Easy21 Discussion

Petros Karampelas

Introduction

This document is dedicated to discussing the questions in the fifth section of the Easy21 assignment. The goal of answering these questions is to understand how some core notions in reinforcement learning affect algorithm efficiency, depending on the environment and the type of the task.

1 What are the pros and cons of bootstrapping in Easy21?

Bootstrapping is the inherent use of expectations (expected values) in a learning algorithm to estimate a Q-value instead of sampling rewards realized from past visits to the state action pair. More specifically, temporal difference algorithms bootstrap, but Monte Carlo algorithms don't, since by definition they sample from a sum of observations.
An advantage of bootstrapping over sampling is its faster convergence, therefore less run time and computational cost. Methods such as MC can come really close to the true value of Q but there is notable variance during the learning process, thus requiring a large sample before converging. On the other hand, bootstrapping, if not used with intelligent prior distributions (meaning, rational estimates of Q- values before running the environment), it can result to considerable bias. Therefore, the bias-variance tradeoff in choosing from these two algorithm families is apparent. TD(lambda) methods are algorithms that bridge the gap between MC and TD, enabling balance between the two.

2 Would you expect bootstrapping to help more in blackjack or Easy21 ? Why?

Blackjack does not have cards with negative values, however Easy21's half card deck subtracts from the player's sum. Thus, it is safe to assume longer episodes in Easy21. This raises the computational cost of sampling methods, rendering bootstrapping algorithms more efficient for Easy21.

3 What are the pros and cons of function approximation in Easy21?

The pro is less training time as some states are considered dependent, so we do not have to go through all states to come up with the value function. On the other hand, if the value function relationship is very complex-or at least more complex than our estimator- we will get bad predictions.

4 How would you modify the function approximator suggested in this section to get better results in Easy21?

From a visual point of view, a coarse function approximator makes the Q(s,a) shape less smooth, since it is grouping neighbouring state-action pairs together. Q-values estimated by a course function have larger and more evident incontinuities, as shown in the Figures below. What I would do to improve its predictability is manually change the intervals for the binary feature activation according to the results from Monte Carlo. Nonetheless, We can input the interval limits as hyperparameters and use an optimization algorithm to find the best value using MSE with respect to MC as a criterion.

5 Figures

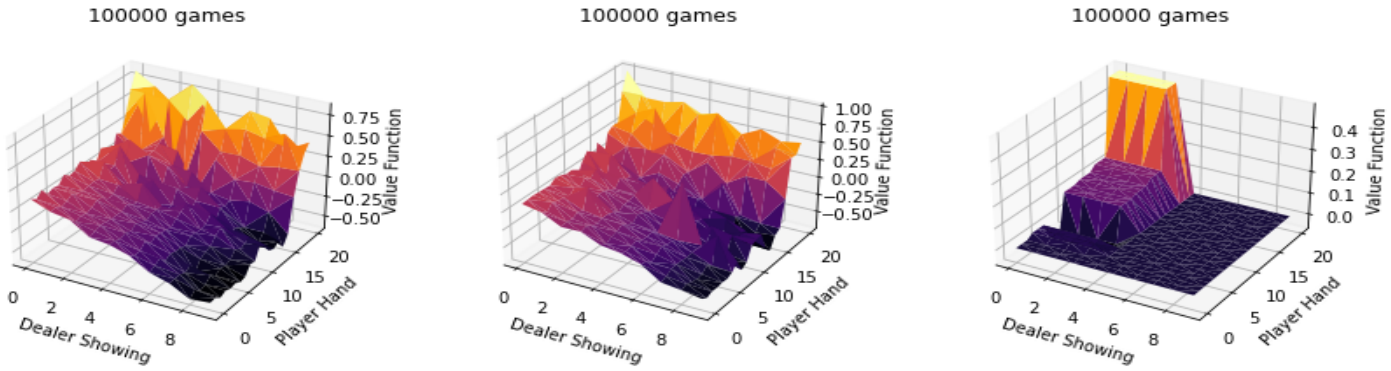

Figure 1: From left to right, this figure shows the MC, TD, and Coarse value functions in this order. While the TD value function is very close to the MC, the coarse function appears to be very different. The coarse approximator "captures" the values of states where the dealer has a low sum and the player has a high sum, but lacks in its predictability for the rest. Perhaps, it needs more episodes in order to converge to the true values.
[MSE Figure below]

