Harnessing Structures for Value-Based Planning and Reinforcement Learning

Yuzhe Yang

Guo Zhang, Zhi Xu, Dina Katabi
New Planning and Deep RL Framework that exploits the “global structure” in tasks
Motivation
Motivation
Motivation
Motivation
Motivation

Can structure help?
What Structure?
What Structure?

- Focus on $Q$-value
What Structure?

- Focus on \textit{Q-value}

![Diagram showing states and actions]

- States
- Actions

$Q(s, a)$
What Structure?

- Focus on *Q*-value

```
    s1
    s2
    ...
```

```
    a1    a2    ...
```

- Actions
- States

Global structure: low-rank
Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration
Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration

\[
Q^{(t+1)}(s, a) = \sum_{s' \in S} P(s'|s, a) \left[ r(s, a) + \gamma \max_{a' \in A} Q^{(t)}(s', a') \right], \quad \forall (s, a) \in S \times A,
\]
Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration

\[
\text{(Approx. rank: first } k \text{ SVs capture > 99\% variance, i.e., } \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_j \sigma_j^2} \geq 0.99)\]

Warm-up: Toy Example

- Randomly sampled deterministic MDP and Q-value iteration

Exploit the structure during the learning process? Enforce/regularize such a structure throughout the iterations?
How Do We Exploit the Structure?
Idea?

Q matrix
Adversarial Examples

<table>
<thead>
<tr>
<th>Q matrix</th>
<th>Sample</th>
<th>Reconstruct</th>
</tr>
</thead>
</table>

Idea?
Idea: Compute few and reconstruct the rest
Idea: Compute few and reconstruct the rest

Q matrix → Sample → [pattern] → [question mark] → Reconstruct
Idea: Compute few and reconstruct the rest

Low-rank Matrix Estimation (ME)
Adversarial Examples

**Idea:** Compute few and reconstruct the rest

**Q matrix** → **Sample** → **ME** → **Reconstruct**

**Low-rank Matrix Estimation (ME)**

\[
\min_{\hat{M} \in \mathbb{R}^{n \times m}} \frac{1}{2} \sum_{(i,j) \in \Omega} \left( \hat{M}_{ij} - X_{ij} \right)^2 + \lambda \|\hat{M}\|_*
\]
Idea: Compute few and reconstruct the rest

Low-rank Matrix Estimation (ME)

\[
\min_{\hat{M} \in \mathbb{R}^{n \times m}} \frac{1}{2} \sum_{(i,j) \in \Omega} \left( \hat{M}_{ij} - X_{ij} \right)^2 + \lambda \| \hat{M} \|_*
\]

ME as a principled reconstruction oracle to exploit the low-rank structure
1. Structured Value-based Planning (SVP)
2. Structured Value-based Deep RL (SV-RL)
1. Structured Value-based Planning (SVP)

2. Structured Value-based Deep RL (SV-RL)
SVP: Structured Value-based Planning

State-action value function: $Q^{(t)}$

Reconstructed value function: $Q^{(t+1)}$

Next iteration
SVP: Structured Value-based Planning
SVP: Structured Value-based Planning

Next iteration

State-action value function

$Q^{(t)}$

Sample

Compute

Incomplete observation

$\hat{Q}_\Omega$

Reconstruct

Reconstructed value function

$Q^{(t+1)}$

$$\hat{Q}(s, a) \leftarrow \sum_{s'} P(s'|s, a) \left( r(s, a) + \gamma \max_{a'} Q^{(t)}(s', a') \right), \quad \forall (s, a) \in \Omega.$$
SVP: Structured Value-based Planning

\[ Q^{(t+1)} = \text{ME} \left( \{ \hat{Q}(s, a) \} \right) \]
Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000
Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000
- Verify low-rank structure:
Stochastic Control: Inverted Pendulum

- Discretization: Q matrix = 2500 * 1000
- Verify low-rank structure:

Approximate rank of Q* = 7

Desired low-rank property for SVP
Stochastic Control: Inverted Pendulum

- Policy visualization:
Stochastic Control: Inverted Pendulum

- Policy visualization:
Stochastic Control: Inverted Pendulum

- Policy visualization:
Stochastic Control: Inverted Pendulum

- Policy visualization:

Success of SVP: a small amount of observations is sufficient!
1. Structured Value-based Planning (SVP)

2. Structured Value-based Deep RL (SV-RL)
Extend to Deep RL?

- Intuition and development of SVP
Extend to Deep RL?

- Intuition and development of SVP
- Naive extension? Issues?

With images as states...
Idea: Batch of States as Proxy

- Intuition and development of SVP
- Naive extension? Issues?

With images as states...
Evidence of low-rank structures

Idea: Batch of States as Proxy

- Intuition and development of SVP
- Naive extension? Issues?

With images as states...
Idea: Batch of States as Proxy

- Intuition and development of SVP
- Naive extension? Issues?

