Optimal discounting for offline input-driven MDP

Presented at Reinforcement Learning Conference 2025

Randy Lefebvre

Audrey Durand

UNIVERSITÉ LAVAL

Mila

Institut intelligence et données

Contents

- Introduction and some preliminary knowledge
- Optimal discounting for offline Input-driven MDPs
- Q & A

The goal of this presentation is to spark curiosity in Reinforcement Learning (RL) for people with a more traditional ML background.

Introduction

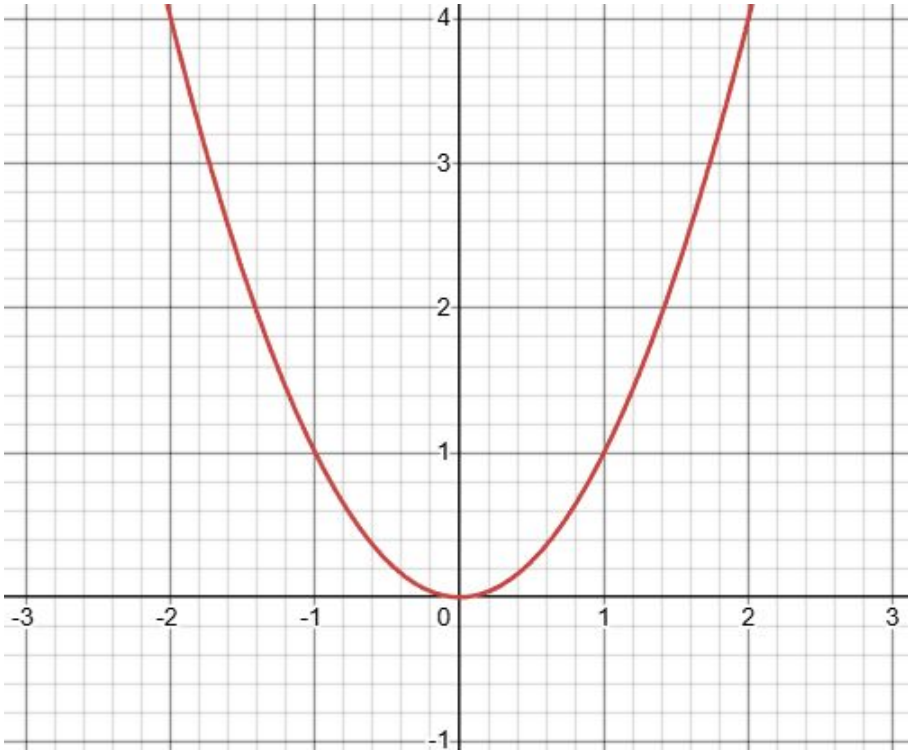
Reinforcement learning (RL) is a framework that enables control in complex environments

(Silver, David, et al. 2016)

(Ma, Yecheng Jason, et al. 2023)

