Reinforcement Learning

Homework #2

The λ Return

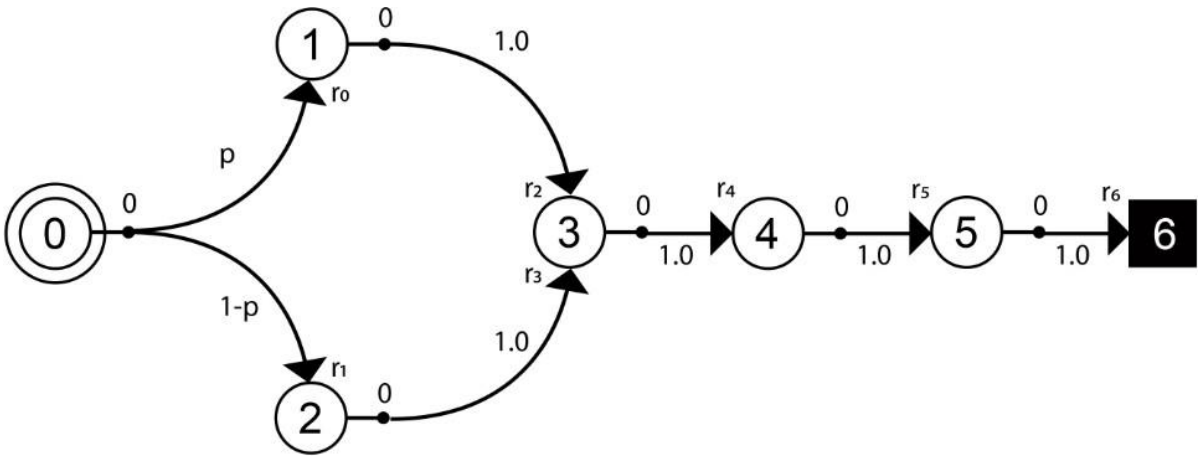
Description

Given a MDP and a particular time step t of a task (continuing or episodic), the λ-return, G_t^λ, $0 \leq \lambda \leq 1$, is a weighted combination of the n-step returns $G_{t:t+n}$, $n \geq 1$:

$$G_t^\lambda = \sum_{n=1}^{\infty} (1-\lambda)\,\lambda^{n-1} G_{t:t+n}$$

While the n-step returns $G_{t:t+n}$ can be viewed as the target of an n -step TD update rule, the λ-return can be viewed as the target of the update rule for the TD(λ) prediction algorithm, which you will become familiar with in Project 1.

Consider the Markov reward process described by the following state diagram and assume the agent is in state 0 at the time t (also assume the discount rate is $\gamma = 1$). A Markov reward process can be considered an MDP with only one action possible from each state (denoted as action 0 in the figure below).



Procedure

- You will implement your solution using the $solve()$ method in the code below.
- You will be given p, the probability of transitioning from state 0 to state 1, V, the estimate of the value function at the time t, represented as a vector $[V(0), V(1), V(2), V(3), V(4), V(5), V(6)]$, and $rewards$, a vector of the rewards $[r_0, r_1, r_2, r_3, r_4, r_5, r_6]$ corresponding to the MDP.
- Your return value should be a value of λ, strictly less than 1, such that the expected value of the λ-return equals the expected Monte-Carlo return at the time t.
- Your answer must be correct to 3 decimal places, truncated (e.g. 3.14159265 becomes 3.141).

Resources

The concepts explored in this homework are covered by:

- Lecture Lesson 3: Temporal Difference Learning
- Chapter 7 (7.1 n-step TD Prediction) and Chapter 12 (12.1 The λ-return) of http://incompleteideas.net/book/the-book-2nd.html
- 'Learning to Predict by the Method of Temporal Differences', R. Sutton, 1988

Submission

- The due date for this assignment is indicated on the syllabus page.
- Use the template code to implement your work.
- Please use *Python 3.6.x* or a more recent version and *numpy==1.18.0* or a more recent version, and you can use any core library (i.e., anything in the Python standard library). No other library can be used.