NEURAL INFORMATION PROCESSING SYSTEMS

Symbolic Model-Based Reinforcement Learning

Pierre-Alexandre Kamienny [1,2] Sylvain Lamprier [3]

université angers [3] SORBONNE UNIVERSITÉ [2] Meta AI [1]

Motivation

Reinforcement learning (RL):

Agent interacts with Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r)$.

State $s_t \in \mathcal{S}$ evolves after agent takes action $a_t \in \mathcal{S}$ according to the transition function p.

The goal of RL is to find policies that **maximise cumulated rewards** $r(s_t, a_t)$.

In many real-world RL applications, e.g. robotic control, system dynamics are governed by **physical laws** that can be expressed as **mathematical expressions** involving the system's state variables and a set of **operators**, e.g. $+, \cos, \exp, \sqrt{}, \text{pow}, \frac{d}{dt}$.

For instance in CartPole, state $s_t = (x_t, \dot{x}_t, \theta_t, \dot{\theta}_t)$ is modified by the agent's actions a_t according to the following laws:

$$\ddot{\theta} = \frac{g \sin\theta + \cos\theta \left(\frac{-K_{mag} a - m_p l \dot{\theta}^2 \sin\theta}{m_c + m_p} \right)}{l \left(\frac{4}{3} - \frac{m_p \cos^2\theta}{m_c + m_p} \right)} \qquad \ddot{x} = \frac{K_{mag} a + m_p l \left(\dot{\theta}^2 \sin\theta - \ddot{\theta} \cos\theta \right)}{m_c + m_p}$$

Main idea: Leverage prior knowledge about dynamics in Model-Based Reinforcement Learning

Model-Based Reinforcement Learning

MBRL: class of RL algorithms that ground policies on learned models of the environment dynamics.

Work with the following phases:

1. Collect data $\mathcal{D}$ with current policy

2. Learn approximate model f with supervised learning (SL):

$$f^* = argmax_{f \in \mathcal{F}} \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} \mathcal{L}(s_{t+1}, f(s_t, a_t))$$

$\mathcal{F}$ can be the class of neural networks (NNs) or Gaussian Processes (GPs).

3. Improve policy

As in SL, phase 2. faces the classic problem of **under/over-fitting**.

NNs: 👍 Can express complex dynamics

👎 Overfit in small (in quantity and quality) data regimes

Solutions?

a) more data (needs good exploration) b) regularisation c) uncertainty-awareness

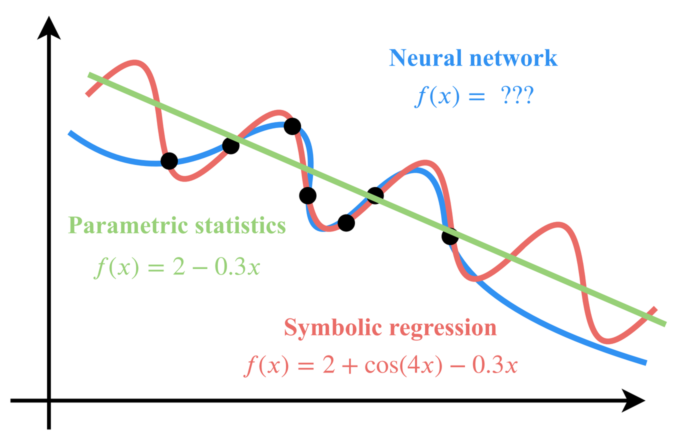
Symbolic Regression

Symbolic regression (SR): Problem of searching a function form and numerical parameters $f(x, \theta)$ that best fits data by composition of symbols (operators, variables, constants).

👍 Interpretability 👍 Extrapolation (OOD+small data)

Dominant approach: **Genetic Programming**

Evolves population of expressions with i) selection, ii) mutation and cross-overs



Symbolic-MBRL

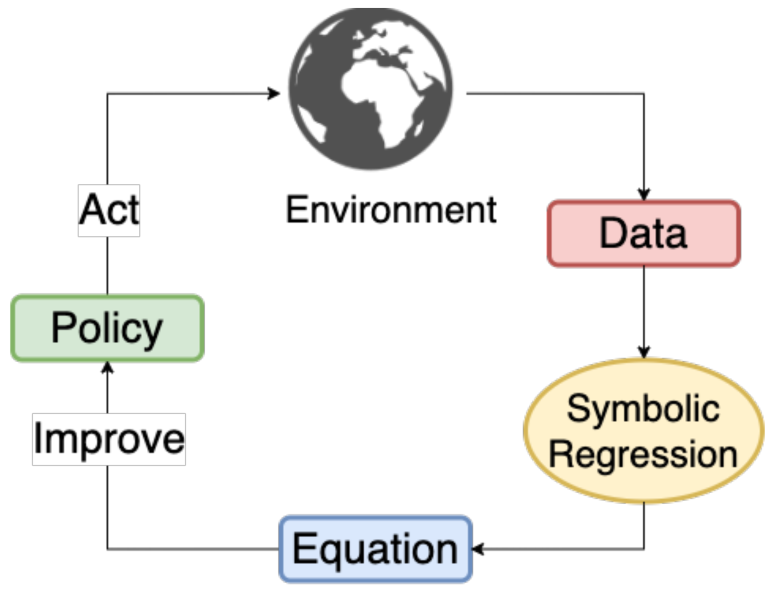
Main idea: Replace the neural dynamics model with expressions optimised via SR using pairs $([s_t, a_t], s_{t+1})$ from $\mathcal{D}$.