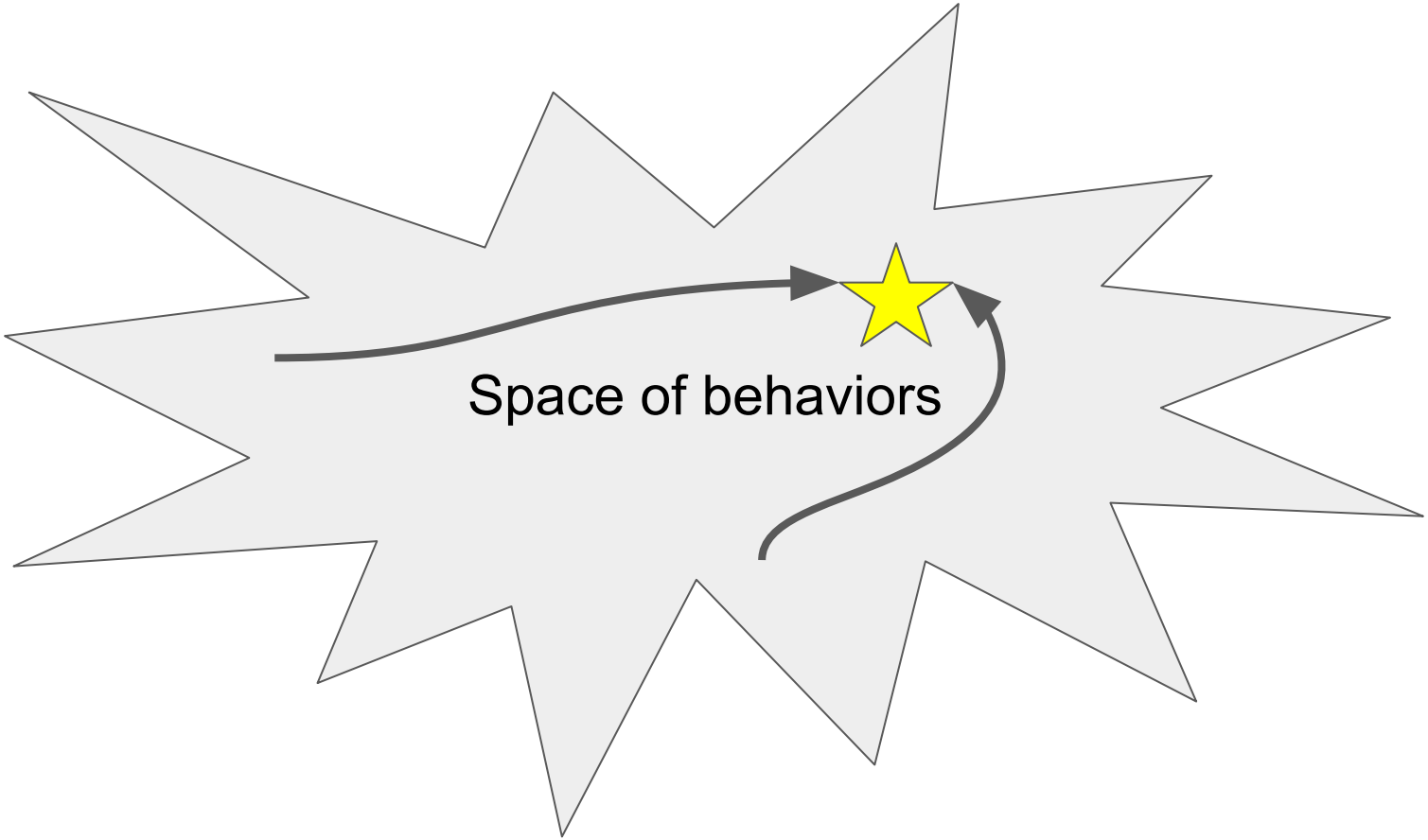
Walking on a yoga ball?

Introduction



But what even is control?!?

Introduction



Introduction

At its core, RL describes an optimization loop in an effort to do control.

The optimization loop is created, in an effort to be **domain agnostic** and **modular**

Introduction

Because RL strives to be modular and domain agnostic, we get various forms:

- **Model**/Value-Based RL
- **Offline**/Online-RL
- Safe RL
- And multiple other

Background knowledge

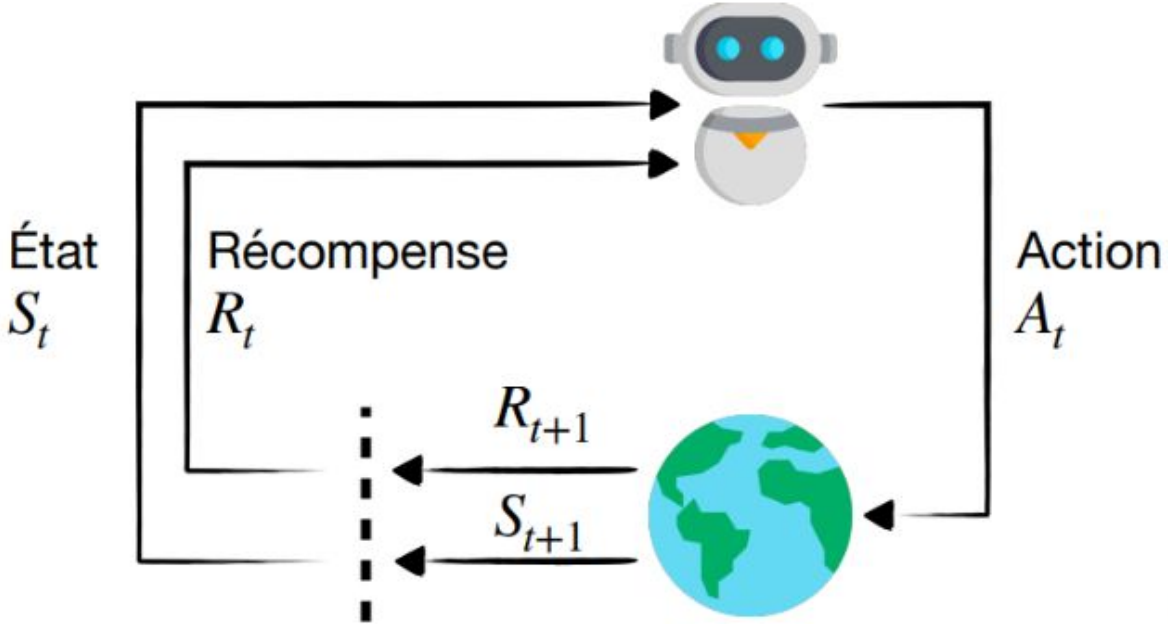
Usually, RL is formalized by a Markov Decision Process (MDP) described by the tuple:

$$M = (\mathcal{S}, \mathcal{A}, R, P, \gamma)$$

- S : The set of states
- A : The set of actions
- $P : S \times A \times S \to [0, 1]$, the transition probabilities
- $R : S \times A \to \mathbb{R}$, the reward distribution
- $\gamma \in [0, 1]$: The discount rate

Background knowledge

Interaction loop:



Background knowledge

The goal: Maximize the expected sum of discounted rewards for all states $s \in S$

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid s_0 = s \right]$$

The optimization loop is the following: Search the space of behavioral policies $\pi : S \to A$ that dictates the action to take for any given state, and find the optimal policy π^* which maximizes $V_\pi(s)$

Blackwell discount factor

Consider the environment

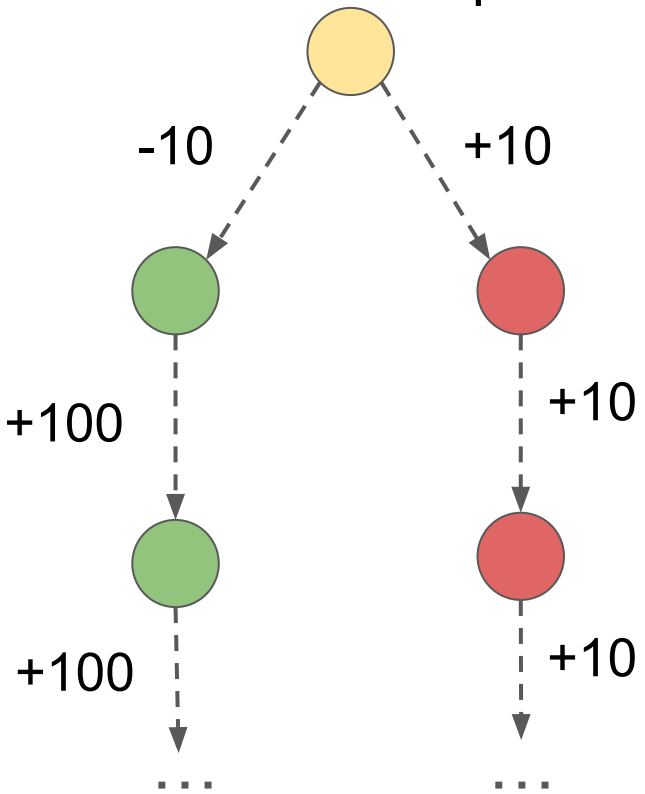
$$M = (\mathcal{S}, \mathcal{A}, R, P, \gamma)$$

Where γ controls the planning horizon

Training policy $\pi^*_{M,\gamma}$ using $\gamma > \gamma_{Bw}$ returns the same policy as $\pi^*_{M,\gamma_{Bw}}$

Training a policy using $\gamma < \gamma_{Bw}$ is called **shallow planning**

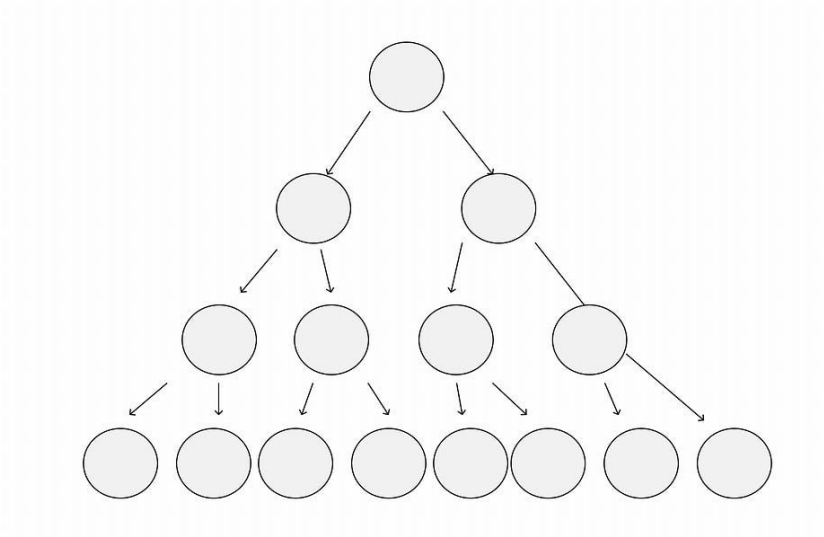
$\pi^*_{M,\gamma}$ With $\gamma > \gamma_{Bw}$
Will find the left path



