Solving MDPs and Multi-Armed Bandit Problems

Joe Shymanski

December 7, 2023

1 MDPs

The gridworlds and Wumpus world have an 80% probability of the intended action and a 10% probability each for either of the adjacent movements.

For all experiments, $\gamma = .9$. Additionally, both α and ϵ are decayed at an exponential rate according to the episode number. This ensures proper exploration in the beginning and exploitation by the end of the trials. Each experiment features 5000 episodes of the agent navigating from the start state to a terminal state.

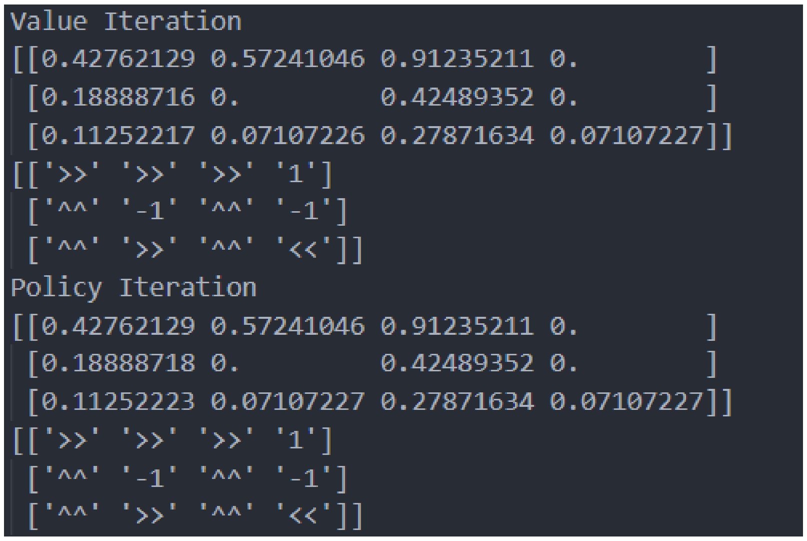
For the 4×3 gridworld, a pit with reward -1 is placed in the middle of the grid instead of a wall, since pits are already implemented in the provided code but not internal wall obstacles.

1.1 World Model Known

Figure 1 shows the results of value and policy iteration for the 4×3 gridworld. As you can see, both algorithms found the exact same policies with nearly identical Q-tables.

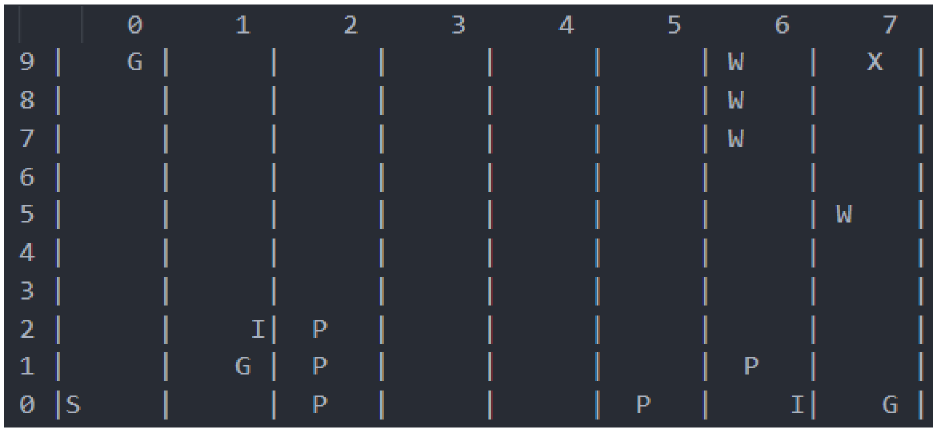
The Wumpus world is shown in Figure 2. The policies learned by both value and policy iteration are shown in Figures 3 and 4, respectively.

Across all the experiments, value iteration ran significantly faster than policy iteration. However, the gap between the two algorithms narrowed as the world size increased.



