Regret Bounds of Model-Based Reinforcement Learning

Joint work with Alex Ayoub, Chengzhuo Ni, Zeyu Jia, Csaba Szepesvari, Lin Yang

Mengdi Wang

PRINCETON UNIVERSITY DeepMind

Model-Based Reinforcement Learning

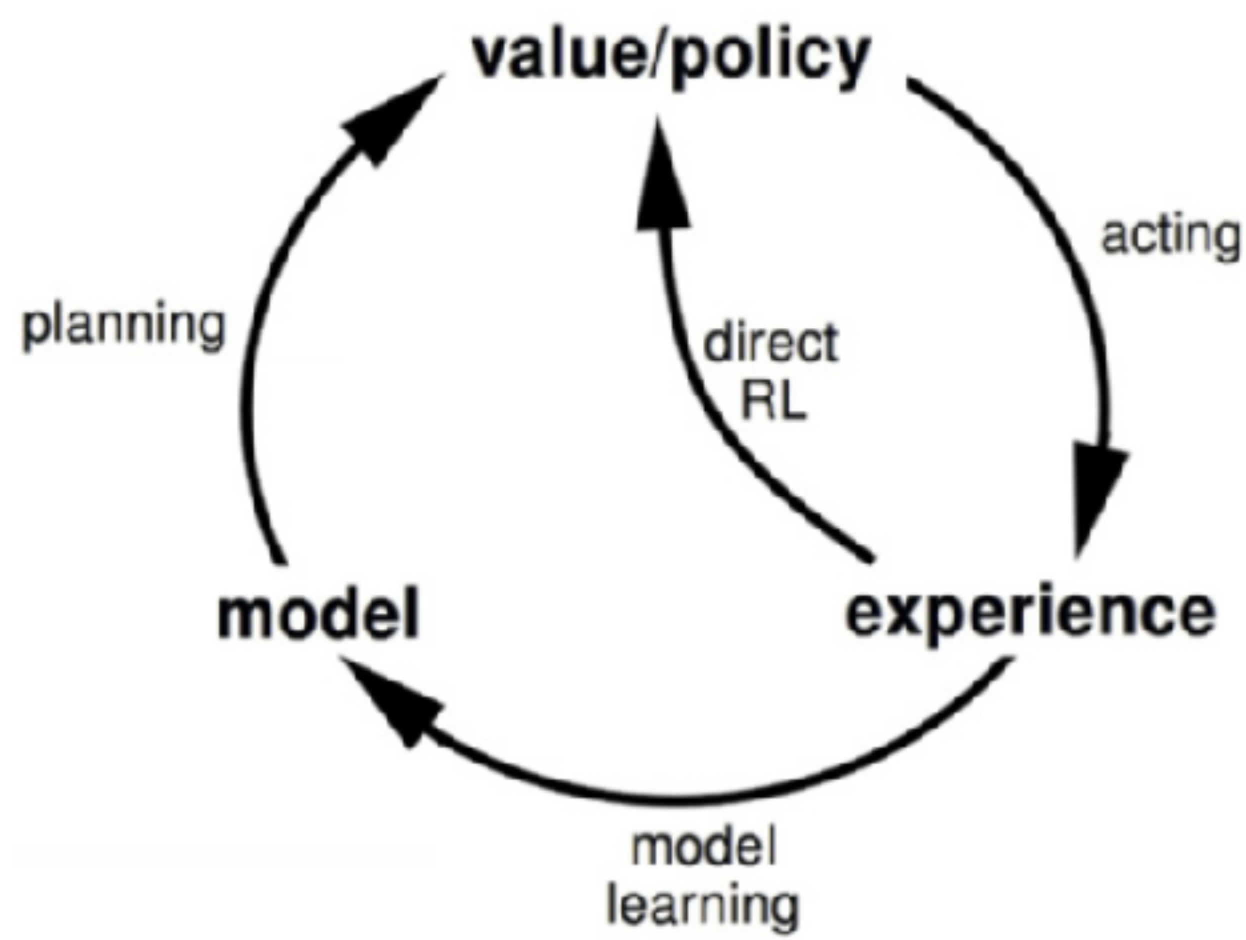
- We fit a model from some family

$$P(s' \mid s, a), \qquad P \in \mathscr{P}$$

to experiences

$$(s_t, a_t, s_{t+1}, r_{t+1})$$

- Then use the learned model for planning and acting



Sutton and Barto (2018)

We ask:

- How to "fit a model"?
- Regret guarantee?

