Reinforcement Learning In Feature Space
From Small Data

Mengdi Wang
Reinforcement learning achieves phenomenal empirical successes
What if the data/trial is limited and costly
How many samples are needed to learn an 90%-optimal policy?

How much regret to pay when learning to control on-the-fly?
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 it "tabular MDP" if there is no structural knowledge at all
What does a sample mean?

Samples are state-transition triplets (s, a, s')
Use empirical risk minimization for RL?

**Data:** Sample state-transition triplets \( \{(s, a, s')\} \)

**Step 1:** Estimate the transition model and compute empirical transition density

\[
\hat{P}(s' | s, s) = \frac{\text{# times } (s, a, s') \text{ appeared}}{\text{# times } (s, a) \text{ appeared}}
\]

**Step 2:** Solve the empirical MDP problem by dynamic programming

\[
\hat{\pi} = \arg\max_\pi \mathbb{E}_\hat{P} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t) \right]
\]

- **Hard to analyze:** tons of dependencies and nonlinearity [AMK13, AKY19]

- **Hard to implement:** it is a model-based approach (large memory overhead + computation bottleneck)

- **Which are model-free:** Q-learning, actor-critic, policy gradients
Prior efforts: algorithms and sample complexity results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sample Complexity</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phased Q-Learning</td>
<td>$\tilde{O}(C\frac{</td>
<td>S</td>
</tr>
<tr>
<td>Empirical QVI</td>
<td>$\tilde{O}(\frac{</td>
<td>S</td>
</tr>
<tr>
<td>Empirical QVI</td>
<td>$\tilde{O}(\frac{</td>
<td>S</td>
</tr>
<tr>
<td>Randomized Primal-Dual Method</td>
<td>$\tilde{O}(C\frac{</td>
<td>S</td>
</tr>
<tr>
<td>Sublinear Randomized Value Iteration</td>
<td>$\tilde{O}\left(\frac{</td>
<td>S</td>
</tr>
<tr>
<td>Sublinear Randomized QVI</td>
<td>$\tilde{O}\left(\frac{</td>
<td>S</td>
</tr>
</tbody>
</table>

$1/(1-\gamma)=1+\gamma+\gamma^2+...$ is the effective horizon
Lots of efforts about $1/(1-\gamma)$
Prior efforts: algorithms and sample complexity results

\[
\frac{1}{1 - \gamma} = 1 + \gamma + \gamma^2 + \ldots
\]

is the effective horizon.

Lots of efforts about \(\frac{1}{1 - \gamma}\)

(slide stolen from L Yang)
Complexity and Regret for Tabular MDP

• **Information-theoretical limit** (Azar et al. 2013): Any method finding an \( \varepsilon \)-optimal policy with probability 2/3 needs at least sample size

\[
\Omega \left( \frac{|SA|}{(1 - \gamma)^3 \varepsilon^2} \right)
\]

• **The optimal sampling-based algorithm** (with Sidford, Yang, Ye, 2018, Agarwal et al, 2019): With a generative model, finding \( \varepsilon \)-optimal policy with probability 1-\( \delta \) using sample size

\[
O \left( \frac{|SA|}{(1 - \gamma)^3 \varepsilon^2 \log \frac{1}{\delta}} \right)
\]

**Statistical complexity of RL (in this basic setting) is finally well understood**
S is way too big

Suppose states are vectors of dimension $d$

Vanilla discretization of state space gives $|S| = 2^d$

Size of policy space $= |A|^{|S|}$

Log of policy space size $= |S| \log(|A|) > 2^d$
Bellman equation is the optimality principal for MDP (in the average-reward case, where $\gamma=1$)

$$\ddot{v}^* + v^*(s) = \max_a \left\{ \sum_{s' \in S} P_a(s, s')v^*(s') + r_a(s) \right\}, \quad \forall s \in S$$

- The $\max$ operation applies to every state-action pair $\rightarrow$ nonlinearity + high dim

Bellman equation is equivalent to a bilinear saddle point problem (Wang 2017)

$$\min_v \max_{\mu \in \Delta} \left\{ L(v, \mu) = \sum_a \left( \mu_a^T((I - P_a)v + r_a) \right) \right\}$$

- Strong duality between value function and invariant measure
- SA x S linear program
State Feature Map

• Suppose we are given a state feature map

\[ \text{state} \mapsto [\phi_1(\text{state}), \ldots, \phi_N(\text{state})] \in \mathbb{R}^N \]

• Can we do better?

