CS747 - Assignment 2

Bhavya Bahl - 150050110

September 9, 2018

1 MDP Encoding of Gambler's problem

- The states in the MDP represent the amount that the gambler can hold at any point of time. Thus there are 99 states for amounts 1 to 99. The actions represent the amount that he bets. The maximum amount that a gambler can bet from any state is 50 and minimum is 1. Hence, there are 50 actions denoting the bets. **Note: In the output of the optimal policy the actions represent the indices and hence, have to be incremented by 1 to get the betting amount**

- There are two additional states, 0 and 100. The value of state 0 is 0 and the value of state 100 is 1. To make sure that we get these values in the optimal policy, all the actions from state 100 end up in state 0 with probability 1 and reward 1 and all the actions from state 0 go back to state 0 with probability 1 and reward 0.

- The reward of all other actions, irrespective of where they originate and terminate, is 0. For any state s, other that 0 and 100, action a goes to state $s + a$ with probability ph and to state $s - a$ with probability $1 - ph$. If $a > min(s, 100 - s)$, then that action goes to terminal state with probability 1 without getting any reward. This makes sure that illegal actions are not allowed in optimal policy.

- The discount factor has been set to 0.999 and the MDP is episodic. Ideally, this should have been 1, but while solving the bellman equations to evaluate a policy, we get a singular matrix if it is 1. This is because all transitions from terminal state end up in that state without any reward and the solution behaves weirdly.

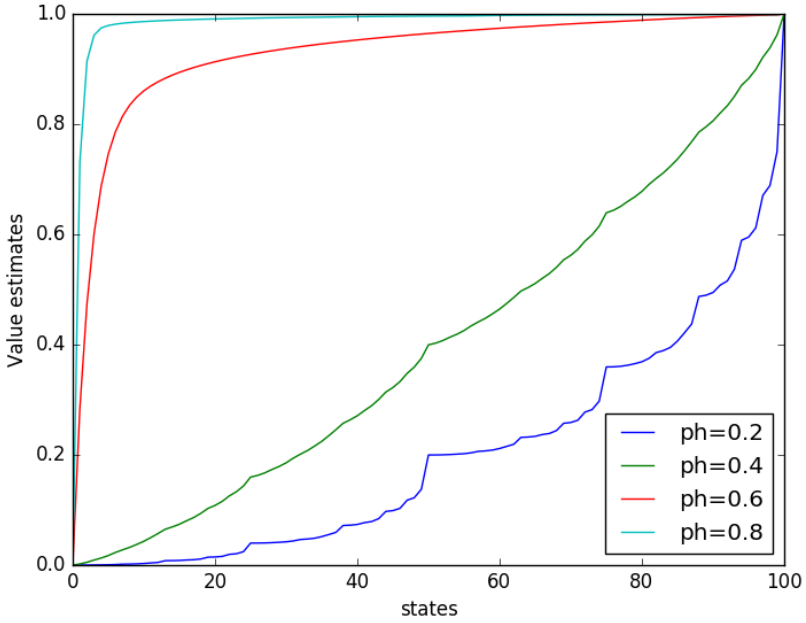

Figure 1: Value function of the optimal policy for various value of ph.

2 Results and Observation

- It is intuitive that as the probability of heads coming up on a coin toss increases, probability to reach the state 100 from each state which is the value function also, increases. This is because at each time step the probability of the gambler to win the bet increases.
- As ph increases, not only the value of states close to 100 becomes close to 1, but also of the states below 50. At each state(i), even if the gambler bids just \$1, the probability of winning from that state is equal to

$$\frac{1-(\frac{1-ph}{ph})^i}{1-(\frac{1-ph}{ph})^{100}}$$

  [3], which is very close to one if $ph > 0.5$. Hence, optimal policy will definitely be better.
- For states close to hundred, gambler will have to lose a lot more times to reach 0 than to reach 100. That is why the probability of success from these states is high. Similarly for states close to 0.

References

[1] Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto, 2nd edition 2018.

[2] Python PuLP documentation

[3] Gambler's Ruin