Tabular Markov decision process

- A finite set of states S
- A finite set of actions A
- Reward is given at each state-action pair (s,a):

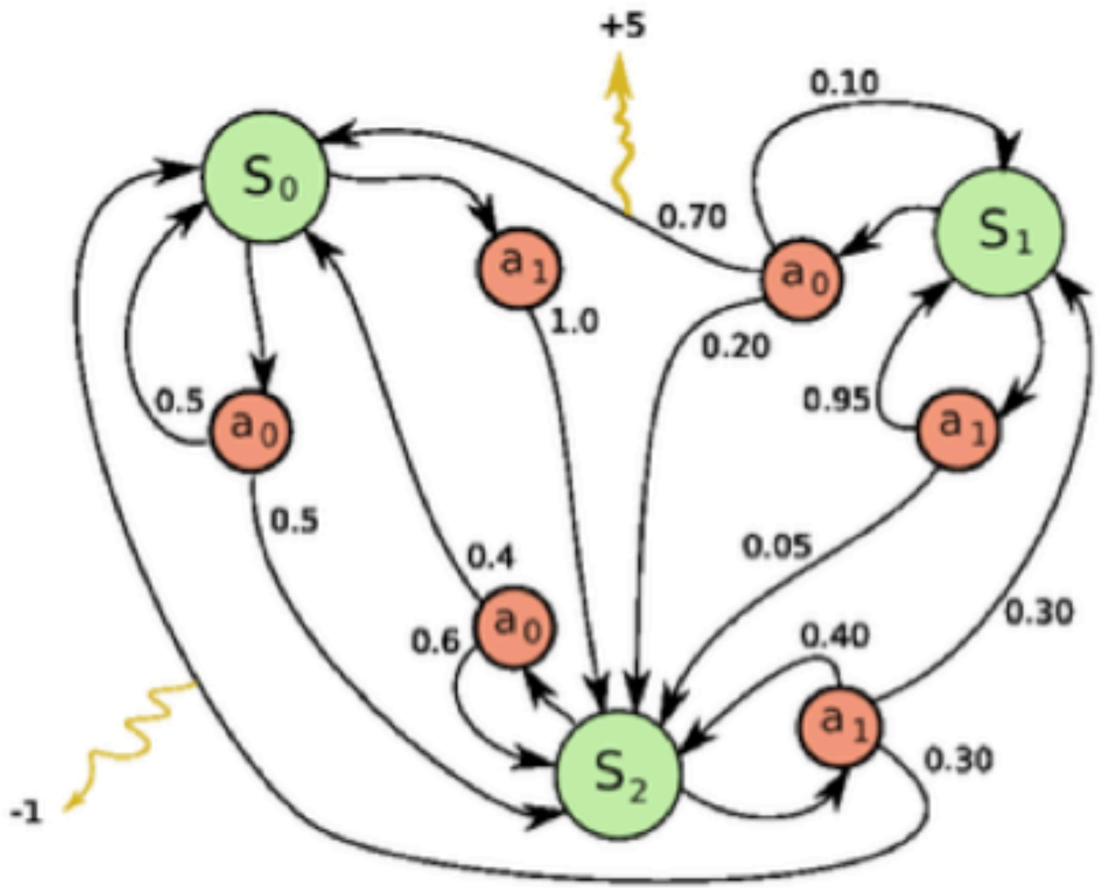
  $r(s,a) \in [0,1]$
- State transits to s' with prob.