Can be applied to **any MBRL algorithms** in principle!

We consider Operon [1] as our base SR algorithm with the following operators:

add, sub, mul, div, sin, cos, pow

As our base MBRL algorithm, we use **Probabilistic Ensembles with Trajectory Sampling** (PETS) [2] with an ensemble of **7 models.**



$$x_{t+1} = \cos(2\pi x_t)\exp(|x_t|/3)$$

At each step , it computes the action that maximise rewards on trajectories simulated via the learned model.

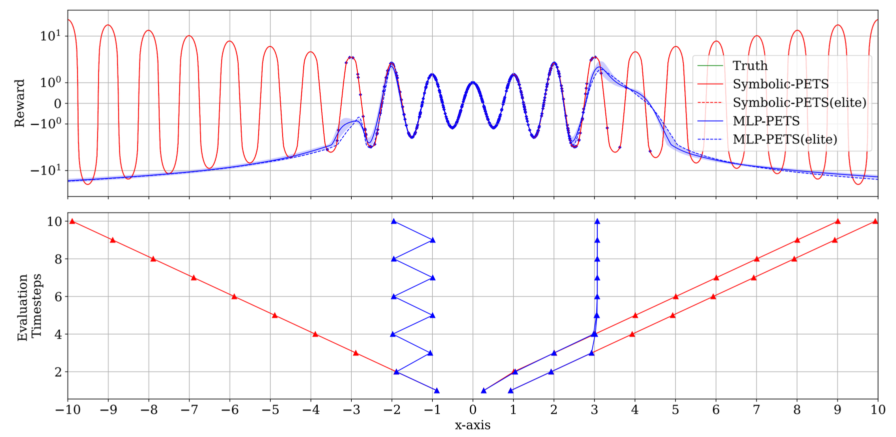
We call **Symbolic-PETS** our model and **MLP-PETS** the base algorithm.

Illustrating Example

Agent moves on the horizontal axis with the following $\mathcal{S} = [-\infty, +\infty], \mathcal{A} = [-1,1]$ (horizon 10)

p, r are given by: $s_{t+1} = s_t + a_t, \quad r_t = \cos(2\pi s_{t+1})\exp(|s_{t+1}|/3)$

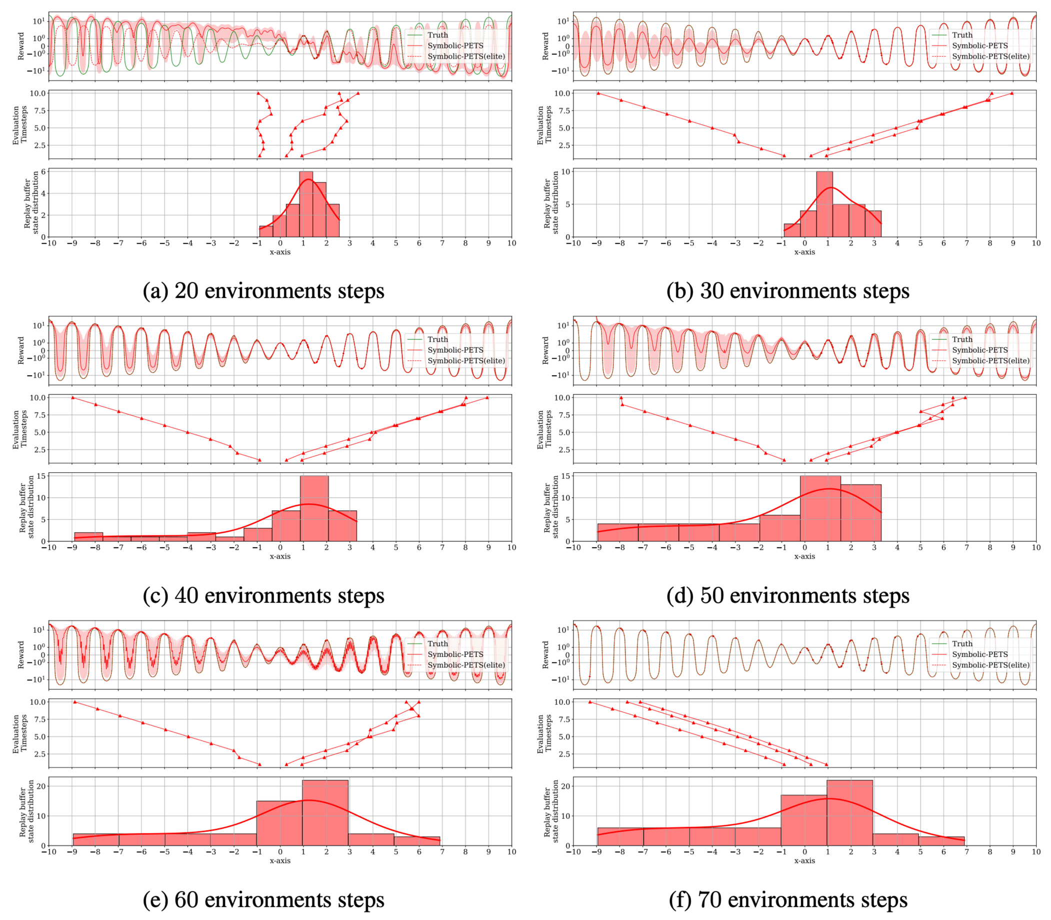
Collect 500 transitions with random policy then follow 2. and 3.



