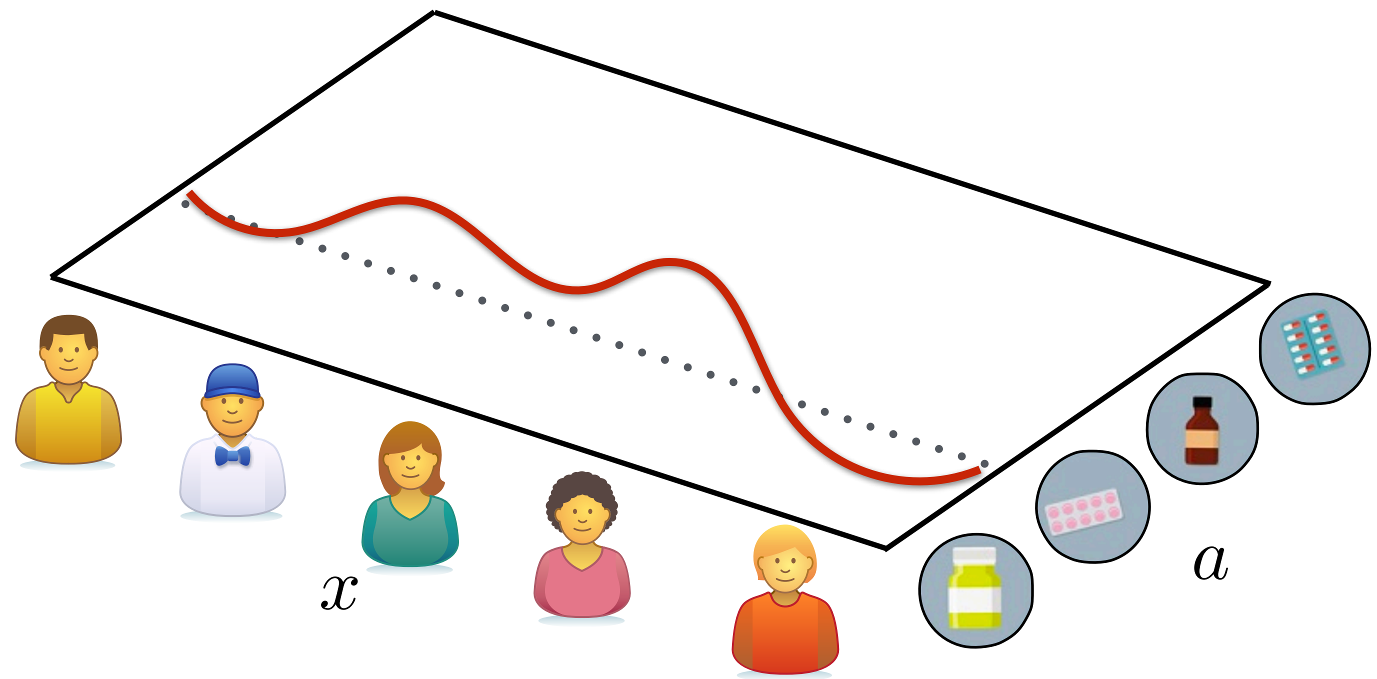
Exploration + Generalization: Contextual Bandits



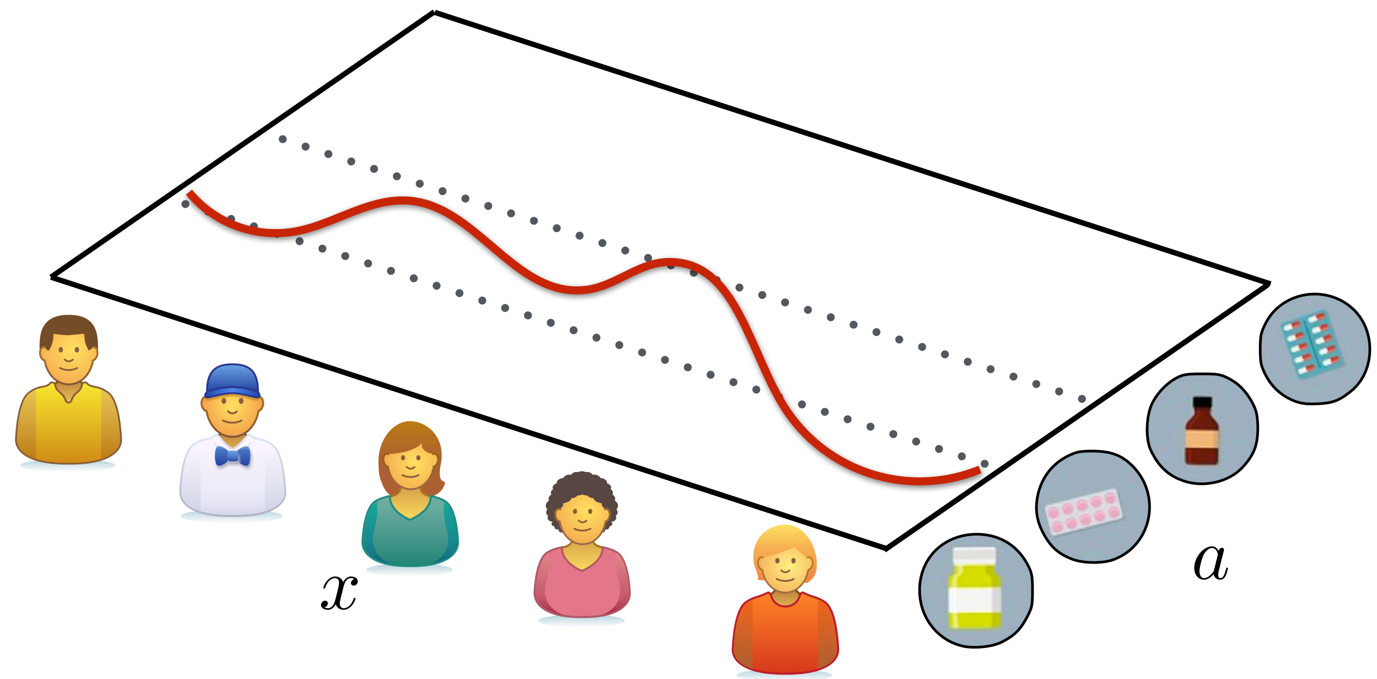
Exploration + Generalization: Contextual Bandits



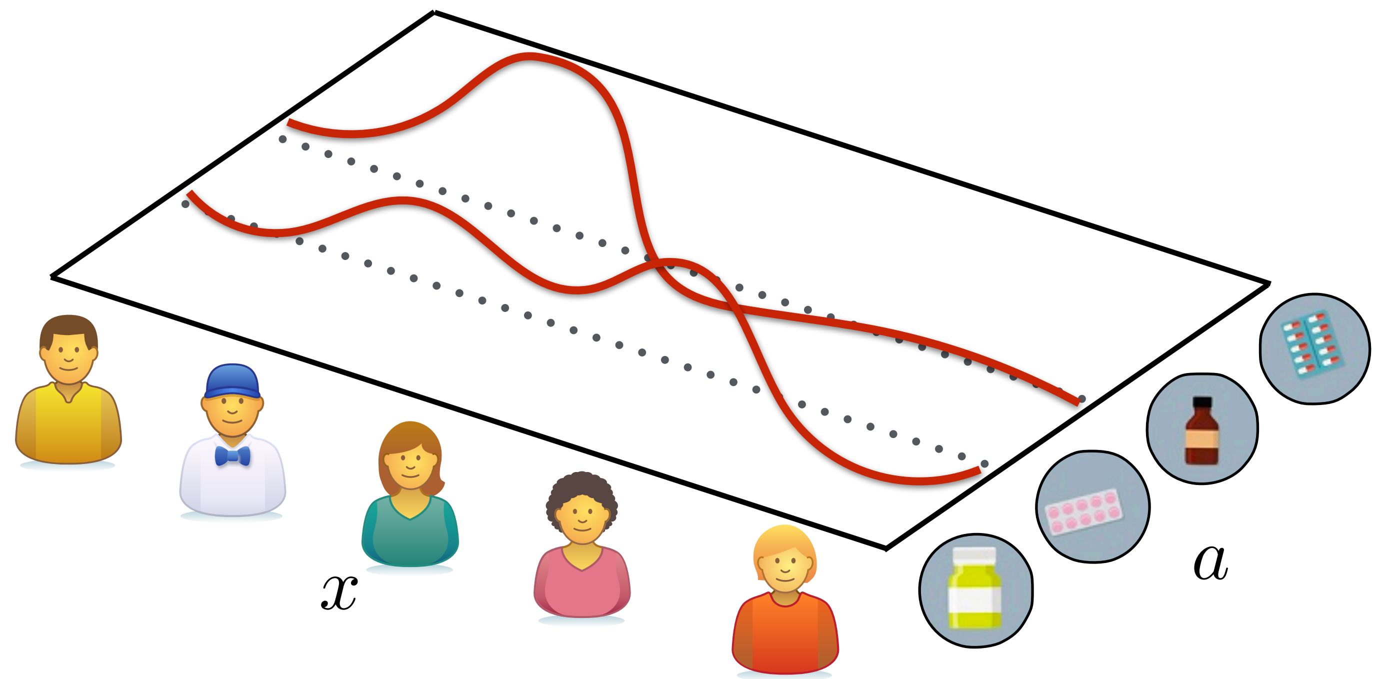
Exploration + Generalization: Contextual Bandits



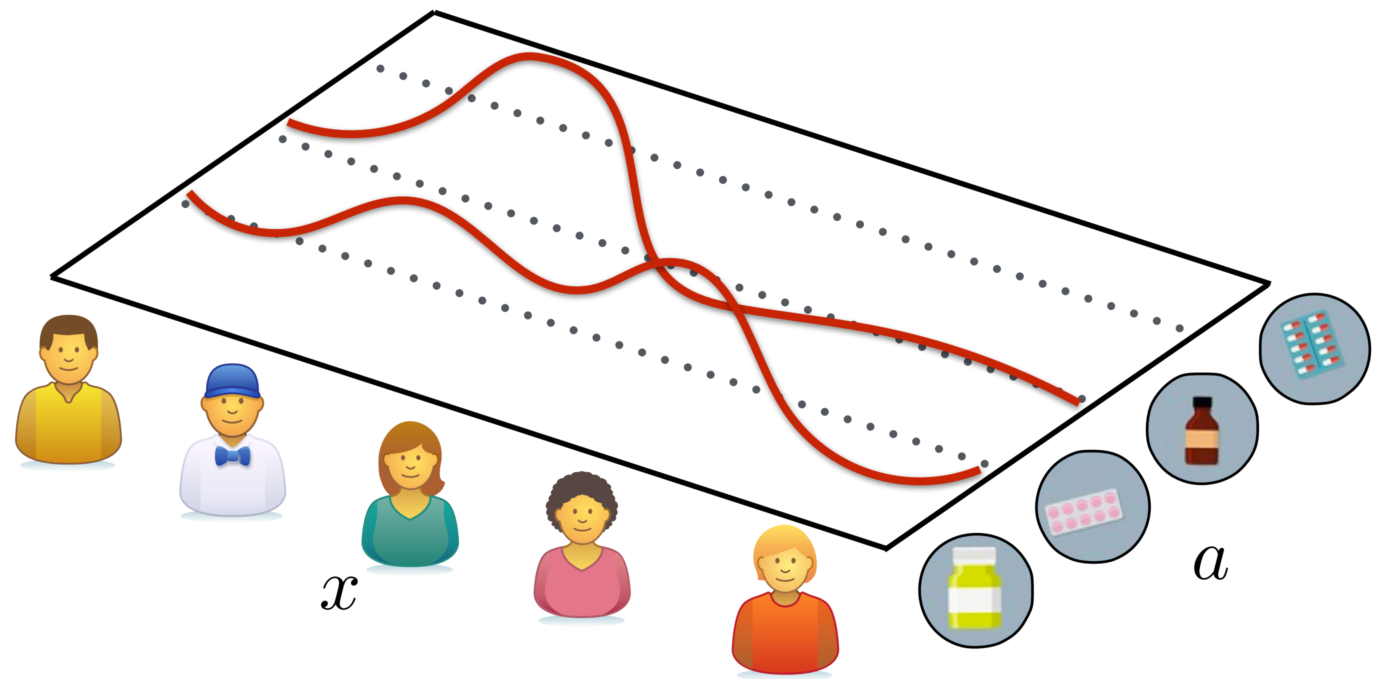
Exploration + Generalization: Contextual Bandits



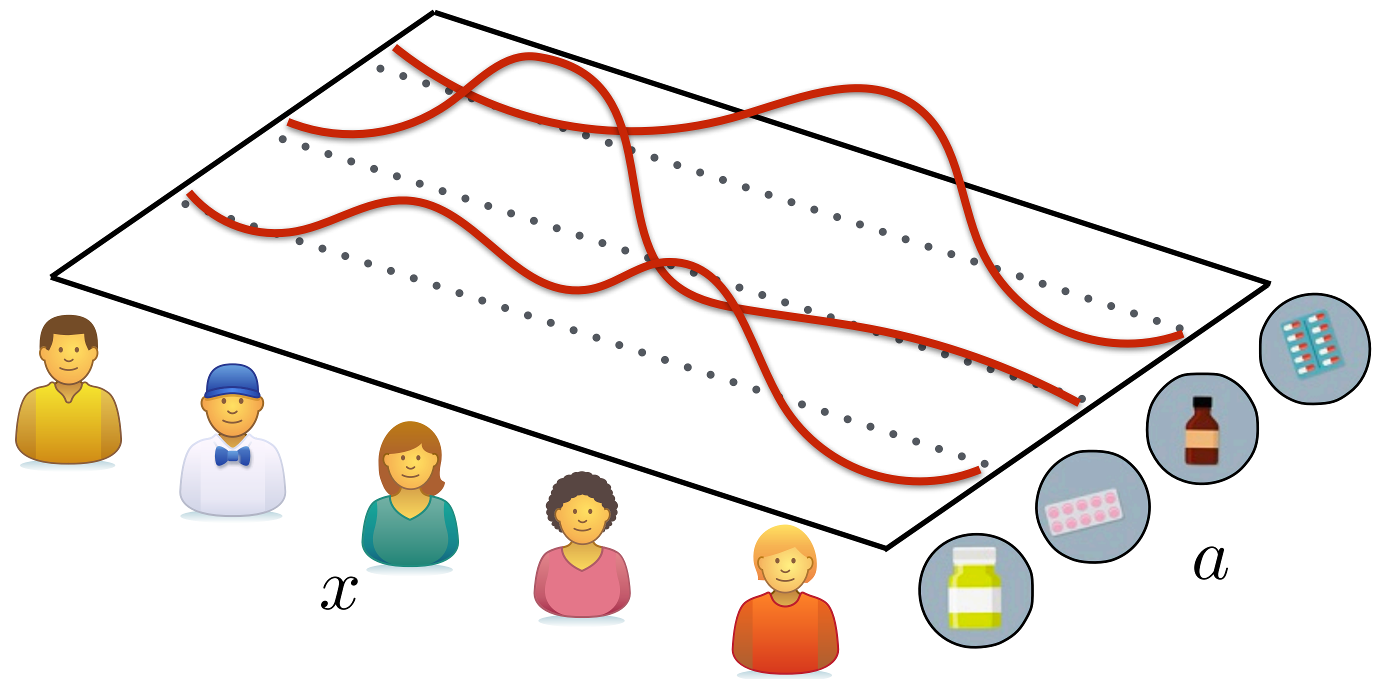
Exploration + Generalization: Contextual Bandits



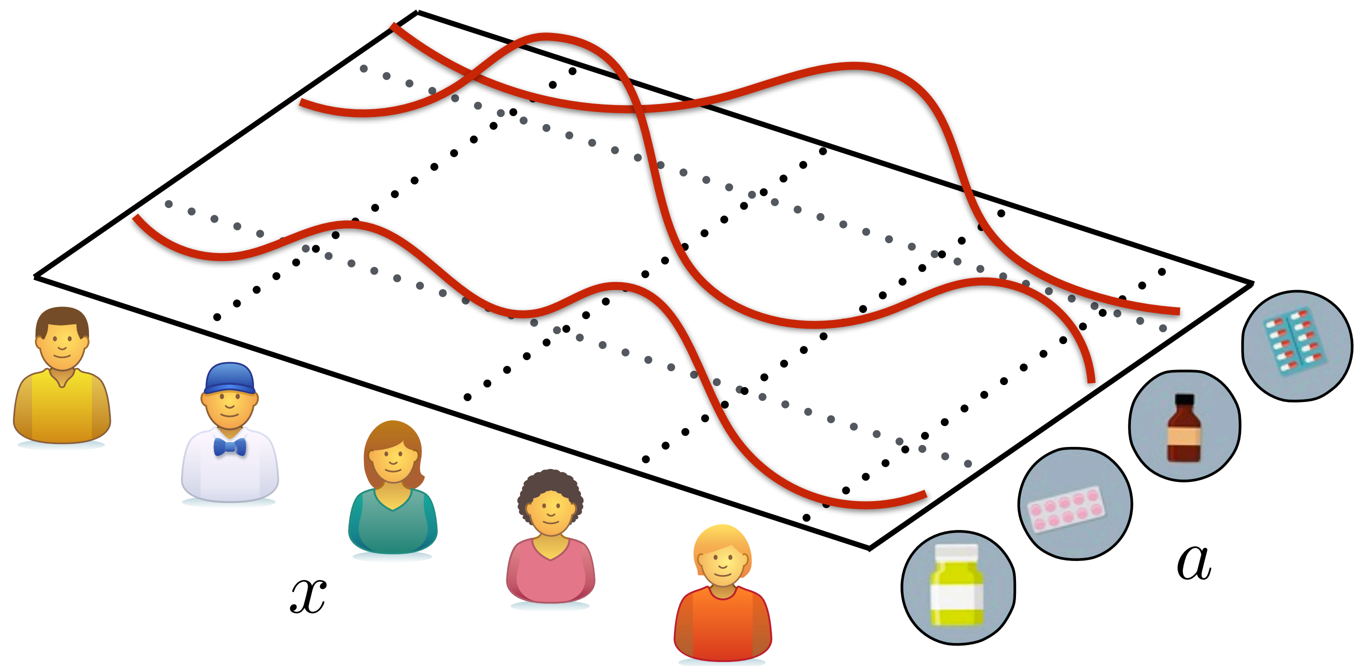
Exploration + Generalization: Contextual Bandits



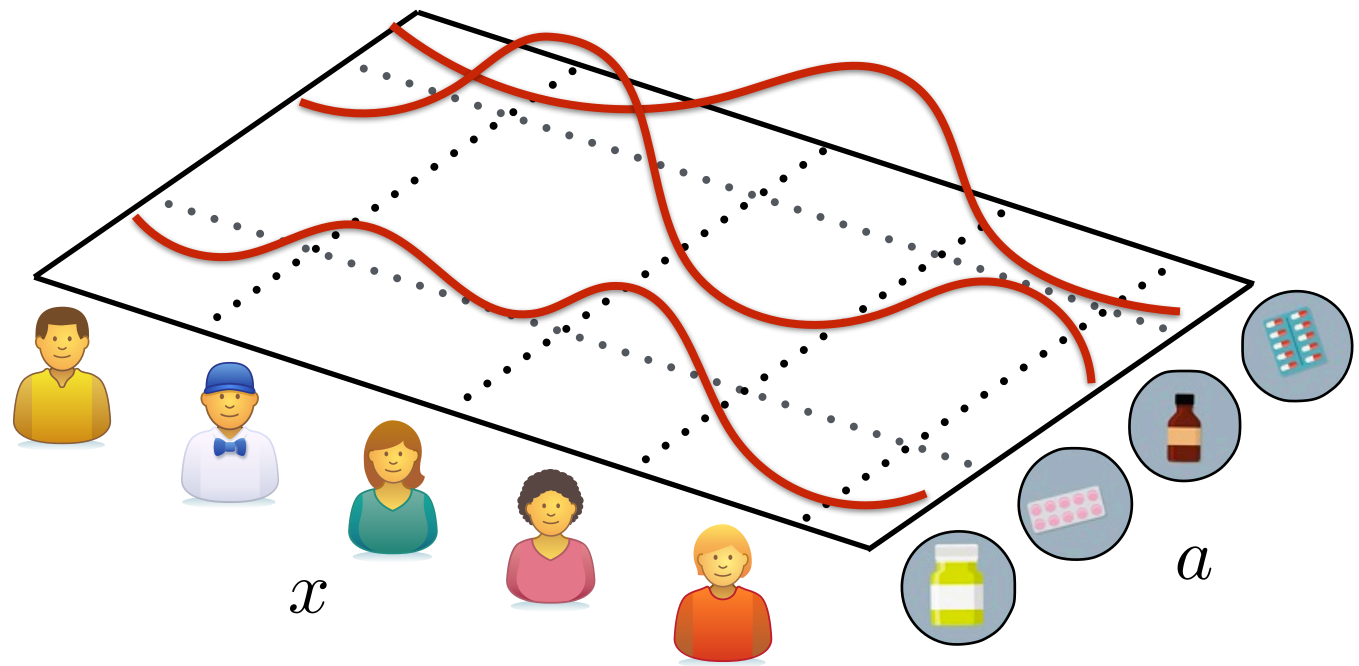
Exploration + Generalization: Contextual Bandits



