Techniques for solving Markov Decision Processes & Multi-Arm Bandit Problems

Luke Runnels

December 7, 2023

1 Domains

1.1 Markov Decision Processes modelled as an available world

The following world domains were used for evaluating the effectiveness and scale-up ability for known algorithms with solving model-available Markov Decision Processes.

Each world features a starting location and terminal location, with either rewards or punishments. These worlds are implemented using a MDP library developed by Chad Crawford in the TU Master's Group. [1]

1.1.1 4x3 Grid World

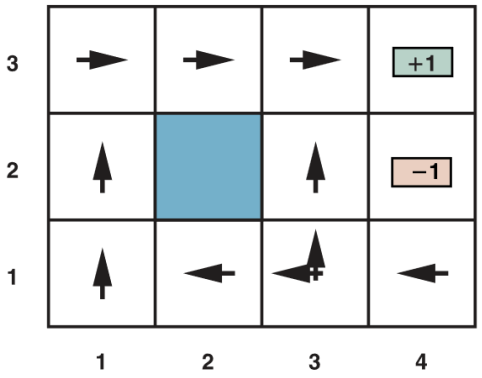

Figure 1: A 4x3 world with a +1 reward in (4, 3), and -1 punishment in (4, 2)

This world is a simple 4x3 grid world proposed by Russell el at. [3]. It features a starting location at (1, 1) with a +1 reward at (4, 3) and a -1 punishment at (4, 2). For every non terminal location, a -0.01 punishment is applied, with the actions of going up, down, left, or right. It should be noted that this paper does will not include a wall at (2, 2) due to technical challenges with the given codebase.

1.1.2 Wumpus World

This world is a 7x9 world grid world. It features two types of punishment terminals. There are pits at locations (2, 0), (2, 1), (5, 0), and (6, 1) that apply a -1 punishment. There are wumpus creatures at locations (6, 8), (6, 7), and (7, 5) that apply a -10 punishment. For some variability, there is also weaker pit of -0.5 at (2, 2) and super hurtful wumpus at (6, 9). A goal location is located at (7, 9) with a reward of +10.

The world also features two non-terminal objects. There is gold at (0, 9), (7, 0), and (1, 1), which will apply a +10 reward. There is immunity at (6, 0) and (1, 2), which will protect the agent if it hits a wumpus.

The world finally features its starting location at (0, 0). For every non terminal, a -0.01 reward is applied with actions up, down, left, or right. At the special non terminal locations, there is an additional action of 'pick-up' with rewards specified above.

1.2 Markov Decision Processes modelled as an unknown world

The following world domains were used for evaluating the effectiveness and scale-up ability for known algorithms with solving model-unknown Markov Decision Processes.

Just like with the known worlds, each world features a starting location and terminal locations, which were implemented using the MDP library by the TU Master's group [1].

1.2.1 4x3 Grid World

The simple 4x3 proposed by Russell el at. [3] was also used for evaulating unknown world algorithms.

1.2.2 10x10 Grid World with a +1 reward at (10, 10)

This is a simple grid 10x10 world with a starting location at (2, 2) and a terminal location at (10, 10) with a reward of +1.

1.2.3 10x10 Grid World with a +1 reward at (5, 5)

This is a simple grid 10x10 world with a starting location at (2, 2) and a terminal location at (5, 5) with a reward of +1.

1.3 Multi-Arm Bandit Problem

For evaluating known algorithms for solving Multi-Arm Bandit Problems, a simple simulator was used with n arms. Each payout for pulling the ith arm is modelled as a Gaussian Distribution $N(\mu_i, \sigma_i^2)$,

$$\mu_i = \frac{i+1}{n\_arms+1} \tag{1}$$

and σ_i^2 is a tunable parameter between 0 and 1. The simulator was developed by Robert Geraughty and the TU Masters Group. [2]

2 Algorithm Explanations

2.1 Solving available worlds

The following algorithms aim to solve known world Markov Decision Processes, or extract the optimal set of actions and utilities at every state in the world, given the known transition function $P(s'|s,a)$ and the rewards $R(s,a,s')$.