Natural to understand the rank of batches of states for the learned Q value
Evidence of low-rank structures

- Batch size = 32; Sample 10,000 sub-matrices from DQN
Evidence of low-rank structures

- Batch size = 32; Sample 10,000 sub-matrices from DQN

Structure widely exists: Majority of games (> 40)!
Evidence of low-rank structures

- Batch size = 32; Sample 10,000 sub-matrices from DQN

Structure widely exists: Majority of games (> 40)!

Harness the structure within the batch of states during the learning process
SV-RL: Structured Value-based Deep RL
SV-RL: Structured Value-based Deep RL

- Original value-based RL
SV-RL: Structured Value-based Deep RL

- **Original value-based RL**

  \[ y(i) = r(i) + \gamma \max_{a'} Q(s(i+1), a') \]

  \[ \sum_{i=1}^{g} (y(i) - Q(s(i), a(i); \theta))^2 \]

- **SV-RL**

  \[ y(i) = r(i) + \gamma \max_{a'} Q^+(s(i+1), a') \]

  \[ \sum_{i=1}^{g} (y(i) - Q(s(i), a(i); \theta))^2 \]
SV-RL: Structured Value-based Deep RL

- Original value-based RL

\[ \hat{Q}(S_B, A) \]

\[ y^{(i)} = r_i^{(i)} + \gamma \max_{a'} \hat{Q}(s_{t+1}^{(i)}, a') \]

\[ \sum_{i=1}^{g} \left( y^{(i)} - Q(s_i^{(i)}, a_i^{(i)}; \theta) \right)^2 \]

- SV-RL

\[ \hat{Q}(S_B, A) \]

\[ \hat{Q}_{\Omega} \]

\[ Q^+(S_B, A) \]

\[ y^{(i)} = r_i^{(i)} + \gamma \max_{a'} Q^+(s_{t+1}^{(i)}, a') \]

\[ \sum_{i=1}^{g} \left( y^{(i)} - Q(s_i^{(i)}, a_i^{(i)}; \theta) \right)^2 \]
Empirical Evaluation: Atari

- Apply SV-RL on three representative value-base deep RL
Consistent Benefits for “Structured” Games
Empirical Evaluation: Atari

- Apply SV-RL on three representative value-base deep RL

- Consistent benefits for “structured” games:
  1. games that possess low-rank structure benefit from SV-RL
  2. consistent improvements across different RL techniques
  3. more games - see paper
Empirical Evaluation: Atari

Further observations? Performance gains vary.
# Diagnose & Interpret Performance

<table>
<thead>
<tr>
<th></th>
<th>Frostbite</th>
<th>Krull</th>
<th>Alien</th>
<th>Seaquest</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-RL</td>
<td>Better</td>
<td>Better</td>
<td>Slightly Better</td>
<td>Worse</td>
</tr>
</tbody>
</table>
Diagnose & Interpret Performance

(a) Frostbite (better)  (b) Krull (better)  (c) Alien (slightly better)  (d) Seaquest (worse)
Diagnose & Interpret Performance

(a) Frostbite (better)  (b) Krull (better)  (c) Alien (slightly better)  (d) Seaquest (worse)
Diagnose & Interpret Performance

Consistent results on rank vs. improvement across games & RL methods

(a) Frostbite (better)  (b) Krull (better)  (c) Alien (slightly better)  (d) Seaquest (worse)
Diagnose & Interpret Performance

<table>
<thead>
<tr>
<th></th>
<th>Frostbite</th>
<th>Krull</th>
<th>Alien</th>
<th>Seaquest</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-RL</td>
<td>Better</td>
<td>Better</td>
<td>Slightly Better</td>
<td>Worse</td>
</tr>
<tr>
<td>Rank</td>
<td>~2</td>
<td>~2</td>
<td>~5</td>
<td>~10</td>
</tr>
</tbody>
</table>

- Consistent interpretations:
Diagnose & Interpret Performance

<table>
<thead>
<tr>
<th></th>
<th>Frostbite</th>
<th>Krull</th>
<th>Alien</th>
<th>Seaquest</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV-RL</td>
<td>Better</td>
<td>Better</td>
<td>Slightly Better</td>
<td>Worse</td>
</tr>
<tr>
<td>Rank</td>
<td>~2</td>
<td>~2</td>
<td>~5</td>
<td>~10</td>
</tr>
</tbody>
</table>

- Consistent interpretations:
  
  If the learned Q function contains low-rank structure

SV-RL is able to exploit the structure!
Summary of Contributions

- Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning
Summary of Contributions

- Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning
- Demonstrate the effectiveness of our approach on classical stochastic control tasks
Summary of Contributions

● Propose a generic framework that exploits the low-rank structures, for planning and deep reinforcement learning

● Demonstrate the effectiveness of our approach on classical stochastic control tasks

● Extend our scheme to deep RL, which is naturally applicable for value-based techniques, and obtain consistent improvements across a variety of methods
Poster Sessions (New York time):

Apr. 28th: 12 AM - 2 AM
Apr. 29th: 12 PM - 2 PM

Source Code: https://github.com/YyzHarry/SV-RL
Project Page: http://svrl.csail.mit.edu