• Tetris can be solved well using 22 features and linear models

• Feature 1: Height of wall

• Feature 2: Number of holes
Representing value function using linear combination of features

- The value function of a policy is the expected cumulative reward as the initial state varies:

\[
V^\pi : S \rightarrow \mathbb{R}, \quad V^\pi(s) = \mathbb{E}^{\pi} \left[ \sum_{t=0}^{H} r(s_t, a_t) \mid s_0 = s \right]
\]

- Suppose that the high-dimensional value vector admits a linear model:

\[
V^\pi(s) \approx w_1 \phi_1(s) + \ldots + w_N \phi_N(s)
\]

- Value of \( V^\pi(s) \approx w_1 \times \text{Height of Wall} + w_2 \times \# \text{Holes} + \ldots \)

- Linear model for value function approximation has lots of limitations (later)
Reducing Bellman equation using features

\[
\bar{v}^* + v^*(s) = \max_a \left\{ \sum_{s' \in S} P_{a}(s, s')v^*(s') + r_{a}(s) \right\}, \forall s
\]

\[
\min \max_{\nu, \mu} \left\{ L(\nu, \mu) = \sum_{a} (\mu_{a}^{T}((I - P_{a})\nu + r_{a})) \right\}
\]

\[
\min_{\nu} \max_{\mu} L(\nu, \mu)
\]

\[
\min_{w} \max_{u} \sum_{a} (\Psi_{a}u_{a}^{T}\Phi^{T}(I - P_{a})\Phi\bar{v} + r_{a})
\]

Bellman eq:
- High-dim
- Nonlinear

Bellman saddle point:
- High-dim

\[
v(\cdot) \approx \sum_{i=1}^{r_{S}} w_{i}\phi_{i}(\cdot)
\]

\[
\mu(s, a) \approx \sum_{i=1}^{r_{S}} \sum_{j=1}^{r_{A}} u_{ij}\phi_{i}(s)\psi_{j}(a)
\]

Low-dim
- Convex-concave
- Strong duality
- Parametric
Sample complexity of RL with features

Suppose that good state and action features are known

- For average-reward RL, a primal-dual policy learning method finds the optimal policy using sample size (with YC, LL, 2018)

\[ \Theta \left( C \cdot \frac{|N_S N_A|}{\epsilon^2} \right) \]

where C is polynomial in mixing and ergodicity parameters

- Sample-Optimal Parametric Q-Learning for discounted RL (with LY, 2019)

\[ \Theta \left( \frac{|N_S N_A|}{\epsilon^2(1 - \gamma)^3} \right) \]

- Matching the information-theoretic minimax lower bound.

- Reduced S to Ns Na (# state-action features)
Learning to Control On-The-Fly

• Prior sample complexity analysis assumes a generative model:
  - One can draw transitions from any pre-specified state-action pair (enough exploration guaranteed)
  - Sample-optimal algorithms draw the same number of samples per state or per representative state (w. Sidford, Yang, Ye18, w. Yang Jia 19, Agarwal metal 19)

• In practice, we have to learn on-the-fly:
  - H-horizon stochastic control problem, starting at a fixed state $s_0$
  - A learning algorithm learns to control by repeatedly acting in the real world
  - It would act in realtime, observe state transitions, and adapt its control policy every episode
  - Impossible to visit all states frequently
Episodic Reinforcement Learning

• Regret of a learning algorithm $\mathcal{K}$

$$\text{Regret}_{\mathcal{K}}(T) = \mathbb{E}_{\mathcal{K}} \left[ \sum_{n=1}^{N} \left( V^*(s_0) - \sum_{h=1}^{H} r(s_{n,h}, a_{n,h}) \right) \right],$$

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

• Challenges:
  
  - Long-term effect of a single wrong decision
  
  - Data dependency: Almost all the transition samples are dependent
  
  - Exploration-exploitation tradeoff
  
  - More complicated than multi-arm bandit (naive reduction yields $A^S$ arms)
Hilbert space embedding of transition kernel

• Suppose we are given state-action feature maps

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

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

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

\[
P(s' \mid s, a) = \phi(s, a)^\top M^* \psi(s')
\]

• The decomposition structure is equivalent to using linear model for value function approximation with 0 Bellman error (w LY 2019)

• Low-dim assumption on \( V \) is closely related to low-dim assumption on \( P \)
The MatrixRL Algorithm

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

\[
\{(s_{n,h}, a_{n,h}), s_{n,h+1}\} \rightarrow \{\phi_{n,h}, \psi_{n,h}\} := \{\phi(s_{n,h}, a_{n,h}), \psi(s_{n,h+1})\}
\]

- 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 \(K_{\psi}\) 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

