COMP-597: Reinforcement Learning - Assignment 2

Posted Thursday February 24, 2022
Due Tuesday March 8, 2022

The assignment can be carried out individually or in teams of two. Further instructions about how to submit will be provided by the TAs.

1. [25 points] **Problem formulation**

   Consider a sandwich shop in a small town with a fixed population of N people. Customers arrive at times governed by an unknown probability distribution. Each customer can order a sandwich with a certain type of bread (chosen from N types) and filling (chosen from M types). Customers pay a given price for each sandwich, which depends on the type of bread and filling. If a customer cannot get the desired sandwich, he or she will not make a purchase and will never come back to the store again. Ingredients need to be discarded 3 days after purchase. The store can hold K units of bread and L units of filling (of all types combined) on any day.The store owner wants to figure out a policy for buying ingredients in such a way as to maximize long-term profit, and hopes to use reinforcement learning for this task.

   (a) [5 points[What is the state space and action space for this problem? What is the reward function?

   (b) [5 points] For your problem formulation, what is the size of the state space?

   (c) [5 points] Would you use dynamic programming or reinforcement learning for this problem? Justify your answer

   (d) [5 points] Between Monte Carlo and temporal-difference learning, which method would you prefer? Justify your answer.

   (e) [5 points] Is function approximation required to solve this problem? Justify your answer

2. [30 points] **Bellman equations and dynamic programming**

   Suppose we are in an MDP and the reward function has the structure $r(s, a) = ar_1(s, a) + br_2(s, a)$ where a and b are real numbers. In this question, we investigate whether we could solve the problems defined by reward functions r_1 and r_2 independently and then somehow combine them in order to solve the problem defined by r. We are using the discounted setting with a given discount γ.

   (a) [10 points] Suppose you are given the action-value functions q_1^π and q_2^π corresponding to the action-value function of an arbitrary, fixed policy π under the two reward functions. Using the Bellman equation, show if it is possible or not to combine these value functions in a simple manner to obtain q^π corresponding to reward function r.

(b) [10 points] Suppose you are given the optimal action-value functions q_1^* and q_2^*. Using the Bellman equation, explain if it is possible or not to combine these value functions in a simple manner to obtain q^* which optimizes reward function r

(c) [10 points] Suppose you are given the optimal policies π_1^* and π_2^*. Explain if it is possible or not to combine these policies in a simple manner to obtain policy π^* which optimizes reward function r.

3. [25 points] **An alternative learning algorithm**

In this question, we will consider a learning algorithm which attempts to learn a Q-function, but instead of using the usual Q-learning target, it uses as target a mixture of $(1-\epsilon)\max_a Q(s',a) + \epsilon \sum_a \pi(s',a)Q(s',a)$, discounted by γ, where $\epsilon \in (0,1)$ is a hyper-parameter. Assume that π is a soft, greedifying policy.

(a) [5 points] Is this algorithm on-policy or off-policy? Justify your answer.

(b) [5 points] Write the two-step version of the algorithm.

(c) [5 points] Write the single-step function approximation version of the algorithm

(d) [5 points] Will this algorithm converge in the limit in the tabular setting? Justify your answer based on your knowledge about Q-learning and Expected Sarsa convergence

(e) [5 points] How would you expect the performance of this algorithm to be compared to Q-learning and Sarsa? Discuss bias, variance and maximization bias.,

4. [5 points] **Function approximation**

Consider two function approximators, one which discretizes the state space into k bins, and one which takes each of those bins and splits it in half (hence producing a feature vector of size $2k$). Suppose we want to use the approximators to estimate the value function of a fixed policy, using on-policy data. Suppose you have a small number of samples n. Explain the impact that you expect to see on the two algorithms by using $TD(\lambda)$ with $\lambda > 0$ compared to $TD(0)$.

5. [15 points] k**-order Markovian assumption**

Consider a k-order MDP, in which the distribution of the next state and reward depend on the k previous states and actions:

$$P(S_{t+1} = s', R_{t+1} = r | S_0 A_0 \dots S_t A_t) = P(S_{t+1} = s', R_{t+1} = r | S_{t-k+1} A_{t-k+1} \dots S_t A_t)$$

(a) [10 points] Define a value iteration algorithm for this setup, assuming k is given and constant, and show that it will converge to a unique, optimal value function

(b) [5 points] What impact would an unknown k have on your algorithm? Can you still have a convergent value iteration algorithm in that case? Justify your answer