RLPy: A Value-Function-Based Reinforcement Learning Framework for Education and Research
Alborz Geramifard®, Christoph Dann, Robert H. Klein, William Dabney®, Jonathan P. How

Abstract

RLPy is an object-oriented reinforcement learning (RL) software package with focus on value-function-based methods using linear function approximation and discrete actions. The framework was designed for both education and research purposes. It provides a rich library of fine-grained, easily exchangeable components for learning agents (e.g., policies or representations of value functions), facilitating recent increased specialization in RL. RLPy is written in Python to allow fast prototyping but is also suitable for large-scale experiments through its inbuilt support for optimized numerical libraries and parallelization. Code profiling, domain visualizations, and data analysis are integrated in a self-contained package available under the Modified BSD License. All these properties allow users to compare various RL algorithms with little effort.

Problem

An easy to use RL framework for both research and education

Interest in Value-based RL with linear function approximation

Increased Specialization in RL ➔ More Granular Framework

Comparison with State-of-the-art Techniques in Various Domains

Existing Gap

Lack of granularity to accommodate recent advances in RL

Challenging for entry level due to programming languages (e.g. C++)

Not Self-contained

Easy Installation

> pip install -U RLPy

Rapid Prototyping in Python

Improved Granularity of the agent using OO Python

Built-in Profiling

Improved Hyperparameter Optimization

Improved Experimentation

Reproducible

Parallel Execution

Optimized Implementation (Cython, C++)

Batteries Included!

20 Domains, 8 Learning Algorithms

4 Policies, 7 Representations

RLPy is a new python based open-source RL framework.

Simplifies the construction of new RL ideas

Accessible for both novice and expert users

Realizes reproducible experiments

Examples

Grid World

Cart Pole

Plotting

Learning Steps

Learning Episodes

Learning Time

Return

Number of Features

Termination

Conclusion