2.1.1 Value Iteration

The premise behind value iteration is simply to compute a utility for every state s in the Markov Decision Process. According to Russell el at [3], for every state s, the utility function U(s) is updated as

$$U(s) = max_{a \in A(s)} \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma U(s')) \tag{2}$$

While this is a simple update equation, a major question is when should value iteration stop?

One option for a stopping criterion is proposed by Russell el at. [3] is to compute the maximum relative change between a U(s) and U(s'), denoted δ. Then, value iteration should be stopped once $\delta \leq \epsilon \frac{(1-\gamma)}{\gamma}$. This stopping criterion will be used in this paper.

2.1.2 Policy Iteration

The premise behind policy iteration is to extract the set of optimal actions from every state s, denoted $\pi^*(s)$, with the computed utility function.

According to Russell el at [3], for every state s, the policy is extracted as

$$\pi^*(s) = argmax_{a \in A(s)} \sum_{s'} P(s'|s,a)(R(s,a,s') + \gamma U(s')) \tag{3}$$

The same stopping criterion used in value iteration is applied here.

2.2 Solving unknown worlds

Although value and policy iteration can be effective algorithms for solving Markov Decision Processes, there is a major caveat. They both require transition knowledge of the world, with $P(s'|s,a)$, hence there are called 'known' worlds. Fortunately, there are algorithms that can still learn the optimal actions of a world without pre-emptive knowledge of the world. Often times, these algorithms are denoted as learning in a reinforcement learning 'agent' environment, where an agent keeps track of the known utility that is associated with state and action pairs. This utility is usually denoted as Q(s, a).

2.2.1 Addressing Exploration vs Exploitation Tradeoff

The premise behind the exploration and exploitation tradeoff is between maximizing on the agent's current knowledge for the best reward, or exploring unknown states in order to gain information about the world. Although there are numerous interpretations for this tradeoff, Russell el at [3] propose a simple scheme. There is an exploration function f,

$$f(u, n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise} \end{cases} \tag{4}$$

where u is known utility of a state-action pair, n is the number of times a state-action pair has been tried, and N_e is an estimate for the maximum reward in the world.

Whenever an agent is choosing the next action, the action that maximizes the exploration function is chosen. The major caveat with this approach is determining how a state is choosen 'n' number of times. This leads to different representations that an agent can use.

2.2.2 Tabular representation of an agent

In a tabular representation, there are simply two tables that the state-action pairs (s, a). There is a Q(s, a) table for the utility, and a N(s, a) table for the number of times that (s, a) has been tried. N(s, a) is simply incremented every time (s, a) is tried.

2.2.3 Function Approximation Representation of an agent

Although a tabular representation seems intuitive, it is not scalable for worlds that hae large state spaces, which can include continuous state spaces or discrete state spaces with a large amount of states. Therefore, it is common practice to approximate Q(s, a) using features of the world. For the worlds described in section 1, Russell el at.[3] proposes a simple approximation function.

$$Q(s, a) = \theta_1 + \theta_2 X' + \theta_3 Y' \tag{5}$$

where X' and Y' are the coordinates of s', or the result of acting on s with the action a.

It should be noted that this paper still interprets a N(s, a) table for a function approximation agent, in order to use the exploration function detailed in section 2.2.1. However, there is a radius parameter r that tests if there is a (s, a) pair that is within the distance of r, given an incoming (s', a') pair. If it is in within distance, then N(s, a) is incremented. If not, then (s', a') is logged in the N(s, a) table.

2.2.4 Q-Learning

The scheme for an agent learning by Q-learning is

$$sample = (R(s, a, s') + \gamma + max_{a'} Q(s', a') - Q(s, a)) \tag{6}$$

where (s, a) is the previous state-action pair, s' is the current state.

2.2.5 SARSA

The scheme for an agent learning by SARSA is

$$sample = (R(s, a, s') + \gamma + Q(s', a') - Q(s, a)) \tag{7}$$

where (s, a) is the previous state-action pair, and s' is the current state.

In this paper, a' is chosen by an ϵ greedy approach. That is, for ϵ probability, the a' that maximizes $Q(s, a)$ is chosen. Otherwise, the a' that maximizes the exploration function from section 2.2.1 is chosen.