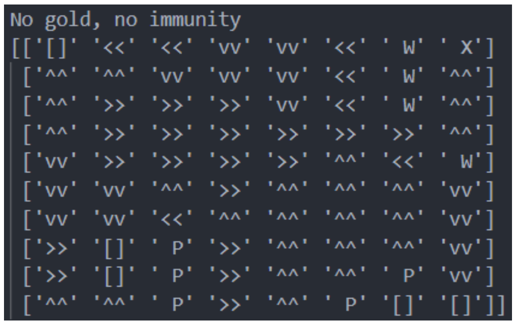
```
Value Iteration
[[0.42762129 0.57241046 0.91235211 0.        ]
 [0.18888716 0.         0.42489352 0.        ]
 [0.11252217 0.07107226 0.27871634 0.07107227]]
[['>>' '>>' '>>' '1']
 ['^^' '-1' '^^' '-1']
 ['^^' '>>' '^^' '<<']]
Policy Iteration
[[0.42762129 0.57241046 0.91235211 0.        ]
 [0.18888718 0.         0.42489352 0.        ]
 [0.11252223 0.07107227 0.27871634 0.07107227]]
[['>>' '>>' '>>' '1']
 ['^^' '-1' '^^' '-1']
 ['^^' '>>' '^^' '<<']]
```

Figure 1: 4×3 gridworld policies when the world model is known.



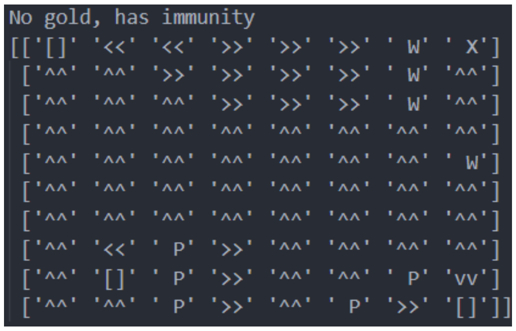
```
      0      1      2      3      4      5      6      7
9 |   G  |      |      |      |      |      | W    |   X  |
8 |      |      |      |      |      |      | W    |      |
7 |      |      |      |      |      |      | W    |      |
6 |      |      |      |      |      |      |      |      |
5 |      |      |      |      |      |      |      | W    |
4 |      |      |      |      |      |      |      |      |
3 |      |      |      |      |      |      |      |      |
2 |      |     I|  P   |      |      |      |      |      |
1 |      |    G |  P   |      |      |      |  P   |      |
0 |S     |      |  P   |      |      |  P   |     I|    G |
```

Figure 2: Wumpus world.



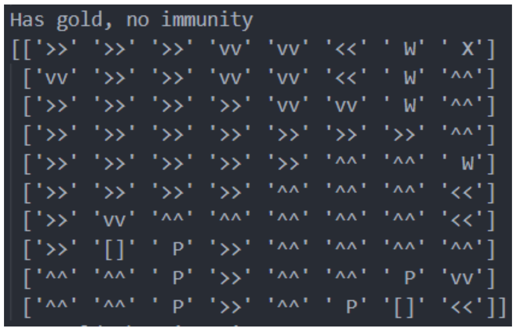
```
No gold, no immunity
[['[]' '<<' '<<' 'vv' 'vv' '<<' ' W' ' X']
 ['^^' '^^' 'vv' 'vv' 'vv' '<<' ' W' '^^']
 ['^^' '>>' '>>' '>>' 'vv' '<<' ' W' '^^']
 ['^^' '>>' '>>' '>>' '>>' '>>' '>>' '^^']
 ['vv' '>>' '>>' '>>' '>>' '^^' '<<' ' W']
 ['vv' 'vv' '^^' '>>' '^^' '^^' '^^' 'vv']
 ['vv' 'vv' '<<' '^^' '^^' '^^' '^^' 'vv']
 ['>>' '[]' ' P' '>>' '^^' '^^' '^^' 'vv']
 ['>>' '[]' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '[]' '[]']]
```

(a) No gold, no immunity



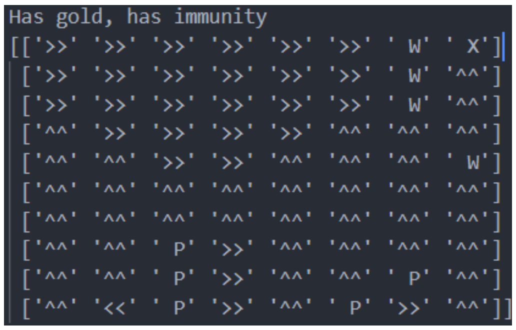
```
No gold, has immunity
[['[]' '<<' '<<' '>>' '>>' '>>' ' W' ' X']
 ['^^' '^^' '>>' '>>' '>>' '>>' ' W' '^^']
 ['^^' '^^' '^^' '>>' '>>' '>>' ' W' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' ' W']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '<<' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '[]' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '>>' '[]']]
```