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

- **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

Regret Analysis

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

\[
\text{Regret}(T) \leq C \cdot dH^2 \cdot \sqrt{T},
\]

• 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 $H$*

The special case where $\Psi = I$

- A nonparametric model where $P$ cannot be encoded using a small # of parameters

\[
P(s' \mid s, a) = \phi(s, a)^T M^* \psi(s'), \quad \text{where} \quad \psi = I.
\]

- It only needs features to describe left principal space of $P$

- **In this case, MatrixRL has closed-form updates:**

\[
Q_{n,h}(s, a) = r(s, a) + \phi(s, a)^T M_n V_{n,h+1} + C\sqrt{\beta_n} \sqrt{\phi_{n,h}^T A_n^{-1} \phi_{n,h}}, \quad Q_{n,H} = 0
\]

- **Theorem** Under the embedding assumption and if $\psi = I$, the T-time-step regret of MatrixBandit is

\[
\text{Regret}(T) \leq C \cdot d^{3/2} H^2 \cdot \sqrt{T},
\]
From feature to kernel

Suppose that we are given a kernel function over the state-action space instead of explicit feature maps

\[ K((s, a), (s', a')) \]

- RL in kernel space? (Ormoneit & San 02, Ormoneit & Glynn 02, …)

- Kernel presents a very flexible framework for extrapolating information from seen states to unseen states

- We consider the generic assumption that the transition kernel belongs to the product Hilbert spaces spanned by these features:

\[ P \in \mathcal{H}_\phi \times \mathcal{H}_\psi \]
MatrixRL has a equivalent kernelization

**Algorithm 2 KernelMatrixRL: Reinforcement Learning with Kernels**

1. **Input:** An episodic MDP environment $M = (\mathcal{S}, \mathcal{A}, P, s_0, r, H)$, kernel functions $k_\phi, k_\psi$;
2. Total number of episodes $N$;
3. **Initialize:** empty reply buffer $B = \{\}$;
4. **for** episode $n = 1, 2, \ldots, N$ **do**
5. **for** $(s, a) \in \mathcal{S} \times \mathcal{A}$, let
   
   \[
   w_n(s, a) := \sqrt{k_\phi((s, a), (s, a)) - k_\phi^T_{n-1}(I + K_{\phi_{n-1}})^{-1}k_\phi_{n-1}(s, a)};
   \]
   
   \[
   x_n(s, a) := k_\psi^T_{n-1}(I + K_{\psi_{n-1}})^{-1}K_{\psi_{n-1}}(K_{\psi_{n-1}}K_{\psi_{n-1}}^T)^{-1}K_{\psi_{n}};
   \]
6. **for** stage $h = 1, 2, \ldots, H$ **do**
7. **for** state $s$, action $a$, $\forall h \in [H]$
8. **for** $(s, a, h) \in \mathcal{S} \times \mathcal{A}$
9. **for** stage $h = 1, 2, \ldots, H$ **do**
10. Let the current state be $s_{n,h}$;
11. Play action $a_{n,h} = \arg \max_{a \in \mathcal{A}} Q_{n,h}(s_{n,h}, a)$;
12. Record the next state $s_{n,h+1}$: $B \leftarrow B \cup \{(s_{n,h}, a_{n,h}, s_{n,h+1})\}$;
13. **end for**
14. **end for**

**Theorem**

**Regret**($T$) $\leq O\left(\|P\|_{\mathcal{H}_\phi \times \mathcal{H}_\psi} \cdot \log(T) \cdot \tilde{d} \cdot H^2 \cdot \sqrt{T}\right)$

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)
Pros and cons for using features for RL

• Deep connection to regression. Theoretical guarantee

• Easy to implement. Not many parameters to tune.

• Rely on good known features

• Pathological policy oscillation and chattering

• Not as rich as nonlinear models

• Not very surprising that good features can reduce the dimensionality of RL … Can we do well without known features?

• Many works in this domain, eg state representation learning (Lesort et al 08), latent state encoding (Du et al 19)
What could be good state features?

