Stanford CME 241 (Winter 2023) - Assignment 7

Assignments:

1. **Optional** Although the provided code contains a number of utilties for Monte-Carlo prediction, you may consider writing re-writing these functions to improve your understanding.

   a. We have written the function mc_prediction in rl/monte_carlo.py as an implementation of Monte-Carlo Prediction with function approximation. You have also learnt that since Tabular MC Prediction is a special case of MC Prediction with Function Approximation and so, writing a separate function for Tabular MC Prediction is not necessary. But for a learning experience, it's a great idea to write a function for Tabular MC Prediction **from scratch**. Think about what the input and output types must be. Be sure to reduce the learning rate appropriately as a function of number of updates (or as a function of number of episodes).

   b. We have written the function td_prediction in rl/td.py as an implementation of Temporal-Difference Prediction with function approximation. You have also learnt that since Tabular TD Prediction is a special case of TD Prediction with Function Approximation and so, writing a separate function for Tabular TD Prediction is not necessary. But for a learning experience, it's a great idea to write a function for Tabular TD Prediction **from scratch**. Think about what the input and output types must be. Be sure to reduce the learning rate appropriately as a function of number of updates.

   c. Test your above implementations of Tabular MC Prediction and Tabular TD Prediction on SimpleInventoryMRPFinite (from rl/chapter2/simple_inventory_mrp.py) by ensuring that your Value Function output matches that produced by the function approximation versions of MC Prediction and TD Prediction.

2. Extend RandomWalkMRP (in rl/chapter10/random_walk_mrp.py) to RandomWalkMRP2D which is a random walk in 2-D with states $\{(i, j) | 0 \leq i \leq B_1, 0 \leq j \leq B_2\}$ with terminal states as $(0, j)$ and (B_1, j) for all j, $(i, 0)$ and (i, B_2) for all i, and with reward of 0 for all $(0, j)$ and for all $(i, 0)$, reward of 1 for all (B_1, j) and for all (i, B_2), and with discrete probabilities of 4 movements - UP, DOWN, LEFT, RIGHT from any non-terminal state. Analyze the convergence of MC and TD on this RandomWalkMRP2D much like how we analyzed it for RandomWalkMRP, along with plots of similar graphs.

3. In the following question, we explore the connection between TD and MC algorithms.

   a. Implement the TD(λ) Prediction algorithm from scratch in Python code. First do it for the Tabular case, then do it for the case of Function Approximation.

   b. Prove that the MC Error can be written as the sum of discounted TD errors, i.e.,

   $$G_t - V(S_t) = \sum_{u=t}^{T-1} \gamma^{u-t} \cdot (R_{u+1} + \gamma \cdot V(S_{u+1}) - V(S_u))$$

   The goal here is for you to practice formal proof-writing of these types of simple yet important identities. So aim to work this out from scratch rather than treating this as a special case of a more general result proved in class or in the textbook.

   c. Test your above implementation of TD(λ) Prediction algorithm by comparing the Value Function of an MRP you have previously developed (or worked with) as obtained by Policy Evaluation (DP) algorithm, as obtained by MC, as obtained by TD, and as obtained by your TD(λ) implementation. Plot graphs of convergence for different values of λ.