Interpretable Reinforcement Learning for Continuous Action Environments

Author
Misha Kaptein
m.z.kaptein@student.tudelft.nl

Supervisor
Daniël Vos

Responsible Professor
Anna Lukina

1 - Introduction

- Reinforcement Learning (RL) trains agents to make decisions in uncertain sequential environments by interacting with them
- Deep RL methods like DQN and PPO perform well but lack interpretability, which is crucial in fields like healthcare
- Decision trees offer transparent policies but are hard to optimize in RL due to their non-differentiability [1, 2]
- DTPO enables direct optimization of decision trees and shows promising results, but only for discrete actions [3]

2 - Research Question

Can DTPO be extended to support continuous action spaces and remain competitive with neural network policies in terms of performance?

RQ1 - How does DTPO perform on discretized continuous action spaces under varying action resolutions?

RQ2 - How can DTPO be extended to directly support continuous actions?

RQ3 - How does the extended DTPO with continuous actions compare in performance and runtime to RPO with neural networks?

3 - Methodology

Discretization Impact on DTPO Performance
- Use the Pendulum-v1 continuous environment
- Convert the action space into uniformly spaced action bins
  - Ranging from 2 - 64 actions
- Evaluate how resolution affects performance

Extending DTPO for Continuous Actions
- Introduce DTPO-c
- Modify each leaf node to output mean and standard deviation
- During training:
  - Sample actions from a Gaussian distribution $N(\mu, \sigma^2)$
- During evaluation:
  - Use mean action for obtaining a deterministic decision tree
- Replace the discrete-action PPO loss function with continuous log-probabilities [4]

Performance Comparison of DTPO-c with RPO
- RPO is similar to PPO, but adds noise to the action mean to encourage exploration
- Same environment (Pendulum-v1), random seeds, and reward metrics
- Compare results in two dimensions:
  - Sample efficiency
    - (return vs. timesteps)
  - Computational efficiency
    - (return vs. runtime)

References

[1] A. Silva, I. D. J. Rodriguez, T. W. Killian, S. Son, and M. C. Gombolay, "Interpretable reinforcement learning via differentiable decision trees," CoRR, vol. abs/1903.09338, 2019.
[2] R. R. Paleja, Y. Niu, A. Silva, C. Ritchie, S. Choi, and M. C. Gombolay, "Learning interpretable, high-performing policies for continuous control problems," CoRR, vol. abs/2202.02352, 2022.
[3] D. Vos and S. Verwer, "Optimizing interpretable decision tree policies for reinforcement learning," CoRR,vol. abs/2408.11632, 2024.
[4] C. M. Bishop, Pattern recognition and machine learning, 5th Edition. Information science and statistics, Springer, 2007.

TU Delft

4 - Results

RQ1 - Discretization Impact on DTPO Performance

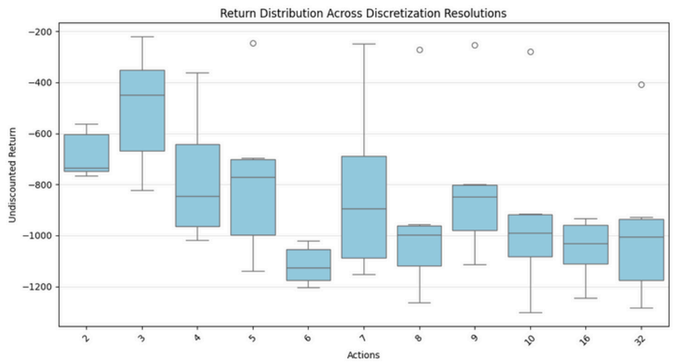

Figure 1: Undiscounted return distribution for varying action resolutions across 6 different seeds

- Finer discretization does not necessarily lead to better performance
  - Decision tree capacity
- Odd-numbered values tend to perform better than nearby even-numbered values on average
- Possibly caused by the 0-action
  - No significant difference after testing