  $P(s'|s,a)$

- Find a best policy $\pi : S \rightarrow A$ such that

$$\max_{\pi} v^{\pi} = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

- $\gamma \in (0,1)$ *is a discount factor*



We call if "tabular MDP" if there is no structural knowledge at all

Episodic Reinforcement Learning

- **Regret of a learning algorithm $\mathscr{K}$**

$$\textbf{Regret}_{\mathscr{K}}(T) = \sum_{n=1}^{N} \left(V^*(s_0) - \sum_{h=1}^{H} r\left(s_{n,h}, a_{n,h}\right)\right),$$

where T= NH, and the sample state-action path $\{s_{n,h}, a_{n,h}\}$ is generated on-the-fly by the learning algorithm

- Many many works: LQR (Abbasi-Yadkori & Szepesvári 2011), (Osband & Van Roy 2014), Deterministic (Zheng and Van Roy 2013), Tabular (Jin et al 2018), (Russo 2019), Q learning with function approximation (Jin et al 2019), among many others

-

Upper Confidence Model-Based RL (UCRL)

- UCRL alternates between two steps:

1. Confidence set construction: construct a confidence set B of the unknown transition model, based on experiences $(s_t, a_t, s_{t+1}, r_{t+1})$

2. Optimistic planning:

$$\hat{\pi} = \text{argmax}_\pi \max_{P \in B} V_P(\pi)$$

Then use this optimistic policy in the next episode

Example 1: Deterministic continuous control

- Consider a deterministic system

$$\textbf{maximize}_\pi \sum_{h=1}^{H} r(s_h, a_h)$$
$$\textbf{subject to } s_{h+1} = f(s_h, a_h), a_h = \pi(s_h, h), s_1 = s_0 .$$

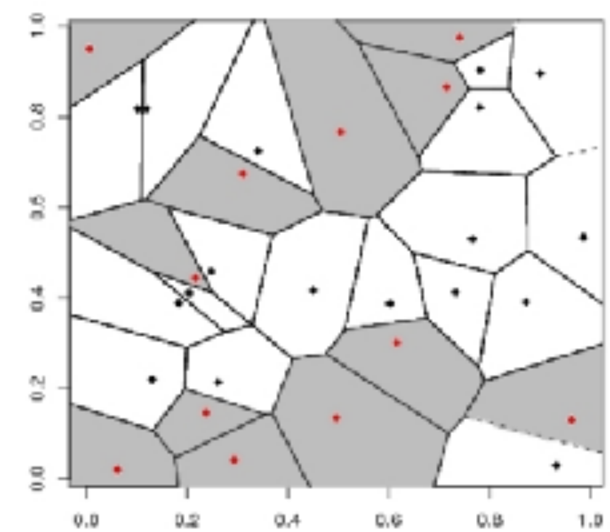
- Metric: Suppose that the only structural knowledge we have is a metric dist over the state-action space

$$dist((s, a), (s', a'))$$

- Let $\mathscr{P}$ be the model class: Set of all deterministic and Lipschitz continuous (w.r.t. to metric $dist$) transition models

A Simple Metric-Based RL Algorithm

- At the beginning of the (n+1)th episode, suppose the samples collected so far are stored in a D_n buffer



- Estimate Q values using **nearest neighbor transitions**

$$Q_H^{(k+1)}(s,a) \leftarrow \min_{(s',a') \in D^{(k+1)}} \left(r(s',a') + L \cdot \textbf{dist}[(s,a),(s',a')] \right)$$

$$Q_h^{(k+1)}(s,a) \leftarrow \min_{(s',a') \in D^{(k+1)}} \left[r(s',a') + \sup_{a''} Q_{h+1}^{(k+1)}(f(s',a'),a'') + L \cdot \textbf{dist}[(s',a'),(s,a)] \right]$$

