ROIL: Robust Offline Imitation Learning

Gersi Doko[1], Guang Yang[2], Daniel S. Brown[2] Marek Petrik[1]

[1]University of New Hampshire, [2]University of Utah

Summary

Motivation

- Compute policies that are robust to parameter uncertainty is very important in many domains, like health care, inventory control or finance.
- Seek policies that maximize the expected return over a distribution of MDP models

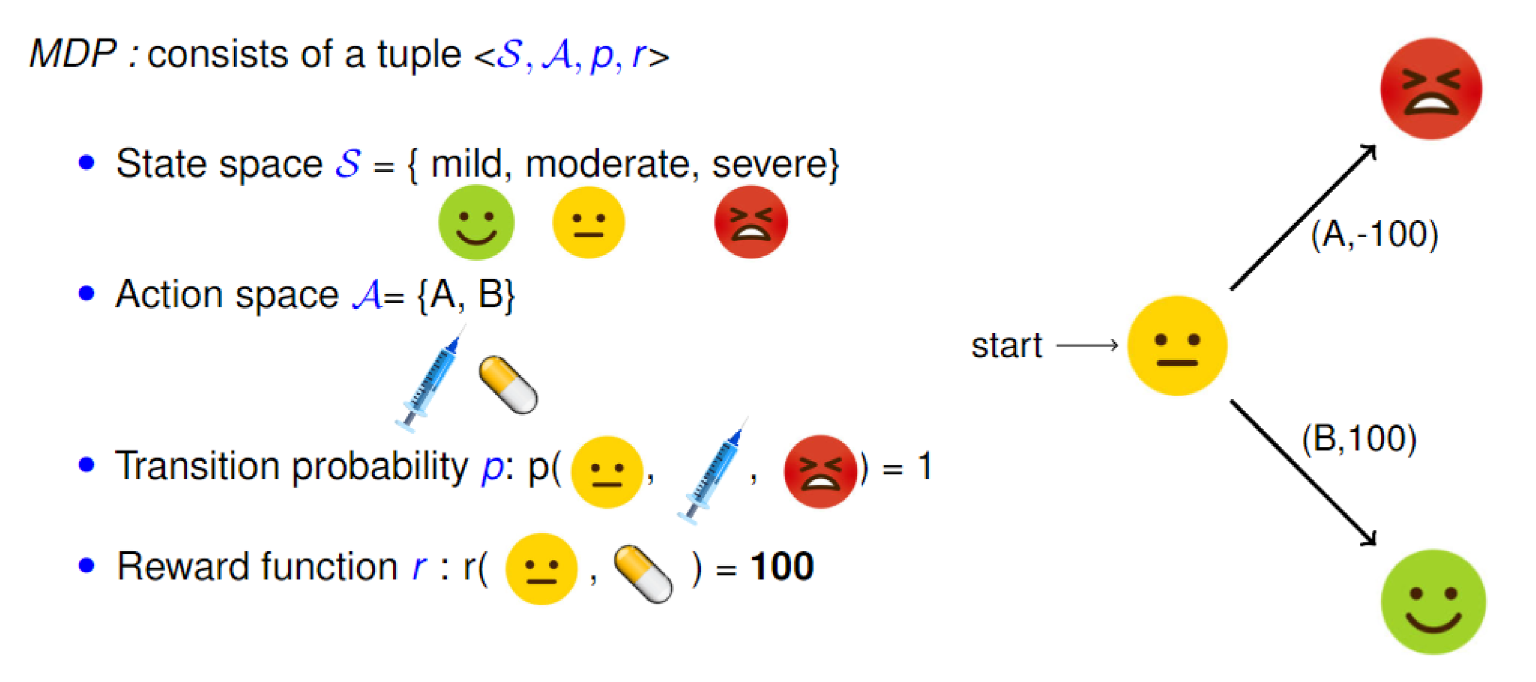
Limitations of existing methods

- Reliance on $\hat{u}_e$ leads to covariate shift for off-policy datasets.
- Inability to specify reliance on $\hat{u}_e$.
- No guarantees of policy convergence to u_e.