RQ2 - Extending DTPO for Continuous Actions
- DTPO-c is successfully learned on Pendulum-v1, which shows feasibility
- Preserved interpretability with visualizable tree policies in a continuous domain

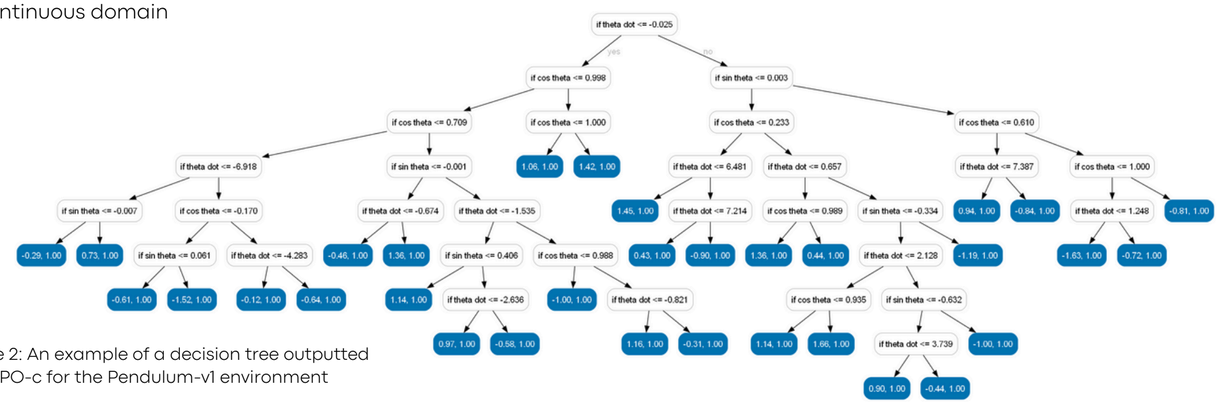

Figure 2: An example of a decision tree outputted by DTPO-c for the Pendulum-v1 environment

RQ3 - Performance Comparison of DTPO-c with RPO

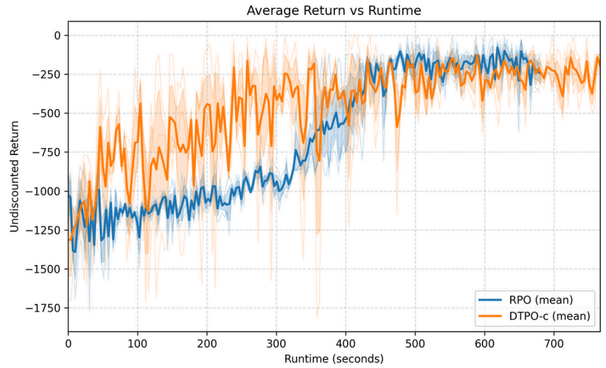

Figure 3: Average undiscounted return across 3 different seeds against the runtime of DTPO-c (orange) and RPO (blue)

- DTPO-c requires significantly more timesteps (12.5 million) than RPO (350k) to reach a performance of around -200
- From runtime perspective, DTPO-c initially outperforms RPO, but then RPO overtakes again
- DTPO-c is computationally heavier per timestep, but can match performance of RPO given sufficient training time

5 - Conclusion

- DTPO-c enables interpretable reinforcement learning in continuous action spaces
- Can achieve competitive performance with RPO, but requires more timesteps and longer runtime
  - Trade-off between transparency and efficiency
  - Useful in safety-critical environments
- Discretization study reveals:
  - More discrete actions do not necessarily yield better performance
  - Performance depends on model capacity

6 - Future Work

- Test DTPO-c on complex environments like HalfCheetah or Walker2d
- Allow each leaf to learn its own variance for better state-dependent exploration
- Explore methods that reuse parts of previous trees to boost sample efficiency
- Explore non-uniform action bins for discrete DTPO