- In the new episode, choose actions greedily by $\max_a Q_{n,h}(s,a)$

Regret Analysis

- **Theorem** The K-episode regret of the metric-RL algorithm satisfies

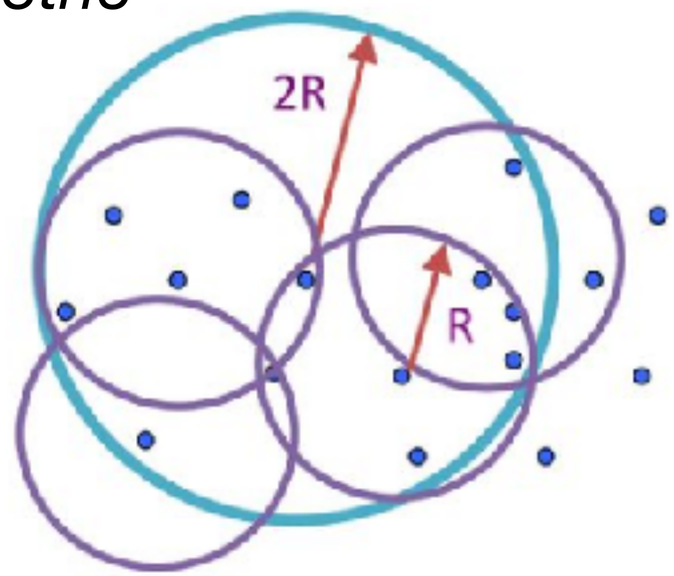
$$\textbf{Regret}(K) = O(DLK)^{\frac{d}{d+1}} \cdot H$$

  - d is the doubling dimension of s-a space
  - D is the diameter of s-a space
- **Theorem** The above regret bound is minimax optimal.

(Learn to Control In Metric Space with Optimal Regret, Allerton, 2019. With Ni and Yang.)

Doubling Dimension d

- Here d be the ***doubling dimension*** *of the state space (smallest positive integer k such that every ball in the metric space can be covered by 2^k balls of half radius)*



- $d \ll$ raw dimension

- For example: raw-pixel images of a video game belong to a smooth manifold and have much smaller d

- Metric-RL *learns the manifold at the same time when solving the dynamic program*. It captures the small intrinsic dimension automatically.

Example 2: Feature space embedding of transition model

- **Suppose we are given state-action feature maps**

$$state, action \mapsto [\phi_1(state, action), \ldots, \phi_d(state, action)] \in \mathbb{R}^N$$

$$state \mapsto [\psi_1(state), \ldots, \psi_{d'}(state)] \in \mathbb{R}^{d'}$$

- Assume that the unknown transition kernel can be fully embedded in the feature space, i.e., there exists a transition core M* such that

$$M^* \phi(s, a) = \mathbb{E}[\psi(s')] \, .$$

- A linear model for state-to-state prediction

-

The MatrixRL Algorithm

- At the beginning of the (n+1)th episode, suppose the samples collected so far are

$$\{(s_{n,h}, a_{n,h}), s_{n,h+1}\} \to \{\phi_{n,h}, \psi_{n,h}\} := \{\phi(s_{n,h}, a_{n,h}), \psi(s_{n,h+1})\}$$

- We will use their corresponding feature vectors.

- Estimate the transition core via matrix ridge regression

$$M_n = \arg\min_{M} \sum_{n'<n, h\leq H} \left\| \psi_{n',h}^{\top} K_{\psi}^{-1} - \phi_{n',h}^{\top} M \right\|_2^2 + \|M\|_F^2.$$

Where K_{ψ} is a precomputed matrix

- However, using empirical estimate greedily would lead to poor exploration

- Borrow ideas from linear bandit (Dani et al 08, Chu et al 11, …)

The MatrixRL Algorithm

- **Construct a matrix confidence ball** around the estimated transition core

$$B_n = \left\{ M \in \mathbb{R}^{d\times d'} : \quad \|(A_n)^{1/2}(M - M_n)\|_F \le \sqrt{\beta_n} \right\}$$

- **Find optimistic Q-function estimate**

$$Q_{n,h}(s,a) = r(s,a) + \max_{M\in B_n} \phi(s,a)^\top M \Psi^\top V_{n,h+1}, \quad Q_{n,H} = 0$$

where the value estimate is given by

$$V_{n,h}(s) = \Pi_{[0,H]}\left[\max_a Q_{n,h}(s,a) \right]$$

- **In the new episode, choose actions greedily by** $\max_a Q_{n,h}(s,a)$

- The optimistic Q encourage exploration: (s,a) with higher uncertainty gets tried more often

(RL in Feature Space: Matrix Bandit, Kernels, and Regret Bounds, Preprint, 2019)

Regret Bound for MatrixRL

- **Theorem** Under the embedding assumption and regularity assumptions, the T-time-step regret of MatrixRL satisfies with high probability thats

$$\textbf{Regret}(T) \leq C \cdot dH^2 \cdot \sqrt{T},$$

- First polynomial regret bound for RL in feature space.