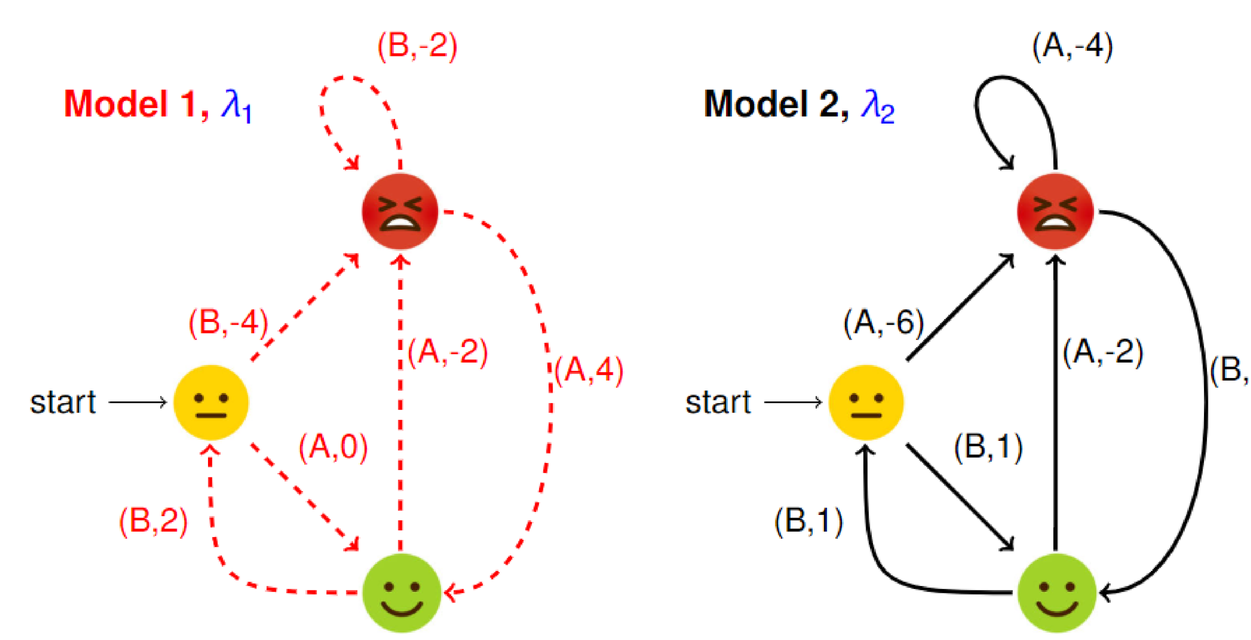
Our contributions

- New algorithm for robust offline imitation learning.
- Guaranteed convergence to the optimal policy for tabular domains.
- Flexibility to define the reliance on $\hat{u}_e$.

Markov Decision Process (MDP)



Multi-model Markov Decision Processes (MMDPs)



$\mathcal{M} = \{1, 2\}, \lambda = \{\lambda_1, \lambda_2\}, \mathcal{S} = \{mild, moderate, severe\}, \mathcal{A} = \{A,\ B\}$

- Mean return across the uncertain true models

$$\rho(\pi) = \mathbb{E}^{\lambda}\left[\mathbb{E}^{\pi, p^{\tilde{m}}, \mu}\left[\sum_{t=1}^{T} r_t^{\tilde{m}}(\tilde{s}_t, \tilde{a}_t) \mid \tilde{m}\right]\right] \qquad (1).$$

- Optimal policy ρ^*

$$\rho^* = \max_{\pi \in \Pi} \rho(\pi).$$

Prior Work: Weight-Select-Update (WSU)

WSU Approximation Algorithm

Input: MMDPs, Model weights λ

Output: $\pi = (\pi_1, \ldots, \pi_T)$

1. Initialize $v^{\pi}_{T+1,m}(s_{T+1}) = 0, \forall m \in \mathcal{M}$
2. For $t = T, T-1, \ldots, 1$ do
3. $\pi_t(s_t) \in \arg\max_{a \in \mathcal{A}} \sum_{m \in \mathcal{M}} \lambda_m \cdot q^{\pi}_{t,m}(s_t, a), \quad \forall s_t \in \mathcal{S}$.
4. $v^{\pi}_{t,m}(s_t) = r^m_t(s_t, \pi(s_t)) + \sum_{s_{t+1} \in \mathcal{S}} p^m_t(s_{t+1} \mid s_t, \pi(s_t)) \cdot v^{\pi}_{t+1,m}(s_t + 1), \forall m \in \mathcal{M}$.
5. end for

