**Question A**

In chapter 4, we found that Eval\_Trunc could be used in place of Eval. Briefly and clearly explain the following:

1. Why is this important for Dynamic Programming?

Ans: Since Dynamic Programming includes policy evaluation in each iteration, but only the first couple iterations have large impact, therefore it’s more efficient to truncate policy evaluation whenever it converges well enough.

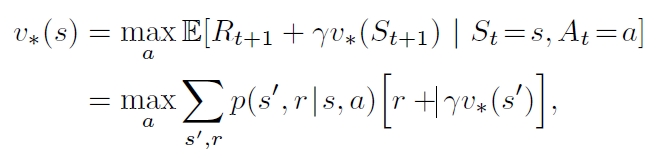
1. Why is this important for Reinforcement Learning? Explain the significance of this result.

This is important for RL because the idea can be applied to other algorithms, it’s result won’t lose convergence guarantees of policy iteration

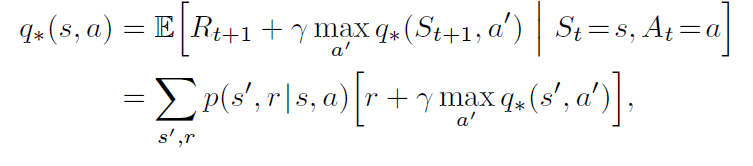
**Question B:**

(10 points) Clearly explain the importance of the Bellman equation in the context of formulating the Dynamic Programming algorithm for prediction? Use equations where needed for clarity.

The bellman equation is important because it allows us to formulate the value of a state and thus we can use dynamic programming to update the evaluation and find policy based on the state value. According to Bellman Equation



we can see we can formulate the value of a state by the sum of next states’ values and reward multiply by the possibility of reaching next state, or we can also use Bellman’s state action equation



to evaluate the value of taking a specific action at certain state.

**Question C:**

Clearly answer the below.

1. (10 points) Explain how exploration concretely manifests itself in Reinforcement Learning as presented for Dynamic Programming and Monte Carlo methods.

In Dynamic Programming since the environment is fully known the exploration is simply iterate over all states and evaluate them and update policy accordingly. But in Monte Carlo method we do not have complete knowledge over environment and thus we need a proactive exploration step, that is, e-greedy or e-soft, to take an exploration step and explore actions we have not tested yet, then we can improve our policy so that we can find optimal action.

1. (5 points) Why is exploration needed?

Because if we always follow the policy we have we will not be able to try new actions, and some of the actions may lead to better result than current action and has potential to be the optimal policy. Without the exploration we will only be updating existing policy repeatedly.

**Question D:**

(10 points) In Figure 4.1 of the text, observe the k=3 value function. Clearly show the calculations that produced row 1 of this value function.

Denote row 1 at k = 3 as s0’, s1’, s2’, s3’, and row 1 at k = 2 as s0, s1, s2, s3

R = -1

S0 = 0, s1 = -1, s2 = -1, s3 = -1

S0’ = 0 because it’s the final state

S1’ = ¼ \* s0 + ¼ \* (s1 + R) + ¼ \* (s2 + R) + ¼ \* (s3 + R)

= 0 + ¼ \* -2.7 + ¼ \* -3 + ¼ \* -3

=

**Question E:**

I have a policy with 10 possible actions {a0, a1, ..., a9 } in state s.

q(a0|s)=q(a1|s) > q(ai | s) forall ai \in {a2, ... a9}

a. (5 points) Show the optimization action selection policy for s.

b. (5 points) Show a fair epsilon-greedy action selection policy for s.

c. (5 points) Show a epsilon-soft action selection policy for s that is distinct from b

d. (5 points) What differentiates e-soft and e-greedy action selection policies?

**Question F:**

In MC, the off-policy methods generate an episode with b which is distinct from pi.

a. (5 points) How are we able to estimate Q\_pi using b as the episode-generating policy?

b. (5 points) In computing the returns, why does the iteration process episodes in reverse-time order?

c. (10 points) Suppose we wrote the iteration in forward-time order. Show the recurrent (sample update) expression for the computation of the return.

That is, how do the first two lines of the inner loop on p 110 change.

d. (5 points) Suppose we wrote the iteration in forward-time order. Show how the inner loop of p 111 would have to be modified to process the episodes.

**Question G:**

Suppose we are using on-policy MC on a simulated environment.

a. (5 points) Explain the two approaches for GPI with on-policy MC.

b. (5 points) Which of the two are most appropriate for our problem and why? Why would you choose this over the other?

c. (10 points) Suppose you used the less appropriate version, while your colleague used the more appropriate one. You both have waited a very long time for your algorithms to converge.

After converging, you find that your policy is performing worse than your colleagues.

Explain the reasons why this has likely occurred, and how would you adjust the algorithm you have chosen (note: you can't use your friend's algorithm) to improve performance.