Background knowledge

$$V_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid s_0 = s \right]$$

γ is important as it controls the complexity of the search space over policies (Jiang et al, 2015)



Background knowledge

In **model-based RL**, we use our knowledge of the transitions P and the rewards R to find the optimal policy.

In **offline RL**, we have a dataset of trajectories collected using a mix of behavioral policies which are suboptimal. The goal is to use the dataset to learn how to act in the real environnement.

Motivation & Problem Statement

- Despite advances in offline RL, it remains hard for policies to generalize to the real world

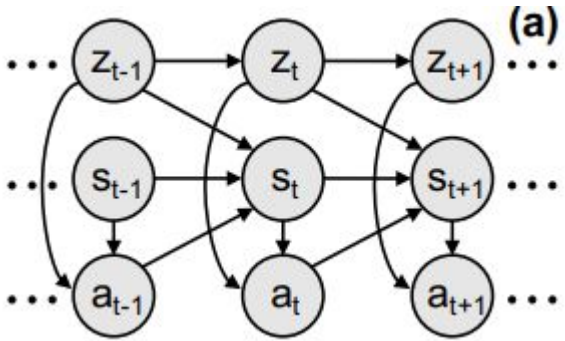
- Real world is chaotic, parts of the state are often uncontrollable and hard to model over long trajectories

- For practical applications with these properties, could we improve generalization by better understanding the impact of the planning horizon?

Input driven MDP

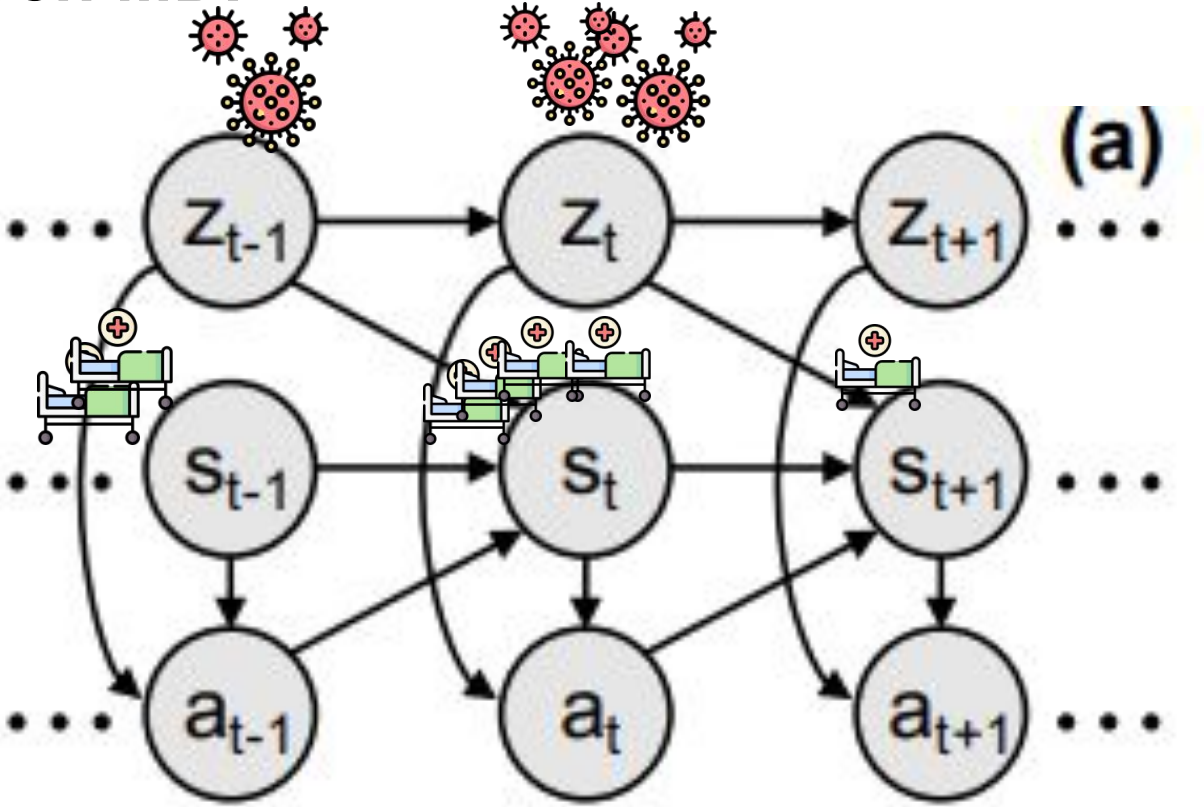
To model practical environments with chaotic tendencies, we use the Input-driven MDP framework (IDMDP)