(b) No gold, has immunity



```
Has gold, no immunity
[['>>' '>>' '>>' 'vv' 'vv' '<<' ' W' ' X']
 ['vv' '>>' '>>' 'vv' 'vv' '<<' ' W' '^^']
 ['>>' '>>' '>>' '>>' 'vv' 'vv' ' W' '^^']
 ['>>' '>>' '>>' '>>' '>>' '>>' '>>' '^^']
 ['>>' '>>' '>>' '>>' '>>' '^^' '^^' ' W']
 ['>>' '>>' '>>' '>>' '^^' '^^' '^^' '<<']
 ['>>' 'vv' '^^' '^^' '^^' '^^' '^^' '<<']
 ['>>' '[]' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '[]' '<<']]
```

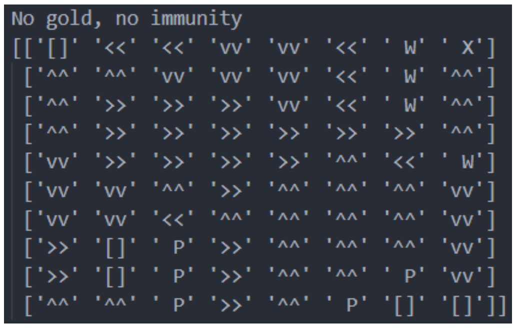
(c) Has gold, no immunity



```
Has gold, has immunity
[['>>' '>>' '>>' '>>' '>>' '>>' ' W' ' X']
 ['>>' '>>' '>>' '>>' '>>' '>>' ' W' '^^']
 ['>>' '>>' '>>' '>>' '>>' '>>' ' W' '^^']
 ['^^' '>>' '>>' '>>' '>>' '^^' '^^' '^^']
 ['^^' '^^' '>>' '>>' '^^' '^^' '^^' ' W']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' ' P' '^^']
 ['^^' '<<' ' P' '>>' '^^' ' P' '>>' '^^']]
```

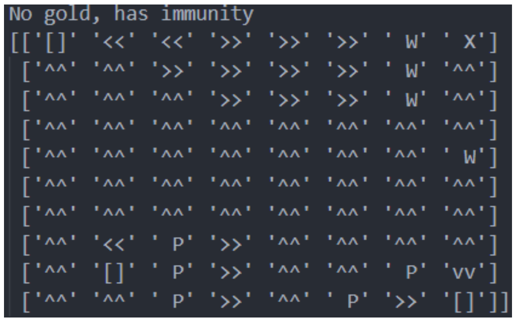
(d) Has gold, has immunity

Figure 3: Value iteration policies for Wumpus world.



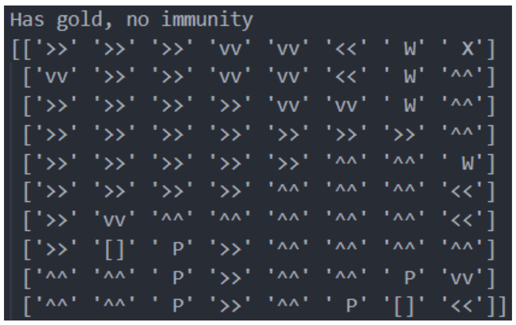
```
No gold, no immunity
[['[]' '<<' '<<' 'vv' 'vv' '<<' ' W' ' X']
 ['^^' '^^' 'vv' 'vv' 'vv' '<<' ' W' '^^']
 ['^^' '>>' '>>' '>>' 'vv' '<<' ' W' '^^']
 ['^^' '>>' '>>' '>>' '>>' '>>' '>>' '^^']
 ['vv' '>>' '>>' '>>' '>>' '^^' '<<' ' W']
 ['vv' 'vv' '^^' '>>' '^^' '^^' '^^' 'vv']
 ['vv' 'vv' '<<' '^^' '^^' '^^' '^^' 'vv']
 ['>>' '[]' ' P' '>>' '^^' '^^' '^^' 'vv']
 ['>>' '[]' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '[]' '[]']]
```

(a) No gold, no immunity



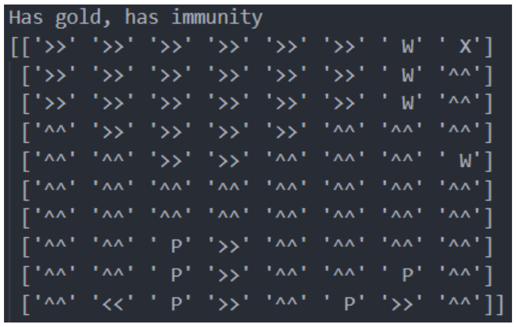
```
No gold, has immunity
[['[]' '<<' '<<' '>>' '>>' '>>' ' W' ' X']
 ['^^' '^^' '>>' '>>' '>>' '>>' ' W' '^^']
 ['^^' '^^' '^^' '>>' '>>' '>>' ' W' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' ' W']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '<<' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '[]' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '>>' '[]']]
```