- *Independent of S*

- Minimax optimal?

- *It is optimal in d and T, close to optimal in H*

(RL in Feature Space: Matrix Bandit, Kernels, and Regret Bounds, ICML, 2019)

From Feature to Kernel Embedding of Transition Model

- Consider the more generic assumption:
- The unknown transition probability kernel belongs to the product Hilbert spaces spanned by state/action features:

$$P \in \mathcal{H}_\phi \times \mathcal{H}_\psi$$

Algorithm 2 KernelMatrixRL: Reinforcement Learning with Kernels

Input: An episodic MDP environment $M = (\mathcal{S}, \mathcal{A}, P, s_0, r, H)$, kernel functions k_ϕ, k_ψ;
Total number of episodes N;
Initialize: empty reply buffer $\mathcal{B} = \{\}$;
for episode $n = 1, 2, \ldots, N$ **do**
For $(s, a) \in \mathcal{S} \times \mathcal{A}$, let

$$w_n(s,a) := \sqrt{k_\phi[(s,a),(s,a)] - \mathbf{k}_{\Phi_{n-1},s,a}^\top (I + \mathbf{K}_{\Phi_{n-1}})^{-1} \mathbf{k}_{\Phi_{n-1},s,a}};$$

$$x_n(s,a) := \mathbf{k}_{\Phi_{n-1},s,a}^\top (I + \mathbf{K}_{\Phi_{n-1}})^{-1} \mathbf{K}_{\Psi_{n-1}} (\mathbf{K}_{\Psi_{n-1}} \mathbf{K}_{\Psi_{n-1}}^\top)^{-1} \mathbf{K}_{\Psi_n};$$

Let $\{Q_{n,h}\}$ be defined as follows:

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad Q_{n,H+1}(s,a) := 0 \quad \text{and}$$

$$\forall h \in [H]: \quad Q_{n,h}(s,a) := r(s,a) + x_n(s,a)^\top V_{n,h+1} + \eta_n w_n(s,a), \tag{9}$$

where

$$V_{n,h}(s) = \Pi_{[0,H]}\big[\max_a Q_{n,h}(s,a)\big] \qquad \forall s, a, n, h;$$

and η_n is a parameter to be determined;
for stage $h = 1, 2, \ldots, H$ **do**
Let the current state be $s_{n,h}$;
Play action $a_{n,h} = \arg\max_{a \in \mathcal{A}} Q_{n,h}(s_{n,h}, a)$;
Record the next state $s_{n,h+1}$: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_{n,h}, a_{n,h}, s_{n,h+1})\}$;
end for
end for

Theorem $\quad$ $\textbf{Regret}(T) \le O\Big(\|P\|_{\mathcal{H}_\phi \times \mathcal{H}_\psi} \cdot \log(T) \cdot \tilde{d} \cdot H^2 \cdot \sqrt{T}\Big)$