$$M = \{\mathcal{S}, \mathcal{Z}, \mathcal{A}, R, P_z, P_s, \gamma\}$$



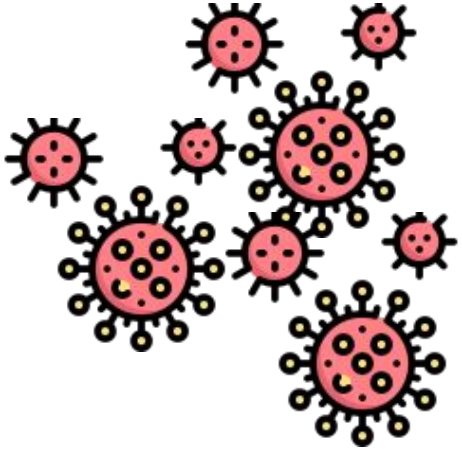
There is often a big part of the states space over which the agent has little or no control

Input driven MDP

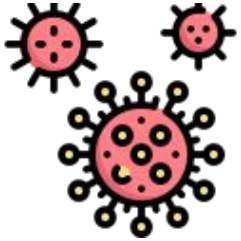


IDMDP variance

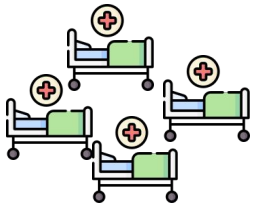
Expectation



Reality



Result = too much!



Bias-variance tradeoff

When planning on an imperfect model of inputs $\widehat{M} = \{\mathcal{S}, \mathcal{Z}, \mathcal{A}, R, \widehat{P_z}, P_s, \gamma\}$
We get the following decomposition:

$$\|V^{\pi^\star_{M,\gamma_{\text{Bw}}}}_{M,\gamma_{\text{Bw}}} - V^{\pi^\star_{\widehat{M},\gamma}}_{M,\gamma_{\text{Bw}}}\|_\infty \cdot \leqslant \underbrace{\|V^{\pi^\star_{M,\gamma_{\text{Bw}}}}_{M,\gamma_{\text{Bw}}} - V^{\pi^\star_{M,\gamma_{\text{Bw}}}}_{M,\gamma}\|_\infty}_{\text{bias}} + \underbrace{\|V^{\pi^\star_{M,\gamma}}_{M,\gamma} - V^{\pi^\star_{\widehat{M},\gamma}}_{M,\gamma}\|_\infty}_{\text{variance}} \cdot$$

Where the components depend on the structure of the problem.
Before digging into the bound, we need a few definitions first.

Input metric

With $\bar{s}_i = (z_i, s)$ the augmented state (IDMDP rewritten as MDP),

$$d^{\pi}_{states,\gamma}(\bar{s}_i, \bar{s}_j) := \left[\left|\bar{R}^{\pi}(\bar{s}_i) - \bar{R}^{\pi}(\bar{s}_j)\right| + \gamma W_1(d^{\pi}_{states,\gamma})\left(\bar{P}^{\pi}(\bar{s}_i), \bar{P}^{\pi}(\bar{s}_j)\right)\right]$$

$$d^{\pi}_{input,\gamma}(z_i, z_j) := \max_{s \in \mathcal{S}} d^{\pi}_{states,\gamma}((z_i, s), (z_j, s)),$$

This is what we mean by “structure” of the IDMDP:

$$|V^{\pi}_{M,\gamma}(s, z_i) - V^{\pi}_{M,\gamma}(s, z_j)| \leqslant d^{\pi}_{input,\gamma}(z_i, z_j)$$

Lemma 2, variance

Intuition: If inputs are chaotic and differ from one another, model error are more costly.

$$\|V_{M,\gamma}^{\pi^\star_{M,\gamma}} - V_{M,\gamma}^{\pi^\star_{\widehat{M},\gamma}}\|_\infty \leqslant \frac{\gamma}{(1-\gamma)} \max_z \|\widehat{P}_z(\cdot|z) - P_z(\cdot|z)\|_1 \max_{z_i, z_j \in \mathcal{Z}} \max_{\pi: \mathcal{Z}\times\mathcal{S}\mapsto\mathcal{A}} d^\pi_{input,\gamma}(z_i, z_j).$$

Importantly, the error can be collapsed to 0 by reducing the discount factor!