2.2.6 How *sample* is used to update $Q(s, a)$

In a tabular representation, *sample* is applied by

$$Q(s, a) = Q(s, a) + \alpha * sample \tag{8}$$

In the function approximation representation, *sample* is applied by

$$\theta_i = \theta_i + \alpha * sample * \frac{\partial Q_\theta(s, a)}{\partial \theta_i} \tag{9}$$

Based on (5), this gradient is simplified to

$$\begin{aligned} \theta_1 &= \theta_1 + \alpha * sample \\ \theta_2 &= \theta_2 + \alpha * sample * X \\ \theta_3 &= \theta_3 + \alpha * sample * Y \end{aligned}$$

where X and Y is the location of the previous state s.

The parameter α is denoted as the learning rate. For more dynamic exploration and exploitation, this paper will α as an inverse of N(s, a) for every transition between a current and next state.

2.3 Solving Multi-Arm Bandit Problems

2.3.1 Upper Confidence Bound

Upper Confidence Bound, or UCB, attempts to determine which arm in Multi-Arm Bandit simulation will give the best rewards, while balancing the exploration and exploitation tradeoff.

For every trial in a Multi-Arm Bandit simulation, UCB will pull the arm a, where

$$argmax_a \tilde{R}(a) + \sqrt{\frac{2 \log n}{n_a}} \tag{10}$$

where $\tilde{R}(a)$ are the observed reward means for each arm a, n_a is the number of times a has been pulled, and n is the current sample.

2.3.2 ϵ Greedy

The implementation for ϵ greedy is somewhat similar to UCB, except there is the probability of ϵ. With probability ϵ, arms are pulled at random. However, with probability $1-\epsilon$, ϵ greedy will the pull a, where

$$argmax_a \tilde{R}(a) \tag{11}$$

where $\tilde{R}(a)$ are the observed reward means for each arm a.

3 Performance and Analysis

3.1 Known Worlds

The following two subsections display the performance of value iteration and policy iteration on the two known world domains described in section 1.

3.1.1 Performance of the 4x3 Grid World

The following table show the summation of utility and the average utility over all the states in the grid world.

4x3 Grid World			
γ	ϵ	$utilitySum$	$avgUtility$
0.100	0.010	-15.448	-1.287
0.010	0.900	0.000	0.000
0.010	0.500	0.000	0.000
0.010	0.100	-15.344	-1.279
0.100	0.900	-15.344	-1.279
0.500	0.900	-15.344	-1.279
0.100	0.500	-15.344	-1.279
0.500	0.010	-15.133	-1.261
0.010	0.010	-15.344	-1.279
0.900	0.500	-8.648	-0.721
0.900	0.010	-7.154	-0.596
0.500	0.500	-15.671	-1.306
0.500	0.100	-15.360	-1.280
0.100	0.100	-15.344	-1.279
0.900	0.100	-7.442	-0.620
0.900	0.900	-9.779	-0.815

Table 1: The summation and average utility over all states in the 4x3 grid world

The best configuration of the 4x3 world with $\gamma = 0.9$ and $\epsilon = 0.01$ gives the following policy map.

