COMP-767: Reinforcement Learning - Assignment 2

Posted Tuesday, February 19, 2019
Due Friday, March 15, 2019 (Revised from Tuesday March 12)

The assignment can be carried out individually or in teams of two. You have choices on both parts of the assignment.

1. **Prediction and control in RL [50 points]**

   Choose **one** of the following topics.

   (a) In this task, you will compare the performance of SARSA, expected SARSA and Q-learning on the Taxi domain from the Gym environment suite:

   `https://gym.openai.com/envs/Taxi-v2/`

   Use a tabular representation of the state space, and ensure that the starting and end location of the passenger are random. Exploration should be softmax (Boltzmann). You will need to run the following protocol. You will do 10 independent runs. Each run consists of 100 segments, in each segment there are 10 episodes of training, followed by 1 episode in which you simply run the optimal policy so far (i.e. you pick actions greedily based on the current value estimates). Pick 3 settings of the temperature parameter used in the exploration and 3 settings of the learning rate. You need to plot:

   - One u-shaped graph that shows the effect of the parameters on the final training performance, expressed as the return of the agent (averaged over the last 10 training episodes and the 10 runs); note that this will typically end up as an upside-down u.
   - One u-shaped graph that shows the effect of the parameters on the final testing performance, expressed as the return of the agent (during the final testing episode, averaged over the 10 runs)
   - Learning curves (mean and standard deviation computed based on the 10 runs) for what you pick as the best parameter setting for each algorithm

   Write a small report that describes your experiment, your choices of parameters, and the conclusions you draw from the graphs.

   (b) We discussed in class some work the complexity of exploration in reinforcement learning. The E^3 algorithm is one of the first attempts to provide sample complexity results for tabular reinforcement learning algorithms that do control.

   `https://www.cis.upenn.edu/~mkearns/papers/reinforcement.pdf`

   You need to write a short summary (max 3 pages in format of your choice) of the result and the main steps and ideas in the proof presented in this paper. Feel free to add some background if you think it would be necessary to understand their approach. Explain why (or why not) in your opinion this is an algorithm that generalizes to function approximation.

2. **Function approximation [50 points]**

Choose **one** of the following topics.

(a) Implement and compare empirically Monte Carlo and TD-learning with eligibility traces and linear function approximation on the Pendulum-v0 domain from the Gym environment suite:

`https://gym.openai.com/envs/Pendulum-v0/`

You should evaluate the fixed policy that produces torque in the same direction as the current velocity with probability 0.9 and in the opposite direction with probability 0.1. If velocity is 0, you can torque in a random direction. For this experiment, you should use a tile coding function approximator, in which you discretize the angular position and angular velocity into 10 bins each, and use 5 overlapping tilings, whose weights start initialized randomly between -0.001 and 0.001. You will need to use the same seed for this initialization for all parameters settings, but will have 10 different seeds (for the different runs). You should use values of $\lambda = \{0, 0.3, 0.7, 0.9, 1\}$. For each value, use 3 settings of the learning rate parameter $\alpha = 1/4, 1/8, 1/16$. Remember that the learning rate per parameter needs to be divided by the number of overlapping tilings. Perform 10 independent runs, each of 200 episodes. Each episode should start art state (0,0). Plot 5 graphs, one for each of the λ values, showing the value of the start state, using each of the α values (each of the 5 graphs has 3 curves). Explain briefly what you can conclude from these graphs, in terms of the speed of convergence and stability of these algorithms.

(b) In 2000, Geoff Gordon proved an interesting result that on-policy SARSA with function approximation converges to a region. This result was in contrast with Leemon Baird's earlier counterexample on Q-learning divergence, which is discussed in the book.

`https://papers.nips.cc/paper/1911-reinforcement-learning-with-function-approximation-converges-to-a-regi`
`pdf`

You need to write a short summary (max 3 pages in latex format of your choice) of the intuition of the proof, and what is the intuition behind the difference in behavior of these two algorithms. Do you see any possibility to improve Gordon's result? Explain your answer.