Optimizing Interpretable Decision Tree Policies for Reinforcement Learning

Daniël Vos and Sicco Verwer

Delft University of Technology
{d.a.vos, s.e.verwer}@tudelft.nl

1 Introduction

In recent years, many successful neural network-based techniques have been proposed for reinforcement learning [3,4]. However, due to the size and structure of these models, the resulting policies cannot be interpreted, which limits their use in real-life applications. Decision trees are a popular model type in supervised learning as they can be directly interpreted and efficiently learned with heuristics [2]. Using decision trees as reinforcement learning policies is, therefore, a promising research direction. Unfortunately, decision trees are difficult to optimize for reinforcement learning because existing algorithms require models to be differentiable, which is not possible for decision trees due to their discontinuity.

Some methods have been proposed for optimizing decision tree policies by working around their non-differentiability. In particular, imitation learning techniques such as VIPER have proven successful. VIPER [1] first trains a Deep Q-Network [3] and then learns a decision tree from the neural network using imitation learning based on the Q-values. While this often results in good decision tree policies, it is time-consuming and relies on a good teacher model.

We propose Decision Tree Policy Optimization (DTPO), an iterative method that uses regression tree learning heuristics to optimize interpretable policies for reinforcement learning directly. DTPO is inspired by the success of PPO [4], a policy gradient style method that has gained widespread attention for its strong performance and relative simplicity. A high-level overview of the method is given

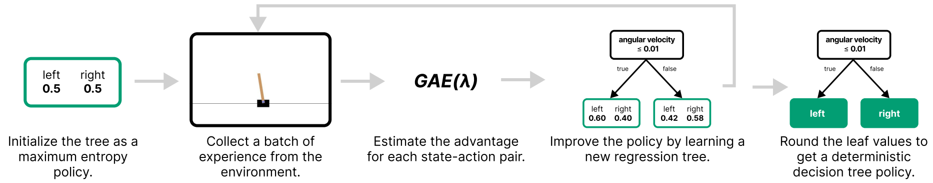
left
right
0.5
0.5
Initialize the tree as a maximum entropy policy.
Collect a batch of experience from the environment.
GAE(λ)
Estimate the advantage for each state-action pair.
angular velocity ≤ 0.01
true
false
left
right
0.60
0.40
left
right
0.42
0.58
Improve the policy by learning a new regression tree.
angular velocity ≤ 0.01
true
false
left
right
Round the leaf values to get a deterministic decision tree policy.

Fig. 1: High-level overview of DTPO. The tree is initialized as a single leaf with equal probability for each action and iteratively refined by minimizing the loss with regression tree heuristics on batches of environment experience. DTPO optimizes decision tree policies without imitating neural networks.

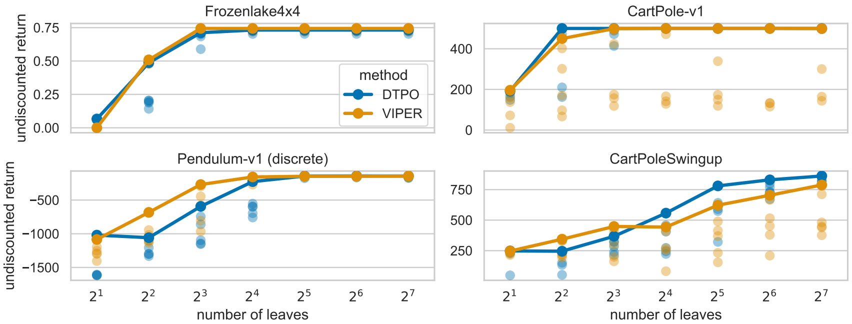

Fig. 2: Returns of policies with varying decision tree sizes. Each tree size was run with six random seeds, and best-performing policies are highlighted. DTPO and VIPER perform similarly on average, but this varies depending on the environment. DTPO finds strong and simple policies for the complex *CartPoleSwingup* problem and identifies a tree with 128 leaves that performs near-optimally.

in Figure 1. DTPO allows us to use gradient-based algorithms to optimize decision tree policies for reinforcement learning without imitation learning or altering the model class.

2 Results and Discussion

In our paper, we evaluate DTPO on various control tasks and discrete MDPs and compare its performance to VIPER and the neural network-based techniques DQN and PPO. An important consideration in interpretable machine learning is the tradeoff between simplicity and performance. Therefore, in Figure 2, we visualize the best returns out of 6 random seeds on various environments and decision tree sizes. In these environments, DTPO can find near-optimal policies that perform as well as neural networks, sometimes with trees with as few as eight leaves. On the complex *CartPoleSwingup* problem, DTPO trains significantly stronger policies for the same tree size (simplicity) compared to VIPER. We also performed a larger benchmark with 17 environments. We found that when we limit the number of leaves to 16, DTPO or VIPER outperformed both neural network-based methods on 4 of the 17 tested environments.

In conclusion, DTPO can train policies that are small enough to be human-interpreted and performed competitively compared to existing algorithms that extract decision trees from neural network policies. Our experiments on classic control tasks and discrete MDPs demonstrate that small decision trees can sometimes directly replace neural networks to improve interpretability without losing performance. We published our code on GitHub[1].

[1] https://github.com/tudelft-cda-lab/DTPO

References

1. Bastani, O., Pu, Y., Solar-Lezama, A.: Verifiable reinforcement learning via policy extraction. Advances in neural information processing systems **31** (2018)
2. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and Regression Trees. Taylor & Francis (1984), https://books.google.nl/books?id=JwQx-WOmSyQC
3. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)