Contextual bandits: Challenges

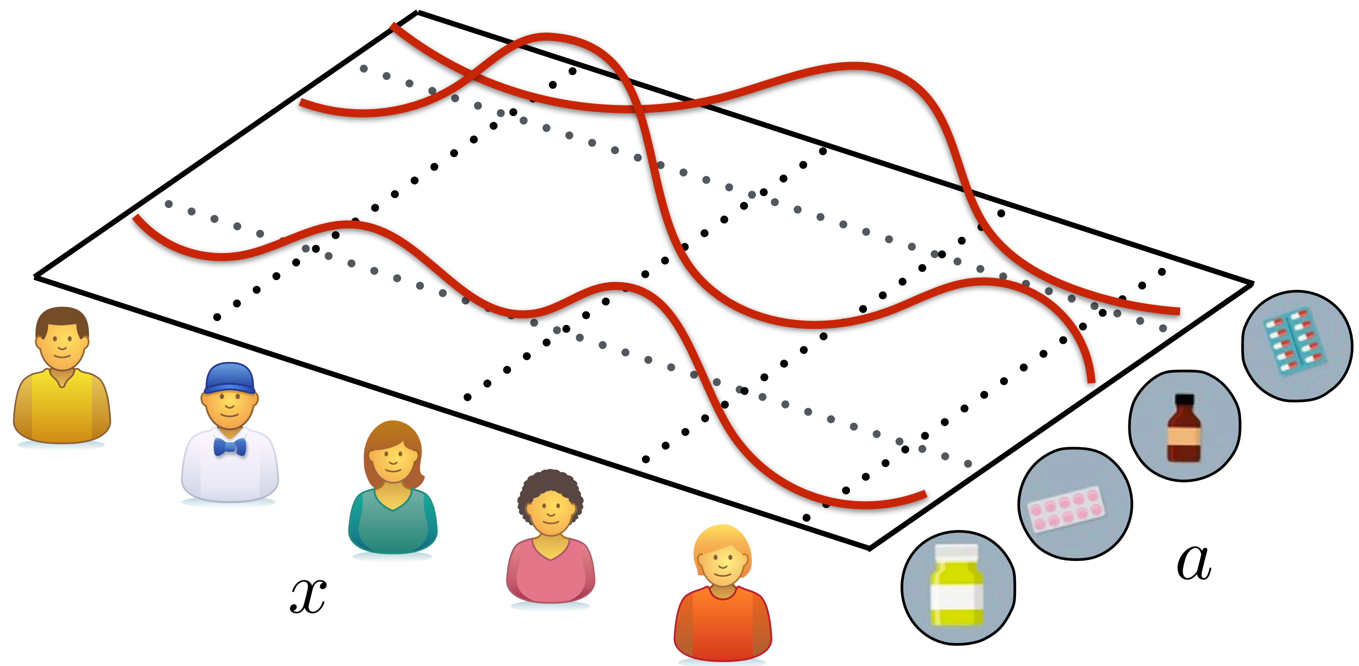


Contextual bandits: Challenges



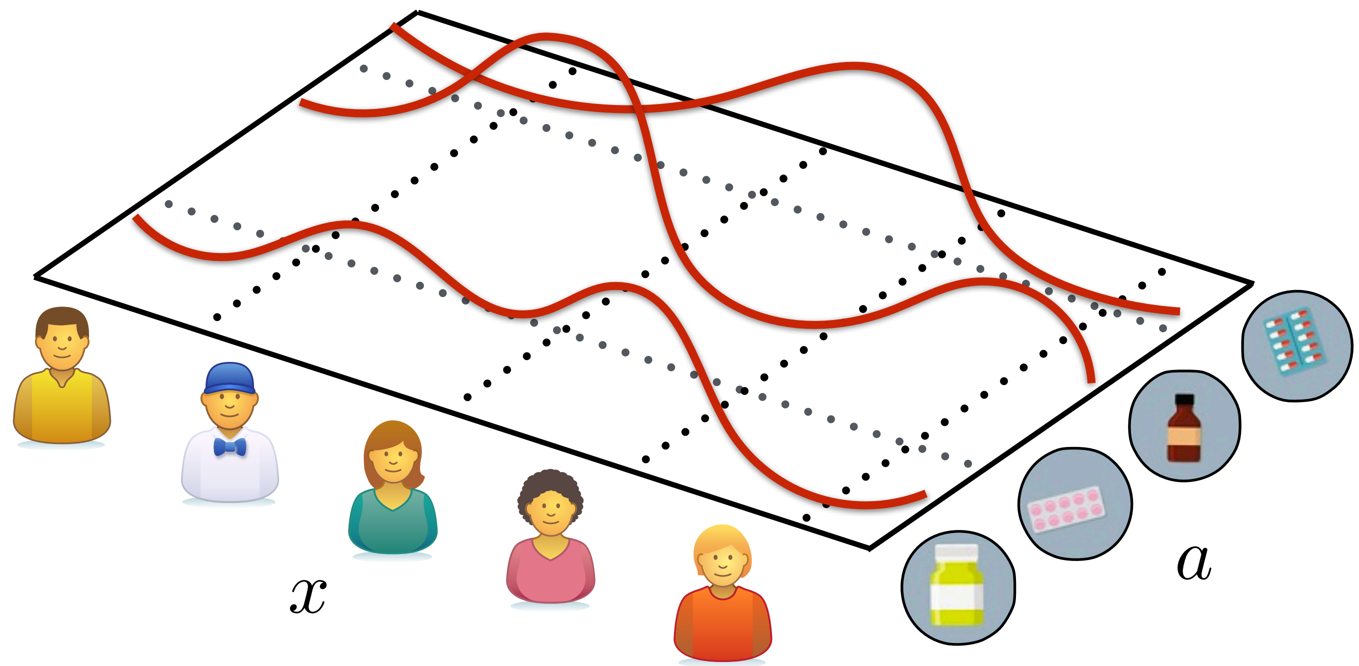
- **Exploration:** Bandit feedback; data collection introduces bias.

Contextual bandits: Challenges



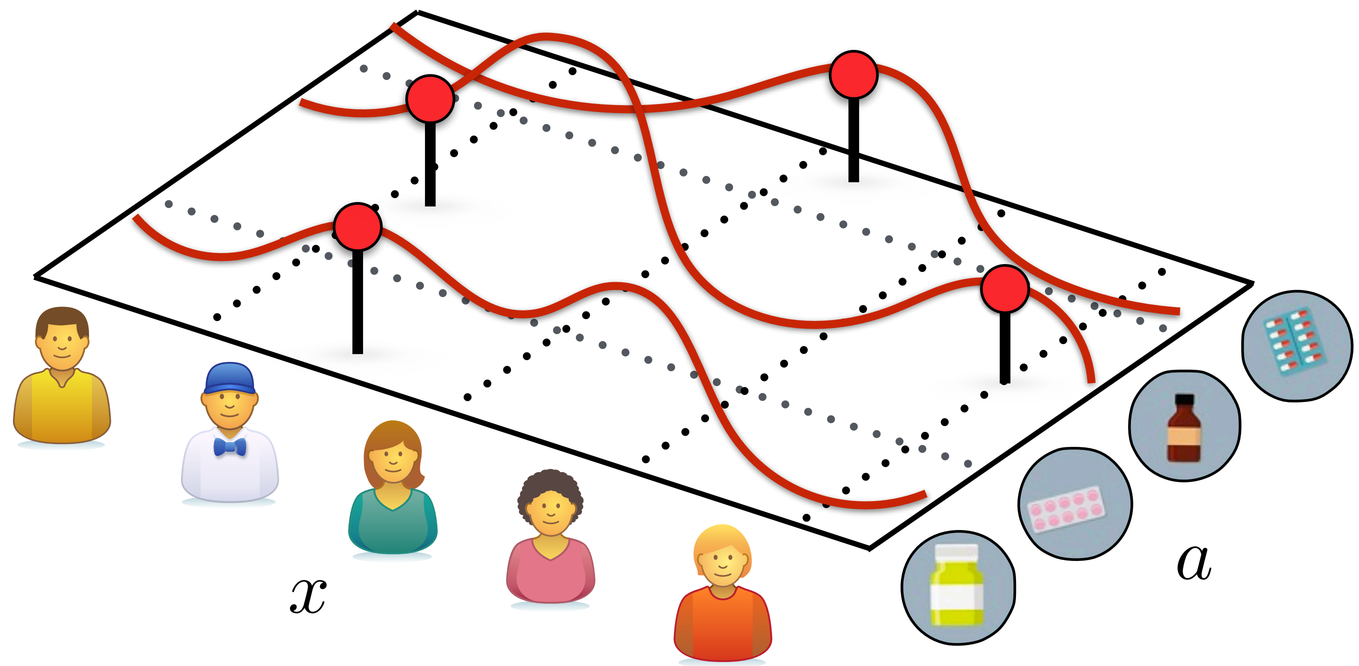
- **Exploration:** Bandit feedback; data collection introduces bias.

Contextual bandits: Challenges



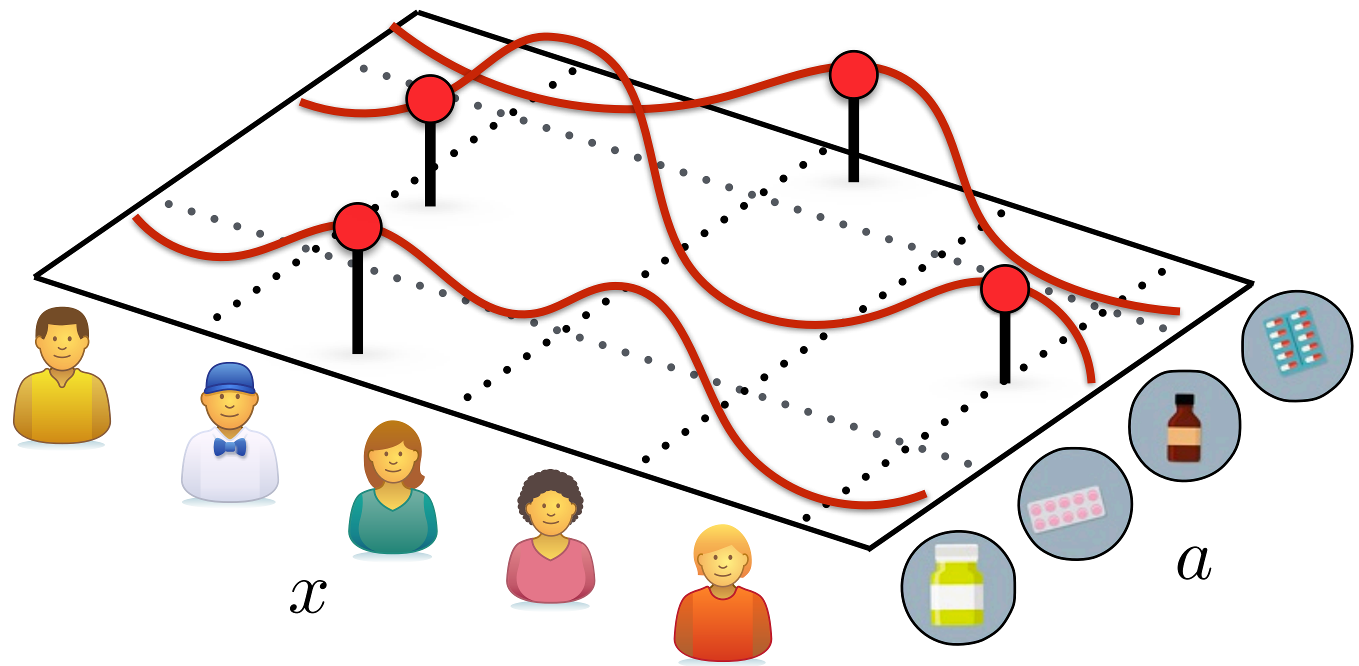
- **Exploration:** Bandit feedback; data collection introduces bias.

Contextual bandits: Challenges



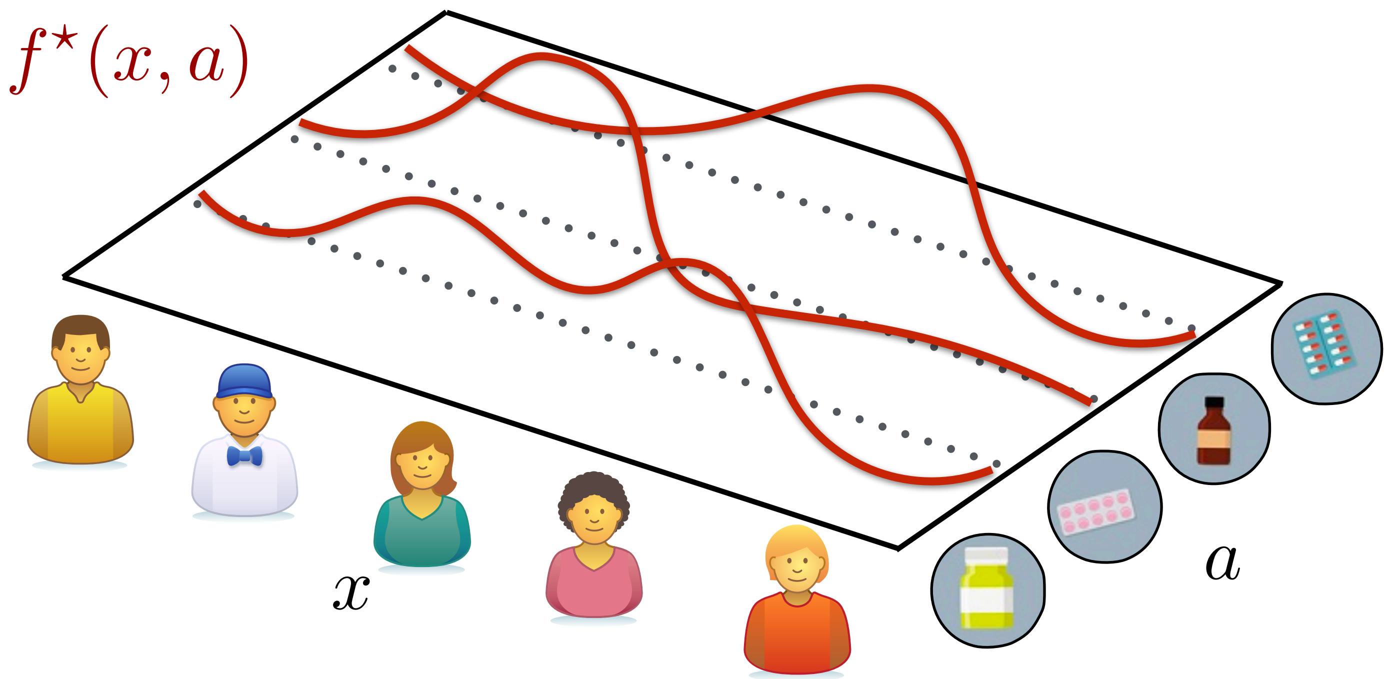
- **Exploration:** Bandit feedback; data collection introduces bias.

Contextual bandits: Challenges



- **Exploration:** Bandit feedback; data collection introduces bias.
- **Generalization:** May not see same context $x^{(t)}$ twice.
  - Can't afford to solve separate bandit problem for each $x^{(t)}$.
  - Need to generalize/extrapolate across contexts.
- How to propagate information across contexts?

Exploration + Generalization: Contextual Bandits

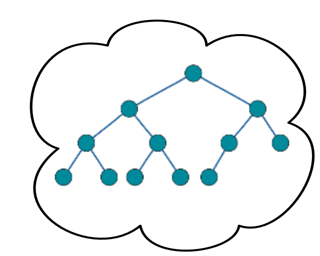


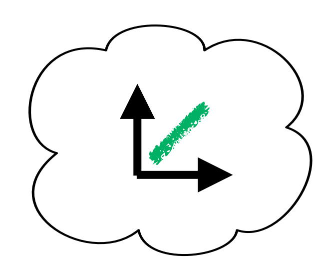
Assumption: Realizability

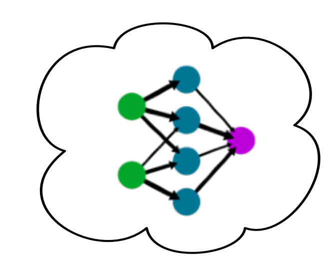
Given hypothesis class $\mathcal{F}$ such that

$$\mathbb{E}[r \mid x, a] = f^\star(x,a)$$

for unknown $f^\star \in \mathcal{F}$. (e.g., $r = f(x,a) + \varepsilon$)

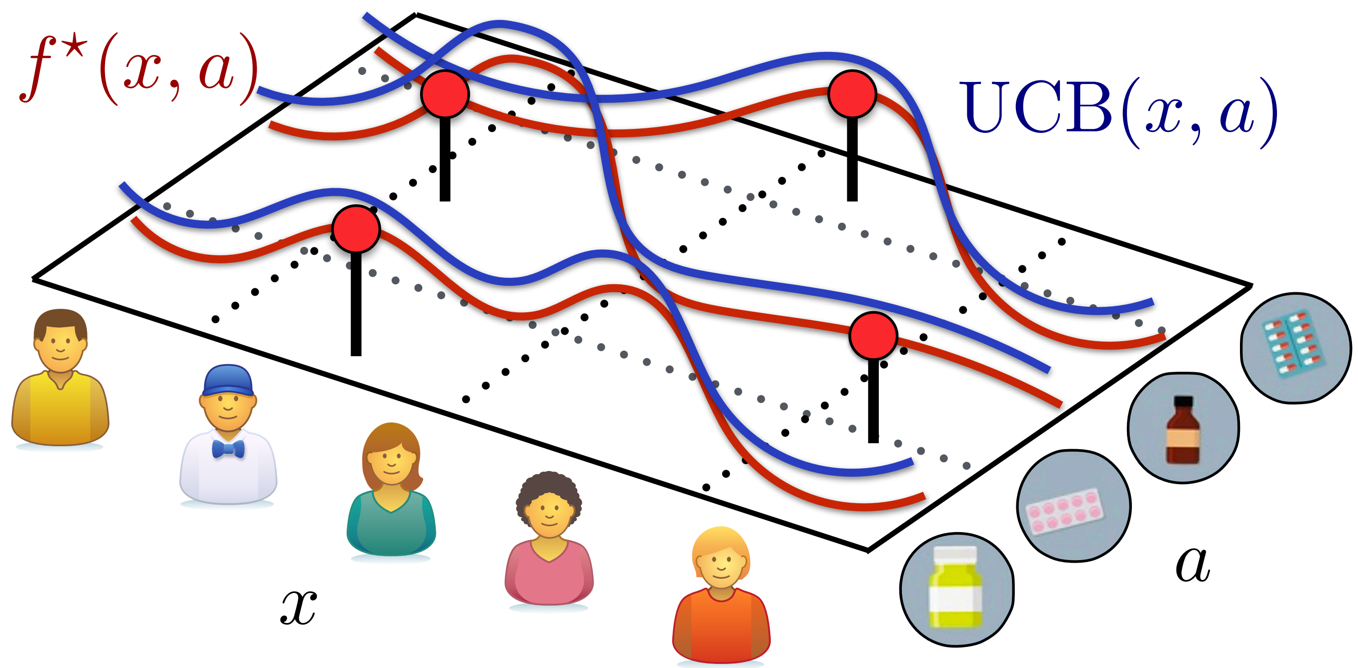




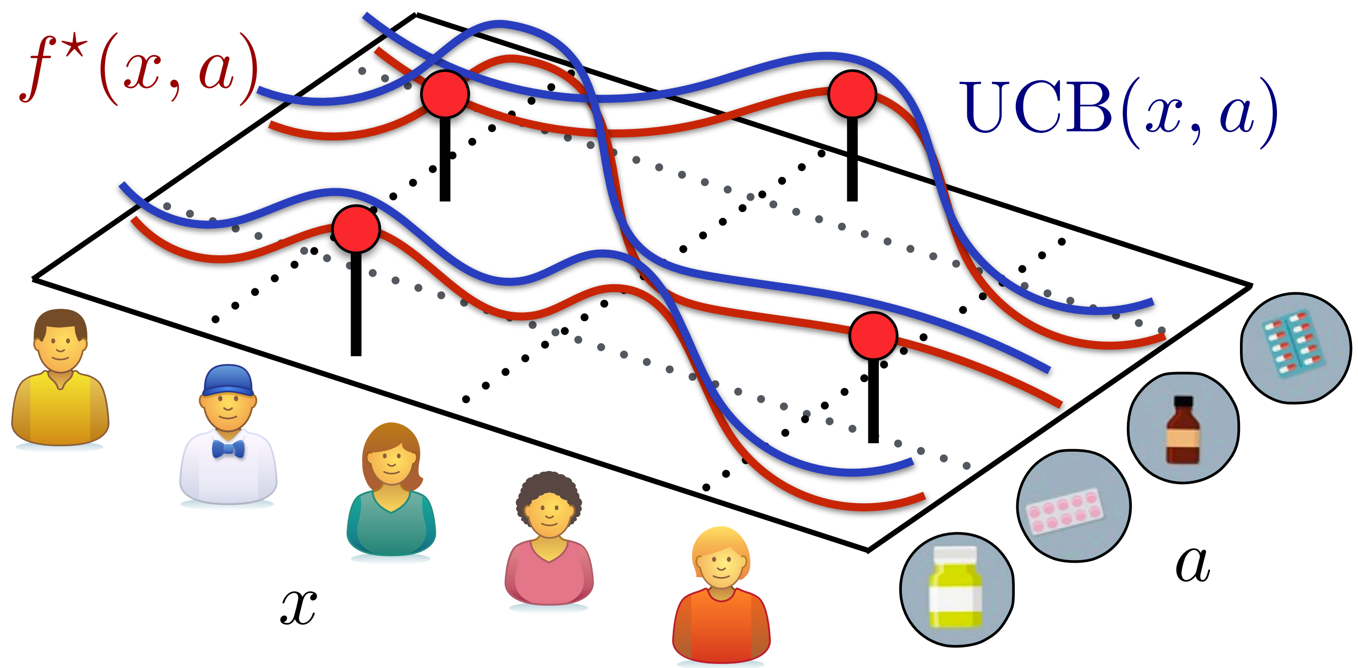


Class $\mathcal{F}$ might consist of linear models, deep neural networks, forests, kernels, ...

Contextual bandits: Upper confidence bound

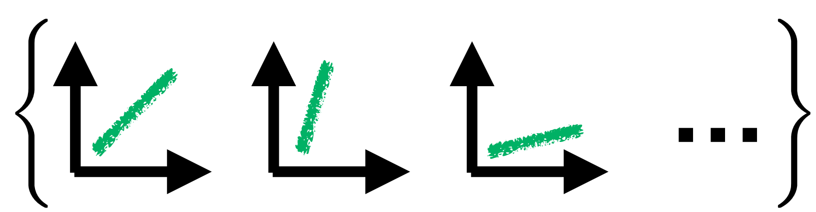


Contextual bandits: Upper confidence bound

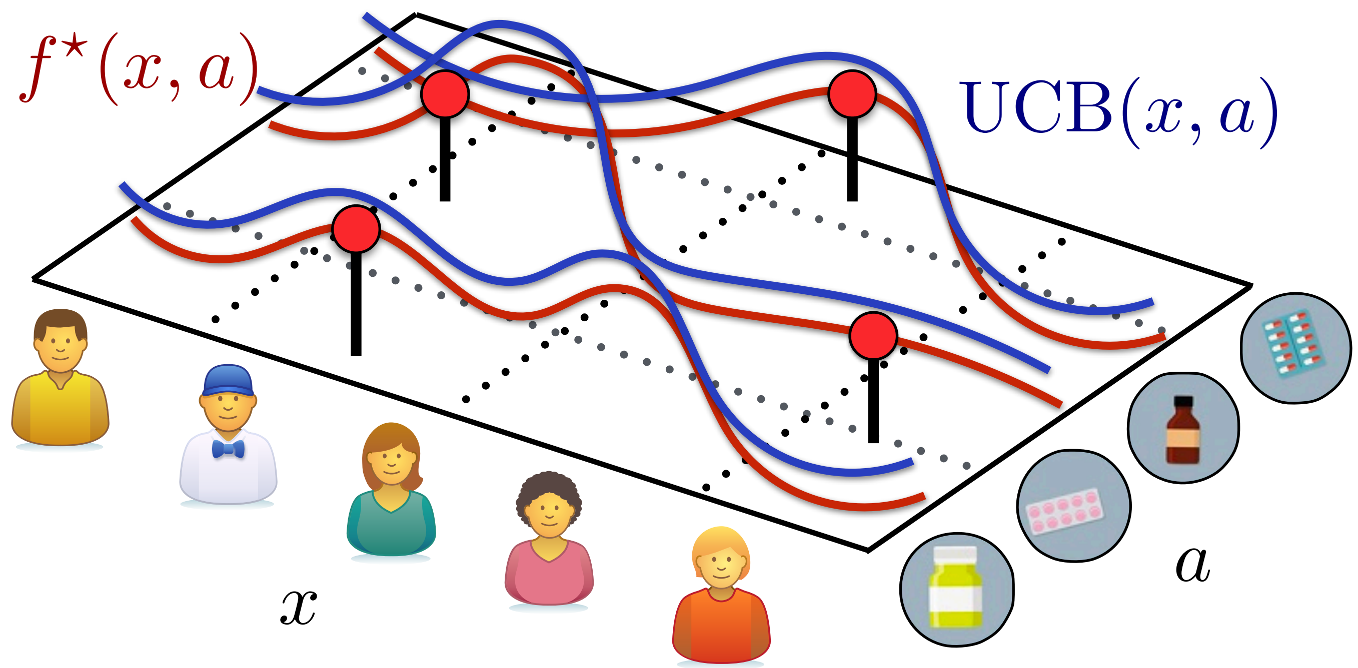


Example: LinUCB [Auer ’02, Chu et al. ’10, Abbasi-Yadkori et al. ’11]

Linear models w/ $f^\star(x,a) = \langle \theta^\star, \phi(x,a) \rangle$, where $\phi(x,a) \in \mathbb{R}^d$: $\mathbf{Reg}(T) \leq d\sqrt{T}$.

