Noah Poirson

A2 Report

1. The Inputs are read from input.txt. To change inputs change or replace input.txt
2. The inputs are parsed into a list of objects for each state-action-state-probability event
3. The input is passed to the function fit_model
4. The model fitting function runs for six thousand iterations. For every iteration our start state is always the Fairway. While the state has not reached the goal state ("In"), an action is chosen at random from the list of available actions for the current state. Every 500 iterations, the probabilities in the state, action, result tuples probabilities are updated. We used the Bellman Equation to calculate Utility values
5. In the end we print out the final policy for all states

Questions:

1. How did changing the initial value for exploration (starting close to 1 versus starting closer to 0) change the resulting computed policy?

With a higher exploration value the model explores more policies leading to slower convergence. We did not observe a notable difference in policies when adjusting the exploration values in our code.

2. How did you decide when to stop learning?

When we saw that the utility values converged we set our iterations a bit past the point where we saw convergence

3. How did changing the discount value (starting close to 1 versus starting closer to 0) change the resulting policy?

With a discount factor closer to one the policy cares more about the long-term success of the policy. With discount value closer to zero it optimizes for immediate gain. When we changed the discount from 0.9 to 0.5 the fairway and ravine policy switched from shooting at the pin to shooting over the pin. This is because shooting over the pin is a better short-term strategy.