MOReL: Model-Based Offline Reinforcement Learning

Rahul Kidambi
Cornell University

Aravind Rajeswaran
University of Washington

Praneeth Netrapalli
Microsoft Research India

Thorsten Joachims
Cornell University
(Deep) Reinforcement Learning

Many application domains

▪ Robotics/Bio-mechanics
  [Lillicrap et al. `16, OpenAI `18]
▪ Ranking/Recommender Systems
  [Covington et al. `18, Chen et al. `18]
▪ Games
  [Mnih et al. `15, Silver et al 2016]
▪ Graphics (The Incredibles!!)
  [Mordatch et al. `15, Peng et al. `18]
MDP: Preliminaries

\[ \mathcal{M} \equiv (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho_0, T) \]

State - \( \mathcal{S} \subset \mathbb{R}^d \) - grid position
Action - \( \mathcal{A} \subset \mathbb{R}^{d'} \) (up, down, left, right)
Transition - \( \mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \) – next grid position
Reward \( R : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \) – Distance to target
Start state distribution \( \rho_0 \in \mathcal{S} \) – where you begin.

Policy - \( \pi : \mathcal{S} \to \mathcal{A} \)

Goal: Compute

\[ \pi^* = \arg\max_{\pi \in \Pi} V(\pi) = \mathbb{E}[r_0 + \gamma \cdot r_1 + \cdots \gamma^{T-1} \cdot r_{T-1} | \pi] \]
Approaches to solve RL

Model Based

- (Generalized) Policy Iteration
- Policy Gradient + Critic (value)
- (Asynchronous) Advantage Actor-Critic (A2C/A3C)
- Soft Actor-Critic (SAC)

Value Based

- (Fitted) Value Iteration
- SARSA
- (Deep) Q-learning

Actor Critic

Policy Based

- Policy Gradient
- NPG/TRPO/PPO etc.
- Zeroth order optimization

Model Free

Adapted from David Silver’s slides
Plan($\pi_0, \mathcal{M}, K$):

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

Paths $\leftarrow$ Interact($\pi, \mathcal{M}$) $K$ times

$\nabla V(\pi) = \text{GetPolicyGradient}($Paths$)$

$\pi \leftarrow \pi + \gamma \cdot \nabla V(\pi)$

Return $\pi$

Positives:

-- Monotonic progress (in theory)
-- Global optimality [Agarwal et al. '19]

Issues:

-- Sample efficiency
-- Non-monotonicity in practice
(baselines, func. approx., noisy gradients)
(Online) Policy Gradient

Plan($\pi_0, \mathcal{M}, K$):

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

Paths $\leftarrow$ Interact($\pi, \mathcal{M}$) $K$ times

$\nabla V(\pi) = \text{GetPolicyGradient}(\text{Paths})$

$\pi \leftarrow \pi + \gamma \cdot \nabla V(\pi)$

Return $\pi$

Positives:
-- Monotonic progress (in theory)
-- Global optimality [Agarwal et al. '19]

Issues:
-- Sample efficiency
-- Non-monotonicity in practice (baselines, func. approx., noisy gradients)

PG [Sutton et al. 2000], NPG [Kakade '01], TRPO [Schulman et al. '15], PPO [Schulman et al. '17]
Offline Reinforcement Learning

OfflineRL($\pi_b, N, M$):
For $i = 1 \cdots N$:
    Traj[$i$] = Interact($\pi_b, M$)
$\pi_{out} \leftarrow$ OfflinePlanningAlgorithm(Traj)
Return $\pi_{out}$

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. How to reuse the dataset?
2. Exploration?

Online Reinforcement Learning
Update dataset after every (noisy) policy update

Offline Planning Algorithm

Online Planning Algorithm

Offline Reinforcement Learning

OfflineRL($\pi_b, N, \mathcal{M}$):
For $i = 1 \cdots N$:
\[
\text{Traj}[i] = \text{Interact}(\pi_b, \mathcal{M})
\]
$\pi_{out} \leftarrow \text{OfflinePlanningAlgorithm}($\text{Traj}$)$
Return $\pi_{out}$

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. How to reuse the dataset?
2. Exploration?

Offline RL - strictly harder
1. Sequential Decision Making
2. Partial (Bandit) Feedback
Challenges In Offline RL

**OfflineRL($\pi_b, N, M$):**

For $i = 1 \cdots N$:

$\text{Traj}[i] = \text{Interact}(\pi_b, M)$

$\pi_{\text{out}} \leftarrow \text{OfflinePlanningAlgorithm}(\text{Traj})$

Return $\pi_{\text{out}}$

**Advantages:**
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

**Challenges:**
1. Exploration?
2. How to reuse the dataset?
3. **Distribution Shift**
4. Impact of the logging policy?

**Distribution shift issues**

- No understanding of how the world behaves

- $\pi_b$’s state visitation
- $\pi_{\text{out}}$’s state visitation

- States in the dataset
- Unknown states

Failure owing to Distribution shift.