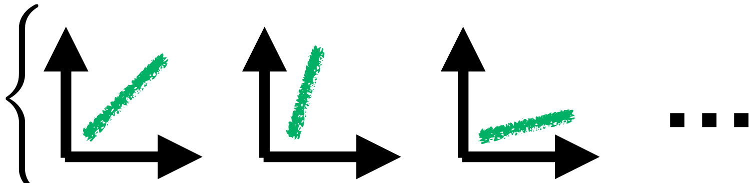
$\mathcal{F} =$ 

Contextual bandits: Upper confidence bound



Example: LinUCB [Auer ’02, Chu et al. ’10, Abbasi-Yadkori et al. ’11]

Linear models w/ $f^\star(x,a) = \langle \theta^\star, \phi(x,a) \rangle$, where $\phi(x,a) \in \mathbb{R}^d$: $\mathbf{Reg}(T) \leq d\sqrt{T}$.

$\mathcal{F} = \{$  $\}$

In general, no hope of constructing valid/shrinking confidence intervals for all (x,a).

- Good cases: Linear models, nonparametric models.
- Bad cases: Sparse linear, single ReLU [LK**F**S’21], neural networks, ...

Idea: Reduce contextual bandits to supervised learning.

$\Longrightarrow$ Leverage existing algorithms and generalization bounds

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Assign probability p_a to each action based on $\widehat{f}^{(t)}(x^{(t)}, a)$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Assign probability p_a to each action based on $\widehat{f}^{(t)}(x^{(t)}, a)$.
- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW):
- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.
- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \dots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.

$$p_a = \frac{1}{|\mathcal{A}| \;+\; \gamma \;\times\; (\widehat{f}^{(t)}(x^{(t)}, b) - \widehat{f}^{(t)}(x^{(t)}, a))} \quad \forall a \neq b$$

- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.

$$p_a = \frac{1}{|\mathcal{A}| + \gamma \times \underbrace{(\widehat{f}^{(t)}(x^{(t)}, b) - \widehat{f}^{(t)}(x^{(t)}, a))}_{\text{reward gap between } b \text{ and } a}} \quad \forall a \neq b$$

- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.

$$p_a = \frac{1}{|\mathcal{A}| + \underbrace{\gamma}_{\text{learning rate}} \times \underbrace{(\widehat{f}^{(t)}(x^{(t)}, b) - \widehat{f}^{(t)}(x^{(t)}, a))}_{\text{reward gap between } b \text{ and } a}} \quad \forall a \neq b$$

- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.

$$p_a = \frac{1}{\underbrace{|\mathcal{A}|}_{\text{\# actions}} + \underbrace{\gamma}_{\text{learning rate}} \times \underbrace{(\widehat{f}^{(t)}(x^{(t)}, b) - \widehat{f}^{(t)}(x^{(t)}, a))}_{\text{reward gap between } b \text{ and } a}} \quad \forall a \neq b$$

- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB [**F** and Rakhlin'20]

For $t = 1, \ldots, T$:

- Receive context $x^{(t)}$.
- Get reward estimate $\widehat{f}^{(t)}(x, a)$ from learning algorithm.
- Inverse Gap Weighting (IGW): Let $b = \arg\max_a \widehat{f}^{(t)}(x^{(t)}, a)$.

$$p_a = \frac{1}{\underbrace{|\mathcal{A}|}_{\text{\# actions}} + \underbrace{\gamma}_{\text{learning rate}} \times \underbrace{(\widehat{f}^{(t)}(x^{(t)}, b) - \widehat{f}^{(t)}(x^{(t)}, a))}_{\text{reward gap between } b \text{ and } a}} \quad \forall a \neq b$$

  with $p_b =$ remaining probability.

- Sample $a^{(t)} \sim p$, update learning algorithm w/ $(x^{(t)}, a^{(t)}, r^{(t)}(a^{(t)}))$.

Contextual bandits: The SquareCB algorithm

SquareCB algorithm: [**F** & Rakhlin '20]

Optimally solve regression $\Longrightarrow$ Optimally solve contextual bandits

- Can form estimates $\widehat{f}^{(t)}$ using online regression.
- Theorem: SquareCB attains optimal rate for any $\mathcal{F}$.

Contextual bandits: The SquareCB algorithm

SquareCB algorithm: [**F** & Rakhlin '20]

Optimally solve regression $\implies$ Optimally solve contextual bandits

- Can form estimates $\widehat{f}^{(t)}$ using online regression.
- Theorem: SquareCB attains optimal rate for any $\mathcal{F}$.

Regret bound: With appropriate learning rate $\gamma > 0$, SquareCB has

$$\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \mathbf{Est}_{\mathsf{Sq}}(T)}, \quad \text{w/} \quad \mathbf{Est}_{\mathsf{Sq}}(T) := \sum_{t=1}^{T} \left(\widehat{f}^{(t)}(x^{(t)}, a^{(t)}) - f^{\star}(x^{(t)}, a^{(t)})\right)^2.$$

Contextual bandits: The SquareCB algorithm

SquareCB algorithm: [**F** & Rakhlin '20]

Optimally solve regression $\implies$ Optimally solve contextual bandits

- Can form estimates $\widehat{f}^{(t)}$ using online regression.
- Theorem: SquareCB attains optimal rate for any $\mathcal{F}$.

Regret bound: With appropriate learning rate $\gamma > 0$, SquareCB has

$$\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \mathbf{Est}_{\mathsf{Sq}}(T)}, \quad \text{w/} \quad \mathbf{Est}_{\mathsf{Sq}}(T) := \textstyle\sum_{t=1}^{T} \big(\widehat{f}^{(t)}(x^{(t)}, a^{(t)}) - f^{\star}(x^{(t)}, a^{(t)})\big)^2.$$

Examples:

- $\mathbf{Est}_{\mathsf{Sq}}(T) \leq \log|\mathcal{F}|$ for finite $\mathcal{F}$ $\implies$ $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \log|\mathcal{F}|}$.
- $\mathbf{Est}_{\mathsf{Sq}} \leq \widetilde{O}(d)$ for linear models $\implies$ $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot d}$.

Contextual bandits: The SquareCB algorithm

SquareCB algorithm: [**F** & Rakhlin '20]

Optimally solve regression $\Longrightarrow$ Optimally solve contextual bandits

- Can form estimates $\widehat{f}^{(t)}$ using online regression.
- Theorem: SquareCB attains optimal rate for any $\mathcal{F}$.

Regret bound: With appropriate learning rate $\gamma > 0$, SquareCB has

$$\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \mathbf{Est}_{\mathsf{Sq}}(T)}, \quad \text{w/} \quad \mathbf{Est}_{\mathsf{Sq}}(T) := \sum_{t=1}^{T} \left(\widehat{f}^{(t)}(x^{(t)}, a^{(t)}) - f^{\star}(x^{(t)}, a^{(t)})\right)^2.$$

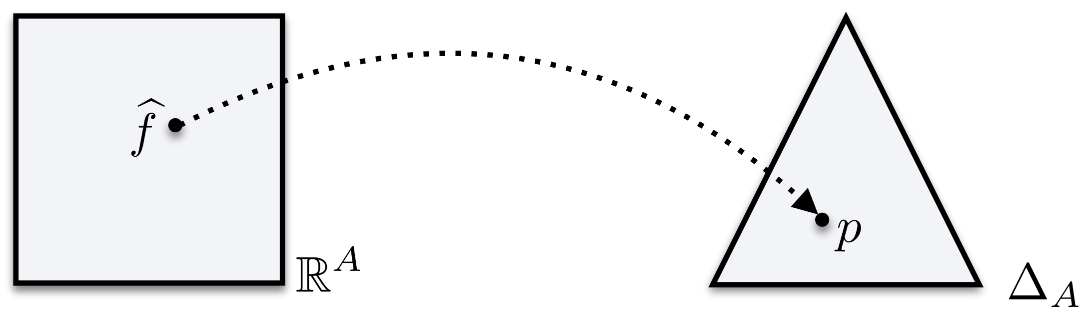
Examples:

- $\mathbf{Est}_{\mathsf{Sq}}(T) \leq \log|\mathcal{F}|$ for finite $\mathcal{F}$ $\Longrightarrow$ $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \log|\mathcal{F}|}$.
- $\mathbf{Est}_{\mathsf{Sq}} \leq \widetilde{O}(d)$ for linear models $\Longrightarrow$ $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot d}$.

In general: $\mathbf{Reg}(T) \leq \sqrt{|\mathcal{A}|T \cdot \mathrm{comp}(\mathcal{F})}$.

(no explicit $|\mathcal{X}|$ dependence!)

Contextual bandits: The SquareCB algorithm

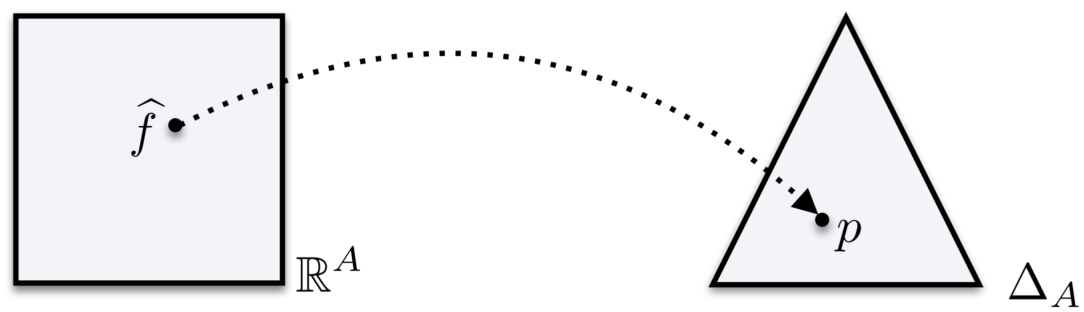


SquareCB solves: For all rounds t, with learning rate γ:

$$\underset{\text{action dist. } p}{\arg\min} \; \underset{\text{reward fn. } f^\star}{\max} \left\{ \mathbb{E}\left[\textsf{CB-Regret}^{(t)}\right] - \gamma \cdot \mathbb{E}\left[\textsf{Est-Error}^{(t)}\right] \right\}.$$

Agnostic to structure of $\mathcal{F}$!

Contextual bandits: The SquareCB algorithm



SquareCB solves: For all rounds t, with learning rate γ:

$$\underset{\text{action dist. } p}{\arg\min} \; \underset{\text{reward fn. } f^\star}{\max} \left\{ \mathbb{E}\left[\text{CB-Regret}^{(t)}\right] - \gamma \cdot \mathbb{E}\left[\text{Est-Error}^{(t)}\right] \right\}.$$

Agnostic to structure of $\mathcal{F}$!

Contextual bandit history:

- Classification reductions: [Langford & Zhang'07, Dudik et al.'11, Agarwal et al.'14]
- Specific models: [Abe & Long'99], [Rigollet & Zeevi'10], [Krause & Ong '11], [Filippi, Cappe, Garivier, Szepesvari '11], [Chu, Li, Reyzin, Schapire'11], [Perchet & Rigollet'13], [Russo & Van Roy '13, '14, '16], [Goldenshluger & Zeevi'13], [Bastani & Bayati '15], [Osband et al. '16], [Sen et al. '17], [GTKM '17], [Jun et al. '17], . . .
- Regression: [**F** & Rakhlin '20], [Simchi-Levi & Xu'20], [**F**RSX'20], [**F**KRQ '21] $\leftarrow$ **RL**

Roadmap

Basic challenges and solutions ✔

- Credit assignment
- Exploration
- Generalization

Intermediate level

- Exploration + credit assignment: Tabular RL ✔
- Exploration + generalization: Contextual bandits ✔
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

Roadmap

Basic challenges and solutions ✔

- Credit assignment
- Exploration
- Generalization

Intermediate level

- Exploration + credit assignment: Tabular RL ✔
- Exploration + generalization: Contextual bandits ✔
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

Credit Assignment + Generalization: Policy Gradient

RL as stochastic optimization

- Parameterize policies via $\theta \mapsto \pi_\theta$, $\theta \in \mathbb{R}^d$.
- Optimization goal: $\max_\theta J(\pi_\theta) = \max_\theta \mathbb{E}^{\pi_\theta}[\sum_{h=1}^{H} r_h]$.

Credit Assignment + Generalization: Policy Gradient

RL as stochastic optimization

- Parameterize policies via $\theta \mapsto \pi_\theta$, $\theta \in \mathbb{R}^d$.
- Optimization goal: $\max_\theta J(\pi_\theta) = \max_\theta \mathbb{E}^{\pi_\theta}[\sum_{h=1}^{H} r_h]$.

Key idea: stochastic policies $\pi_\theta : \mathcal{X} \to \Delta(\mathcal{A})$.

- Typically, $\pi_\theta(a \mid x) \propto \exp(f_\theta(x,a))$.
- Ex: $f_\theta(x,a) = \langle \theta, \phi(x,a) \rangle$ (linear), $f_\theta(x,a) = \mathsf{DNN}(x,a\,;\theta)$ (Deep RL).

Policy gradient methods

- Optimization goal: $\max_\theta J(\pi_\theta)$.
- Gradient ascent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \cdot \nabla_\theta J(\pi_{\theta^{(t)}}).$$

Policy gradient methods

- Optimization goal: $\max_\theta J(\pi_\theta)$.
- Gradient ascent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \cdot \nabla_\theta J(\pi_{\theta^{(t)}}).$$

- **Policy gradient theorem** [Williams '92, Sutton et al. '99]:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}^{\pi_\theta}\left[\left(\sum_{h=1}^{H} r_h\right) \cdot \sum_{h=1}^{H} \nabla_\theta \log \pi_\theta(a_h \mid x_h)\right] \tag{1}$$

Policy gradient methods

- Optimization goal: $\max_\theta J(\pi_\theta)$.
- Gradient ascent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \cdot \nabla_\theta J(\pi_{\theta^{(t)}}).$$

- **Policy gradient theorem** [Williams '92, Sutton et al. '99]:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}^{\pi_\theta}\left[\left(\sum_{h=1}^{H} r_h\right) \cdot \sum_{h=1}^{H} \nabla_\theta \log \pi_\theta(a_h \mid x_h)\right] \tag{1}$$

- REINFORCE [Williams '92]: Approximate (1) w/ trajectories sampled from π_θ.

Policy gradient methods

- Optimization goal: $\max_\theta J(\pi_\theta)$.
- Gradient ascent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \cdot \nabla_\theta J(\pi_{\theta^{(t)}}).$$

- **Policy gradient theorem** [Williams '92, Sutton et al. '99]:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}^{\pi_\theta}\left[\left(\sum_{h=1}^{H} r_h\right) \cdot \sum_{h=1}^{H} \nabla_\theta \log \pi_\theta(a_h \mid x_h)\right] \tag{1}$$

- REINFORCE [Williams '92]: Approximate (1) w/ trajectories sampled from π_θ.

Log Derivative Trick

$$\nabla_\theta g(\theta) = g(\theta) \cdot \nabla_\theta \log g(\theta)$$

Policy gradient theory

Representative result [Agarwal et al. '19]:

Tabular setting, $\pi_\theta(a \mid x) = \theta_{x,a}$.

Policy gradient theory

Representative result [Agarwal et al. '19]:

Tabular setting, $\pi_\theta(a \mid x) = \theta_{x,a}$.

$$J(\pi^\star) - J(\pi_{\theta^{(t)}}) \leq C_{\mathrm{mismatch}}(\theta^{(t)}) \cdot \left\| \nabla_\theta J(\pi_{\theta^{(t)}}) \right\|,$$

where

$$C_{\mathrm{mismatch}}(\theta) := \max_{x,a,h} \frac{\mathbb{P}^{\pi_\theta}(x_h = x, a_h = a)}{\mathbb{P}^{\pi^\star}(x_h = x, a_h = a)}.$$

Policy gradient theory

Representative result [Agarwal et al. '19]:

Tabular setting, $\pi_\theta(a \mid x) = \theta_{x,a}$.

$$J(\pi^\star) - J(\pi_{\theta^{(t)}}) \leq C_{\mathrm{mismatch}}(\theta^{(t)}) \cdot \left\| \nabla_\theta J(\pi_{\theta^{(t)}}) \right\|,$$

where

$$C_{\mathrm{mismatch}}(\theta) := \max_{x,a,h} \frac{\mathbb{P}^{\pi_\theta}(x_h = x, a_h = a)}{\mathbb{P}^{\pi^\star}(x_h = x, a_h = a)}.$$

General function approximation: For appropriate policy gradient variant,

$$J(\pi^\star) - J(\pi_{\theta^{(t)}}) \lesssim C_{\mathrm{mismatch}} \cdot \underbrace{\varepsilon_{\mathrm{opt}}}_{\text{opt/stat error (generalization)}} + \underbrace{\varepsilon_{\mathrm{bias}}}_{\text{quality of function approx.}}.$$

Ideally, $\varepsilon_{\mathrm{opt}} \propto \mathrm{comp}(\mathcal{F})$ (no explicit $|\mathcal{X}|$ dependence).

Policy gradient: History

- **Basic principles:** REINFORCE [Williams '92], function approximation [Sutton et al. '99], actor-critic [Konda & Tsitsiklis '00], natural policy gradient [Kakade '01]
- **Empirical improvements (deep RL)**:
  Trust regions (TRPO, PPO) [Schulman et al. '15, Schulman et al. '17], Regularization (e.g., SAC) [Haarnoja et al. '18], . . .
- **Asymptotic convergence:** [Bellman & Dreyfus '51, Sutton et al. '99]
- **Non-asymptotic guarantees:** [Kakade & Langford '02], [Scherrer & Geist '14], [Fazel et al. '18], [Agarwal et al. '19], . . .

Roadmap

Basic challenges and solutions ✔

- Credit assignment
- Exploration
- Generalization

Intermediate level ✔

- Exploration + credit assignment: Tabular RL
- Exploration + generalization: Contextual bandits
- Generalization + credit assignment: Policy gradient