MMDP Policy Gradient

- Main idea: Take a coordinate ascent perspective to adjust model weights iteratively.
- **Definition 4.1** An *adjustable weight* for each $m \in \mathcal{M}$, $\pi \in \Pi$, $t \in \mathcal{T}$, and $s \in \mathcal{S}$ is

$$b^{\pi}_{t,m}(s) \;=\; \mathbb{P}[\tilde{m} = m, \tilde{s}_t = s],$$

where $S_0 \sim \mu$, $\tilde{m} \sim \lambda$, and $\tilde{s}_1, \ldots, \tilde{s}_T$ are distributed according to $p^{\tilde{m}}$ of policy π.

- **Theorem 4.1**: Gradient of ρ in Eq. (1) for each $t \in \mathcal{T}$, $\hat{s} \in \mathcal{S}$, $\hat{a} \in \mathcal{A}$, and $\pi \in \Pi_R$ is

$$\frac{\partial \rho(\pi)}{\partial \pi_t(\hat{s}, \hat{a})} \;=\; \sum_{m \in \mathcal{M}} b^{\pi}_{t,m}(\hat{s}) \cdot q^{\pi}_{t,m}(\hat{s}, \hat{a}) \,,$$

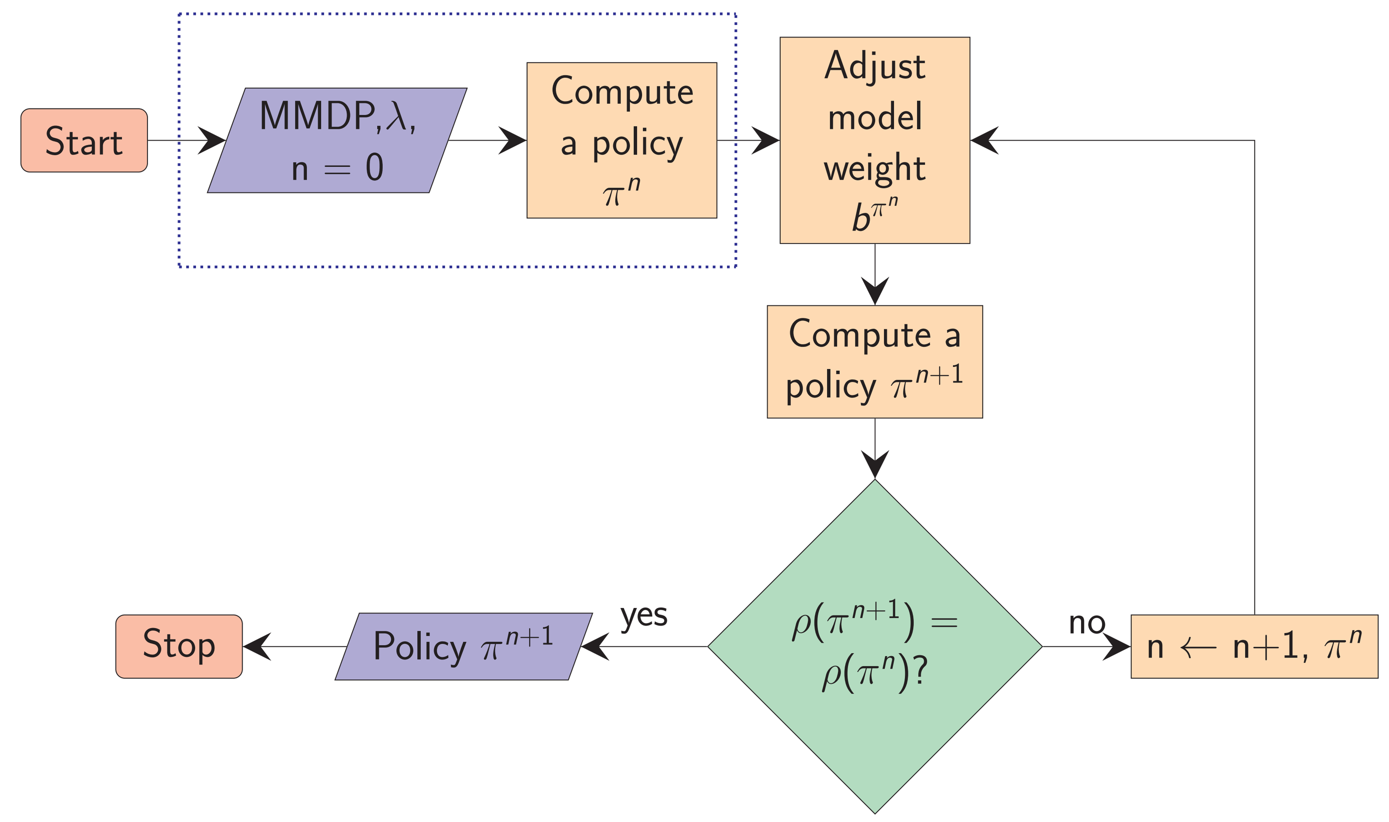
where q is state-action value function and b is an adjustable weight