Theorem 1

Bound on the planning loss:

$$\|V^{\pi^\star_{M,\gamma_{Bw}}}_{M,\gamma_{Bw}} - V^{\pi^\star_{\widehat{M},\gamma}}_{M,\gamma_{Bw}}\|_\infty \leqslant \underbrace{\frac{\gamma_{Bw}-\gamma}{(1-\gamma_{Bw})(1-\gamma)} R_{max}}_{\text{Bias}} + \underbrace{\frac{\gamma}{(1-\gamma)} \max_z \|\widehat{P}_z(\cdot|z) - P_z(\cdot|z)\|_1 \max_{z_i,z_j \in \mathcal{Z}} \max_{\pi:\mathcal{Z}\times\mathcal{S}\mapsto\mathcal{A}} d^\pi_{input,\gamma}(z_i,z_j)}_{\text{Variance}}$$

As the discount factor becomes bigger, the bias becomes smaller, but trades off with a higher variance

Theorem 1 intuition

Take the hospital exemple. Inputs (spread of the disease) have a major impact over transitions and rewards:

$\max_{z_i, z_j \in \mathcal{Z}} \max_{\pi: \mathcal{Z} \times \mathcal{S} \mapsto \mathcal{A}} d^{\pi}_{input,\gamma}(z_i, z_j)$ Is high.

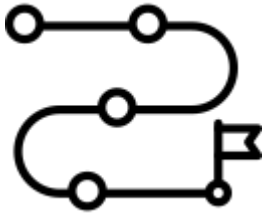
These inputs are hard to model as they depend on chaotic systems like social interactions:

$\max_{z} \|\widehat{P}_z(\cdot|z) - P_z(\cdot|z)\|_1$ Is high.

Theorem 1 intuition

We would like to plan in advance but this could be more costly than being reactive because of the complexity.

The theorem formalizes an argument as to why some control systems should be more reactive to external changes vs proactive.



Input dependent planning

When looking at the definition of the planning loss, there exists a discount factor that minimizes it for a given input. (Finite amount of policies)

$$\gamma^\star(z) := \operatorname*{argmin}_{\gamma \in [0, \gamma_{\mathrm{Bw}}]} \| V^{\pi^\star_{M,\gamma_{\mathrm{Bw}}}}_{M,\gamma_{\mathrm{Bw}}}(\cdot, z) - V^{\pi^\star_{\widehat{M},\gamma}}_{M,\gamma_{\mathrm{Bw}}}(\cdot, z) \|_\infty$$

We can reuse definitions from earlier to have this input-wise bound

$$|f_{z_i}(\gamma) - f_{z_j}(\gamma)| \leqslant 2 \max_{\pi : \mathcal{Z} \times \mathcal{S} \mapsto \mathcal{A}} d^{\pi}_{input, \gamma_{Bw}}(z_i, z_j),$$

With $f_{z_i}(\gamma)$ the planning loss at z_i using γ

Theorem 2

With a convexity assumption and prior results, we get the following theorem:

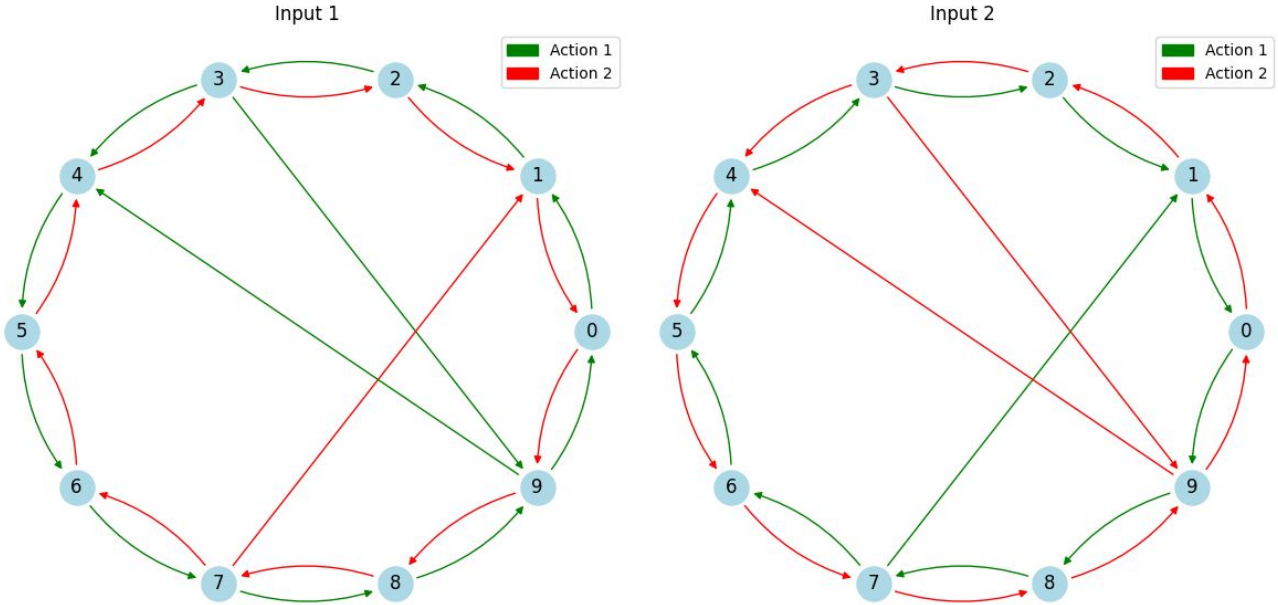
$$|\gamma^{\star}(z_i) - \gamma^{\star}(z_j)| \leqslant \sqrt{\frac{8 \max_{\pi: \mathcal{Z} \times \mathcal{S} \mapsto \mathcal{A}} d^{\pi}_{input, \gamma_{Bw}}(z_i, z_j)}{\mu}}.$$

Intuitively, if two inputs are different, the optimal discount factor on them will also be, affected by the u-shape curvature of the loss.

Theorem 2

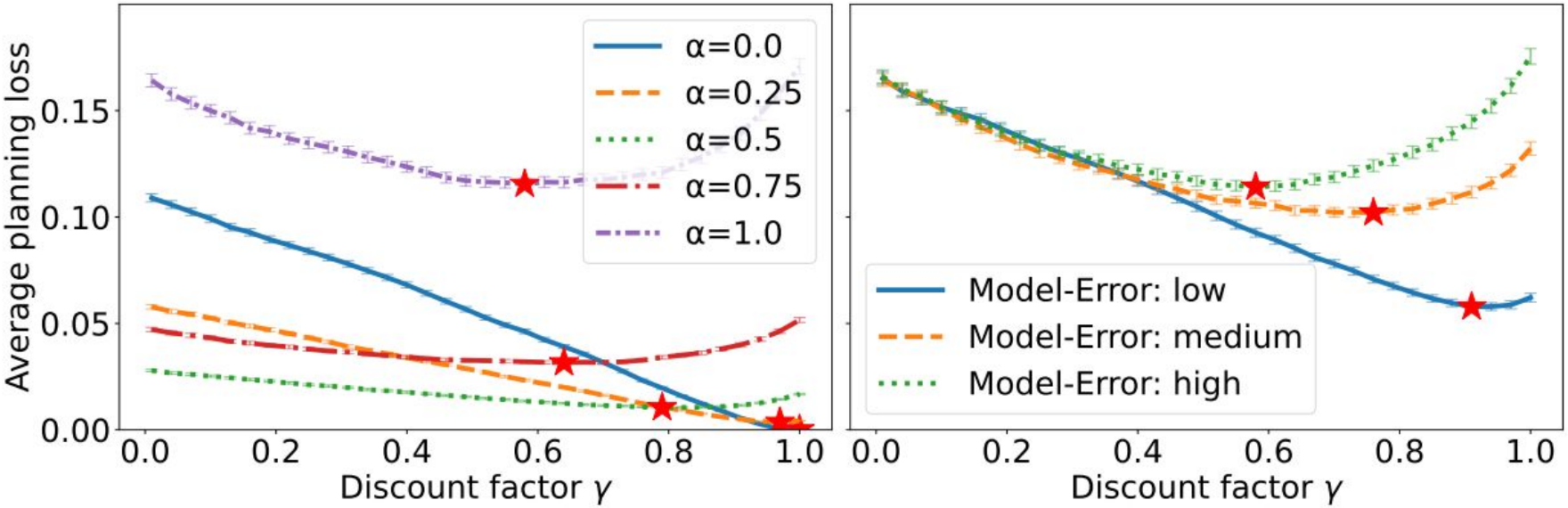
Theorem 2 puts formalism behind the idea that there might be structure behind what the optimal planning horizon is in the environnement.