The frontier: Exploration + generalization + credit assignment

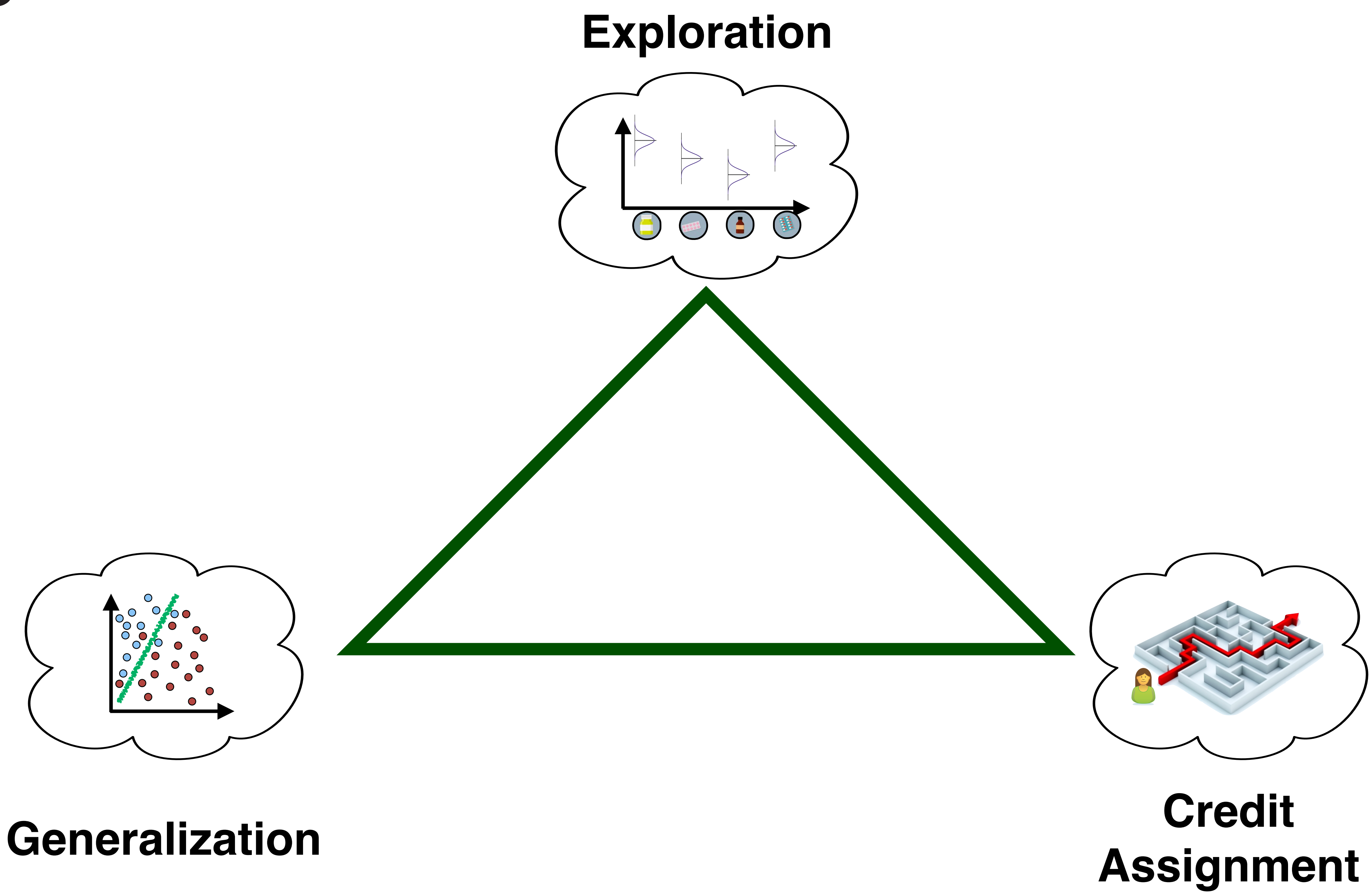
Foundations of Reinforcement Learning

Learning and Games Bootcamp @ Simons Institute

Dylan Foster

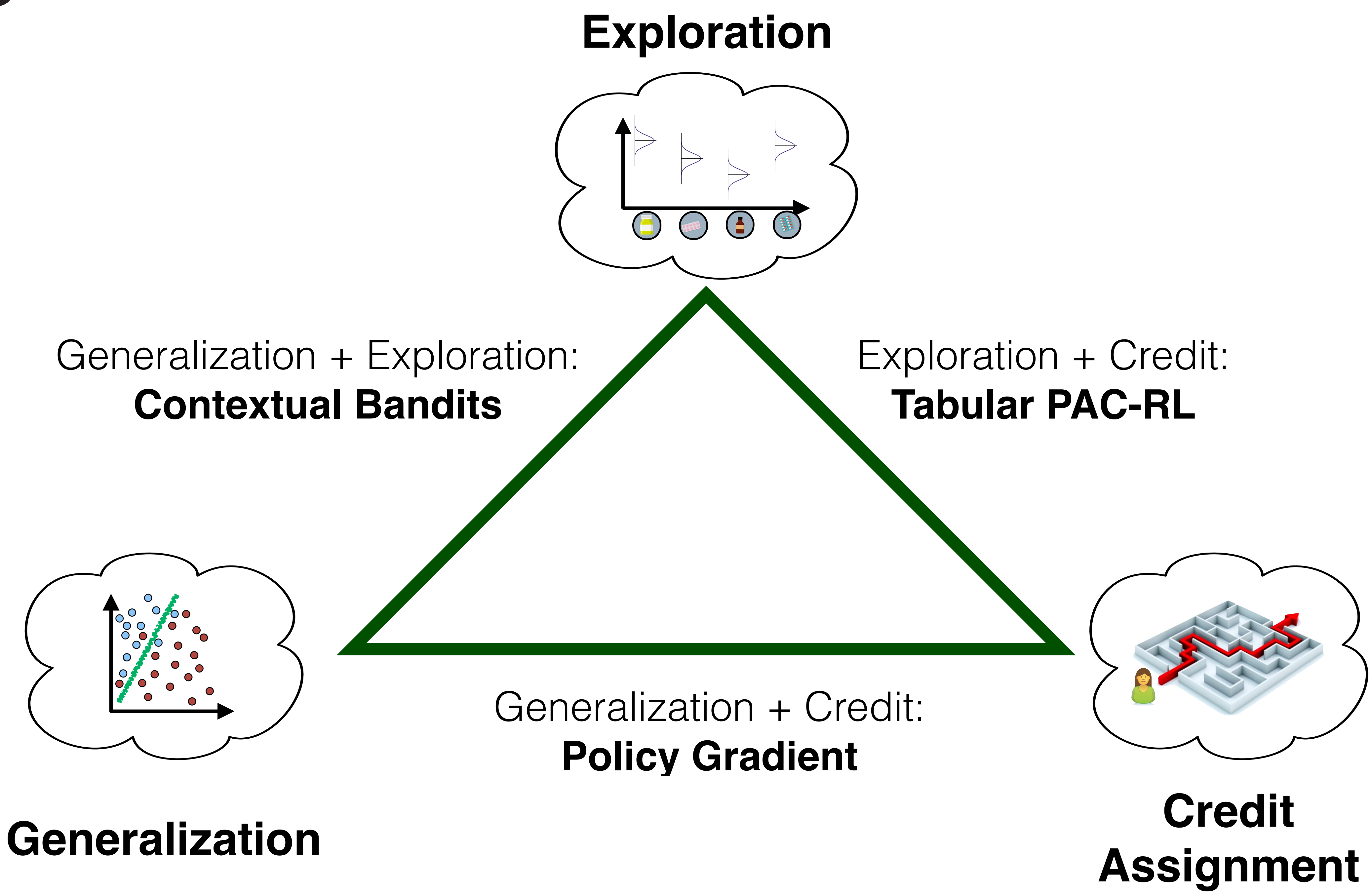
Microsoft Research, New England

Our goal



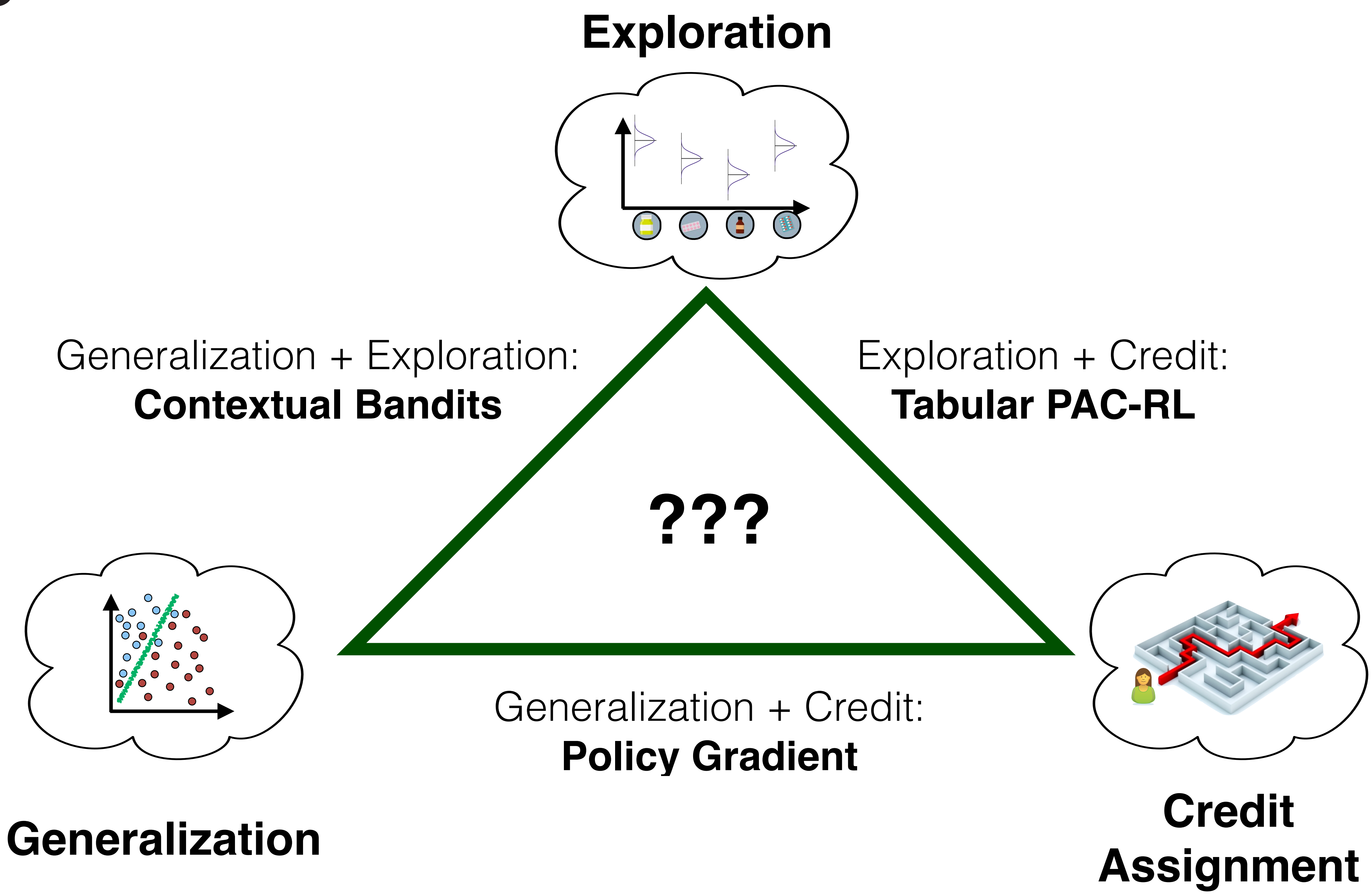
[Credit: John Langford]

Our goal



[Credit: John Langford]

Our goal



Goal: Exploration + credit assignment + generalization:

- Explore unknown systems with long horizon (credit assignment)

  ...while generalizing: No dependence on $|\mathcal{X}|$ (ideally not $|\mathcal{A}|$ either).

[Credit: John Langford]

RL: The need for modeling and generalization

Challenge: States/observations are typically rich/complex/high-dimensional.

- Ex: robotics: x_h = camera image, $\mathcal{X}$ = all possible images

  $\implies |\mathcal{X}|$ = intractably large

Approach: Use hypothesis class $\mathcal{F}$ to model:

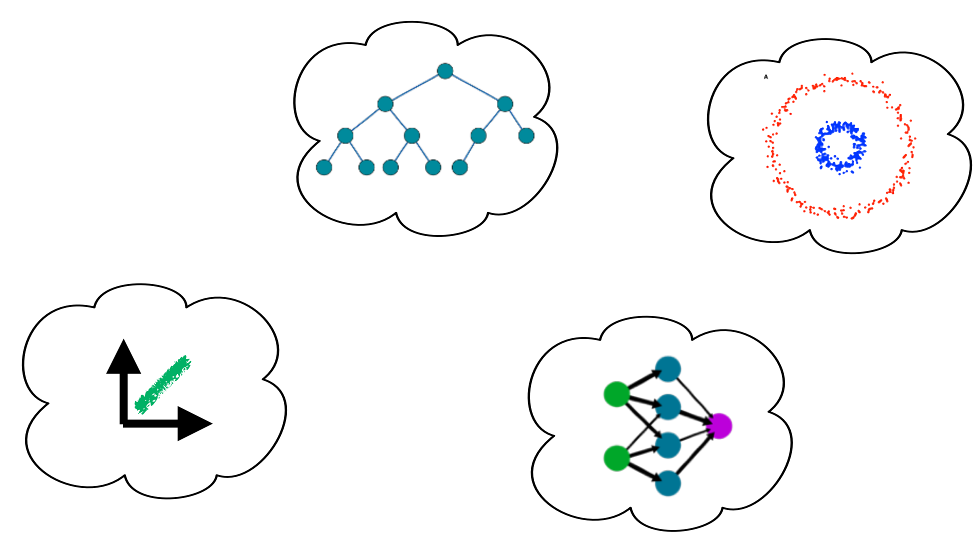
- Rewards/responses/treatment effects
- Dynamics
- Long-term rewards

  $\vdots$

In general, model class $\mathcal{F}$ might consist of:

- Deep neural networks
- Generalized linear models
- Kernels

  $\vdots$



RL: Modeling approaches

State space $\mathcal{X}$ is intractably large. Use hypothesis class $\mathcal{F}$ to restrict soln. space.

RL: Modeling approaches

State space $\mathcal{X}$ is intractably large. Use hypothesis class $\mathcal{F}$ to restrict soln. space.

Policy-based methods: $\mathcal{F} = \textbf{policies}$

- Use restricted policy class $\Pi \subset \{\mathcal{X} \to \mathcal{A}\}$.
  - Ex: Policy gradient with $\theta \mapsto \pi_\theta$ parameterized by neural net.

RL: Modeling approaches

State space $\mathcal{X}$ is intractably large. Use hypothesis class $\mathcal{F}$ to restrict soln. space.

Policy-based methods: $\mathcal{F} =$ **policies**

- Use restricted policy class $\Pi \subset \{\mathcal{X} \to \mathcal{A}\}$.
  - Ex: Policy gradient with $\theta \mapsto \pi_\theta$ parameterized by neural net.

Value-based methods: $\mathcal{F} =$ **value functions**

- Model state-action value functions with value fn. class $\mathcal{Q} \subset \{\mathcal{X} \times \mathcal{A} \to \mathbb{R}\}$.

$$Q_h^\pi(x, a) := \mathbb{E}^\pi\left[\sum_{h' \geq h}^{H} r_{h'} \mid x_h = x, a_h = a\right].$$

- Can use $\mathcal{Q}$ to model Q^π for all π, or just for optimal policy $\pi^\star$.

RL: Modeling approaches

State space $\mathcal{X}$ is intractably large. Use hypothesis class $\mathcal{F}$ to restrict soln. space.

Policy-based methods: $\mathcal{F} = \textbf{policies}$

- Use restricted policy class $\Pi \subset \{\mathcal{X} \to \mathcal{A}\}$.
  - Ex: Policy gradient with $\theta \mapsto \pi_\theta$ parameterized by neural net.

Value-based methods: $\mathcal{F} = \textbf{value functions}$

- Model state-action value functions with value fn. class $\mathcal{Q} \subset \{\mathcal{X} \times \mathcal{A} \to \mathbb{R}\}$.

$$Q_h^\pi(x, a) := \mathbb{E}^\pi\left[\sum_{h' \geq h}^{H} r_{h'} \mid x_h = x, a_h = a\right].$$

- Can use $\mathcal{Q}$ to model Q^π for all π, or just for optimal policy $\pi^\star$.

Model-based methods: $\mathcal{F} = \textbf{transition dynamics}$

- Model class $\mathcal{M}$; MDPs $M = (P, R) \in \mathcal{M}$ parameterize transition dynamics+rewards.

RL: Modeling approaches

State space $\mathcal{X}$ is intractably large. Use hypothesis class $\mathcal{F}$ to restrict soln. space.

Policy-based methods: $\mathcal{F} =$ **policies**

- Use restricted policy class $\Pi \subset \{\mathcal{X} \to \mathcal{A}\}$.
  - Ex: Policy gradient with $\theta \mapsto \pi_\theta$ parameterized by neural net.

Value-based methods: $\mathcal{F} =$ **value functions**

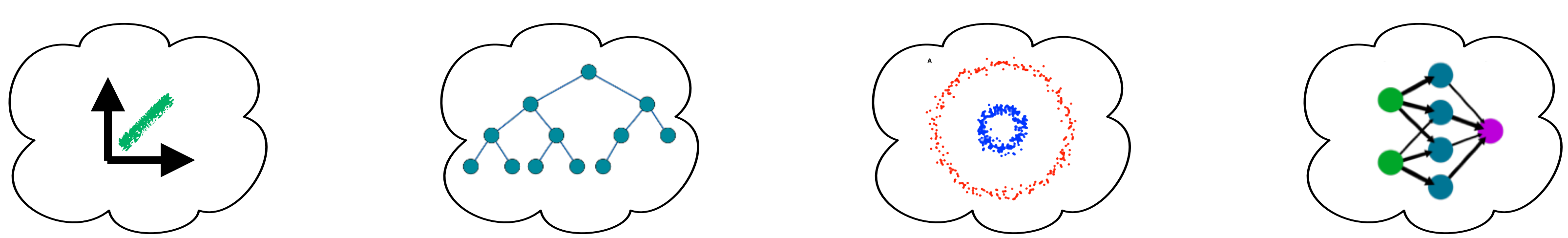
- Model state-action value functions with value fn. class $\mathcal{Q} \subset \{\mathcal{X} \times \mathcal{A} \to \mathbb{R}\}$.

$$Q_h^\pi(x, a) := \mathbb{E}^\pi \left[\sum_{h' \geq h}^{H} r_{h'} \mid x_h = x, a_h = a \right].$$

- Can use $\mathcal{Q}$ to model Q^π for all π, or just for optimal policy $\pi^\star$.

Model-based methods: $\mathcal{F} =$ **transition dynamics**

- Model class $\mathcal{M}$; MDPs $M = (P, R) \in \mathcal{M}$ parameterize transition dynamics+rewards.



RL: Formal setup

For $t = 1, \ldots, T$:

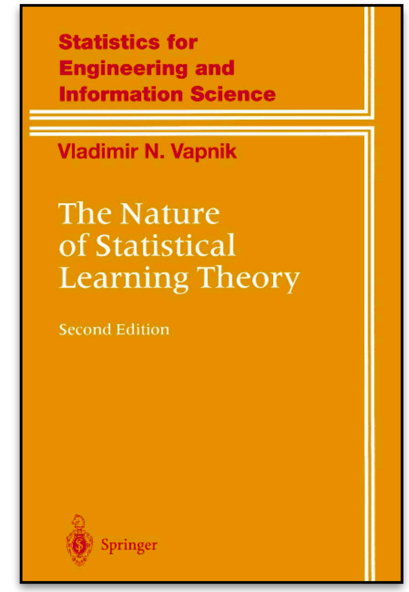
- $x_1^{(t)} \sim d_1$.
- For $h = 1, \ldots, H$: (Markov Decision Process (MDP))
  - Observe $x_h^{(t)} \in \mathcal{X}$. `(Sensor measurement)`
  - Take action $a_h^{(t)} \in \mathcal{A}$. `(Actuator signal)`
  - Observe reward $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$ w/ $r_h^{(t)} \in [0, 1]$. `(Reached goal?)`
  - Transition: $x_{h+1}^{(t)} \sim P(\cdot \mid x_h^{(t)}, a_h^{(t)})$. `(System evolves)`

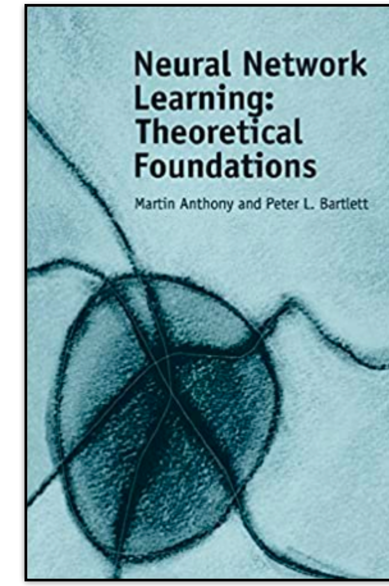
Goal: Given hypothesis class $\mathcal{F} \in \{\textsf{policies}, \textsf{value fns.}, \textsf{dynamics}\}$ + realizability:

Find $\widehat{\pi}$ with $J(\pi^\star) - J(\widehat{\pi}) \leq \varepsilon$ using $\mathrm{poly}(\mathrm{comp}(\mathcal{F}), H, \varepsilon^{-1})$ episodes,

or achieve, e.g., $\mathbf{Reg}(T) \leq \sqrt{\mathrm{poly}(\mathrm{comp}(\mathcal{F}), H) \cdot T}$.

Statistical learning: Complexity measures





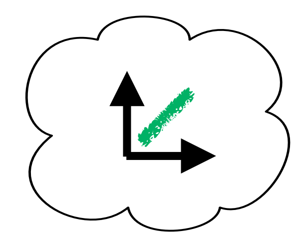
Complexity measures:

- VC Dimension (classification)
- Fat-shattering dimension (regression)
- Rademacher complexity (both)
- Covering numbers (both)

[e.g., Vapnik ’95, Anthony & Bartlett ’99, Bousquet-Boucheron-Lugosi ’03]

Examples:

- Finite class: $\mathrm{comp}(\mathcal{F}) \leq \log|\mathcal{F}|$
- Linear classification: $\mathrm{comp}(\mathcal{F}) \leq \textsf{dimension}$ (VC dim)
- Linear regression: $\mathrm{comp}(\mathcal{F}) \leq (\textsf{weight norm})^2$ (fat-shattering)
- Similar bounds for neural nets, kernels, ...



No explicit dependence on $|\mathcal{X}|$!

RL: Distribution shift

What we would like:

1. Gather data from distribution $\mathcal{D}$ using policy $\pi^{(t)}$.
2. Fit hypothesis $\widehat{f} \in \mathcal{F}$ (e.g., value fn., transition dynamics) using dataset (via supervised learning).
3. Update policy $\pi^{(t+1)}$ using $\widehat{f}$.
4. Performance improves?

RL: Distribution shift

What we would like:

1. Gather data from distribution $\mathcal{D}$ using policy $\pi^{(t)}$.
2. Fit hypothesis $\widehat{f} \in \mathcal{F}$ (e.g., value fn., transition dynamics) using dataset (via supervised learning).
3. Update policy $\pi^{(t+1)}$ using $\widehat{f}$.
4. Performance improves?

Why doesn't this work?

1. Statistical learning gives us

$$\mathsf{Error}_{\mathcal{D}}(\widehat{f}) \leq \sqrt{\frac{\mathrm{comp}(\mathcal{F})}{n}}.$$

2. No guarantee on performance on dataset $\mathcal{D}'$ induced by $\pi^{(t+1)}$.

$\implies$ **fail to improve performance or explore**.

RL: Distribution shift

Solution 1: Control # effective distributions

RL: Distribution shift

Solution 1: Control # effective distributions

- For general contextual bandits, SquareCB has

$$\mathbf{Reg}(T) \leq \sqrt{\underbrace{|\mathcal{A}|}_{\text{\# possible action distributions}} \cdot T \cdot \mathrm{comp}(\mathcal{F})}$$

- Idea: Can only be "suprised" $|\mathcal{A}|$ times if we explore deliberately.
- No assumption on $\mathcal{F}$, but requires strong assumption on $\mathcal{A}$.

RL: Distribution shift

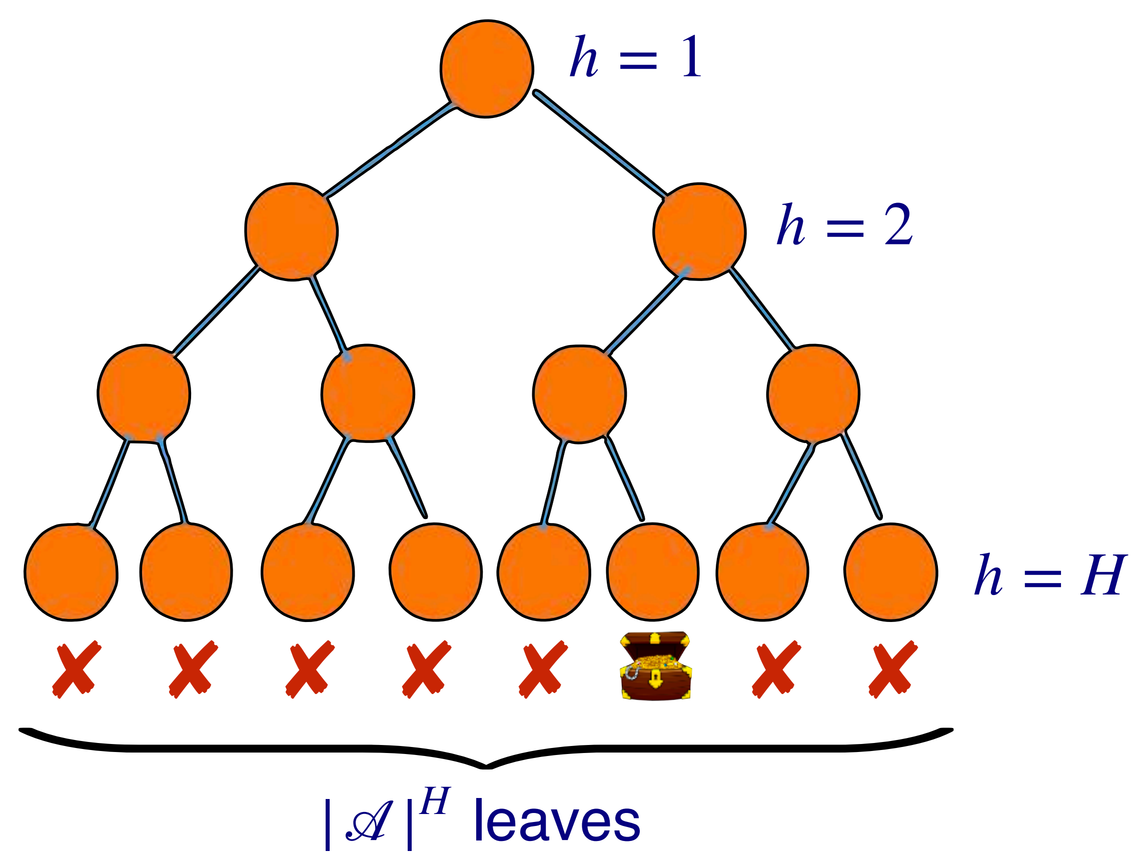
Solution 1: Control # effective distributions

- For general contextual bandits, SquareCB has

$$\mathbf{Reg}(T) \leq \sqrt{\underbrace{|\mathcal{A}|}_{\text{\# possible action distributions}} \cdot T \cdot \mathrm{comp}(\mathcal{F})}$$

- Idea: Can only be "suprised" $|\mathcal{A}|$ times if we explore deliberately.
- No assumption on $\mathcal{F}$, but requires strong assumption on $\mathcal{A}$.

Naively extending reasoning gives $|\mathcal{A}|^H$.



RL: Distribution shift

Solution 1: Control # effective distributions

- For general contextual bandits, $\texttt{SquareCB}$ has

$$\mathbf{Reg}(T) \leq \sqrt{\underbrace{|\mathcal{A}|}_{\text{\# possible action distributions}} \cdot T \cdot \mathrm{comp}(\mathcal{F})}$$

- Idea: Can only be “surprised” $|\mathcal{A}|$ times if we explore deliberately.
- No assumption on $\mathcal{F}$, but requires strong assumption on $\mathcal{A}$.