Praneeth Netrapalli (MSR India) - Model-based Offline Reinforcement Learning
Challenges In Offline RL

OfflineRL($\pi_b, N, M$):
For $i = 1 \cdots N$:

\[
\text{Traj}[i] = \text{Interact}(\pi_b, M)
\]

$\pi_{out} \leftarrow \text{OfflinePlanningAlgorithm}(\text{Traj})$

Return $\pi_{out}$

Advantages:
- Work with already collected datasets
- Deploy after checks
- Avoid non-monotonic behavior

Challenges:
1. Exploration?
2. How to reuse the dataset?
3. Handling distribution shift
4. Impact of the logging policy?

Impact of Logging Policy

Behavior policy $\pi_b^{(1)}$

Behavior policy $\pi_b^{(2)}$

Both policies equally good!!

States in the dataset
Unknown states
Final policy’s state visitation

Both final policies are optimal.
Prior Work
Direction 1: Off-Policy Actor-Critic

• Q-learning [Watkins `89]
  • Good performance in tabular settings [Watkins & Dayan 92]
  • Challenging with function approx. [Baird ‘95, Sutton ‘95]
    • Setting targets for Bellman error minimization
    • Replay buffer dictates performance.

• Actor-Critic methods
  • DDPG [Silver et al`14], DQN [Mnih et al`15], SAC [Haarnoja et al `18]:
  • Similar issues [van Hasselt et al`18, Fujimoto et al`18, Fu et al`19]

• Many recent works [Fujimoto et al`18b, Kumar et al`19, Jaques et al`19, Wu et al`19]
  • Design choices vary on a per-task basis.
Direction 2: Off-policy Policy Gradients

• Use importance weighted policy gradient
  • Importance weights: exponential in horizon.
    • Fix: use state-distribution ratios [Liu et al.`18].
  • Maintain ESS is reasonably large [Fakoor et al. `18].

• Liu et al. (2018)
  • Model-free planning using pessimistic MDP approaches.
  • Yet to be examined for continuous control tasks.

• Many such issues examined in contextual bandit problems
  • For e.g. [Swaminathan & Joachims `15, Joachims et al. `18, Ma et al. `19] and many others.
Our Framework: MOReL

MOReL is an offline RL framework: relies on optimization + supervised learning.

One-line description: plan using a model-based pessimistic MDP construction. Achieves state-of-the-art on several benchmarks.

Features of MOReL

Model-based approach:
- Effective sample reuse
- Unlocks generalization!
- On-policy samples

Caveat: No model is globally accurate

Planning based on a “pessimistic” MDP framework:
- Effective way to handle distribution shift.
- Use model where it is accurate!

Similar algorithm proposed independently by Yu et al. `20
Our Contributions

• MOReL: model-based offline RL
  • Ross and Bagnell (2012) analyzed naïve model-based offline RL

• Pessimistic MDP construction
  • State-action pairs → known/unknown

• Planning on the pessimistic MDP
  • Policy discouraged from visiting unknown states

• MOReL - minimax optimal for offline RL
  • Model score approx. lower bounds true score

• Experiments on deep RL benchmarks
  • Continuous state/action spaces with function approx.
**Review: Model-Based RL**

- Replace $\mathcal{M} \equiv (S, A, T, R, \rho_0, T_h)$ with $\hat{\mathcal{M}} \equiv (S, A, \hat{T}, R, \rho_0, T_h)$.
- Where, $\hat{T} : S \times A \rightarrow S$ is estimated through MLE on the logged dataset.
- Plan using $\hat{\mathcal{M}}$.

**Advantages:**
- Sample efficiency & reuse.

---

Praneeth Netrapalli (MSR India) - Model-based Offline Reinforcement Learning
Review: Model-Based RL

- Replace $\mathcal{M} \equiv (S, \mathcal{A}, T, R, \rho_0, T_h)$ with $\hat{\mathcal{M}} \equiv (S, \mathcal{A}, \hat{T}, R, \rho_0, T_h)$.
- Where, $\hat{T} : S \times \mathcal{A} \rightarrow S$ is estimated through MLE on the logged dataset.
- Plan using $\hat{\mathcal{M}}$.