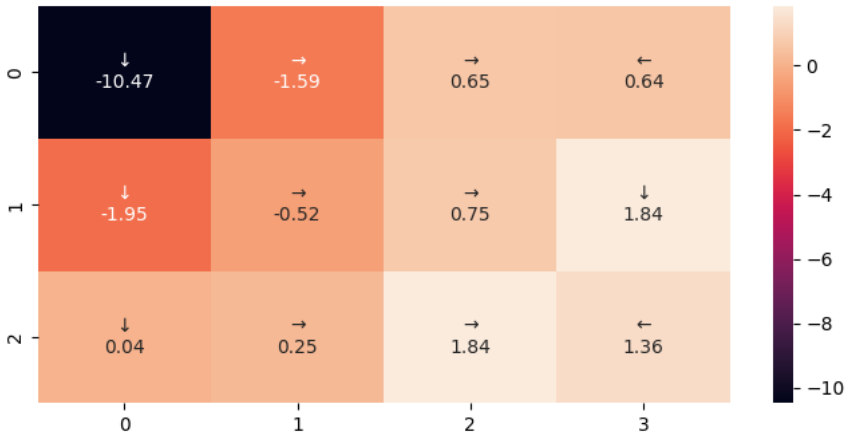

Figure 2: The policy map for the 4x3 grid world

3.1.2 Performance of the Wumpus world

9x7 Wumpus world			
γ	ϵ	$utilitySum$	$avgUtility$
0.010	0.010	58.100	0.182
0.010	0.100	58.100	0.182
0.100	0.900	58.100	0.182
0.100	0.500	58.100	0.182
0.100	0.010	65.209	0.204
0.010	0.500	0.000	0.000
0.010	0.900	0.000	0.000
0.100	0.100	64.524	0.202
0.500	0.100	161.468	0.505
0.500	0.500	151.775	0.474
0.500	0.010	165.536	0.517
0.500	0.900	141.681	0.443
0.900	0.900	3011.464	9.411
0.900	0.010	3050.370	9.532
0.900	0.500	3028.382	9.464
0.900	0.100	3046.928	9.522

Table 2: The summation and average utility over all states in the 9x7 wumpus world

The best configuration of the wumpus with $\gamma = 0.9$ and $\epsilon = 0.01$ gives the policy maps

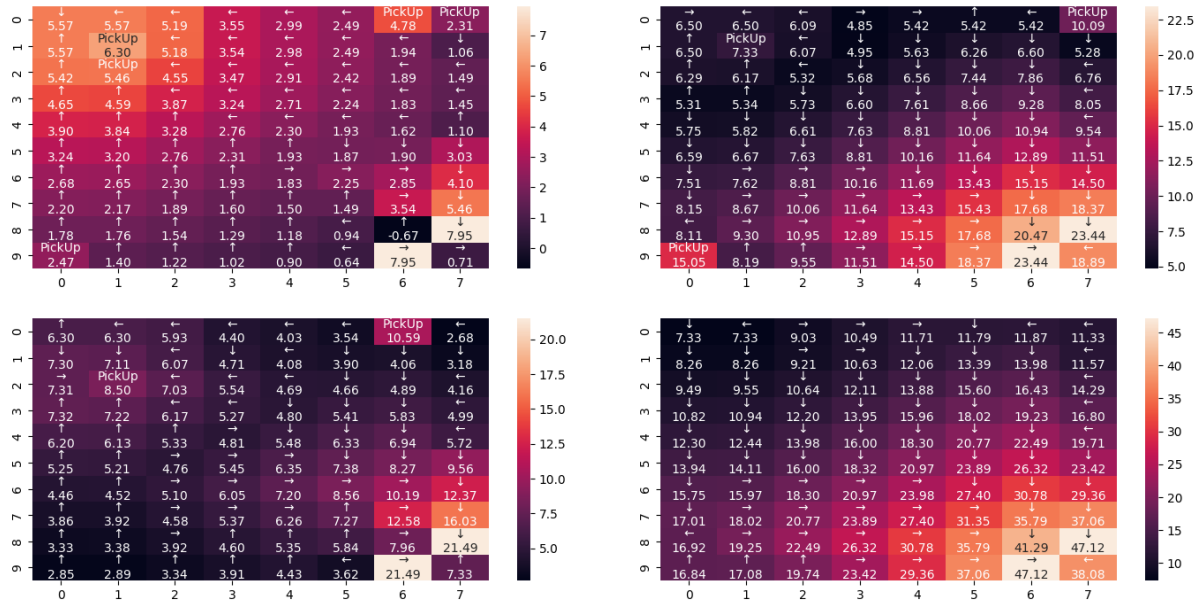

Figure 3: The policy maps for the Wumpus world. Top left(hasGold: False, hasImmunity: False). Top right(hasGold: False, hasImmunity: True) Bottom left(hasGold: True, hasImmunity: False). Bottom right(hasGold: True, hasImmunity: True)

3.2 Unknown Worlds

The following three subsections showcase samples of the performance that both table Q agents and function approximation Q agents had with the three unknown world domains described in section 1. The mean and standard deviation reward tables for Q learning and SARSA with $\epsilon = 0.5, 0.75$ are in the appendix. It should be noted that table Q agents explicitly run on discrete versions of the grid worlds, while function approximation agents run on continuous versions of the grid worlds.

