Machine Problem 1 - Reinforcement Learning
Part 2 - Experimentation

Nathan Havens
havens2
9/29/14

1. **a)** The agent simply travels to the top of the world and then sits there, so the end reward is 0.0. The reason is because it doesn't perform any random actions. The agent sees that traveling up yields a reward of 0.0, but that is better than crossing the slippery slope which gives a negative reward. Since there is no exploration that will cause the agent to explore an area that seems worse than other options, it will never find the goal states and will therefore sit in a sub-optimal space thinking that it has the best net reward.

b) Now the agent makes it to the goal states and ends up with a net reward at the end of the run. Since epsilon is now set to 0.1, the agent will occasionally explore and because of that, it will find the goals and get a positive reward. The value of the total reward at the end of each episode varies a little bit. This is because the agent will sometimes explore a random path even though it knows a good path and sometimes the path it takes is not a good one and ends with a very low reward.

c) The performance of the agent at epsilon = 0.5 is not good at all. Because epsilon is so high, the agent explores random actions roughly half the time. This means that a lot of times, it picks a non-optimal route and ends up with a negative reward. In fact, the end reward is rarely positive, and even when it is, its quite small.

2. a) The agent ends up sitting in the top left space. With epsilon set low, it won't explore much and for that reason, it followed the same path that it did when epsilon was 0.0 in the WorldWithoutThief scenario. Its rewards are entirely negative, so it performs terribly.

b) With KnowsThief = y, the agent now performs really well. It quickly finds a good path to follow and while the thief's position allows it, the agent keeps to the path to gain good rewards. I does have to take time to avoid the thief a lot of times, but that doesn't cause it to get a bad reward because it moves back onto the optimal path quickly.

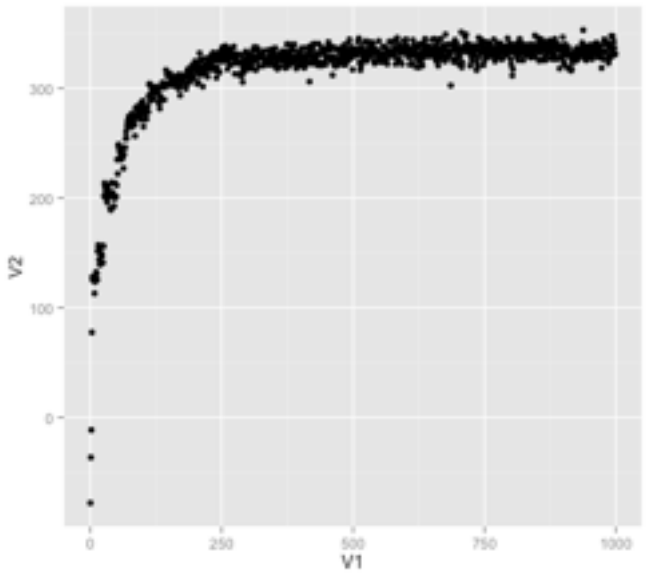
c) In order to find both the best alpha and best epsilon value, I started by forming bounds around the best value and narrowing the range in half every iteration. I maximized the value of each variable independently because once one is maximized, the other one will be based off a max value to begin with, so in the end it will fully maximized. I started by calculating the average of the total reward and then averaged that 4 times to get a consistent, more accurate result.

Alpha: 0.0 => -104
1.0 => 89.5
0.5 => 238.5
0.25 => 266.9
0.125 => 273.4
0.0625 => 270.5
0.09375 => 273.6
0.109 => 273.7
0.117 => 273.8
0.101 => 273.6
The 0.09375 to 0.117 is a very small range, so the best alpha is approximately 0.1

Epsilon: 0.0 => 100.5
1.0 => -103.6
0.5 => -179.5
0.25 => 35.6
0.125 => 179.9
0.0625 => 258.4
0.03125 => 294.5
0.015625 => 306.9
0.007813 => 306.4
0.0117 => 308.7
0.01 => 315
So the best epsilon is approximately 0.01.

So, alpha = 0.1, epsilon = 0.01.

3. The first plot is the reward over 1000 episodes. The x-axis is the episode number and the y-axis is the reward value.



This plot is the average over 10 iterations of the episodes. Again, the x-axis is the episode number and the y-axis is the reward value.