Plan ($\pi_0, \hat{\mathcal{M}}, K$):

1. $\pi \leftarrow \pi_0$
2. REPEAT Till CONVERGENCE:
   - Paths $\leftarrow$ Interact($\pi, \hat{\mathcal{M}}$) $K$ times
   - $\text{PG}(\pi) = \text{GetPolicyGradient}(\text{Paths})$
   - $\pi \leftarrow \pi_n + \gamma \cdot \text{PG}(\pi)$
3. Return $\pi$

Advantages:
- Sample efficiency & reuse.
Review: Model-Based RL

- Replace $\mathcal{M} \equiv (S, A, T, R, \rho_0, T_h)$ with $\hat{\mathcal{M}} \equiv (S, A, \hat{T}, R, \rho_0, T_h)$.
- Where, $\hat{T} : S \times A \to S$ is estimated through MLE on the logged dataset.
- Plan using $\hat{\mathcal{M}}$.

Advantages:
- Sample efficiency & reuse.

Issues:
- Short horizon bias (compounding errors) [Levine et al. 2020]
- Continuous state space => model not accurate everywhere.

Plan($\pi_0, \hat{\mathcal{M}}, K$):

$\pi \leftarrow \pi_0$

REPEAT Till CONVERGENCE:

Paths $\leftarrow$ Interact($\pi, \hat{\mathcal{M}}$) $K$ times

Return $\pi$

Clip trajectories to fixed length - Shorter horizon.

Suppose $|T - \hat{T}| < \alpha$ in t.v., Trajectory length $\approx O(1/\alpha)$

Large

Dynamics Model Errors

Low

Praneeth Netrapalli (MSR India) - Model-Based Offline Reinforcement Learning
Pessimistic MDP Formulation

Instead of $\hat{M} \equiv (S, A, T, R, \rho_0, T)$ consider

$\hat{M}_p \equiv (S_k \cup \{\text{UNK}\}, A, \hat{T}, R, \rho_0, T),$

- $S_k$ -- set of known states [Kakade et al. `03]

Defining the UNK state

Ones that aren’t observed in the dataset.

**UNK**: absorbing with large negative reward.

---

Advantages:
- Planner restricted to states where model is accurate.
- **Dynamic** pruning of trajectory based on when it encounters UNK.

Question: How do we truncate in practice?
Deciding UNK states

- Require an oracle $O$:
  
  $$O(s, \text{dataset}) = \begin{cases} 
  \text{True} & \text{if } s \in S_k \\
  \text{False} & \text{if } s \in \text{UNK} 
  \end{cases}$$

- Examples:
  - Bayesian approaches: if $\hat{T}(s, a)$ has large uncertainties, $O(s, \text{dataset}) = \text{True}$
  - (Random) Ensembles: Train multiple $\{\hat{T}_i\}$.

  If these disagree on $(s, a)$, then, $O(s, \text{dataset}) = \text{True}$

- We use the latter, but the framework is flexible.
**Framework: MOReL**

MOReL(Dataset, $\mathcal{M}$):
- Learn an ensemble of generative models:
  \[ O \equiv \{ \mathcal{T}_i \} \leftarrow \text{MLE} \text{(Dataset, seed = } i) \]
- Define the model based pessimistic MDP:
  \[ \overline{\mathcal{M}}_p = \{ S_k \cup \{ \text{UNK} \}, A, \{ \mathcal{T}_i \}, R, \rho_0 = \text{dataset, } T_h \} \]
- $\pi_{\text{new}} \leftarrow \text{Plan}(\pi_0, \overline{\mathcal{M}}_p, K, O)$

Plan($\pi_0, \mathcal{M}, K, O$):
\[
\pi \leftarrow \pi_0 \\
\text{REPEAT Till CONVERGENCE:} \\
\text{Paths} \leftarrow \text{Interact}(\pi, \mathcal{M}, O) \text{ } K - \text{times} \\
\text{PG}(\pi) = \text{GetPolicyGradient}(\text{Paths}) \\
\pi \leftarrow \pi + \gamma \cdot \text{PG}(\pi)
\]

Return $\pi$

Interact($\pi, \mathcal{M}, O$):
\[
\begin{align*}
\text{(start state)} \\
\text{S}_0 \sim \rho_0 \\
\text{For } t = 0, 1, \ldots, T_h - 1: \\
\text{If } O(s_t) = \text{True} \\
\text{\quad } a_t \sim \pi(\cdot | s_t) \\
\text{\quad } r_t \leftarrow r(s_t, a_t) \\
\text{\quad } s_{t+1} \sim T(\cdot | s_t, a_t) \\
\text{Else:} \\
\text{\quad } r_t = r_{\text{min}} \\
\text{\quad } \text{Break} \end{align*}
\]