3.2.1 Performance of the 4x3 grid world

The best configurations for the table Q agent seem to be $\gamma = 0.9$, $R^+ = 1$, and $N_e = 10$. The average rewards per episode and a solvable policy are displayed as follows

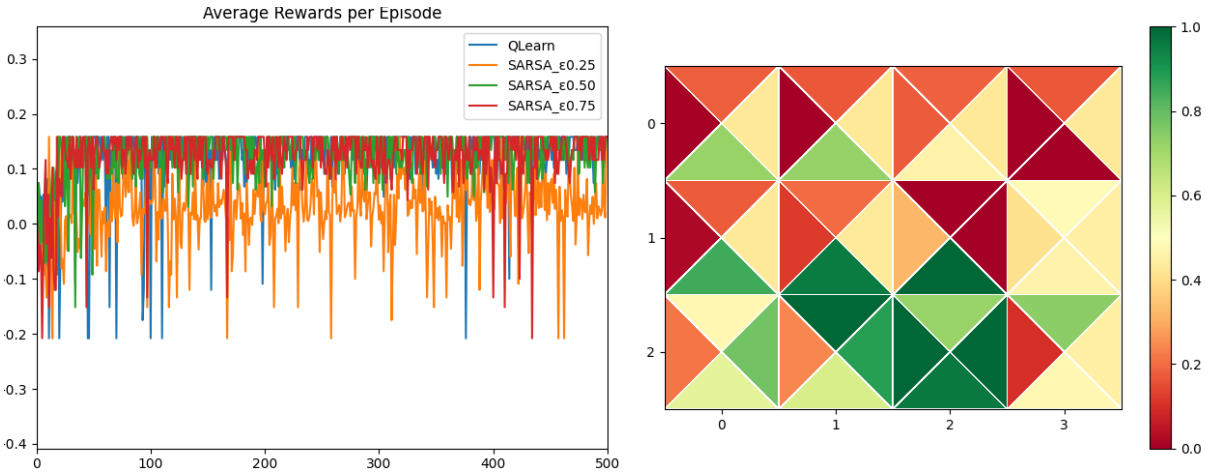

Figure 4: The graph on the left showcases the average rewards per episode. The policy on the right was produced by SARSA with a $\epsilon = 0.75$ strategy.

Meanwhile, a configuration that produced a solvable policy in the function approximation agent is $\gamma = 0.9$, $R^+ = 2.5$, and $N_e = 1000$. This policy was generated with SARSA and an $\epsilon = 0.75$ strategy.

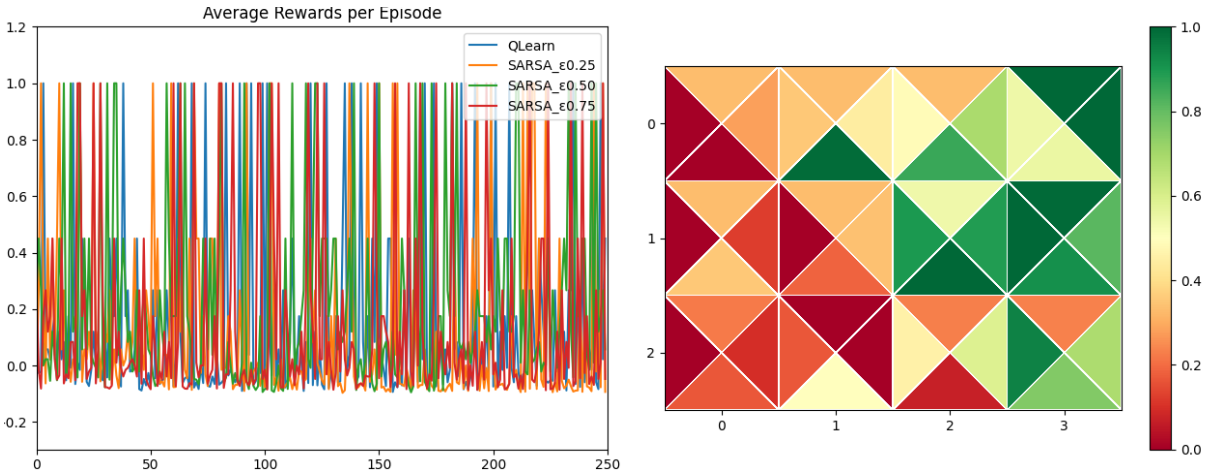

Figure 5: The graph on the left showcases the average rewards per episode. The policy on the right was produced by SARSA with a $\epsilon = 0.75$ strategy.