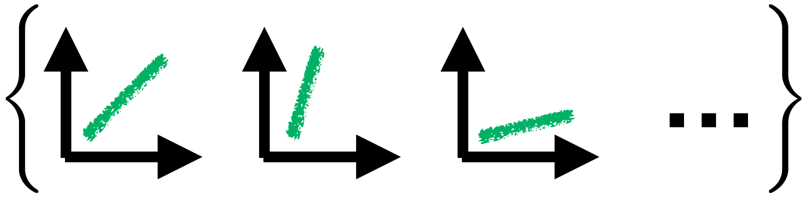
Naively extending reasoning gives $|\mathcal{A}|^H$.

Solution 2: Extrapolation

- For linear contextual bandits ($\mathbb{E}[r(a) \mid x, a] = \langle \phi(x, a), \theta \rangle$), LinUCB has

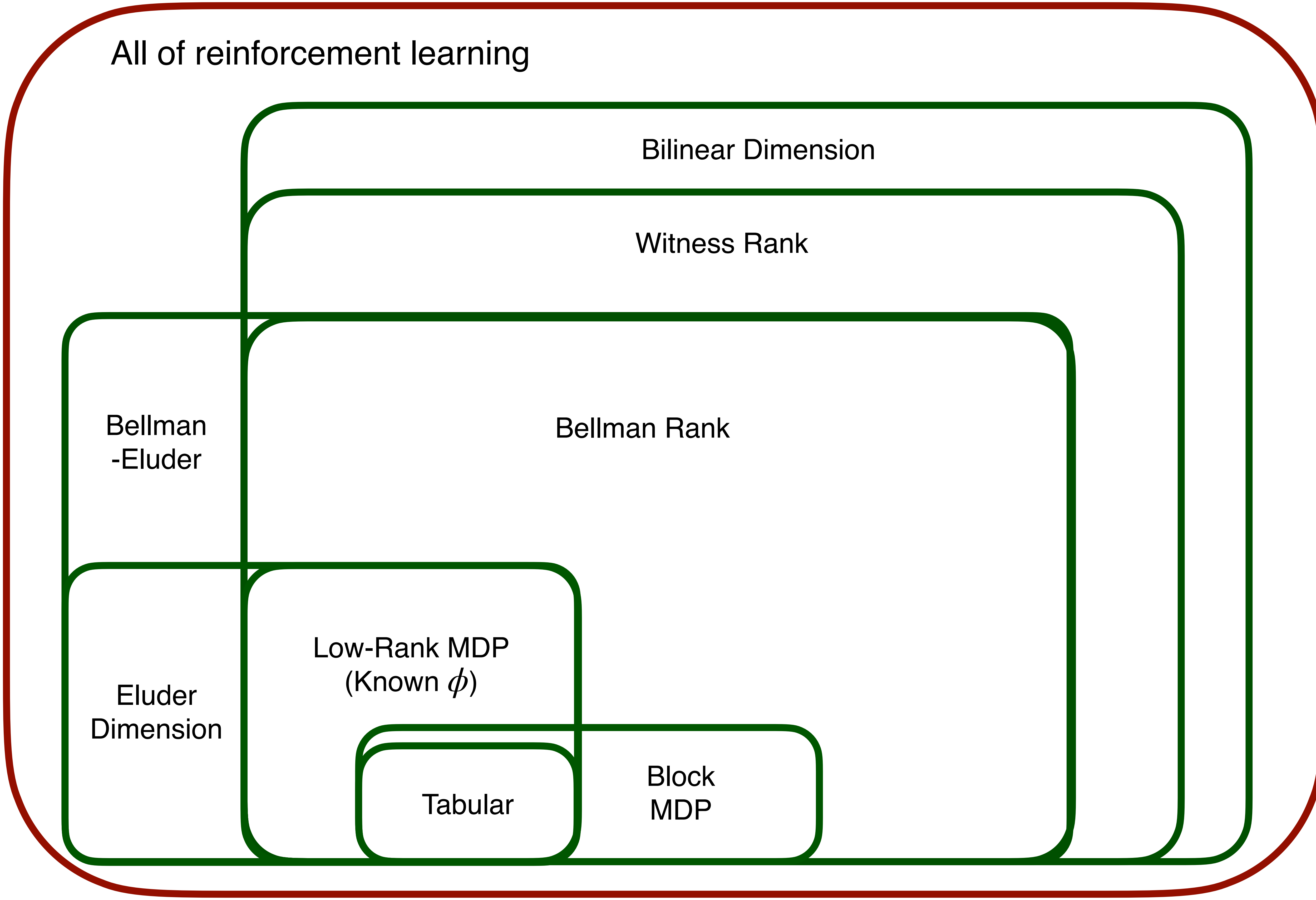
$$\mathbf{Reg}(T) \leq d \cdot \sqrt{T}$$

- Idea: Can extrapolate once we have info from d dimensions.
- No assumption on $\mathcal{A}$, but strong assumption on $\mathcal{F}$.



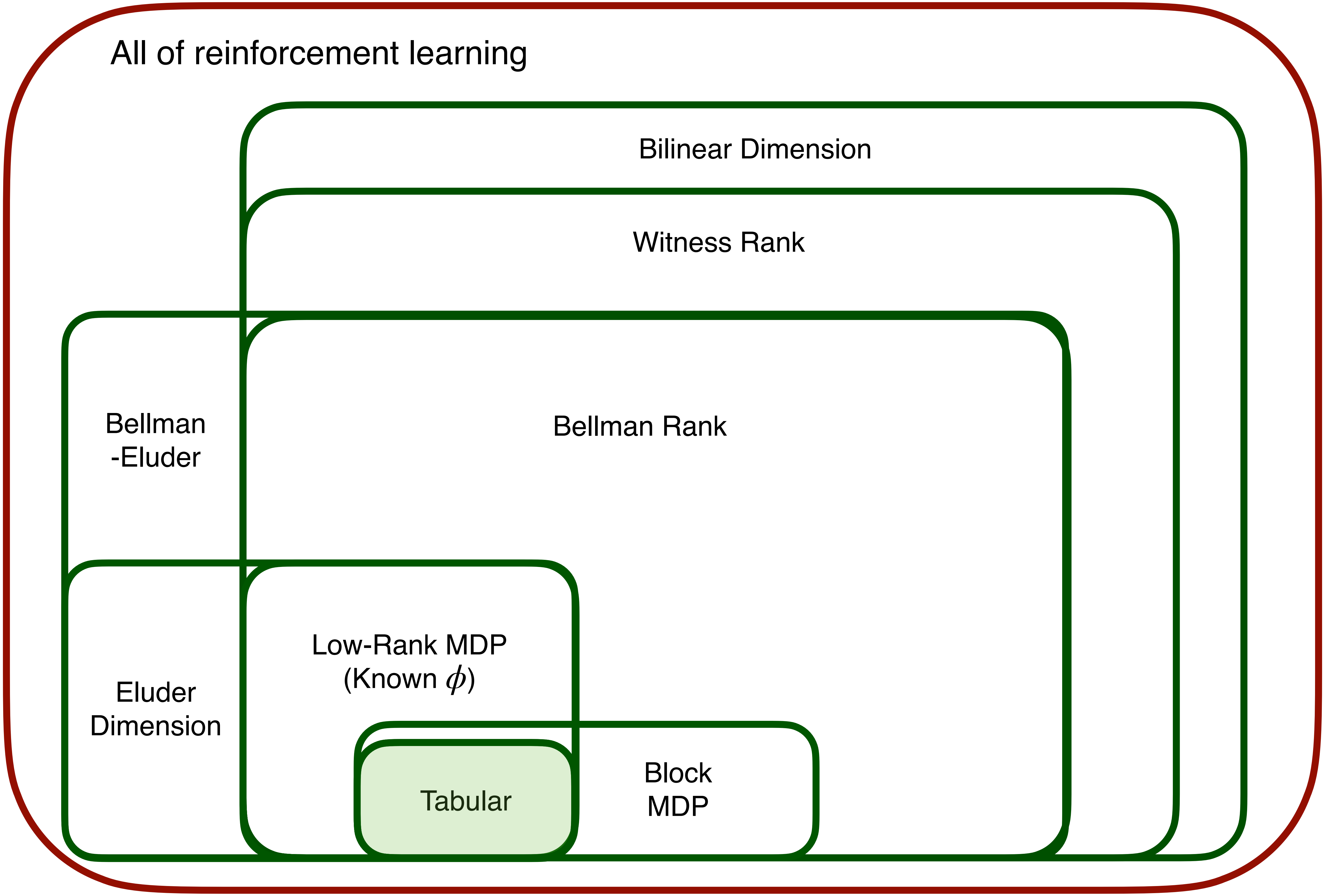
Landscape of RL

Landscape of RL



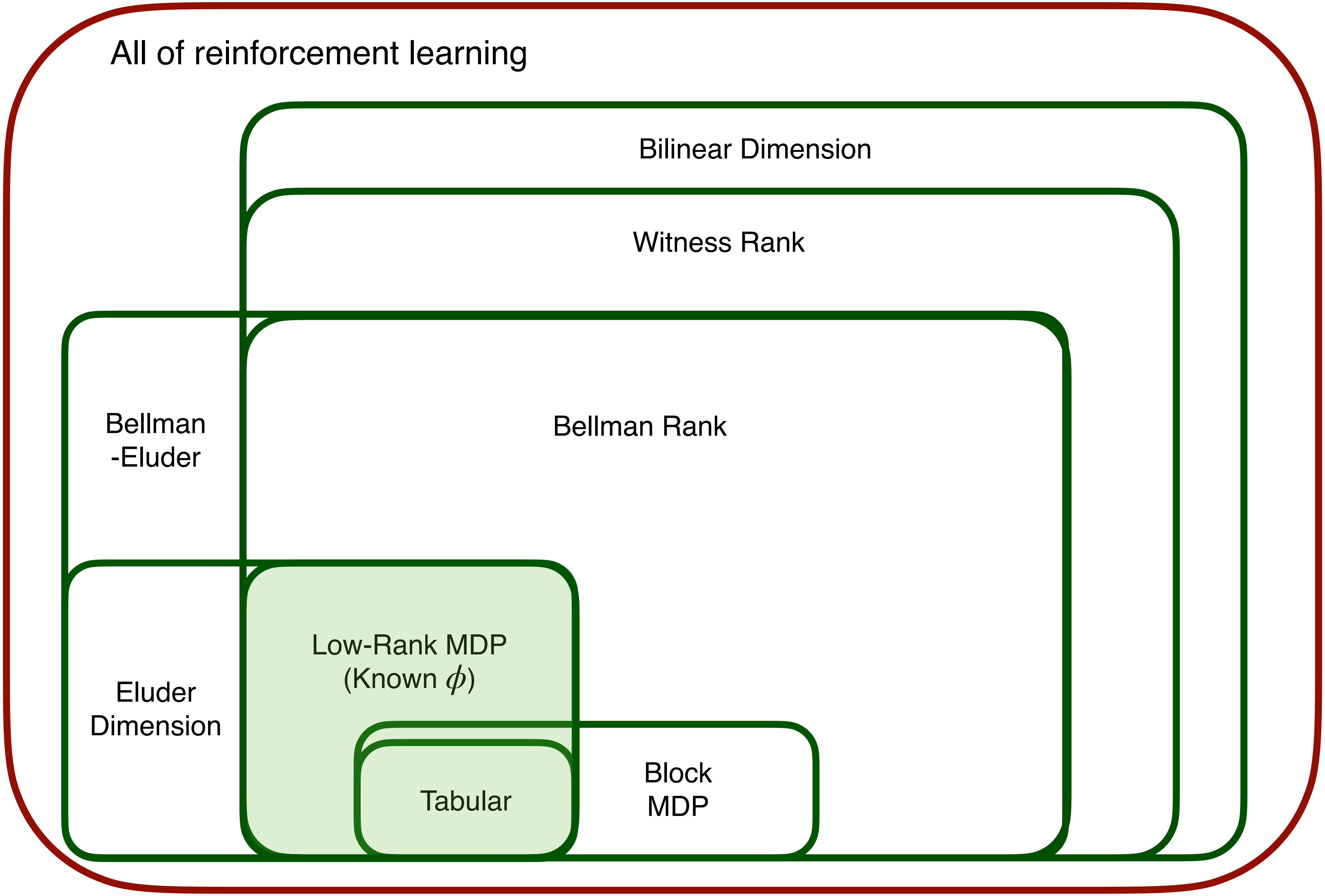
[Credit: Akshay Krishnamurthy]

Landscape of RL



[Credit: Akshay Krishnamurthy]

Landscape of RL



[Credit: Akshay Krishnamurthy]

RL: Linear hypothesis classes

Valued-based setting. Hypothesis class:

$$\mathcal{Q} = \Big\{ Q_h(x,a) = \big\langle \phi(x,a), \theta_h \big\rangle \mid \theta_h \in \mathbb{R}^d \Big\}$$

for fixed feature map $\phi(x,a) \in \mathbb{R}^d$.

RL: Linear hypothesis classes

Valued-based setting. Hypothesis class:

$$\mathcal{Q} = \left\{ Q_h(x,a) = \langle \phi(x,a), \theta_h \rangle \mid \theta_h \in \mathbb{R}^d \right\}$$

for fixed feature map $\phi(x,a) \in \mathbb{R}^d$.

Assumption: Realizability.

Assume $Q^\star \in \mathcal{Q}$.

RL: Linear hypothesis classes

Valued-based setting. Hypothesis class:

$$\mathcal{Q} = \left\{ Q_h(x,a) = \langle \phi(x,a), \theta_h \rangle \mid \theta_h \in \mathbb{R}^d \right\}$$

for fixed feature map $\phi(x,a) \in \mathbb{R}^d$.

Assumption: Realizability.

Assume $Q^\star \in \mathcal{Q}$.

- Contextual bandits ($H = 1$): $\mathbf{Reg}(T) \leq d\sqrt{T}$.

RL: Linear hypothesis classes

Valued-based setting. Hypothesis class:

$$\mathcal{Q} = \left\{ Q_h(x,a) = \langle \phi(x,a), \theta_h \rangle \mid \theta_h \in \mathbb{R}^d \right\}$$

for fixed feature map $\phi(x,a) \in \mathbb{R}^d$.

Assumption: Realizability.

Assume $Q^\star \in \mathcal{Q}$.

- Contextual bandits ($H = 1$): $\mathbf{Reg}(T) \le d\sqrt{T}$.
- RL: $\mathbf{Reg}(T) \ge \min\{\exp(d), \exp(H)\}$. [Weisz et al. '20, '21]

RL: Linear hypothesis classes

Valued-based setting. Hypothesis class:

$$\mathcal{Q} = \left\{ Q_h(x,a) = \langle \phi(x,a), \theta_h \rangle \mid \theta_h \in \mathbb{R}^d \right\}$$

for fixed feature map $\phi(x,a) \in \mathbb{R}^d$.

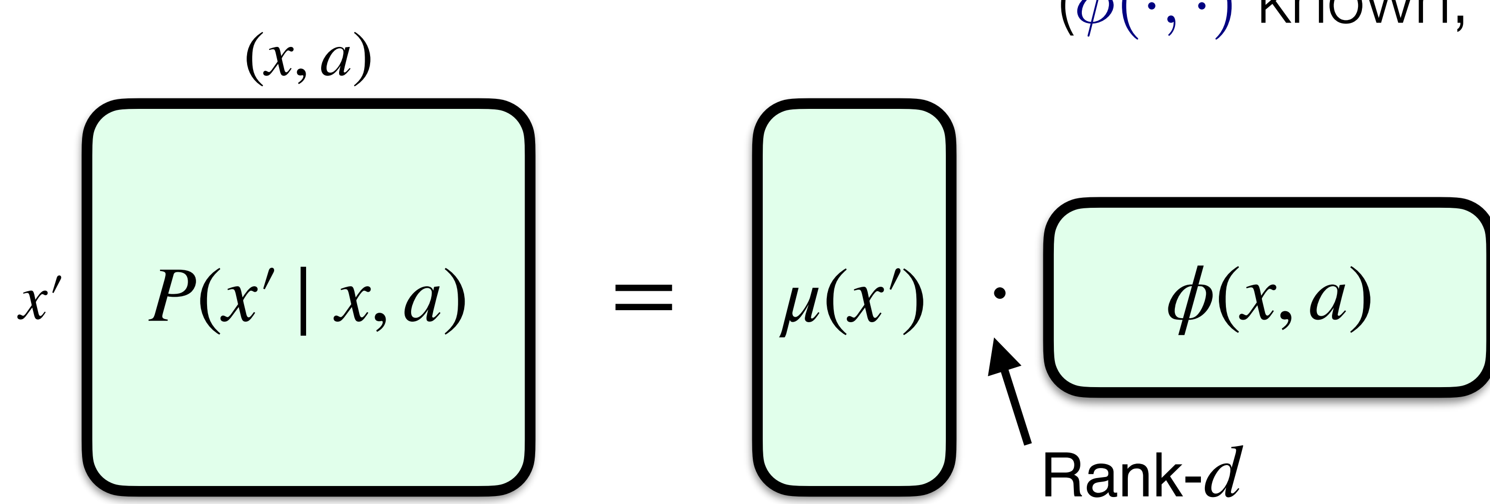
Assumption: Realizability.

Assume $Q^\star \in \mathcal{Q}$.

- Contextual bandits ($H = 1$): $\mathbf{Reg}(T) \leq d\sqrt{T}$.
- RL: $\mathbf{Reg}(T) \geq \min\{\exp(d), \exp(H)\}$. [Weisz et al. '20, '21]

Low-Rank MDP. Have (i) $P(x' \mid x,a) = \langle \phi(x,a), \mu(x') \rangle$, (ii) $R(x,a) = \langle \phi(x,a), \theta \rangle$.

($\phi(\cdot,\cdot)$ known, $\mu(\cdot)$ & θ unknown)



Linear/Low Rank MDPs: Upper confidence bounds

LSVI-UCB [Jin et al. '20]

- With $\overline{Q}^{(t)}_{H+1}(x,a) = 0$, solve

$$\widehat{\theta}^{(t)}_h = \arg\min_{\theta} \sum_{i<t} \left(\langle \phi(x^{(i)}_h, a^{(i)}_h), \theta \rangle - \left(r^{(i)}_h + \max_a \overline{Q}^{(t)}_{h+1}(x^{(i)}_{h+1}, a) \right) \right)^2 .$$

- $\overline{Q}^{(t)}_h(x,a) = \left\langle \phi(x,a), \widehat{\theta}^{(t)}_h \right\rangle + \mathsf{bon}^{(t)}_h(x,a)$.
- Play $\pi^{(t)}_h(x) = \arg\max_a \overline{Q}^{(t)}_h(x,a)$.

Linear/Low Rank MDPs: Upper confidence bounds

LSVI-UCB [Jin et al. '20]

Motivation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + \max_{a'} Q_{h+1}^\star(x_{h+1},a') \mid x_h = x, a_h = a\right]$$

- With $\overline{Q}_{H+1}^{(t)}(x,a) = 0$, solve

$$\widehat{\theta}_h^{(t)} = \arg\min_\theta \sum_{i<t} \left(\langle \phi(x_h^{(i)}, a_h^{(i)}), \theta \rangle - \left(r_h^{(i)} + \max_a \overline{Q}_{h+1}^{(t)}(x_{h+1}^{(i)}, a) \right) \right)^2.$$

- $\overline{Q}_h^{(t)}(x,a) = \left\langle \phi(x,a), \widehat{\theta}_h^{(t)} \right\rangle + \mathsf{bon}_h^{(t)}(x,a)$.
- Play $\pi_h^{(t)}(x) = \arg\max_a \overline{Q}_h^{(t)}(x,a)$.

Linear/Low Rank MDPs: Upper confidence bounds

Motivation

$$Q_h^\star(x,a) = \mathbb{E}\left[r_h + \max_{a'} Q_{h+1}^\star(x_{h+1}, a') \mid x_h = x, a_h = a\right]$$

LSVI-UCB [Jin et al. '20]

- With $\overline{Q}_{H+1}^{(t)}(x,a) = 0$, solve

$$\widehat{\theta}_h^{(t)} = \arg\min_\theta \sum_{i<t} \left(\langle \phi(x_h^{(i)}, a_h^{(i)}), \theta \rangle - \left(r_h^{(i)} + \max_a \overline{Q}_{h+1}^{(t)}(x_{h+1}^{(i)}, a) \right) \right)^2.$$

- $\overline{Q}_h^{(t)}(x,a) = \left\langle \phi(x,a), \widehat{\theta}_h^{(t)} \right\rangle + \mathsf{bon}_h^{(t)}(x,a)$.
- Play $\pi_h^{(t)}(x) = \arg\max_a \overline{Q}_h^{(t)}(x,a)$.