Synthetic experiments setting



- α Is a parameter input, it ranges from 0 to 1. When it's 0 both inputs are the same ring mdp. When it's one, they are totally inverted in terms of rewards and transitions

Synthetic experiments results

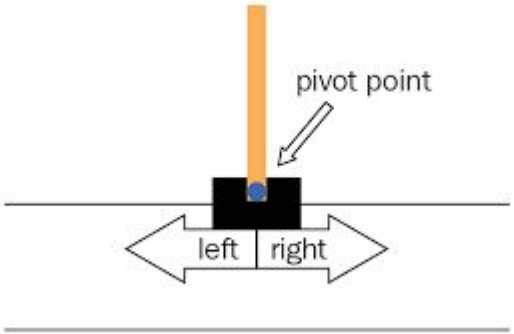


The greater the distance between inputs, the lower the optimal discount factor. When alpha is kept constant, the lower the model error, the more we should plan far ahead.

Deep RL experiment

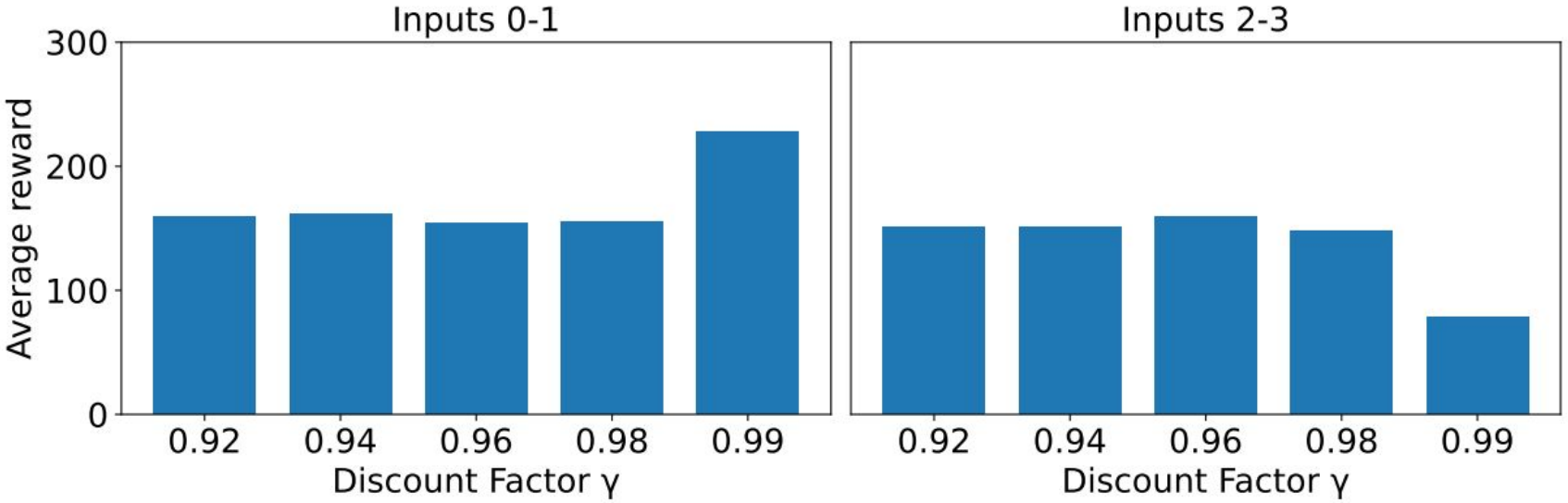
The goal is to quickly test our thesis on a setting we can't control.

We modify cartpole to have a safe zone which grants modest reward in the middle and reward zones on the side.



Input 0-1, reward zones stays static
Input 2-3, they move around

Deep RL experiment



When inputs don’t move (inputs 0 and 1), having a poor model over inputs don’t matter. When they switch, the high discount factor overfits.

Conclusion

RL describes an **optimization loop** that is **domain agnostic** and **modular**

A variant is the **IDMDP**

We formalized the **bias-variance tradeoff** for **model based offline RL** in this setting

The existence of s**tructure across inputs** could lead to input-specific planning schemes and **algorithms**.

Q & A

Thank you for listening!