3.2.2 Performance of the 10x10 world with a +1 reward at (10, 10)

The best configurations for the table Q agent seem to be $\gamma = 0.9$, $R^+ = 1$, and $N_e = 10$ with the table Q agent performing Q learning. The average rewards per episode and the Q learning policy are displayed as follows

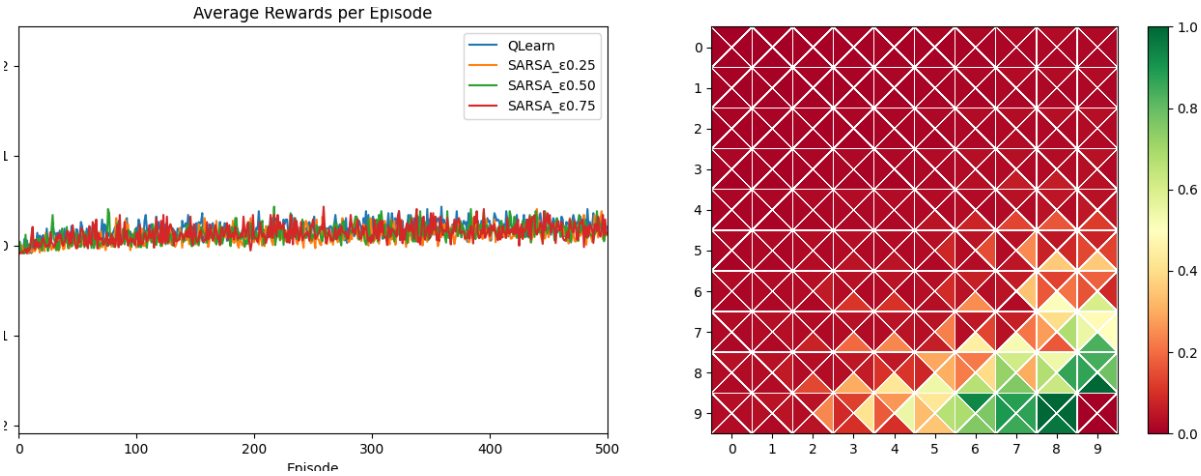

Figure 6: The graph on the left showcases the average rewards per episode. The policy on the right was produced by Q-Learning.

The configurations for the function approximation Q agent is nearly indistinguishable. In fact, upon observing the reward trends, the function Q agent struggled to learn anything. One exception is SARSA with $\epsilon = 0.5$ with the configuration of $\gamma = 0.1$, $R^+ = 2.5$, $N_e = 100$, and a radius bound of 5.