Theorem: LSVI-UCB has

$$\mathbf{Reg}(T) \le \sqrt{d^3 H^4 T}.$$

Analysis for LSVI-UCB

Optimism. With high probability (least squares + low rank MDP structure),

$$\overline{Q}_h^{(t)}(x,a) \geq Q_h^{\star}(x,a) \quad \forall x, a.$$

Analysis for LSVI-UCB

Optimism. With high probability (least squares + low rank MDP structure),

$$\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \quad \forall x, a.$$

Bonus: Let $\Sigma_h^{(t)} = \sum_{i<t} \phi(x_h^{(i)}, a_h^{(i)})\phi(x_h^{(i)}, a_h^{(i)})^\top + \varepsilon \cdot I_{d\times d}$ and set

$$\mathsf{bon}_h^{(t)}(x,a) \propto \sqrt{\phi(x,a)^\top (\Sigma_h^{(t)})^{-1}\phi(x,a)} =: \|\phi(x,a)\|_{(\Sigma_h^{(t)})^{-1}}.$$

Analysis for LSVI-UCB

Optimism. With high probability (least squares + low rank MDP structure),

$$\bar{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \quad \forall x, a.$$

Bonus: Let $\Sigma_h^{(t)} = \sum_{i<t} \phi(x_h^{(i)}, a_h^{(i)})\phi(x_h^{(i)}, a_h^{(i)})^\top + \varepsilon \cdot I_{d\times d}$ and set

$$\mathsf{bon}_h^{(t)}(x,a) \propto \sqrt{\phi(x,a)^\top(\Sigma_h^{(t)})^{-1}\phi(x,a)} =: \|\phi(x,a)\|_{(\Sigma_h^{(t)})^{-1}}.$$

Regret decomposition. As in tabular setting, $\bar{Q}_h^{(t)} \geq Q_h^\star$ pointwise implies

$$\mathbf{Reg}(T) \lesssim \mathrm{poly}(H) \cdot \sum_{t=1}^{T}\sum_{h=1}^{H} \mathsf{bon}_h^{(t)}(x_h^{(t)}, a_h^{(t)}).$$

Analysis for LSVI-UCB

Optimism. With high probability (least squares + low rank MDP structure),

$$\overline{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \quad \forall x, a.$$

Bonus: Let $\Sigma_h^{(t)} = \sum_{i<t} \phi(x_h^{(i)}, a_h^{(i)})\phi(x_h^{(i)}, a_h^{(i)})^\top + \varepsilon \cdot I_{d\times d}$ and set

$$\mathsf{bon}_h^{(t)}(x,a) \propto \sqrt{\phi(x,a)^\top(\Sigma_h^{(t)})^{-1}\phi(x,a)} =: \|\phi(x,a)\|_{(\Sigma_h^{(t)})^{-1}}.$$

Regret decomposition. As in tabular setting, $\overline{Q}_h^{(t)} \geq Q_h^\star$ pointwise implies

$$\mathbf{Reg}(T) \lesssim \mathrm{poly}(H) \cdot \sum_{t=1}^{T}\sum_{h=1}^{H} \mathsf{bon}_h^{(t)}(x_h^{(t)}, a_h^{(t)}).$$

Potential argument.

$$\sum_{t=1}^{T} \mathsf{bon}_h^{(t)}(x_h^{(t)}, a_h^{(t)}) \approx \sum_{t=1}^{T} \|\phi(x_h^{(t)}, a_h^{(t)})\|_{(\Sigma_h^{(t)})^{-1}} \lesssim \sqrt{dT}.$$

Analysis for LSVI-UCB

Optimism. With high probability (least squares + low rank MDP structure),

$$\bar{Q}_h^{(t)}(x,a) \geq Q_h^\star(x,a) \quad \forall x,a.$$

Bonus: Let $\Sigma_h^{(t)} = \sum_{i<t} \phi(x_h^{(i)}, a_h^{(i)})\phi(x_h^{(i)}, a_h^{(i)})^\top + \varepsilon \cdot I_{d\times d}$ and set

$$\mathsf{bon}_h^{(t)}(x,a) \propto \sqrt{\phi(x,a)^\top (\Sigma_h^{(t)})^{-1} \phi(x,a)} =: \|\phi(x,a)\|_{(\Sigma_h^{(t)})^{-1}}.$$

Regret decomposition. As in tabular setting, $\bar{Q}_h^{(t)} \geq Q_h^\star$ pointwise implies

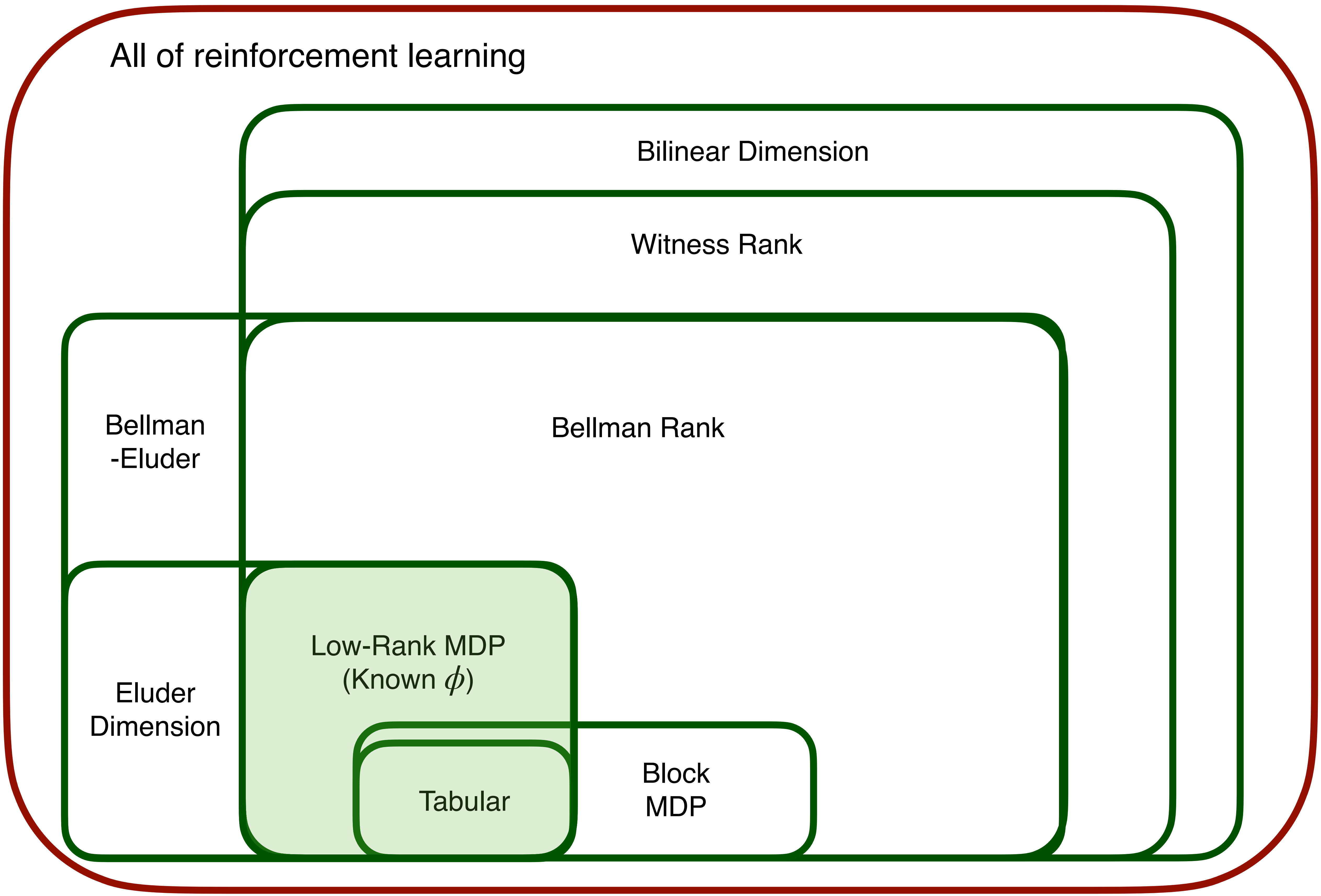
$$\mathbf{Reg}(T) \lesssim \mathrm{poly}(H) \cdot \sum_{t=1}^{T} \sum_{h=1}^{H} \mathsf{bon}_h^{(t)}(x_h^{(t)}, a_h^{(t)}).$$

Potential argument.

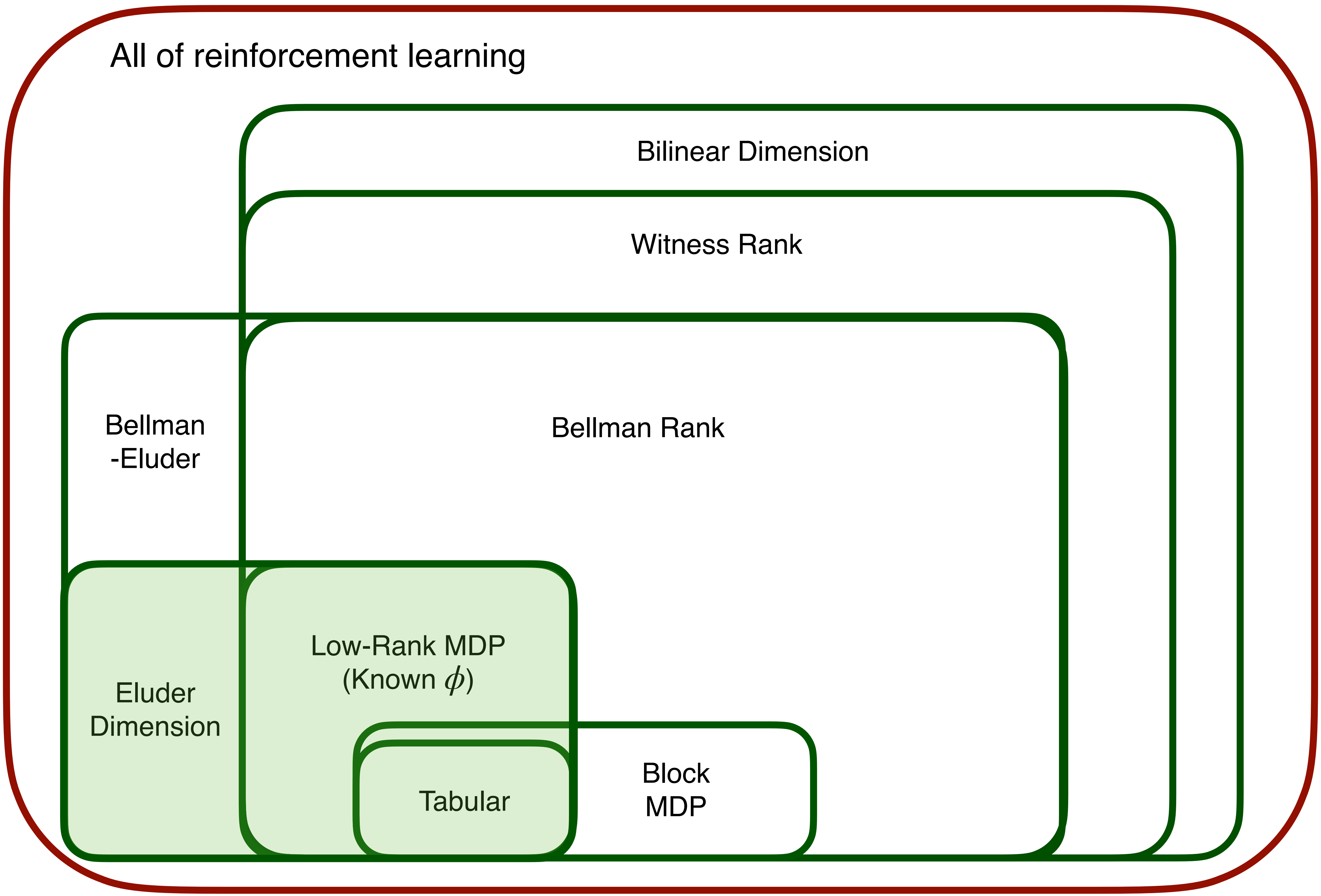
$$\sum_{t=1}^{T} \mathsf{bon}_h^{(t)}(x_h^{(t)}, a_h^{(t)}) \approx \sum_{t=1}^{T} \|\phi(x_h^{(t)}, a_h^{(t)})\|_{(\Sigma_h^{(t)})^{-1}} \lesssim \sqrt{dT}.$$

Intuition: $\Sigma_h^{(t+1)} \leftarrow \Sigma_h^{(t)} + \phi(x_h^{(t)}, a_h^{(t)})\phi(x_h^{(t)}, a_h^{(t)})^\top$.

Landscape of RL



Landscape of RL



Eluder dimension

Eluder dimension: Combinatorial parameter controlling extrapolation.

Eluder dimension

Eluder dimension: Combinatorial parameter controlling extrapolation.

For a class $\mathcal{F} \subseteq (\mathcal{Z} \to \mathbb{R})$, *eluder dimension* $d_E(\mathcal{F}, \varepsilon)$ is the length of the longest sequence $z^{(1)}, \ldots, z^{(N)}$ such that for all $t \leq N$,

$$\exists f, f' \in \mathcal{F} : \quad \left| f(z^{(t)}) - f'(z^{(t)}) \right| > \varepsilon, \quad \text{and} \quad \sqrt{\sum_{i<t} \left| f(z^{(i)}) - f'(z^{(i)}) \right|^2} \leq \varepsilon.$$

Eluder dimension

Eluder dimension: Combinatorial parameter controlling extrapolation.

For a class $\mathcal{F} \subseteq (\mathcal{Z} \to \mathbb{R})$, *eluder dimension* $d_{\mathsf{E}}(\mathcal{F}, \varepsilon)$ is the length of the longest sequence $z^{(1)}, \ldots, z^{(N)}$ such that for all $t \leq N$,

$$\exists f, f' \in \mathcal{F} : \quad \left| f(z^{(t)}) - f'(z^{(t)}) \right| > \varepsilon, \quad \text{and} \quad \sqrt{\sum_{i<t} \left| f(z^{(i)}) - f'(z^{(i)}) \right|^2} \leq \varepsilon.$$

Results:

- Russo & Van Roy '13: $\sqrt{d_{\mathsf{E}}(\mathcal{Q}) \cdot T}$ regret for bandits.
- Wang et al '20: $\sqrt{\mathrm{poly}(d_{\mathsf{E}}(\mathcal{Q}), H) \cdot T}$ regret for RL (w/ additional assumptions).

Eluder dimension

Eluder dimension: Combinatorial parameter controlling extrapolation.

For a class $\mathcal{F} \subseteq (\mathcal{Z} \to \mathbb{R})$, *eluder dimension* $d_{\mathsf{E}}(\mathcal{F}, \varepsilon)$ is the length of the longest sequence $z^{(1)}, \ldots, z^{(N)}$ such that for all $t \leq N$,

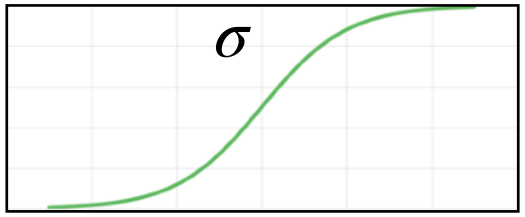
$$\exists f, f' \in \mathcal{F} : \quad \left|f(z^{(t)}) - f'(z^{(t)})\right| > \varepsilon, \quad \text{and} \quad \sqrt{\sum_{i<t} \left|f(z^{(i)}) - f'(z^{(i)})\right|^2} \leq \varepsilon.$$

Results:

- Russo & Van Roy '13: $\sqrt{d_{\mathsf{E}}(\mathcal{Q}) \cdot T}$ regret for bandits.
- Wang et al '20: $\sqrt{\mathrm{poly}(d_{\mathsf{E}}(\mathcal{Q}), H) \cdot T}$ regret for RL (w/ additional assumptions).

Examples:

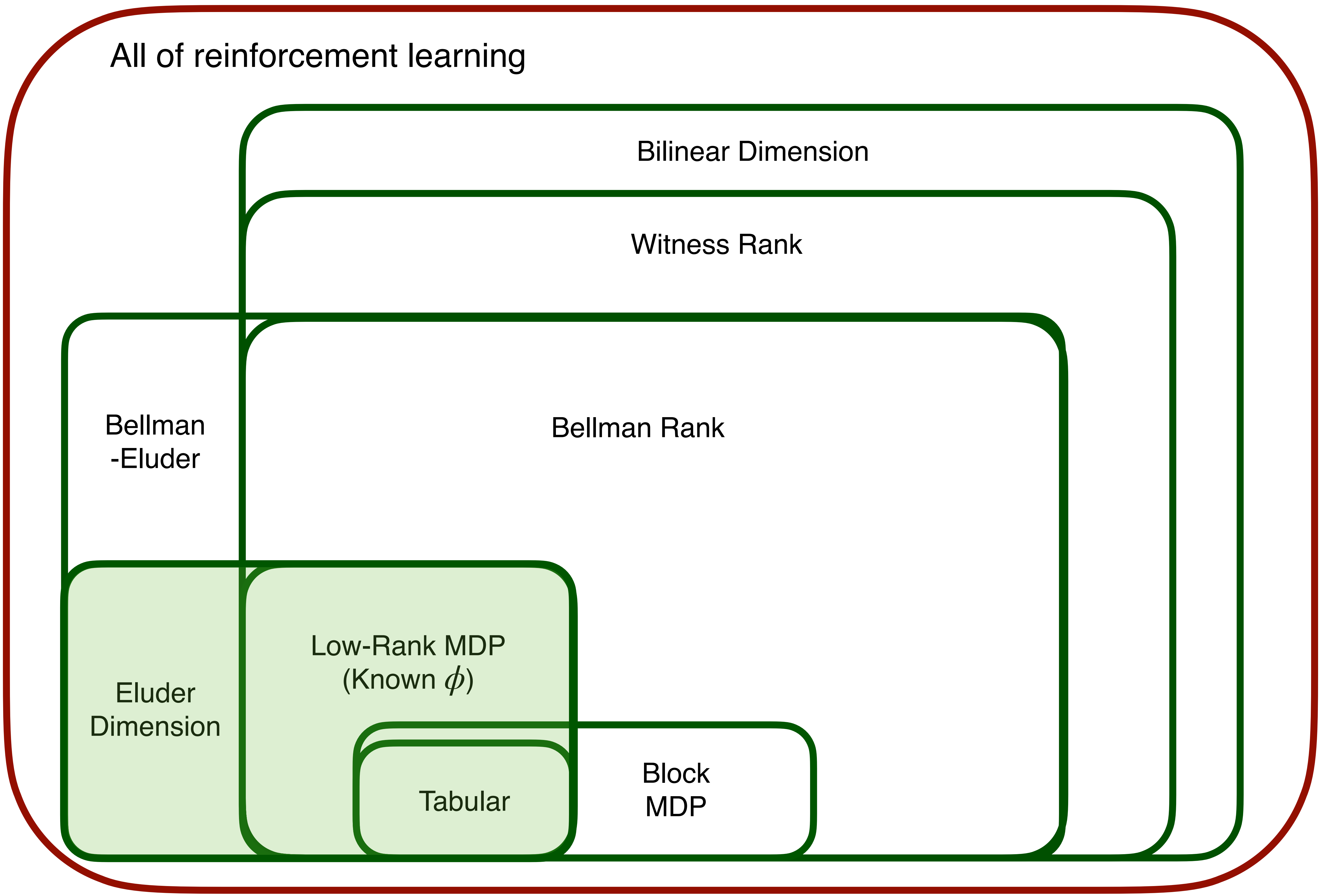
- Linear: $d_{\mathsf{E}}(\mathcal{Q}, \varepsilon) = \widetilde{O}(d)$.
- Generalized linear:
  - $Q(x, a) = \sigma(\langle \phi(x, a), \theta \rangle)$ for $\sigma : \mathbb{R} \to \mathbb{R}$
  - $d_{\mathsf{E}}(\mathcal{Q}, \varepsilon) = \widetilde{O}(d)$ when $0 < c \leq \sigma' \leq C$



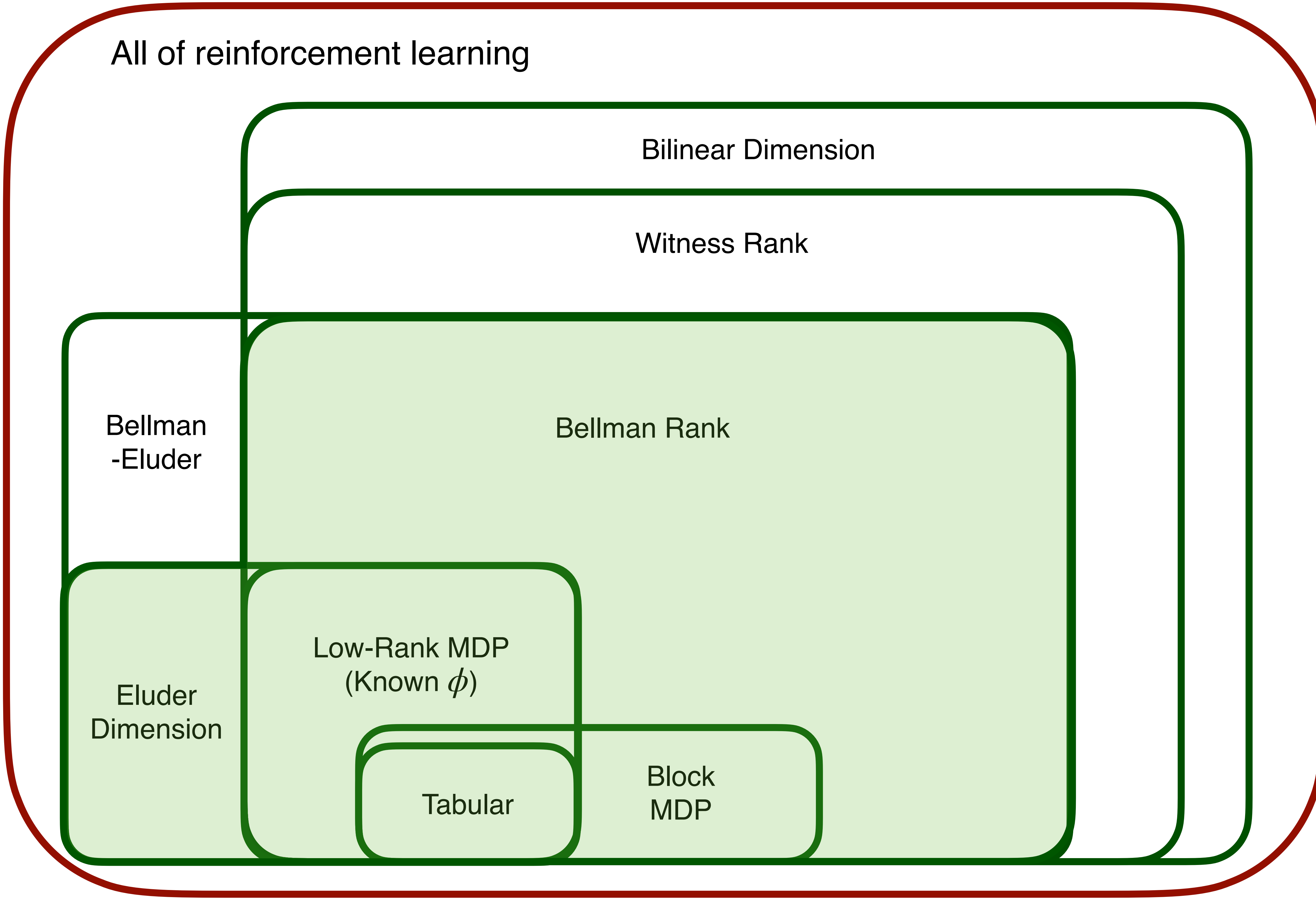
- ReLU: $d_{\mathsf{E}}(\mathcal{Q}, \varepsilon) = \exp(d)$ [LKFS'21]. $(\sigma(z) = \max\{z, 0\})$

Tighter variants: [FRSX'20], [FKQR'21]. Connection to RKHS: [Huang et al '21]

Landscape of RL



Landscape of RL



Bellman rank

$P(x' \mid x, a)$ $=$ $\mu(x')$ $\cdot$ $\phi(x, a)$

Observation: In a low rank MDP, for any function $f(x)$, can write $\mathbb{E}^{\pi}[f(x_h)]$ as

$$\mathbb{E}^{\pi}\left[\mathbb{E}\left[f(x_h) \mid x_{h-1}, a_{h-1}\right]\right] = \mathbb{E}^{\pi}\left[\int \langle \phi(x_{h-1}, a_{h-1}), \mu(x) f(x) \rangle dx\right]$$

$$= \left\langle \mathbb{E}^{\pi}\left[\phi(x_{h-1}, a_{h-1})\right], \int \mu(x) f(x) dx \right\rangle = \langle X(\pi), W(f) \rangle.$$

Bellman rank

$$P(x' \mid x, a) = \mu(x') \cdot \phi(x, a)$$

Observation: In a low rank MDP, for any function $f(x)$, can write $\mathbb{E}^{\pi}[f(x_h)]$ as

$$\mathbb{E}^{\pi}\left[\mathbb{E}\left[f(x_h) \mid x_{h-1}, a_{h-1}\right]\right] = \mathbb{E}^{\pi}\left[\int \langle \phi(x_{h-1}, a_{h-1}), \mu(x) f(x) \rangle dx\right]$$

$$= \left\langle \mathbb{E}^{\pi}\left[\phi(x_{h-1}, a_{h-1})\right], \int \mu(x) f(x) dx \right\rangle = \langle X(\pi), W(f) \rangle.$$

Bellman residual: For $Q \in \mathcal{Q}$ and π, define (π_Q = opt policy for Q)

$$\mathcal{E}_h(\pi, Q) = \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - \left(r_h + \max_a Q_{h+1}(x_{h+1}, a)\right)\right].$$