- **Corollary 4.2** For any $\bar{\pi} \in \Pi$ and $t \in \mathcal{T}$, function $\pi_t \mapsto \rho(\bar{\pi}_1, \ldots, \pi_t, \ldots, \bar{\pi}_T)$ is *linear*.
- Linearity implies that we can solve the maximization over $\pi_t(s)$ as

$$\pi^n_t(s) \in \arg\max_{a \in \mathcal{A}} \sum_{m \in \mathcal{M}} b^{\pi^{n-1}}_{t,m}(s) \cdot q^{\pi^n}_{t,m}(s, a).$$

Coordinate Ascent Dynamic Programming (CADP)

- Main idea: Combine coordinate ascent method and DP to solve MMDPs.
- Corresponds to: Replace the fixed model weights λ_m in WSU by adjustable weights $b^{\pi}_{t,m}$
- Blue dotted rectangle is to compute an initial policy (for example by WSU, MVP)

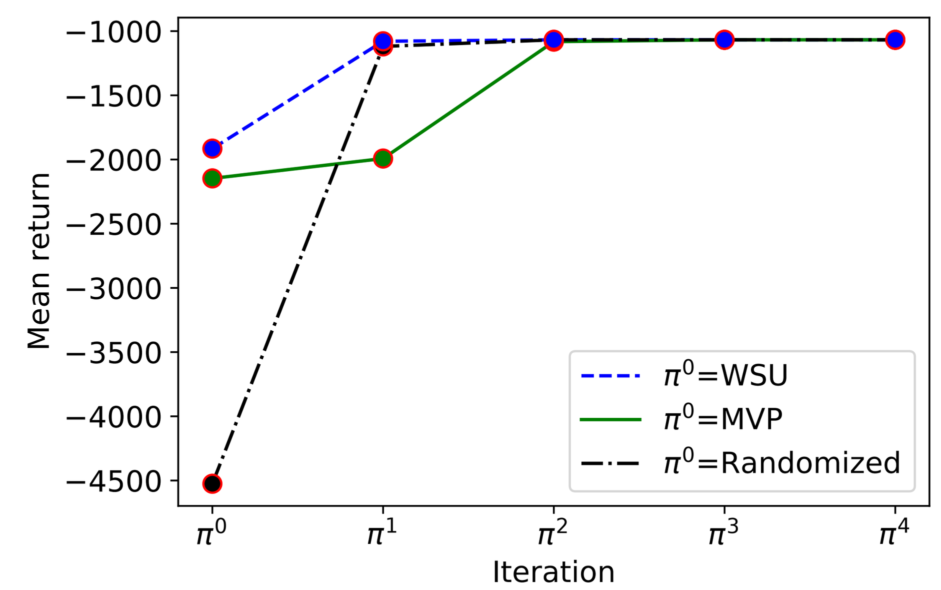


Related Algorithms

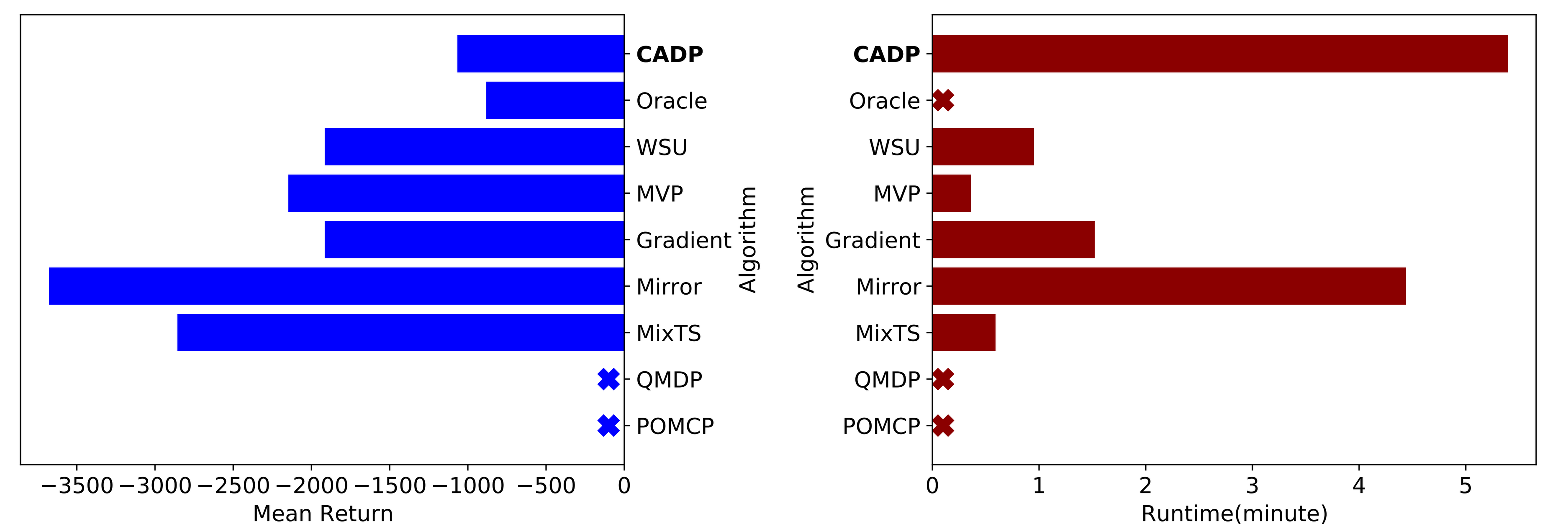
- Prior MMDP algorithms: WSU and MVP
- Gradient-based MMDP methods: Mirror and Gradient
- Thompson sampling-based algorithms: MixTS
- POMDP formulations: QMDP and POMCP

Simulation Results: Pest Control

- Time horizon $T = 50$, Domain: Pest control simulation