(b) No gold, has immunity



```
Has gold, no immunity
[['>>' '>>' '>>' 'vv' 'vv' '<<' ' W' ' X']
 ['vv' '>>' '>>' 'vv' 'vv' '<<' ' W' '^^']
 ['>>' '>>' '>>' '>>' 'vv' 'vv' ' W' '^^']
 ['>>' '>>' '>>' '>>' '>>' '>>' '>>' '^^']
 ['>>' '>>' '>>' '>>' '>>' '^^' '^^' ' W']
 ['>>' '>>' '>>' '>>' '^^' '^^' '^^' '<<']
 ['>>' 'vv' '^^' '^^' '^^' '^^' '^^' '<<']
 ['>>' '[]' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' ' P' 'vv']
 ['^^' '^^' ' P' '>>' '^^' ' P' '[]' '<<']]
```

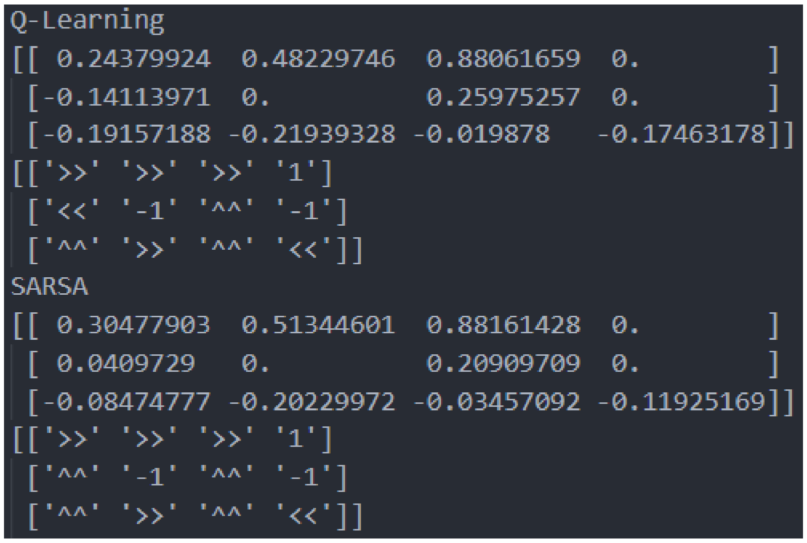
(c) Has gold, no immunity



```
Has gold, has immunity
[['>>' '>>' '>>' '>>' '>>' '>>' ' W' ' X']
 ['>>' '>>' '>>' '>>' '>>' '>>' ' W' '^^']
 ['>>' '>>' '>>' '>>' '>>' '>>' ' W' '^^']
 ['^^' '>>' '>>' '>>' '>>' '^^' '^^' '^^']
 ['^^' '^^' '>>' '>>' '^^' '^^' '^^' ' W']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '^^' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' '^^' '^^']
 ['^^' '^^' ' P' '>>' '^^' '^^' ' P' '^^']
 ['^^' '<<' ' P' '>>' '^^' ' P' '>>' '^^']]
```

(d) Has gold, has immunity

Figure 4: Policy iteration policies for Wumpus world.



```
Q-Learning
[[ 0.24379924  0.48229746  0.88061659  0.        ]
 [-0.14113971  0.          0.25975257  0.        ]
 [-0.19157188 -0.21939328 -0.019878   -0.17463178]]
[['>>' '>>' '>>' '1']
 ['<<' '-1' '^^' '-1']
 ['^^' '>>' '^^' '<<']]
SARSA
[[ 0.30477903  0.51344601  0.88161428  0.        ]
 [ 0.0409729   0.          0.20909709  0.        ]
 [-0.08474777 -0.20229972 -0.03457092 -0.11925169]]
[['>>' '>>' '>>' '1']
 ['^^' '-1' '^^' '-1']
 ['^^' '>>' '^^' '<<']]
```

Figure 5: 4×3 gridworld policies when the world model is unknown.

1.2 World Model Unknown

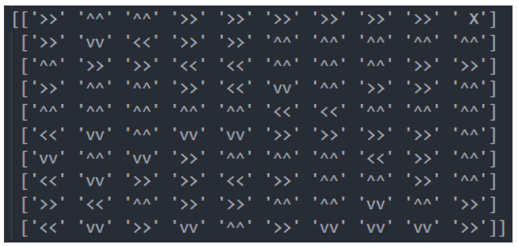
The results of Q-learning and SARSA in the 4x3 gridworld can be found in Figure 5. Both agents learned similar Q-tables with values close to those found earlier in Figure 1 using value and policy iteration. While the SARSA agent was able to obtain the optimal policy, the Q-learning agent only made one mistake and it is the best alternative action to the optimal one.