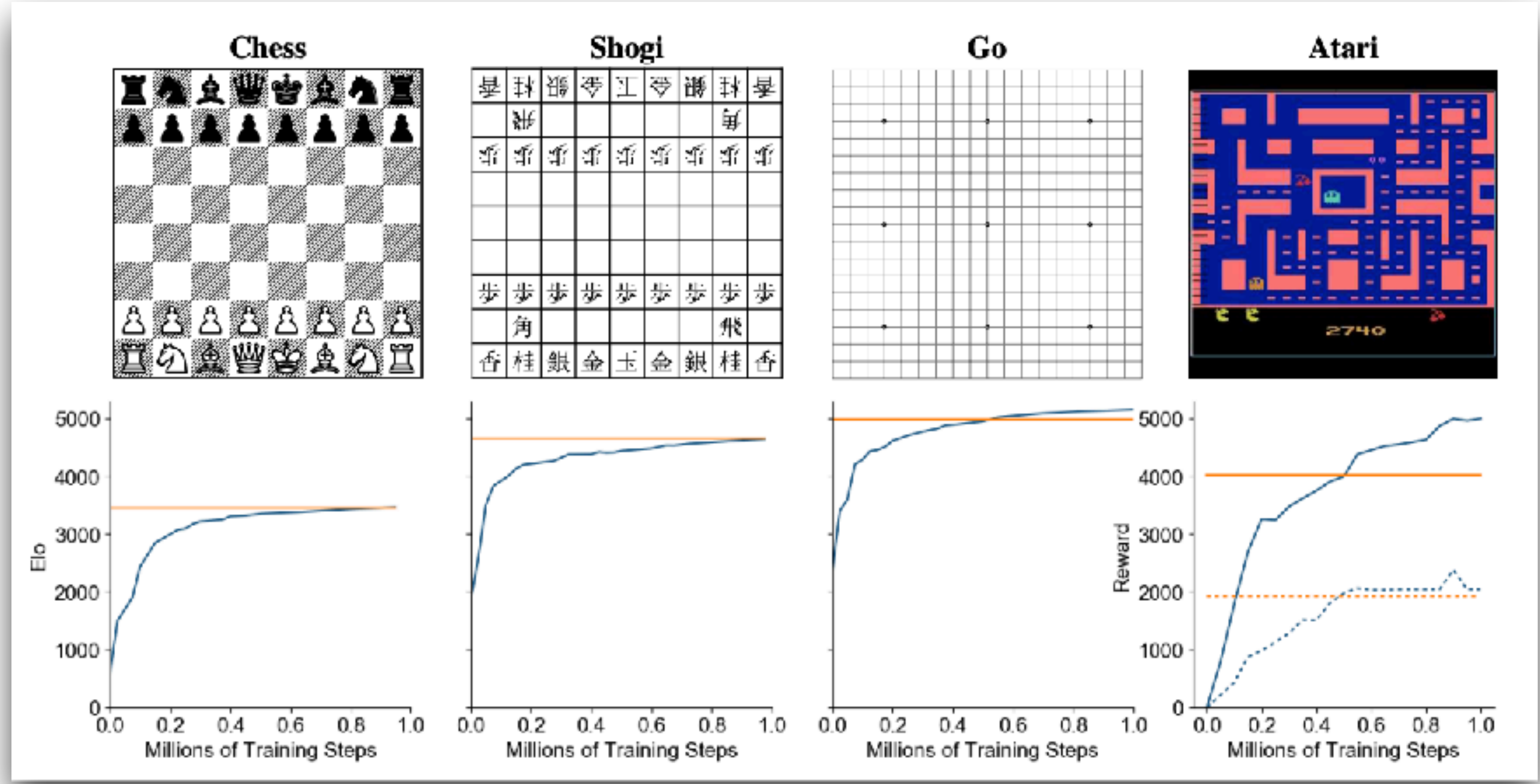
RL regret in kernel space depends on Hilbert space norm of the transition kernel and effective dimension of the kernel space

(RL in Feature Space: Matrix Bandit, Kernels, and Regret Bounds, w. Lin Yang, 2019)