Bellman rank

$$P(x' \mid x, a) = \mu(x') \cdot \phi(x, a)$$

Observation: In a low rank MDP, for any function $f(x)$, can write $\mathbb{E}^{\pi}[f(x_h)]$ as

$$\mathbb{E}^{\pi}\left[\mathbb{E}\left[f(x_h) \mid x_{h-1}, a_{h-1}\right]\right] = \mathbb{E}^{\pi}\left[\int \langle \phi(x_{h-1}, a_{h-1}), \mu(x) f(x) \rangle dx\right]$$

$$= \left\langle \mathbb{E}^{\pi}\left[\phi(x_{h-1}, a_{h-1})\right], \int \mu(x) f(x) dx \right\rangle = \langle X(\pi), W(f) \rangle.$$

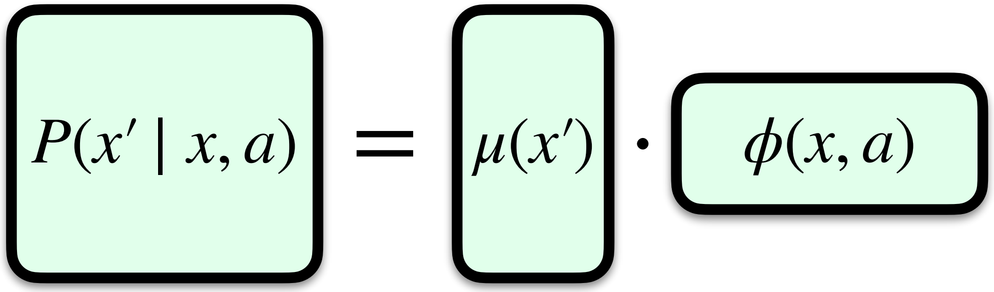
Bellman residual: For $Q \in \mathcal{Q}$ and π, define (π_Q = opt policy for Q)

$$\mathcal{E}_h(\pi, Q) = \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - \left(r_h + \max_a Q_{h+1}(x_{h+1}, a)\right)\right].$$

Motivation

$$Q_h^{\star}(x, a) = \mathbb{E}\left[r_h + \max_{a'} Q_{h+1}^{\star}(x_{h+1}, a') \mid x_h = x, a_h = a\right]$$

Bellman rank



Observation: In a low rank MDP, for any function $f(x)$, can write $\mathbb{E}^{\pi}[f(x_h)]$ as

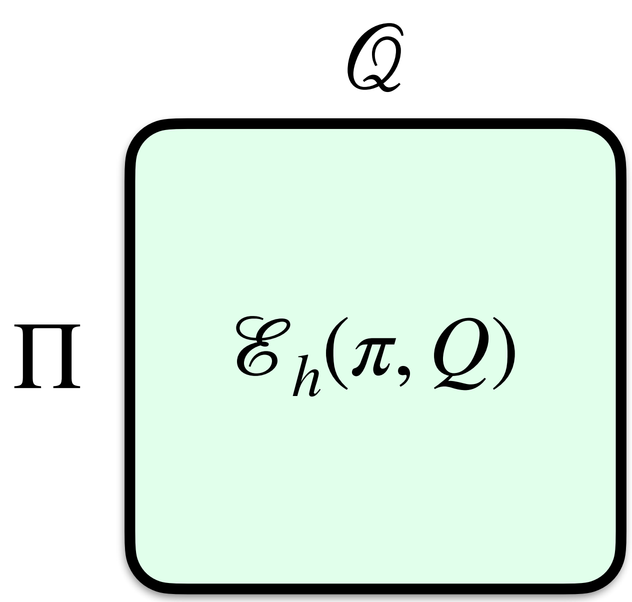
$$\mathbb{E}^{\pi}\left[\mathbb{E}[f(x_h) \mid x_{h-1}, a_{h-1}]\right] = \mathbb{E}^{\pi}\left[\int \langle \phi(x_{h-1}, a_{h-1}), \mu(x) f(x) \rangle dx\right]$$

$$= \left\langle \mathbb{E}^{\pi}[\phi(x_{h-1}, a_{h-1})], \int \mu(x) f(x) dx \right\rangle = \langle X(\pi), W(f) \rangle.$$

Bellman residual: For $Q \in \mathcal{Q}$ and π, define (π_Q = opt policy for Q)

$$\mathcal{E}_h(\pi, Q) = \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - \left(r_h + \max_a Q_{h+1}(x_{h+1}, a)\right)\right].$$

Low Rank MDP has $\mathcal{E}_h(\pi, Q) = \langle X_h(\pi), W_h(Q) \rangle$.



Bellman rank

$$P(x' \mid x, a) = \mu(x') \cdot \phi(x, a)$$

Observation: In a low rank MDP, for any function $f(x)$, can write $\mathbb{E}^{\pi}[f(x_h)]$ as

$$\mathbb{E}^{\pi}\left[\mathbb{E}[f(x_h) \mid x_{h-1}, a_{h-1}]\right] = \mathbb{E}^{\pi}\left[\int \langle \phi(x_{h-1}, a_{h-1}), \mu(x) f(x) \rangle dx\right]$$

$$= \left\langle \mathbb{E}^{\pi}[\phi(x_{h-1}, a_{h-1})], \int \mu(x) f(x) dx \right\rangle = \langle X(\pi), W(f) \rangle.$$

Bellman residual: For $Q \in \mathcal{Q}$ and π, define (π_Q = opt policy for Q)

$$\mathcal{E}_h(\pi, Q) = \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - \left(r_h + \max_a Q_{h+1}(x_{h+1}, a)\right)\right].$$

Low Rank MDP has $\mathcal{E}_h(\pi, Q) = \langle X_h(\pi), W_h(Q) \rangle$.

Bellman rank: [Jiang et al. '17]

$$d_{\text{Be}} = \max_h \operatorname{rank}(\mathcal{E}_h(\cdot, \cdot)).$$

$$\Pi \; \overset{\mathcal{Q}}{\boxed{\mathcal{E}_h(\pi, Q)}}$$

Low Bellman rank implies sample efficiency

Theorem [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

When $Q^\star \in \mathcal{Q}$, can learn an ε-optimal policy with

$$\mathrm{poly}(d_{\mathsf{Be}}, |\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$$

samples.

Low Bellman rank implies sample efficiency

Theorem [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

When $Q^\star \in \mathcal{Q}$, can learn an ε-optimal policy with

$$\mathrm{poly}(d_{\mathsf{Be}}, |\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$$

samples.

Remarks

- $\mathrm{comp}(\mathcal{Q})$ = supervised learning complexity. (e.g., $\log|\mathcal{Q}|$ for finite)

Low Bellman rank implies sample efficiency

Theorem [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

When $Q^\star \in \mathcal{Q}$, can learn an ε-optimal policy with

$$\mathrm{poly}(d_{\mathsf{Be}}, |\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$$

samples.

Remarks

- $\mathrm{comp}(\mathcal{Q}) =$ supervised learning complexity. (e.g., $\log|\mathcal{Q}|$ for finite)
- $|\mathcal{A}|$ can be removed with slightly different variant of d_{Be}. [Jin et al '21, Du et al '21]
- Not computationally efficient in general. [cf. Dann et al. '18]

The BilinUCB algorithm

Variant of OLIVE [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

BilinUCB. [Du et al. '21]

Maintain "plausible" set $\mathcal{Q}^{(t)} \subseteq \mathcal{Q}$.

The BilinUCB algorithm

Variant of OLIVE [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

BilinUCB. [Du et al. '21]

Maintain "plausible" set $\mathcal{Q}^{(t)} \subseteq \mathcal{Q}$.

Repeat:

- Let $\overline{Q}^{(t)} = \arg\max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q)$, where $J_Q(\pi) := \mathbb{E}\big[Q_1(x_1, \pi(x_1))\big]$.
- Set $\pi^{(t)}(x) = \pi_{\overline{Q}^{(t)}}(x)$. **`(opt policy for`** $\overline{Q}^{(t)}$**`)`**

The BilinUCB algorithm

Variant of OLIVE [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

BilinUCB. [Du et al. '21]

Maintain "plausible" set $\mathcal{Q}^{(t)} \subseteq \mathcal{Q}$.

Repeat:

- Let $\overline{Q}^{(t)} = \arg\max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q)$, where $J_Q(\pi) := \mathbb{E}\big[Q_1(x_1, \pi(x_1))\big]$.
- Set $\pi^{(t)}(x) = \pi_{\overline{Q}^{(t)}}(x)$. **`(opt policy for`** $\overline{Q}^{(t)}$**`)`**
- Estimate $\mathcal{E}_h(\pi^{(t)}, Q)$ by running $\pi^{(t)}$ and gathering $O(\varepsilon^{-2})$ trajectories.

The BilinUCB algorithm

Variant of OLIVE [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

BilinUCB. [Du et al. '21]

Maintain "plausible" set $\mathcal{Q}^{(t)} \subseteq \mathcal{Q}$.

Repeat:

- Let $\overline{Q}^{(t)} = \arg\max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q)$, where $J_Q(\pi) := \mathbb{E}\big[Q_1(x_1, \pi(x_1))\big]$.
- Set $\pi^{(t)}(x) = \pi_{\overline{Q}^{(t)}}(x)$. `(opt policy for` $\overline{Q}^{(t)}$`)`
- Estimate $\mathcal{E}_h(\pi^{(t)}, Q)$ by running $\pi^{(t)}$ and gathering $O(\varepsilon^{-2})$ trajectories.
- Set $\mathcal{Q}^{(t+1)} = \Big\{Q \in \mathcal{Q} \mid \sum_{i \le t}(\mathcal{E}_h(\pi^{(i)}, Q)) \lesssim \varepsilon^2 \;\; \forall h\Big\}$

The BilinUCB algorithm

Variant of OLIVE [Jiang, Krishnamurthy, Agarwal, Langford, Schapire '17]

BilinUCB. [Du et al. '21]

Maintain "plausible" set $\mathcal{Q}^{(t)} \subseteq \mathcal{Q}$.

Repeat:

- Let $\overline{Q}^{(t)} = \arg\max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q)$, where $J_Q(\pi) := \mathbb{E}\big[Q_1(x_1, \pi(x_1))\big]$.
- Set $\pi^{(t)}(x) = \pi_{\overline{Q}^{(t)}}(x)$. `(opt policy for` $\overline{Q}^{(t)}$`)`
- Estimate $\mathcal{E}_h(\pi^{(t)}, Q)$ by running $\pi^{(t)}$ and gathering $O(\varepsilon^{-2})$ trajectories.
- Set $\mathcal{Q}^{(t+1)} = \left\{Q \in \mathcal{Q} \mid \sum_{i \leq t}(\mathcal{E}_h(\pi^{(i)}, Q)) \lesssim \varepsilon^2 \ \ \forall h\right\}$

Each iteration requires only $\mathrm{poly}(|\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$ episodes.

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\Big[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\Big] = \big\langle X_h(\pi), W_h(Q)\big\rangle.$$

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \left\langle X_h(\pi), W_h(Q) \right\rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \left\langle X_h(\pi), W_h(Q) \right\rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

Average optimism. As a result, (recall $J_Q(\pi) = \mathbb{E}\left[Q_1(x_1, \pi(x_1))\right]$)

$$J(\pi^\star) = J_{Q^\star}(\pi_{Q^\star}) \leq \max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q) = J_{\overline{Q}^{(t)}}(\pi^{(t)}).$$

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \langle X_h(\pi), W_h(Q)\rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

Average optimism. As a result, (recall $J_Q(\pi) = \mathbb{E}\big[Q_1(x_1, \pi(x_1))\big]$)

$$J(\pi^\star) = J_{Q^\star}(\pi_{Q^\star}) \leq \max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q) = J_{\overline{Q}^{(t)}}(\pi^{(t)}).$$

Regret decomposition. For all Q-functions,

$$J_Q(\pi_Q) - J(\pi_Q) = \sum_{h=1}^{H} \mathcal{E}_h(\pi_Q, Q)$$

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \langle X_h(\pi), W_h(Q)\rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

Average optimism. As a result, (recall $J_Q(\pi) = \mathbb{E}[Q_1(x_1, \pi(x_1))]$)

$$J(\pi^\star) = J_{Q^\star}(\pi_{Q^\star}) \leq \max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q) = J_{\overline{Q}^{(t)}}(\pi^{(t)}).$$

Regret decomposition. For all Q-functions,

$$J_Q(\pi_Q) - J(\pi_Q) = \sum_{h=1}^{H} \mathcal{E}_h(\pi_Q, Q) = \sum_{h=1}^{H} \langle X_h(\pi_Q), W_h(Q)\rangle$$

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h \sim \pi, a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \langle X_h(\pi), W_h(Q) \rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

Average optimism. As a result, (recall $J_Q(\pi) = \mathbb{E}[Q_1(x_1, \pi(x_1))]$)

$$J(\pi^\star) = J_{Q^\star}(\pi_{Q^\star}) \leq \max_{Q \in \mathcal{Q}^{(t)}} J_Q(\pi_Q) = J_{\overline{Q}^{(t)}}(\pi^{(t)}).$$

Regret decomposition. For all Q-functions,

$$J_Q(\pi_Q) - J(\pi_Q) = \sum_{h=1}^{H} \mathcal{E}_h(\pi_Q, Q) = \sum_{h=1}^{H} \langle X_h(\pi_Q), W_h(Q) \rangle$$

so $J(\pi^\star) - J(\pi^{(t)}) \leq \sum_{h=1}^{H} \langle X_h(\pi^{(t)}), W_h(\overline{Q}^{(t)}) \rangle$.

BilinUCB: Analysis

Recall:

$$\mathcal{E}_h(\pi, Q) := \mathbb{E}_{x_h\sim\pi, a_h\sim\pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_{h+1}, a)\right] = \langle X_h(\pi), W_h(Q)\rangle.$$

$Q^\star$ **is never eliminated**. $Q^\star \in \mathcal{Q}^{(t)}\ \forall t$ (Bellman optimality: $\mathcal{E}_h(\pi, Q^\star) = 0$ for all π)

Average optimism. As a result, (recall $J_Q(\pi) = \mathbb{E}[Q_1(x_1, \pi(x_1))]$)

$$J(\pi^\star) = J_{Q^\star}(\pi_{Q^\star}) \leq \max_{Q\in\mathcal{Q}^{(t)}} J_Q(\pi_Q) = J_{\overline{Q}^{(t)}}(\pi^{(t)}).$$

Regret decomposition. For all Q-functions,

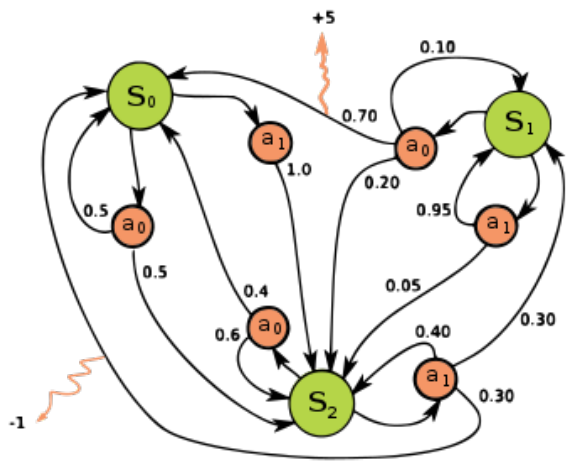
$$J_Q(\pi_Q) - J(\pi_Q) = \sum_{h=1}^H \mathcal{E}_h(\pi_Q, Q) = \sum_{h=1}^H \langle X_h(\pi_Q), W_h(Q)\rangle$$

so $J(\pi^\star) - J(\pi^{(t)}) \leq \sum_{h=1}^H \langle X_h(\pi^{(t)}), W_h(\overline{Q}^{(t)})\rangle$.

Confidence bound. Bound residuals using potential argument.

$$\langle X_h(\pi^{(t)}), W_h(\overline{Q}^{(t)})\rangle \lesssim \left\|X_h(\pi^{(t)})\right\|_{\left(\Sigma_h^{(t)}\right)^{-1}}, \quad \text{w/} \quad \Sigma_h^{(t)} = \sum_{i<t} X_h(\pi^{(i)}) X_h(\pi^{(i)})^\top.$$

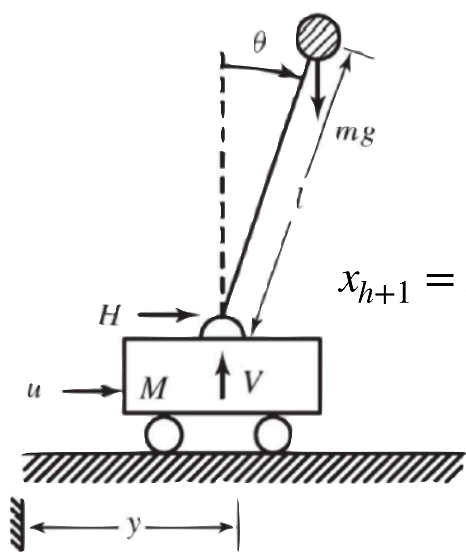
Bellman rank: Examples



Tabular: #states

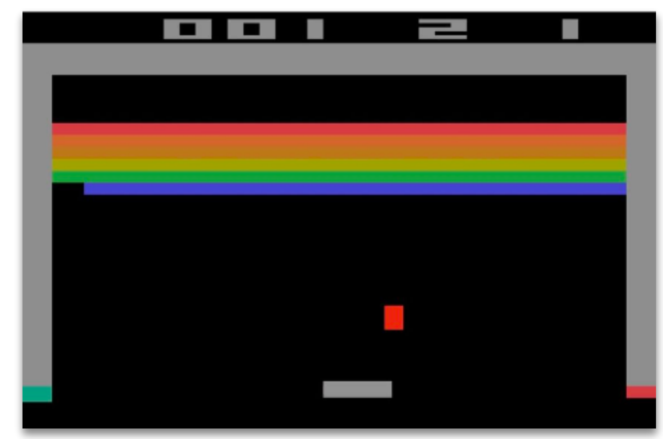
$$P(x' \mid x, a) = \mu(x') \cdot \phi(x, a)$$

Low-Rank MDP: Dimension
(even w/ ϕ unknown)



$$x_{h+1} = Ax_h + Ba_h + w_h$$

Linear-Quadratic Regulator (LQR):
state*action dimension



Block MDP:
latent states

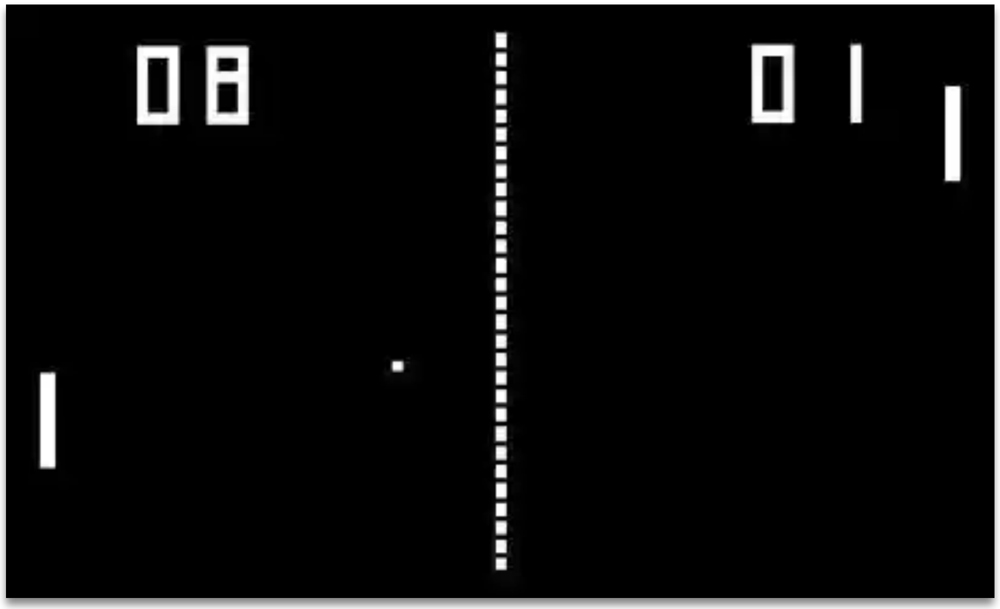
Further examples: [Jiang et al. ’17, Jin et al. ’21, Du et al.’21]

- Low occupancy complexity
- Linear $Q^\star$ & $V^\star$
- State abstraction
- Linear Bellman-Complete
- Predictive state representations
- Reactive POMDP

Example: Block MDP

Rich Observation Markov Decision Process
[Krishnamurthy et al.'16, Jiang et al.'17, Dann et al.'18, Du et al.'19]

- Markov decision process (MDP) with large/high-dimensional state space $\mathcal{X}$.
- **Assumption:** States can be uniquely mapped down into small latent MDP in state space $\mathcal{S}$, with $|\mathcal{S}| < \infty$ states.



$\mathcal{X} = \text{images (pixels)}, \ \mathcal{S} = \text{game state}$

Example: Block MDP

Rich Observation Markov Decision Process
[Krishnamurthy et al.'16, Jiang et al.'17, Dann et al.'18, Du et al.'19]

- Markov decision process (MDP) with large/high-dimensional state space $\mathcal{X}$.
- **Assumption:** States can be uniquely mapped down into small latent MDP in state space $\mathcal{S}$, with $|\mathcal{S}| < \infty$ states.

Bellman rank depends only on # latent states:

$$\text{Bellman Rank} \leq |\mathcal{S}|.$$

Achieve $\mathrm{poly}(|\mathcal{S}|, |\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$ sample complexity. (no $|\mathcal{X}|$ dependence!)

- $\mathrm{comp}(\mathcal{Q})$ will generally depend on mapping from observed to latent states

Example: Block MDP

Rich Observation Markov Decision Process
[Krishnamurthy et al.'16, Jiang et al.'17, Dann et al.'18, Du et al.'19]

- Markov decision process (MDP) with large/high-dimensional state space $\mathcal{X}$.
- **Assumption:** States can be uniquely mapped down into small latent MDP in state space $\mathcal{S}$, with $|\mathcal{S}| < \infty$ states.

Bellman rank depends only on # latent states:

$$\text{Bellman Rank} \leq |\mathcal{S}|.$$

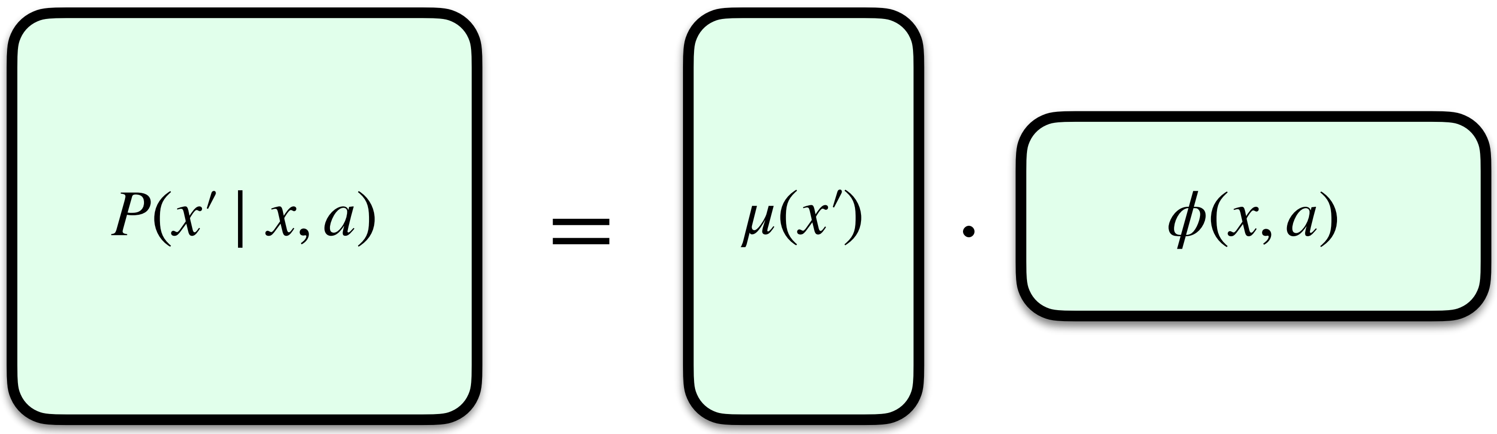
Achieve $\mathrm{poly}(|\mathcal{S}|, |\mathcal{A}|, H, \mathrm{comp}(\mathcal{Q}), \varepsilon^{-1})$ sample complexity. (no $|\mathcal{X}|$ dependence!)