Return Traj = $\{(s_j, a_j, r_j, s_{j+1})\}_{j=0}^{T}$
Theoretical Guarantees

Let $J_{\rho_0}(\pi, \mathcal{M})$ be the expected performance of policy $\pi$ on the MDP $\mathcal{M}$, i.e.,

$$J_{\rho_0}(\pi, \mathcal{M}) = E_{s_0 \sim \rho_0, \mathcal{M}, \pi}[V(s_0)]$$

Then,

**Theorem** (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):

The value of any policy $\pi$ the model-based pessimistic MDP $\hat{\mathcal{M}}_p$ is related to $\mathcal{M}$ as follows:

$$J_{\rho_0}(\pi, \hat{\mathcal{M}}_p) \geq J_{\rho_0}(\pi, \mathcal{M}) - 2\alpha \frac{R_{\max}}{(1 - \gamma)^2} - 2 \frac{R_{\max}}{1 - \gamma} \cdot E \left[ \gamma^{T_{\hat{u}}} \right]$$

$$J_{\rho_0}(\pi, \hat{\mathcal{M}}_p) \leq J_{\rho_0}(\pi, \mathcal{M}) + 2\alpha \frac{R_{\max}}{(1 - \gamma)^2}$$

Where, $\alpha$ is the total variation distance between the true/learnt dynamics on the known states, and $T_{\hat{u}}^{\pi}$ denotes the hitting time of $\pi$ on unknown states in $\mathcal{M}$.
Sub-optimality bound of MOReL

**Corollary** (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):

Suppose the PLANNER offers an $\epsilon_\pi$-sub-optimal policy on the pessimistic MDP $\widehat{\mathcal{M}}_p$. Then,

$$J_{\rho_0}(\pi^*, \mathcal{M}) - J_{\rho_0}(\pi, \mathcal{M}) \leq 4 \frac{R_{\text{max}}}{(1 - \gamma)^2} \cdot \alpha - 2 \frac{R_{\text{max}}}{(1 - \gamma)} \cdot \mathbb{E}[\gamma^{T_{\pi}^\mathcal{U}}] + \epsilon_\pi$$

Where, $\alpha$ is the total variation distance between the true/learnt dynamics on the known states, and $T_{\pi}^{\mathcal{U}}$ denotes the hitting time of $\pi$ on unknown states in $\mathcal{M}$.

**Proposition** (Kidambi, Rajeswaran, Netrapalli, Joachims 2020):

MOReL is minimax optimal for offline RL.
Experimental Analysis

- 4 continuous control tasks from MuJoCo:
  - Hopper
  - HalfCheetah
  - Ant
  - Walker.
- Continuous state/action spaces.
- Record a total of 1M transitions.
Dataset collected with a partially-trained policy

- Unroll 1000 trajectories from a sub-optimal policy.

- We compare the best of 12 recent algorithms against our approach.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Behavior Policy Value</th>
<th>Best of 12 baselines</th>
<th>Our Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hopper-v2</td>
<td>1000</td>
<td>2774</td>
<td>3642 ± 54</td>
</tr>
<tr>
<td>Half Cheetah-v2</td>
<td>4000</td>
<td>6207</td>
<td>6028 ± 192</td>
</tr>
<tr>
<td>Walker2d-v2</td>
<td>1000</td>
<td>2907</td>
<td>3709 ± 159</td>
</tr>
<tr>
<td>Ant-v2</td>
<td>1000</td>
<td>2839</td>
<td>3663 ± 247</td>
</tr>
</tbody>
</table>
Benchmarking with Noisy Experience Datasets

Hopper-v2

- Value: 500, 1500, 2500, 3500
- Behavior policy, Best Baseline, Our Algorithm

HalfCheetah-v2

- Value: 3000, 4000, 5000, 6000, 7000
- Behavior policy, Best Baseline, Our Algorithm

Walker2d-v2

- Value: 500, 1500, 2500, 3500
- Behavior policy, Best Baseline, Our Algorithm

Ant-v2

- Value: 26
- Behavior policy, Best Baseline, Our Algorithm
Future Work

- What happens when model-learning is hard?
  - For e.g., video, NLP (conversational AI), etc..

- Reward centric dynamics/representation learning [Du et al. 2020]

- Instance optimal bounds for offline RL

- Related problems: model learning, uncertainty quantification and planning in online RL
Conclusions

• Proposed a framework for model-based offline RL.

• Relies on supervised learning + optimization.

• Very effective
  • Learning dynamics model is feasible
  • State representation is sufficient

• Future work
  • Model selection, exploration etc.