Example 3: *Can we learn a more generic model?*

A motivating example: *MuZero*

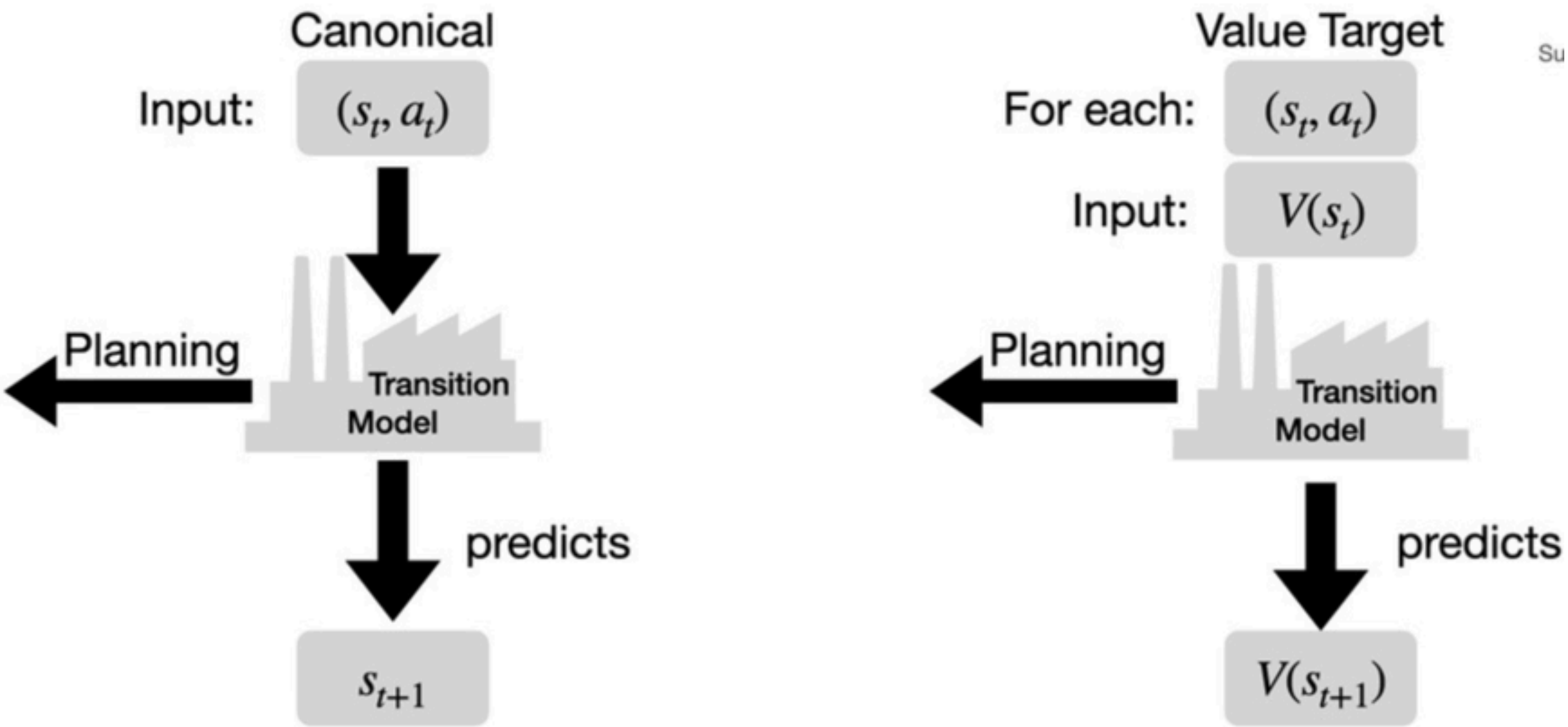
A single algorithm generalizes to 60 games and beats the best player of each



End-to-end training; no prior knowledge of game rules; plan & explore with a learned model

(figure from MuZero paper, by DeepMind, Nature 2020)

- Key idea of Muzero: only try to predict quantities central to the game, e.g., value and policies
- Let's try to predict values only: **Value-Targeted Regression (VTR)**



Assumption of Value-Targeted Regression

- There exists a class of transition model $\mathscr{P}$ such that

$$P \in \mathscr{P}$$

- $\mathscr{P}$ is known

- $\mathscr{P}$ is generic

- Examples: linear models, non-linear models, sparse models, neural network models, physics models, etc.