Neither Q-Learning nor SARSA seemed to perform well in the Wumpus world, which is not pictured in this report. I believe this world is too large and complicated for the algorithms to learn effectively within 5000 iterations.

Across all experiments, the SARSA algorithm ran slightly faster than the Q-learning algorithm, on average.

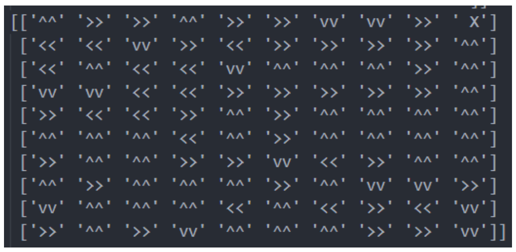
2 Multi-Armed Bandit Problems

The multi-armed bandits in this project are created using a specified number of arms and the standard deviation of the arm payouts. Each arm payout is sampled from a Normal distribution. The mean rewards of the arms are evenly spaced out in the interval $[0, 1]$, each with the same standard deviation. Figures 7 and 8 demonstrate these payout distributions for each arm of the 2-armed bandit and 10-armed bandit, respectively.



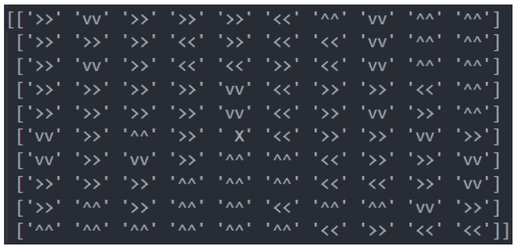
```
[['>>' '^^' '^^' '>>' '>>' '>>' '>>' '>>' '>>' ' X']
 ['>>' 'vv' '<<' '>>' '>>' '^^' '^^' '^^' '^^' '^^']
 ['^^' '>>' '>>' '<<' '<<' '^^' '^^' '^^' '>>' '>>']
 ['>>' '^^' '^^' '>>' '<<' 'vv' '^^' '>>' '>>' '^^']
 ['^^' '^^' '^^' '^^' '^^' '<<' '<<' '^^' '^^' '^^']
 ['<<' 'vv' '^^' 'vv' 'vv' '>>' '>>' '>>' '>>' '^^']
 ['vv' '^^' 'vv' '>>' '^^' '^^' '^^' '<<' '>>' '^^']
 ['<<' 'vv' '>>' '>>' '<<' '>>' '^^' '^^' '>>' '^^']
 ['>>' '<<' '^^' '>>' '>>' '^^' '^^' 'vv' '^^' '>>']
 ['<<' 'vv' '>>' 'vv' '^^' '>>' 'vv' 'vv' 'vv' '>>']]
```

(a) Corner goal, Q-learning



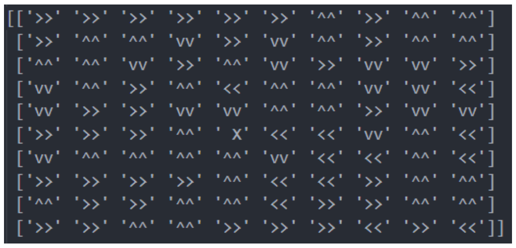
```
[['^^' '>>' '>>' '^^' '>>' '>>' 'vv' 'vv' '>>' ' X']
 ['<<' '<<' 'vv' '>>' '<<' '>>' '>>' '>>' '>>' '^^']
 ['<<' '^^' '<<' '<<' 'vv' '^^' '^^' '^^' '>>' '^^']
 ['vv' 'vv' '<<' '<<' '>>' '>>' '>>' '>>' '>>' '^^']
 ['>>' '<<' '<<' '>>' '^^' '>>' '^^' '^^' '^^' '^^']
 ['^^' '^^' '^^' '<<' '^^' '>>' '^^' '^^' '^^' '^^']
 ['>>' '^^' '^^' '>>' '>>' 'vv' '<<' '>>' '^^' '^^']
 ['^^' '>>' '^^' '^^' '^^' '>>' '^^' 'vv' 'vv' '>>']
 ['vv' '^^' '^^' '^^' '<<' '^^' '<<' '>>' '<<' 'vv']
 ['>>' '^^' '>>' 'vv' '^^' '^^' '^^' '>>' '>>' 'vv']]
```