- Below figure: mean returns of CADP with different initial policies.



- Left figure: mean returns of algorithms, and right figure: runtimes of algorithms.
- Marker X: no single policy available or runtime is greater than 900 minutes



Simulation Results: Other Domains

- Mean returns $\rho(\pi)$ on the test set of policies π computed by each algorithm

Algorithm	RS		POP		POPS		INV		HIV	
	T = 50	T =150	T = 50	T =150	T = 50	T =150	T = 50	T =150	T = 5	T =20
CADP	**204**	**207**	**-361**	**-368**	**-1067**	**-1082**	323	**350**	**33348**	**42566**
WSU	203	206	-542	-551	-1915	-1932	323	349	**33348**	42564
MVP	201	204	-704	-717	-2147	-2179	323	**350**	**33348**	42564
Mirror	181	183	-1650	-1600	-3676	-3800	314	345	**33348**	**42566**
Gradient	203	206	-542	-551	-1915	-1932	323	349	**33348**	42564
MixTS	167	176	-1761	-1711	-2857	-3016	**327**	**350**	293	-1026
QMDP	190	183	-	-	-	-	-	-	30705	39626
POMCP	58	64	-	-	-	-	-	-	25794	30910
Oracle	210	213	-168	-172	-882	-894	332	360	40159	53856