Gambler's Problem

Ashwini Venkatesh - av28895
ashuven63@utexas.edu

Introduction

A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, he wins as many dollars as he has staked on that flip; if it is tails, he loses his stake. The game ends when the gambler wins by reaching his goal of $100, or loses by running out of money. On each flip, the gambler must decide what portion of his capital to stake, in integer numbers of dollars [1]. The assignment looks at modelling this problem as an undiscounted, episodic, finite MDP.

Approach

The gambler's capital represents the state **s ∈ {1, 2, .., 99}.**
The stake which the gambler places on each coin toss is the action he takes. Thus for each state s, the possible actions are **a ∈ {0, 1, 2, .., min(s,100 - s)}.** p_h is the probability the coin toss results in a head. We apply value iteration to obtain the optimal policy for this problem. The following algorithm is implemented to obtain the value of each state of the MDP.

Value iteration

Initialize array V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)

Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)\big[r + \gamma V(s')\big]$

Figure 1: Value Iteration algorithm for optimal policy

Experiments

Probability of coin toss

This experiment looks at how the optimal policy changes with change in the probability of a coin toss. In this experiment the action with min value is selected in case of a tie at the argmax step. This would give us policies which are low risk. The estimate value graphs show how the values of the state improves across each iteration.

Figure 2 shows the behaviour when p_h is less than 0.5. At each step the optimal policy tries to take take an action which you result in state close to 50. Also at each state you place the bet such that in worst case you go back to one of the spikes in the graph which you have just passed.

Figure 3 shows the behaviour when p_h is greater than equal to 0.5. The optimal policy graph shows that the optimal action is always 1 if it's low risk. Since the probability of getting a head on the coin toss is greater than half, its safest to best 1 since even if you lose you just go back a step. Since the number of steps taken here is not penalized, the action taken is one at each state. If we observe the values of the states, almost all the states have values of 1 because of the higher probability of reaching the goal from every state given the high probability of obtaining a heads on the coin toss.

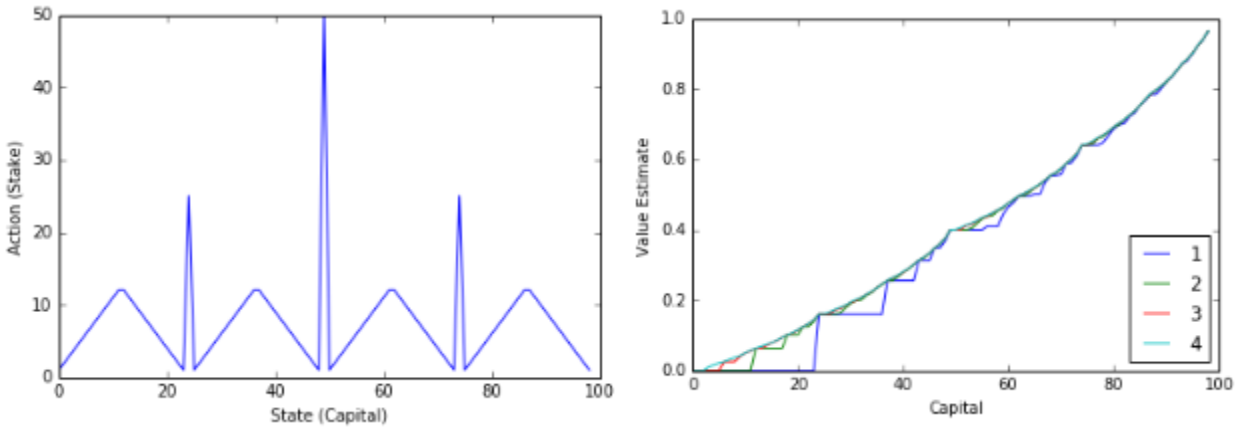

Figure 2: Optimal Policy and value estimates across iterations for $p_h < 0.5$

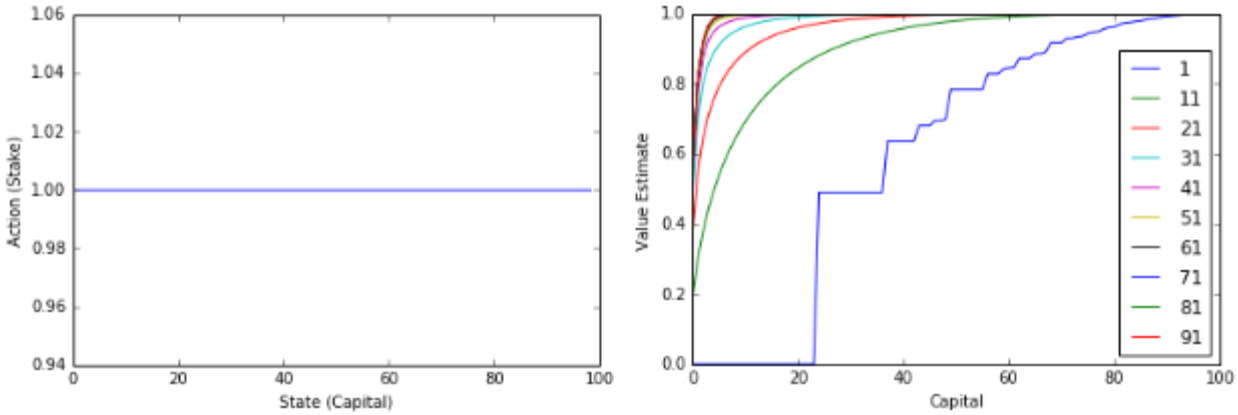

Figure 3: Optimal Policy and value estimates across iterations for $p_h \geq 0.5$

Effect of argmax over action

In the previous experiment, the minimum action was selected, i.e the optimal policy is low risk. When the max action is selected we obtain a high risk policy. Figure 4 shows the optimal policies obtained for different values of $p_{h.}$ When $p_h <= 0.5$, the policy is bet whatever you can since this would take you to a state closer to the goal. However once $p_h > 0.5$, at lower states you bet a low value and start betting higher once you reach higher states when you have confidence that you have achieved enough capital that even if you lose you can go back to a safe state where you can start betting again. However, I am unsure as to why the point at which you start betting higher is different for different probabilities.