- $\mathrm{comp}(\mathcal{Q})$ will generally depend on mapping from observed to latent states

Idea:

$$\mathcal{E}_h(\pi, Q) := \sum_{s \in \mathcal{S}} \mathbb{P}^{\pi}(s_h = s) \cdot \mathbb{E}_{a_h \sim \pi_Q(x_h)}\left[Q_h(x_h, a_h) - r_h - \max_a Q_{h+1}(x_h, a) \mid s_h = s\right]$$

Example: Low-Rank MDP



Already saw:

$$\mathcal{E}_h(\pi, Q) = \left\langle \mathbb{E}^{\pi}\left[\phi(x_{h-1}, a_{h-1})\right], \int \mu(x)\mathsf{err}_h(x; Q)dx \right\rangle$$

Implication: Sample-efficient learning is possible even when ϕ is unknown.

Discussion

Only considered value-based methods (hypothesis class $= \mathcal{Q}$)

- For some classes, modeling transitions (**hypothesis class** $= \mathcal{M}$) is required.
    - Factored MDP, Linear Mixture MDP
- Model-based generalization: “Witness Rank” [Sun et al. ’19, Du et al. ’21]

Discussion

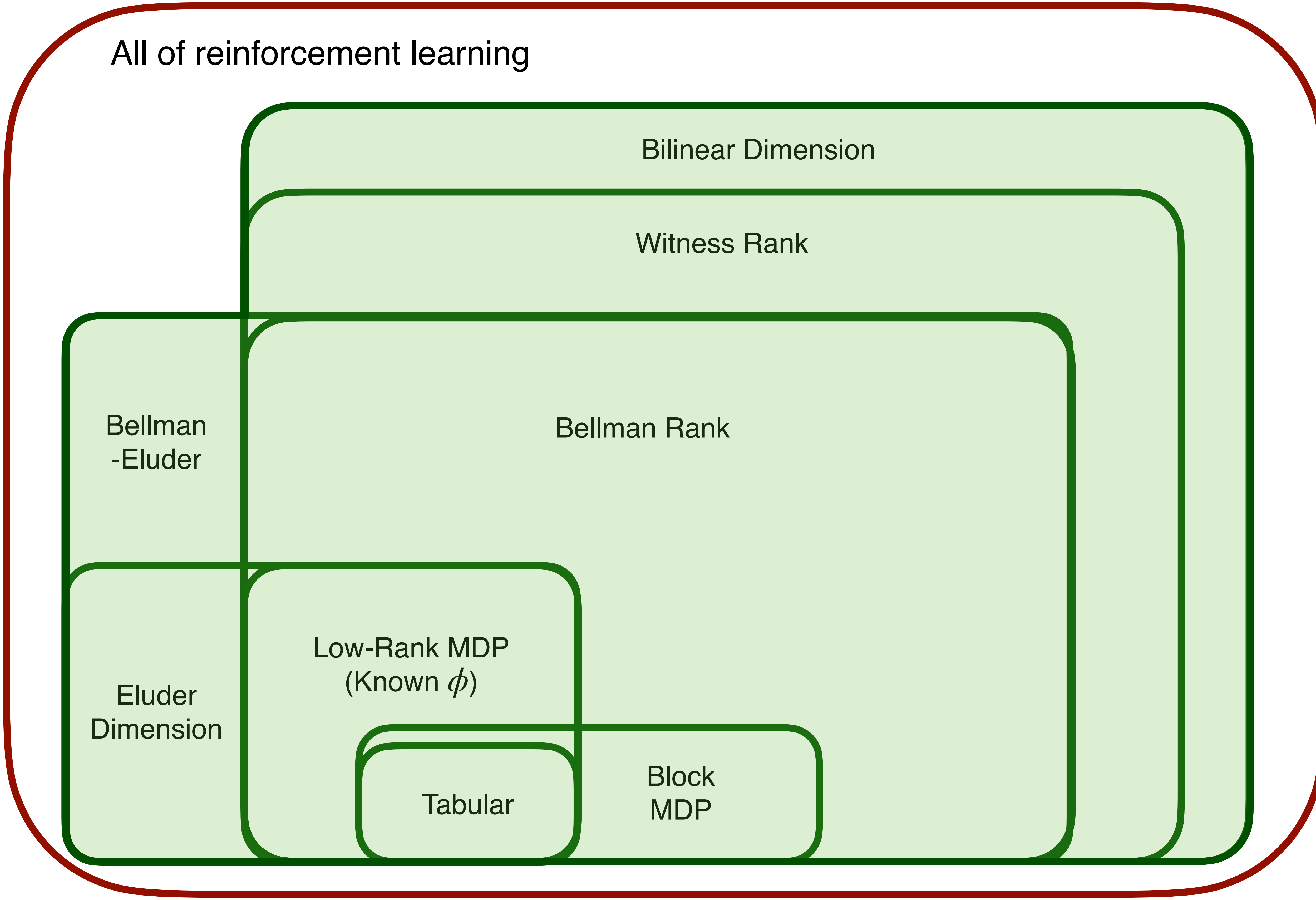
Only considered value-based methods (hypothesis class $= \mathcal{Q}$)

- For some classes, modeling transitions (hypothesis class $= \mathcal{M}$) is required.
  - Factored MDP, Linear Mixture MDP
- Model-based generalization: “Witness Rank” [Sun et al. ’19, Du et al. ’21]

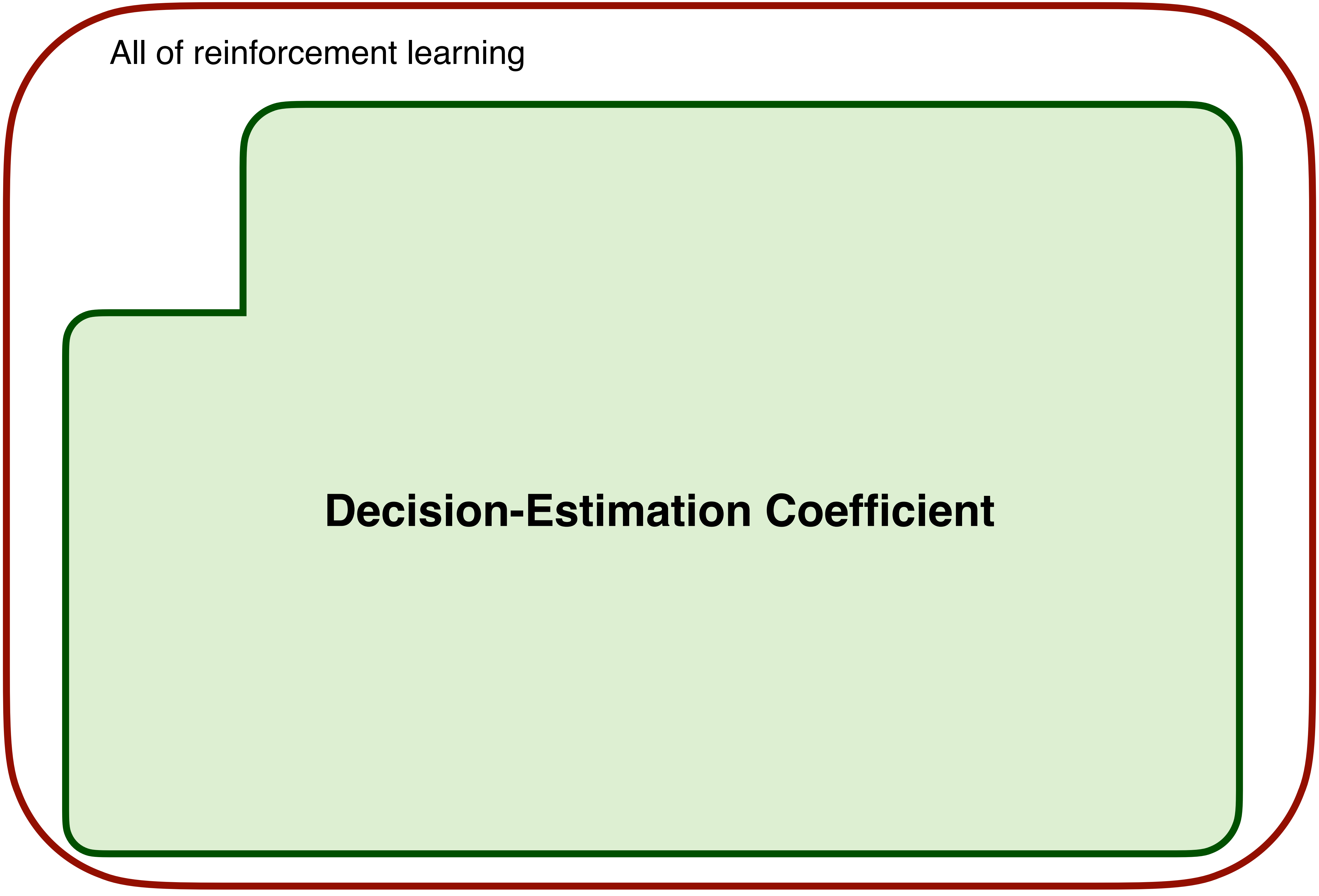
Further generalizations

- Bilinear dimension [Du et al. ’21]
- Bellman rank + eluder [Jin et al. ’21]

Landscape of RL



Landscape of RL



The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^{\star} \in \mathcal{M}$ (realizability)

The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^\star \in \mathcal{M}$ (realizability)
- $M(\pi) =$ distribution over trajectories when we run policy π
- $J_M(\pi) =$ expected reward for π under M
- $\pi^\star_M =$ optimal policy for M

The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^\star \in \mathcal{M}$ (realizability)
- $M(\pi) =$ distribution over trajectories when we run policy π
- $J_M(\pi) =$ expected reward for π under M
- $\pi_M^\star =$ optimal policy for M

The Decision-Estimation Coefficient [**F**, Kakade, Qian, Rakhlin '21]

For $\overline{M} \in \mathcal{M}$ and $\gamma > 0$, define

$$\mathsf{dec}_\gamma(\mathcal{M}, \overline{M}) = \min_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\left[J_M(\pi_M^\star) - J_M(\pi) - \gamma \cdot D_{\mathsf{H}}^2\big(M(\pi), \overline{M}(\pi)\big) \right],$$

where $D_{\mathsf{H}}^2(P, Q) := \int (\sqrt{p(z)} - \sqrt{q(z)})^2 dz$.

The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^\star \in \mathcal{M}$ (realizability)
- $M(\pi) =$ distribution over trajectories when we run policy π
- $J_M(\pi) =$ expected reward for π under M
- $\pi_M^\star =$ optimal policy for M

The Decision-Estimation Coefficient [**F**, Kakade, Qian, Rakhlin '21]

For $\overline{M} \in \mathcal{M}$ and $\gamma > 0$, define

$$\textbf{dec}_\gamma(\mathcal{M}, \overline{M}) = \min_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\Big[\underbrace{J_M(\pi_M^\star) - J_M(\pi)}_{\text{regret of decision}} - \gamma \cdot D_{\mathsf{H}}^2\big(M(\pi), \overline{M}(\pi)\big)\Big],$$

where $D_{\mathsf{H}}^2(P, Q) := \int (\sqrt{p(z)} - \sqrt{q(z)})^2 dz$.

The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^\star \in \mathcal{M}$ (realizability)
- $M(\pi) =$ distribution over trajectories when we run policy π
- $J_M(\pi) =$ expected reward for π under M
- $\pi_M^\star =$ optimal policy for M

The Decision-Estimation Coefficient [**F**, Kakade, Qian, Rakhlin '21]

For $\overline{M} \in \mathcal{M}$ and $\gamma > 0$, define

$$\mathsf{dec}_\gamma(\mathcal{M}, \overline{M}) = \min_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\Big[\underbrace{J_M(\pi_M^\star) - J_M(\pi)}_{\text{regret of decision}} - \gamma \cdot \underbrace{D_{\mathsf{H}}^2\big(M(\pi), \overline{M}(\pi)\big)}_{\text{estimation error for obs.}}\Big],$$

where $D_{\mathsf{H}}^2(P, Q) := \int (\sqrt{p(z)} - \sqrt{q(z)})^2 dz$.

The Decision-Estimation Coefficient

Setup:

- Hypothesis class of MDPs $\mathcal{M}$, $M \in \mathcal{M}$ has $M = (P, R)$.
- $M^{\star} \in \mathcal{M}$ (realizability)
- $M(\pi) =$ distribution over trajectories when we run policy π
- $J_M(\pi) =$ expected reward for π under M
- $\pi_M^{\star} =$ optimal policy for M

The Decision-Estimation Coefficient [**F**, Kakade, Qian, Rakhlin '21]

For $\overline{M} \in \mathcal{M}$ and $\gamma > 0$, define

$$\mathsf{dec}_{\gamma}(\mathcal{M}, \overline{M}) = \min_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\Big[\underbrace{J_M(\pi_M^{\star}) - J_M(\pi)}_{\text{regret of decision}} - \gamma \cdot \underbrace{D_{\mathsf{H}}^2\big(M(\pi), \overline{M}(\pi)\big)}_{\text{estimation error for obs.}}\Big],$$

where $D_{\mathsf{H}}^2(P, Q) := \int (\sqrt{p(z)} - \sqrt{q(z)})^2 dz$.

$$\mathsf{dec}_{\gamma}(\mathcal{M}) := \max_{\overline{M} \in \mathcal{M}} \mathsf{dec}_{\gamma}(\mathcal{M}, \overline{M}).$$

Decision-Estimation Coefficient

DEC: Lower bound [F, Kakade, Qian, Rakhlin '21]

Any algorithm must have

$$\mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\mathsf{dec}_{\gamma}(\mathcal{M}) \cdot T, \gamma\right\}.$$

Decision-Estimation Coefficient

DEC: Lower bound [F, Kakade, Qian, Rakhlin '21]

Any algorithm must have

$$\mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\mathsf{dec}_{\gamma}(\mathcal{M}) \cdot T, \gamma\right\}.$$

Examples:

- Multi-armed bandit:

$$\mathsf{dec}_{\gamma}(\mathcal{M}) \propto \frac{|\mathcal{A}|}{\gamma} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\frac{|\mathcal{A}|T}{\gamma}, \gamma\right\} = \sqrt{|\mathcal{A}|T}.$$

Decision-Estimation Coefficient

DEC: Lower bound [F, Kakade, Qian, Rakhlin '21]

Any algorithm must have

$$\mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\mathsf{dec}_\gamma(\mathcal{M}) \cdot T, \gamma\right\}.$$

Examples:

- Multi-armed bandit:

$$\mathsf{dec}_\gamma(\mathcal{M}) \propto \frac{|\mathcal{A}|}{\gamma} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\frac{|\mathcal{A}|T}{\gamma}, \gamma\right\} = \sqrt{|\mathcal{A}|T}.$$

- Bellman rank d:

$$\mathsf{dec}_\gamma(\mathcal{M}) \geq \frac{d}{\gamma} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \sqrt{d \cdot T}.$$

Decision-Estimation Coefficient

DEC: Lower bound [F, Kakade, Qian, Rakhlin '21]

Any algorithm must have

$$\mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\mathsf{dec}_\gamma(\mathcal{M}) \cdot T, \gamma\right\}.$$

Examples:

- Multi-armed bandit:

$$\mathsf{dec}_\gamma(\mathcal{M}) \propto \frac{|\mathcal{A}|}{\gamma} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \max_{\gamma>0} \min\left\{\frac{|\mathcal{A}|T}{\gamma}, \gamma\right\} = \sqrt{|\mathcal{A}|T}.$$

- Bellman rank d:

$$\mathsf{dec}_\gamma(\mathcal{M}) \geq \frac{d}{\gamma} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \sqrt{d \cdot T}.$$

- Linear $Q^\star$ (dimension d):

$$\mathsf{dec}_\gamma(\mathcal{M}) \geq \mathbb{I}\{\gamma \leq \exp(d)\} \quad \Longrightarrow \quad \mathbf{Reg}(T) \geq \exp(d).$$

(recovers [Weisz et al. '21])

Decision-Estimation Coefficient: Algorithms

Estimation-to-Decisions (E2D):

Decision-Estimation Coefficient: Algorithms

Estimation-to-Decisions (E2D):

For $t = 1, \ldots, T$:

- Get estimator $\widehat{M}^{(t)} \in \mathcal{M}$ from supervised estimation algorithm.

Decision-Estimation Coefficient: Algorithms

Estimation-to-Decisions (E2D):

For $t = 1, \ldots, T$:

- Get estimator $\widehat{M}^{(t)} \in \mathcal{M}$ from supervised estimation algorithm.
- Solve min-max optimization problem: (corresponds to $\mathsf{dec}_\gamma(\mathcal{M}, \widehat{M}^{(t)})$)

$$p^{(t)} = \operatorname*{arg\,min}_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\left[J_M(\pi^\star_M) - J_M(\pi) - \gamma \cdot D^2_{\mathsf{H}}\big(M(\pi), \widehat{M}^{(t)}(\pi)\big) \right].$$

Decision-Estimation Coefficient: Algorithms

Estimation-to-Decisions (E2D):

For $t = 1, \ldots, T$:

- Get estimator $\widehat{M}^{(t)} \in \mathcal{M}$ from supervised estimation algorithm.
- Solve min-max optimization problem: (corresponds to $\mathsf{dec}_\gamma(\mathcal{M}, \widehat{M}^{(t)})$)

$$p^{(t)} = \operatorname*{arg\,min}_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\Big[J_M(\pi^\star_M) - J_M(\pi) - \gamma \cdot D^2_{\mathsf{H}}\big(M(\pi), \widehat{M}^{(t)}(\pi)\big)\Big].$$

- Sample $\pi^{(t)} \sim p^{(t)}$ and update estimation algorithm with trajectory.

Decision-Estimation Coefficient: Algorithms

Estimation-to-Decisions (E2D):

For $t = 1, \ldots, T$:

- Get estimator $\widehat{M}^{(t)} \in \mathcal{M}$ from supervised estimation algorithm.
- Solve min-max optimization problem: (corresponds to $\mathsf{dec}_\gamma(\mathcal{M}, \widehat{M}^{(t)})$)

$$p^{(t)} = \arg\min_{p \in \Delta(\Pi)} \max_{M \in \mathcal{M}} \mathbb{E}_{\pi \sim p}\Big[J_M(\pi^\star_M) - J_M(\pi) - \gamma \cdot D^2_{\mathsf{H}}\big(M(\pi), \widehat{M}^{(t)}(\pi)\big)\Big].$$

- Sample $\pi^{(t)} \sim p^{(t)}$ and update estimation algorithm with trajectory.

DEC: Upper bound [F, Kakade, Qian, Rakhlin '21]

The E2D algorithm has

$$\mathbf{Reg}(T) \leq \min_{\gamma > 0} \max\big\{\mathsf{dec}_\gamma(\mathcal{M}) \cdot T, \gamma \cdot \mathbf{Est}_{\mathsf{H}}(T)\big\},$$

where $\mathbf{Est}_{\mathsf{H}}(T) := \sum_{t=1}^{T} D^2_{\mathsf{H}}\big(M^\star(\pi^{(t)}), \widehat{M}^{(t)}(\pi^{(t)})\big)$.

$\mathbf{Est}_{\mathsf{H}}(T) \leq \mathrm{comp}(\mathcal{M})$:

- $\mathrm{comp}(\mathcal{M}) = \log|\mathcal{M}|$ (finite), $\mathrm{comp}(\mathcal{M}) = \widetilde{O}(d)$ (parametric).

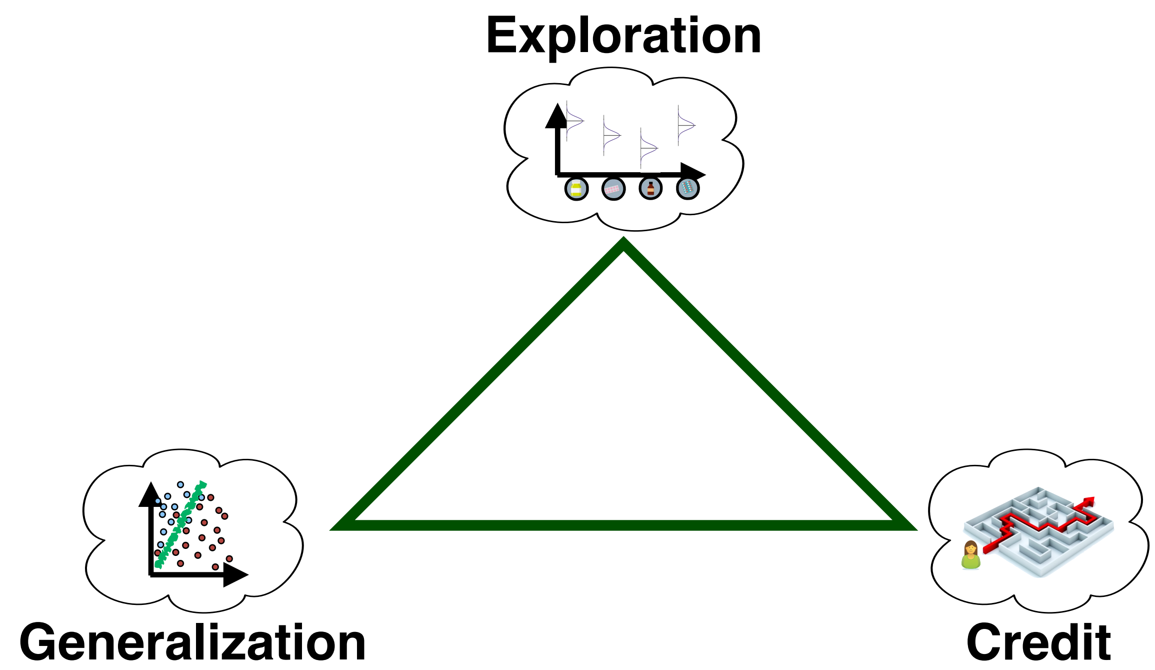
Frontier: Summary

Multiple ways to handle distribution shift:

- Extrapolation: Linear models, eluder dimension.
- Effective # distributions: Bellman rank and friends.

Decision-estimation coefficient provides necessary conditions.

Conclusion



Challenges for RL

- Credit assignment
- Exploration
- Generalization

The frontier: Exploration + generalization + credit assignment

- Lots of room for new theoretical/algorithmic insights.
- Bridging theory + practice.

Multi-agent RL (Markov games/stochastic games)

- What function approximation/modeling assumptions?
  (how well do I need to model my opponent's behavior?)
- Min-max optimization perspective? (policy gradient)
- Competitive vs. cooperative, centralized vs. decentralized, . . .
- Communication
- ⋮