(b) Corner goal, SARSA



```
[['>>' 'vv' '>>' '>>' '>>' '<<' '^^' 'vv' '^^' '^^']
 ['>>' '>>' '>>' '<<' '>>' '<<' '<<' 'vv' '^^' '^^']
 ['>>' 'vv' '>>' '<<' '<<' '>>' '<<' 'vv' '^^' '^^']
 ['>>' '>>' '>>' '>>' 'vv' '<<' '>>' '>>' '<<' '^^']
 ['>>' '>>' '>>' '>>' 'vv' '<<' '>>' 'vv' '>>' '^^']
 ['vv' '>>' '^^' '>>' ' X' '<<' '>>' '>>' 'vv' '>>']
 ['vv' '>>' 'vv' '>>' '^^' '^^' '<<' '>>' '>>' 'vv']
 ['>>' '>>' '>>' '^^' '^^' '^^' '<<' '<<' '>>' 'vv']
 ['>>' '^^' '>>' '^^' '^^' '<<' '^^' '^^' 'vv' '>>']
 ['^^' '^^' '^^' '^^' '^^' '^^' '<<' '>>' '<<' '<<']]
```

(c) Center goal, Q-learning



```
[['>>' '>>' '>>' '>>' '>>' '>>' '^^' '>>' '^^' '^^']
 ['>>' '^^' '^^' 'vv' '>>' 'vv' '^^' '>>' '^^' '^^']
 ['^^' '^^' 'vv' '>>' '^^' 'vv' '>>' 'vv' 'vv' '>>']
 ['vv' '^^' '>>' '^^' '<<' '^^' '^^' 'vv' 'vv' '<<']
 ['vv' '>>' '>>' 'vv' 'vv' '^^' '^^' '>>' 'vv' 'vv']
 ['>>' '>>' '>>' '^^' ' X' '<<' '<<' 'vv' '^^' '<<']
 ['vv' '^^' '^^' '^^' '^^' 'vv' '<<' '<<' '^^' '<<']
 ['>>' '>>' '>>' '>>' '^^' '<<' '<<' '>>' '^^' '^^']
 ['^^' '>>' '>>' '^^' '^^' '<<' '>>' '>>' '^^' '^^']
 ['>>' '>>' '^^' '^^' '>>' '>>' '>>' '<<' '>>' '<<']]
```

(d) Center goal, SARSA

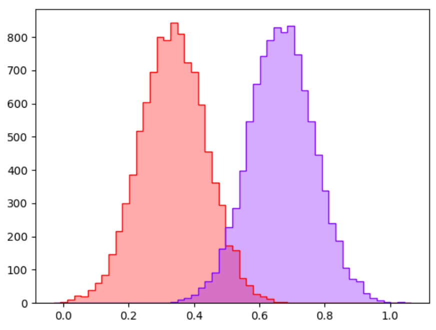
Figure 6: Learned policies in 10x10 gridworlds.

Clearly, as the standard deviation of each arm's payout increases, the overlap of the distributions grows significantly. This causes the overall distributions appear more and more normal. This overlap is also greater when there are more bandit arms present.

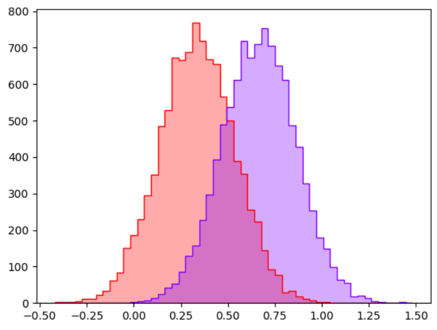
Several simulations were run in order to compare cumulative loss (regret) between four algorithms. The first is the Upper Confidence Bound (UCB) algorithm, and the last three are versions of the ϵ-greedy algorithm with $\epsilon \in \{.1, .2, .3\}$. A simulation consist of m episodes, each with A arm pulls ($A = \texttt{num\_arms}$), to create $m \cdot A = h$ total samples. Their results are shown in Figures 9 and 10. All simulations consisted of 2000 episodes, generating 4000 samples for the 2-armed bandit and 20000 for the 10-armed bandit.

Cumulative regret after n samples is calculated as $\sum_{t=1}^{n}[r^* - R(a_t)]$ where r^* is the largest mean arm reward and $R(a_t)$ is the reward from sampling action a at time t. Thus, one can obtain negative values for cumulative regret, since $R(a_t)$ can sometimes be larger than r^*.