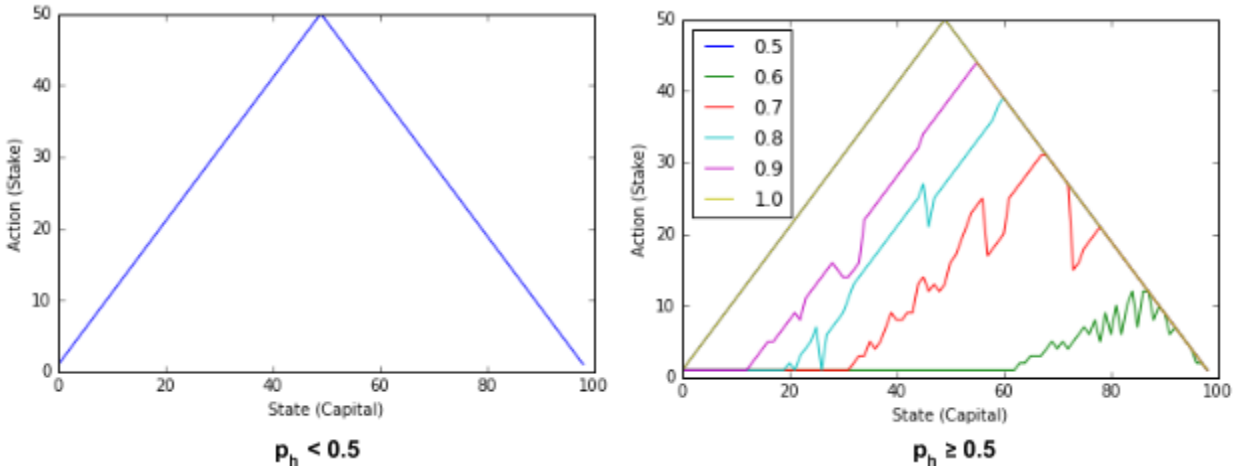

Figure 4: High Risk optimal policies

Penalizing the steps

We can also examine what the optimal policy when each step taken is given a small negative reward. In this experiment the action with min value is selected in case of a tie at the argmax step. When p_h <= 0.5, the optimal policy is same as the high risk optimal policy since we try to reduce the number of steps. When p_h > 0.5, we place lower bets at lower states since we have a higher chance of making it to the goal because of higher p_h and place higher bets in higher states thus minimizing the number of steps taken. Compared to the graph in Figure 4, we start placing higher bets in the earlier states so that we can reduce the number of steps taken to reach the goal.

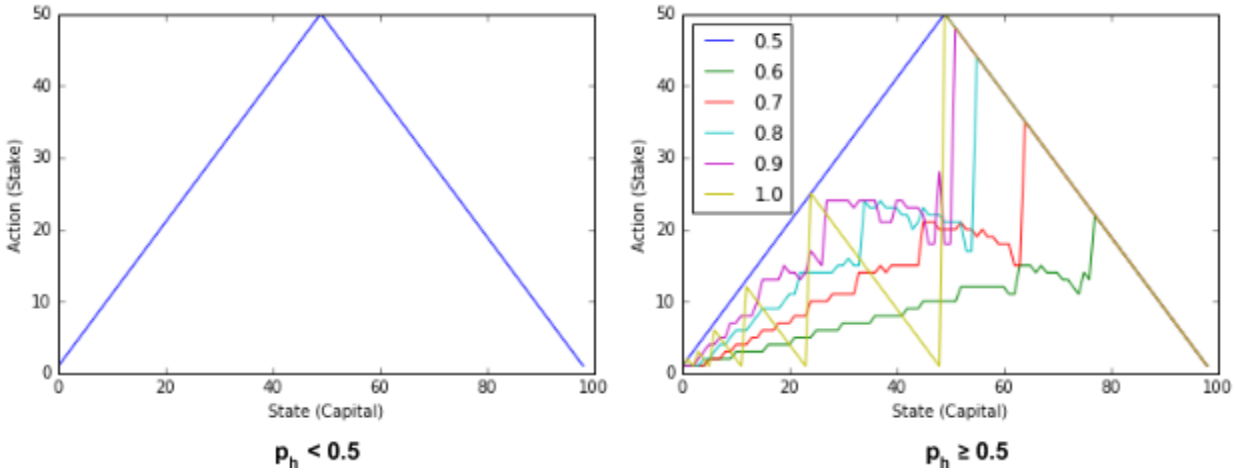

Figure 5: Optimal policy when the number of steps taken to reach the goal is penalized

References

[1] Reinforcement Learning:An Introduction, Second Edition, Richard S. Sutton and Andrew G. Barto