✦MLP-PETS overfits and gets sub-optimal performance

✦Symbolic-PETS learns the perfect dynamics model:

$$1.0\exp(|0.333s_t + 0.333a_t| + 2.14e^{-4})\sin(6.283x_t + 6.283a_t - |0| - 4.712)$$



(a) 20 environments steps (b) 30 environments steps

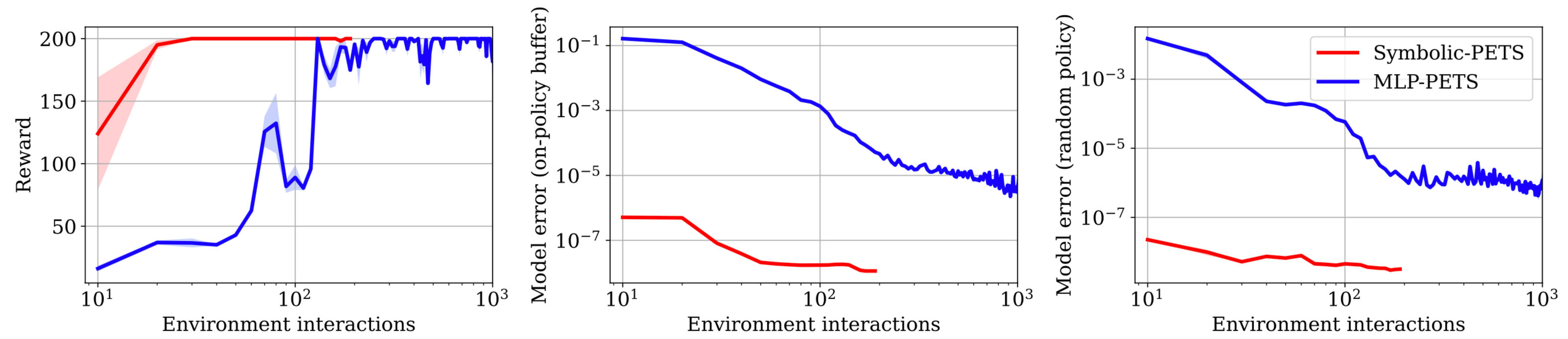
(c) 40 environments steps (d) 50 environments steps

(e) 60 environments steps (f) 70 environments steps

Evolution of the learned model, reward and state visits with respect to the number of environment steps

CartPole

As in [2], reward function is assumed to be given, we just need to learn the transition function. Symbolic-PETS achieves perfect performance in **2 order of magnitude** less interactions than MLP-PETS!



Following equations are found (looks like Taylor expansion):

$$x_{t+1} = x_t + 0.02\dot{x}_t$$

$$\theta_{t+1} = \theta_t + (0.02\dot{\theta}_t + 0.015)/\cos(0.035 * \dot{\theta}_t) - 0.015$$

$$\dot{x}_{t+1} = (0.002\theta_t + 2.34e^{-4}\dot{\theta}_t a_t + 1.0) \times (\dot{x}_t + 0.195a_t - \sin(0.015\theta_t) + 3.23e^{-5})$$

$$\dot{\theta}_{t+1} = \cos(0.195\theta_t)(0.314\theta_t + \dot{\theta}_t - 8.97e^{-1}a_t \times (-0.031\dot{\theta}_t - 2.014)\frac{(0.016\dot{\theta}_t - \cos(1.053\theta_t))}{(6.173 - 0.002\theta_t)})$$

Discussion

SR can **impact** multiple RL research topics:

- Safe RL (interpretable model?)
- Meta- and Continual RL (sample efficient)
- Sim2Real (Real2Sim?)
- Environment Design
- Exploration (extrapolation removes the need for hard exploration?)

Ideally, symbolic regressors should be as modulable than NNs:

- Fast and accurate/reliable expression inference
- Even faster expression fine-tuning
- Scale to high input dimensions (requires great feature selection)
- Batched over output dimension.
- Represent piecewise continuous functions (especially for robotics tasks)
- Represent stochastic functions (aleatoric uncertainty)

References

[1] Burlacu, Bogdan and Kronberger, Gabriel and Kommenda, Michael. "Operon C++: An Efficient Genetic Programming Framework for Symbolic Regression", In ACM, 2020.

[2] Chua, Kurtland and Calandra, Roberto and McAllister, Rowan and Levine, Sergey, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models" In NeurIPS, 2018.