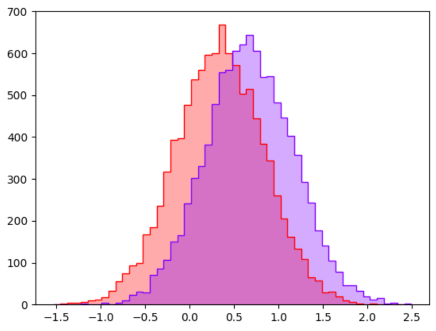
As you can see, increasing the standard deviation of the payouts magnifies the variability in regret within the same number of episodes for all the algorithms. This is because the payout distributions of each arm highly overlap for large standard deviations. Thus, it takes more episodes for the



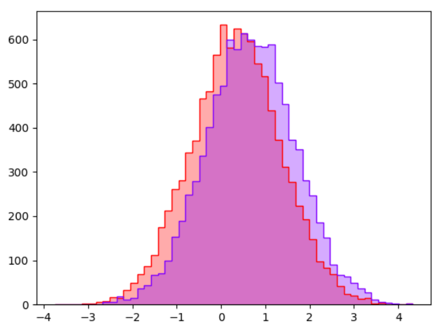
(a) Payout SD = .1



(b) Payout SD = .2

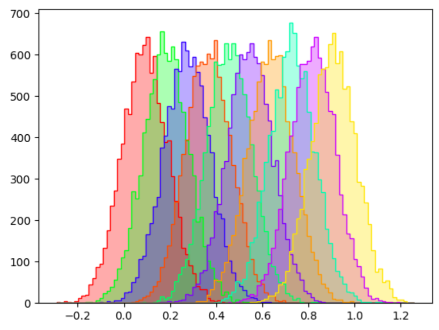


(c) Payout SD = .5

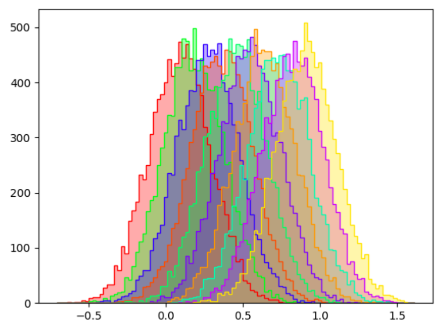


(d) Payout SD = 1

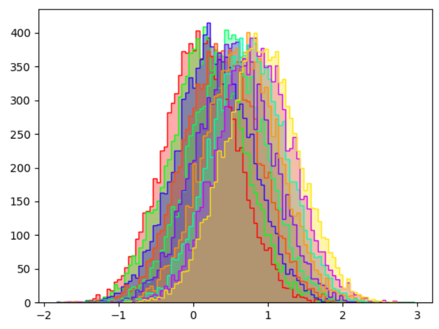
Figure 7: Arm payout distributions using a 2-armed bandit.



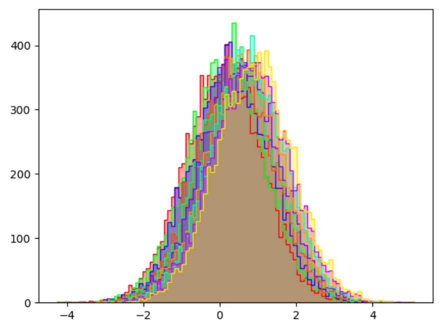
(a) Payout SD = .1



(b) Payout SD = .2

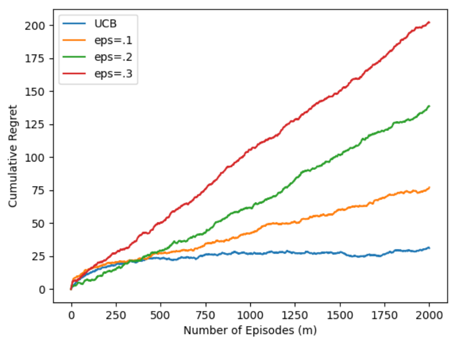


(c) Payout SD = .5

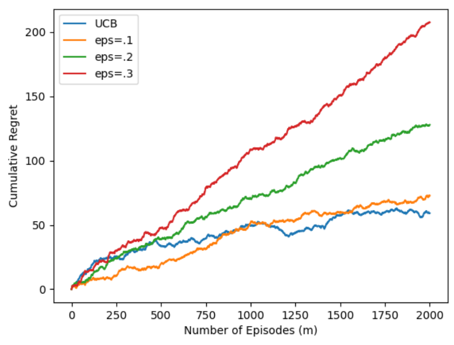


(d) Payout SD = 1

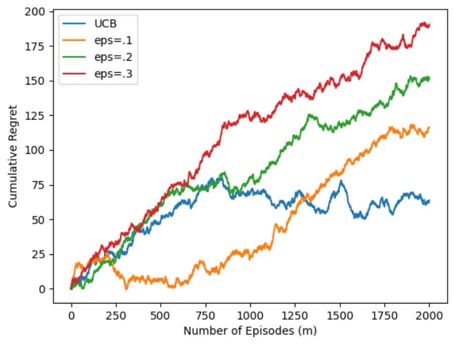
Figure 8: Arm payout distributions using a 10-armed bandit.