Value-Targeted Regression (VTR) for Confidence Set Construction

- Confidence Set

$$B = \{P' \,|\, L(P') \leq \beta\}$$

  - $L(P') = \sum_{t=1}^{T} (\langle P'(\cdot \,|\, s_t, a_t), V_t \rangle - y_t)^2$
  - $y_t := V_t(s_{t+1})$
  - V_t is the agent's real-time value estimate
- The agent is training the model P' to predict estimated value of next state

Full Algorithm of UCRL-VTR

- **Let θ parameterize the state-to-value predictor** (which implies a transition model class $\mathscr{P}$)
- Let $\hat{V}$ be real-time value estimate at the beginning of a new episode

1. Whenever observing a new sample (s, a, r', s'), update data buffer
   $D \leftarrow D \cup \{(x(\cdot), y)\}$ where $x(\theta) = \mathbb{E}_\theta[\hat{V}(s') \mid s, a], y = \hat{V}(s')$
2. **Value-targeted nonlinear regression for model learning** $\hat{\theta} = \text{argmin}_\theta \sum_{(x,y)\in\mathscr{D}} (x(\theta) - y)^2$
3. **Planning using an optimistic learned model**

   $$\theta_{opt} \leftarrow \text{argmax}_{\theta\in\mathscr{B}} V_\theta(s_0), \quad \text{where } \mathscr{B} = \left\{ \theta \;\middle|\; \sum_{(x,y)\in\mathscr{D}} (x(\theta) - x(\hat{\theta}))^2 \leq \beta \right\}$$

   $$\hat{\pi} \leftarrow \text{argmax}_\pi V^\pi_{\theta_{opt}}(s_0), \qquad \hat{V} \leftarrow V^{\hat{\pi}}_{\theta_{opt}},$$

- Implement $\hat{\pi}$ as the policy in the next run
- The target value function $\hat{V}$ keeps changing as the agent learns

(Model-based RL with Value Targeted Regression. with Szepesvari, Yang et al. ICML, 2020)

Regret analysis of UCRL-VTR

Theorem: By choosing confidence levels $\{\beta_k\}$ appropriately, the VTR algorithm's regret satisfies with probability $1-\delta$ that

$$R_K = \sum_{k=1}^{K}\left(V^*(s_0^k) - V^{\hat{\pi}_k}(s_0^k)\right) \leq \tilde{O}(\sqrt{dim_{\mathscr{E}}(\mathscr{P}, 1/KH)\log \mathscr{N}(\mathscr{F}, 1/KH^2, \|\cdot\|_{1,\infty})KH^3})$$

where $dim_{\mathscr{E}}(\mathscr{P}, 1/KH)$ is the Eluder dimension (Russo & Van Roy 2013) of the function class

and $\mathscr{N}(\mathscr{P}, \alpha, \|\cdot\|_{1,\infty})$ denotes the covering number of $\mathscr{F}$ at a the scale α.

- A frequentist regret bound for model-based RL with a generic model family

Value-targeted regression is efficient for exploration in RL

A Special Case

- Linearly parametrized transition model $\mathscr{P} = \left\{ \exists \theta : P = \sum_{j=1}^{d} \theta_j P_j \right\}$

where each P_j is a base model

- In this case, UCRL-VTR has regret bound

$$R(T) \leq d\sqrt{H^3 T}$$

- Sparse linearly parametrized transition model$\mathscr{P} = \left\{ \exists \theta : P = \sum_{j=1}^{d} \theta_j P_j, \|\theta\|_0 \leq s \right\}$

- In this case, UCRL-VTR has regret bound

$$R(T) \leq \sqrt{H^3 dsT}$$

Summary: Upper Confidence Model-Based RL

Use prior knowledges about the model (ie, the model class) to derive appropriate RL algorithms.

Complexity of the model determines the regret.

- Deterministic continuous control:
  $\text{Regret}(K) = O(DLK)^{\frac{d}{d+1}} \cdot H$
- Linear model: $\text{Regret}(T) \leq C \cdot dH^2 \cdot \sqrt{T}$
- More general model:

$$R_K \leq \tilde{O}(\sqrt{dim_{\mathscr{E}}(\mathscr{F}, 1/KH) \log \mathscr{N}(\mathscr{F}, 1/KH^2, \|\cdot\|_{1,\infty})KH^3})$$

Thank you!