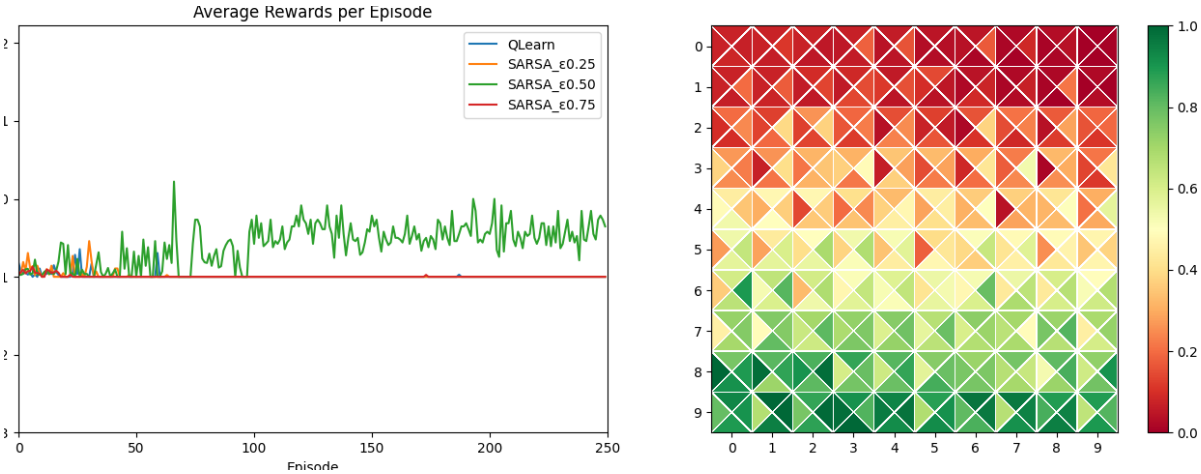

Figure 7: The graph on the left showcases the average rewards per episode. The policy on the right was produced by SARSA with a $\epsilon = 0.5$ strategy.

3.2.3 Performance of the 10x10 world with a +1 reward at (5, 5)

The best configurations for the table Q agent is $\gamma = 0.9$, $R^+ = 1$, and $N_e = 10$, with the table agent performing Q learning. The average rewards per episode and policy of Q learning policy is displayed as follows.

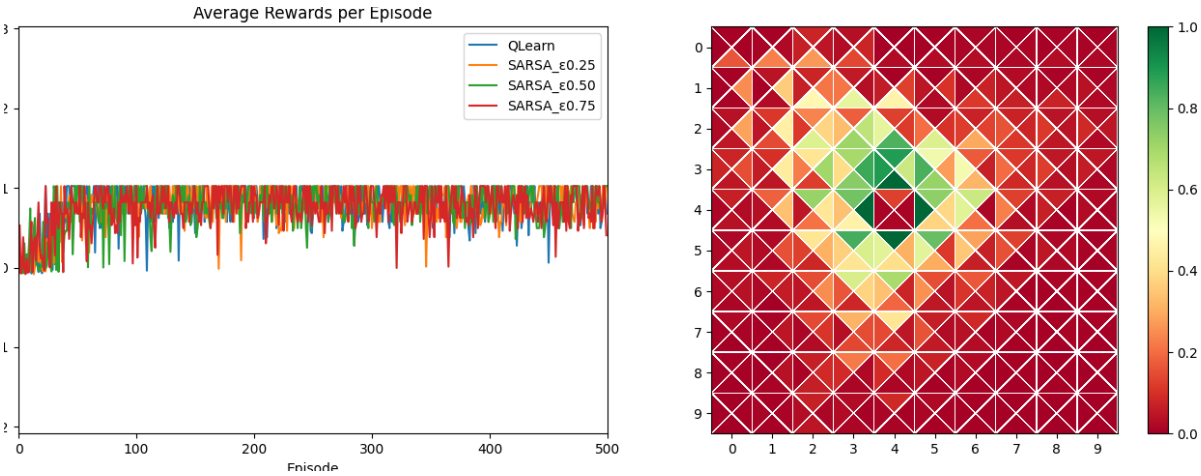

Figure 8: The graph on the left showcases the average rewards per episode. The policy on the right was produced by Q learning.

The best configurations for the function approximation Q agent is $\gamma = 0.9$, $R^+ = 5$, $N_e = 10$, and a radius bound of 5 on a Q learning policy. The average rewards per episode and the policy of Q learning is displayed as follows.