- Given a stationary Markov chain with transition operator $P$ and one-step reward function $r$, the average-reward difference-of-value function is given by

$$v = \lim_{T \to \infty} (r + Pr + P^2r + \cdots + P^Tr - (T\bar{r}) \cdot 1).$$

- Suppose that $P$ admits the decomposition

$$P = \Phi \tilde{P} \Psi^T$$

- Both the value $v$ and the invariant measure $\xi$ lie in low-dim spaces:

$$v \in \text{Span}(\Phi) \quad \xi \in \text{Span}(\Psi)$$

**Good value features** $\phi$ shall span the column space of $P$
Learning features automatically from time series data

- Consider a state-transition trajectory
  \[ X_1, X_2, \ldots, X_t, \ldots \]
- Spectral decomposition of the transition operator
  \[ \mathbb{P}(X_{t+1} \mid X_t) \approx \sum_{i}^{r} u_i(X_t)v_i(X_{t+1}) \]

**Markov features**

- \( u_i(\cdot) \)'s \( v_i(\cdot) \)'s are natural features for RL
- Reward-independent

**Estimate** \( x \to \Psi(x) \) from data to “preserve dynamics” (approximate leading singular functions of \( \mathbb{P} \))

\[
\max_{\Psi: \mathbb{X} \to \mathbb{R}^r, \Psi_j \in \mathcal{H}} \text{Tr} \left( \int \Psi(x)p(x, y)\Psi(y)^T \, dx \, dy \right)
\]

- Statistical error bounds and information-theoretic limits proved (w AZ 2018, w YD, KZ, 2018, w YS, YD, GH, 2019)
Kernelized state embedding from random features

**Data:** A high-dimensional time series and a kernel space with $K$

$$X_1, X_2, \ldots, X_t, \ldots$$  
where $X_t \in \mathbb{R}^d$

**Solution:**

1. Open up the kernel space and approximate with random features

$$K(x, y) \approx \phi(x)^\top \phi(y) \quad \quad \phi(\cdot) = [\phi_1(\cdot), \ldots, \phi_N(\cdot)]^\top$$

2. Estimate a projection matrix of the transition kernel onto the $K$ space

$$\hat{Q} = \frac{1}{T} \sum_{t=1}^T \phi(X_t)\phi(X_{t+1})^\top$$

3. Find the best rank-$r$ approximation

$$\hat{Q} = \hat{U}\Lambda\hat{V}^\top, \quad \hat{Q}_r = \hat{U}_r\Lambda_r\hat{V}_r^\top$$

**Output:** Low-dim state embedding (a kernelized diffusion map)

$$X \mapsto P(\cdot | X) \mapsto \hat{\Psi}(X) := \phi(X)^\top \hat{U}_r \in \mathbb{R}^r$$

- Minimax-optimal error bounds for recovering $P$ proved in (w Sun, Duan, Gong 2019)
Some theory

- The diffusion distance between two states is
  \[ \text{dist}(x, y) = \| p(\cdot|x) - p(\cdot|y) \| \]

- Kernelized state embedding preserves the diffusion distance up to error
  \[ | \text{dist}(x, y) - \| \hat{\Psi}(x) - \hat{\Psi}(x) \| | \leq O \left( \sqrt{\frac{rkt_{\text{mix}}}{n}} \right), \quad \forall x, y \]

where \( r \) is rank, \( k \) is MC’s condition number, \( n \) is the length of trajectory.
Finding Metastable State Clusters

• We want to find a partition of the state space such that that states within the same set shares similar future paths

\[
\min_{\Omega_1, \ldots, \Omega_m, q_1, \ldots, q_m} \min_{\Omega_i} \sum_{i=1}^{m} \int_{\Omega_i} \pi(x) \| p(\cdot | x) - q_i(\cdot) \|_{L_2}^2 dx,
\]

• If the MC is reversible, the problem finds the optimal metastable partition [E 2008]

\[
(A_1^*, \ldots, A_m^*) = \text{argmax}_{A_1, \ldots, A_m} \sum_{k=1}^{m} \frac{p(A_k | A_k)}{A_1 \ldots A_m}
\]

• **Solution:** 1. Estimate state embedding; 2. Solve

\[
\min_{(\Omega_1, \ldots, \Omega_m)} \min_{s_1, \ldots, s_k \in \mathbb{R}'} \sum_{i=1}^{m} \sum_{i \in [N]} \| \hat{\Psi}(x_i) - s_i \|_2^2 dx
\]
Example: stochastic diffusion process

Potential Function

True Invariant Measure $p(x_1, x_2)$
Metastable clusters learned from $P^t$

Learning metastable sets from state trajectories
Example: State Trajectories of Demon Attack

(a) Before Embedding

(b) After Embedding

Visualization of game states before and after embedding in t-SNE plots.
Game states that are close after embedding

About to score; both moving to the left

New demons appearing

Waiting for new targets; moving to center from opposite ends

State embedding identifies states as similar in low-dim space if they share similar future paths
Collaborators

Aaron Sidford
Yinyu Ye
Anru Zhang
Lin Yang
Tracy Ke

Yichen Chen
Yaqi Duan
Hao Gong
Yifan Sun
Zeyu Jia

Thank you!