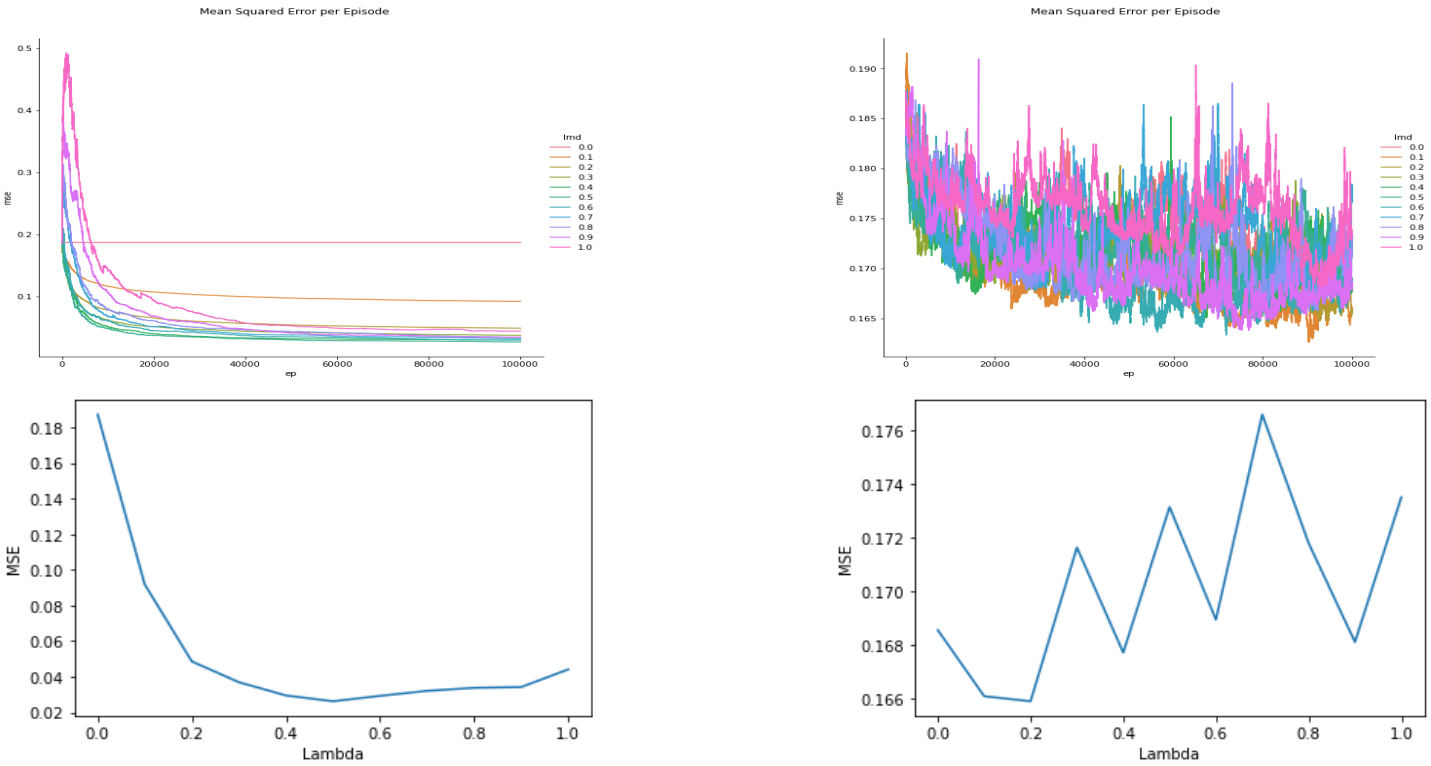

Figure 2: On the left of the figure, the TD MSE per episode and per lambda are visualized. We can see that for many lambda values, MSE is minimized after 40000 episodes. Lambda=0.1 MSE is sub-optimal, while a lambda value of 0 never lowers. The MSE per episode evolution for the coarse function is very noisy. Regarding the MSE per lambda, it appears to circle between low values in even lambda and high values in odd lambda with the exception of 0.9. The lowest MSE is achieved when lambda=0.2.