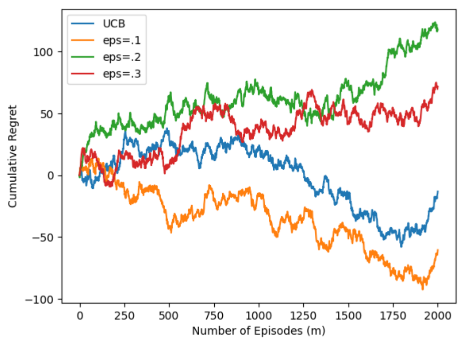
(a) Payout SD = .1



(b) Payout SD = .2

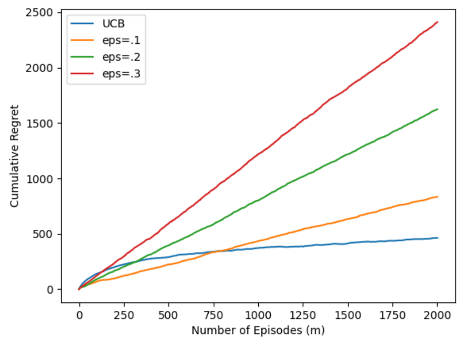


(c) Payout SD = .5

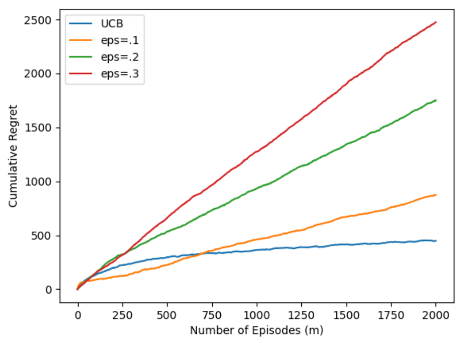


(d) Payout SD = 1

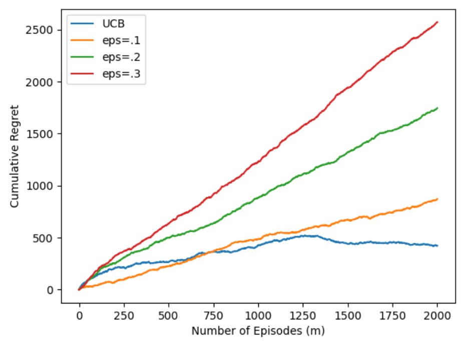
Figure 9: Cumulative regret using a 2-armed bandit.



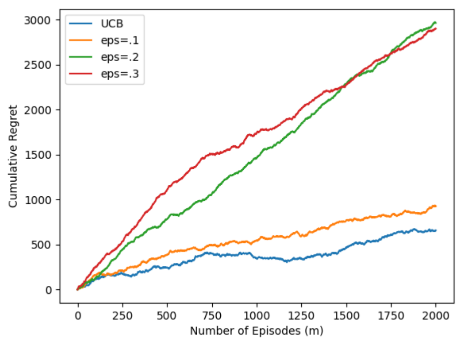
(a) Payout SD = .1



(b) Payout SD = .2



(c) Payout SD = .5



(d) Payout SD = 1

Figure 10: Cumulative regret using a 10-armed bandit.

relative ordering of the mean arm rewards to be apparent.

Additional arms decrease regret variance within the same number of episodes. This may seem counterintuitive, since more arms means more distribution overlap. However, I believe there are two factors which lead to this observation. The first is that the number of samples per episode increases with the number of bandit arms. The second is that the increased number of arms smooths out the suboptimal payouts from the regrettable arm pulls, creating a more continuous spectrum of payouts rather than a harsher discrete set.

The UCB algorithm has lower cumulative regret than all of the ϵ-greedy variations in nearly every simulation. In fact, Figure 10a seems to suggest that, after a sufficiently large number of episodes, the UCB algorithm will always win out. It appears the cumulative regret of the UCB algorithm is correlated to the number of episodes in either a square root or logarithmic fashion. In contrast, the regret accumulated from the ϵ-greedy algorithm is linearly correlated to the number of episodes, with larger ϵ values producing larger slopes.

Figure 9d shows the only scenario in which UCB did not finish with the lowest regret. Using the observations made early, we can deduce that the high payout standard deviation and low number of arms causes this increased variation. Undoubtedly, more episodes would yield trends similar to all of the other plots.