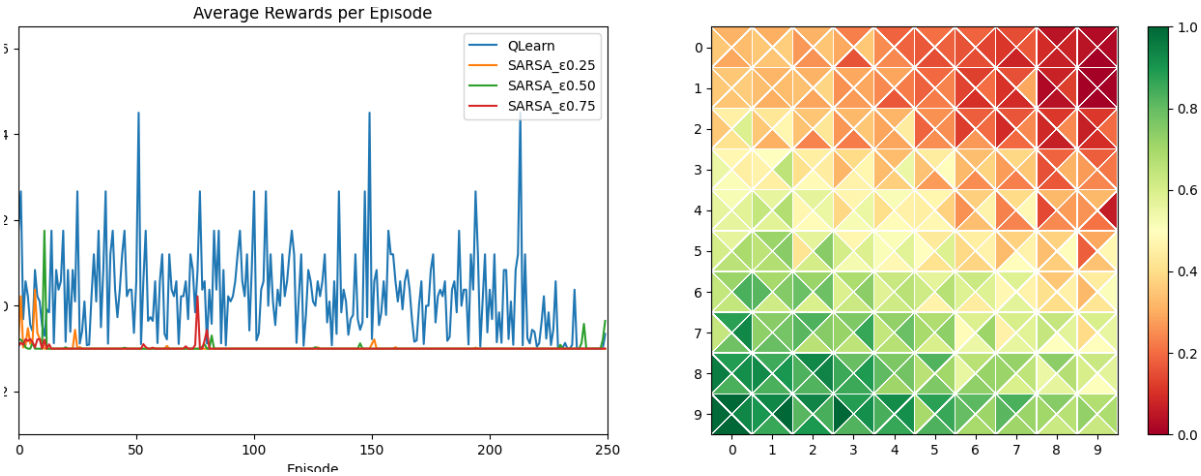

Figure 9: The graph on the left showcases the average rewards per episode. The policy on the right was produced by Q learning.

3.3 Multi-Arm Bandit Problems

3.3.1 Decision Accuracy Performance on UCB

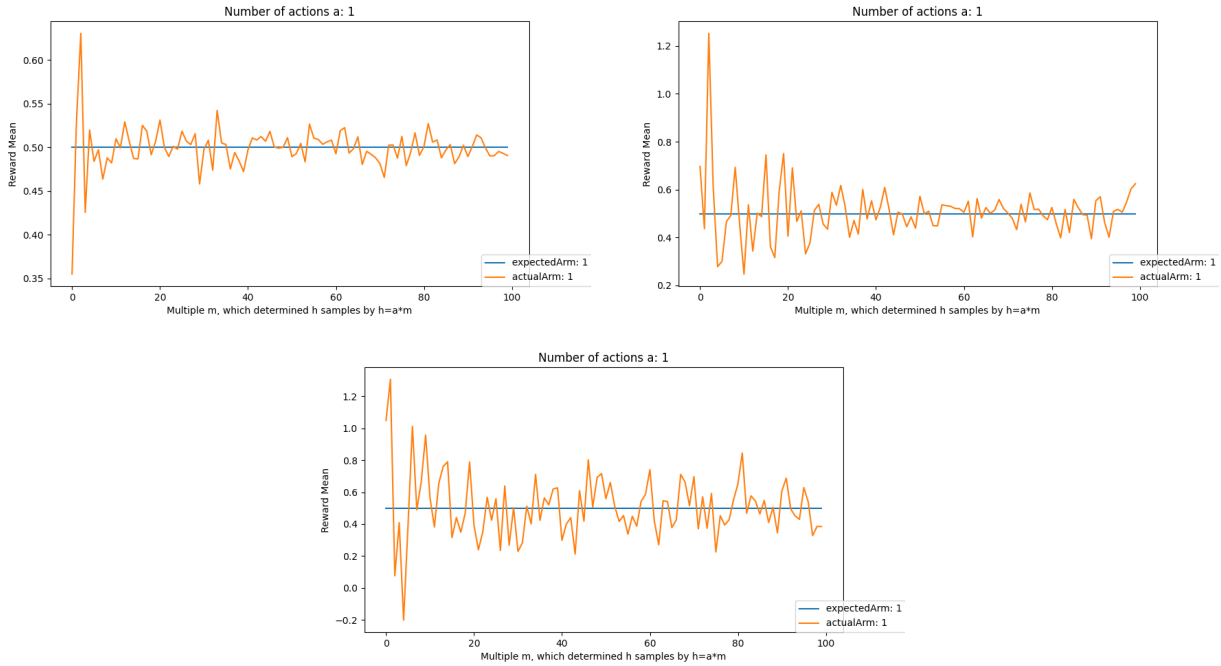

Figure 10: Comparison between the expected and actual reward means on UCB with 1 arm. Top left has a payout STD=0.1. Top right has a payout STD=0.5. Bottom has a payout STD=1

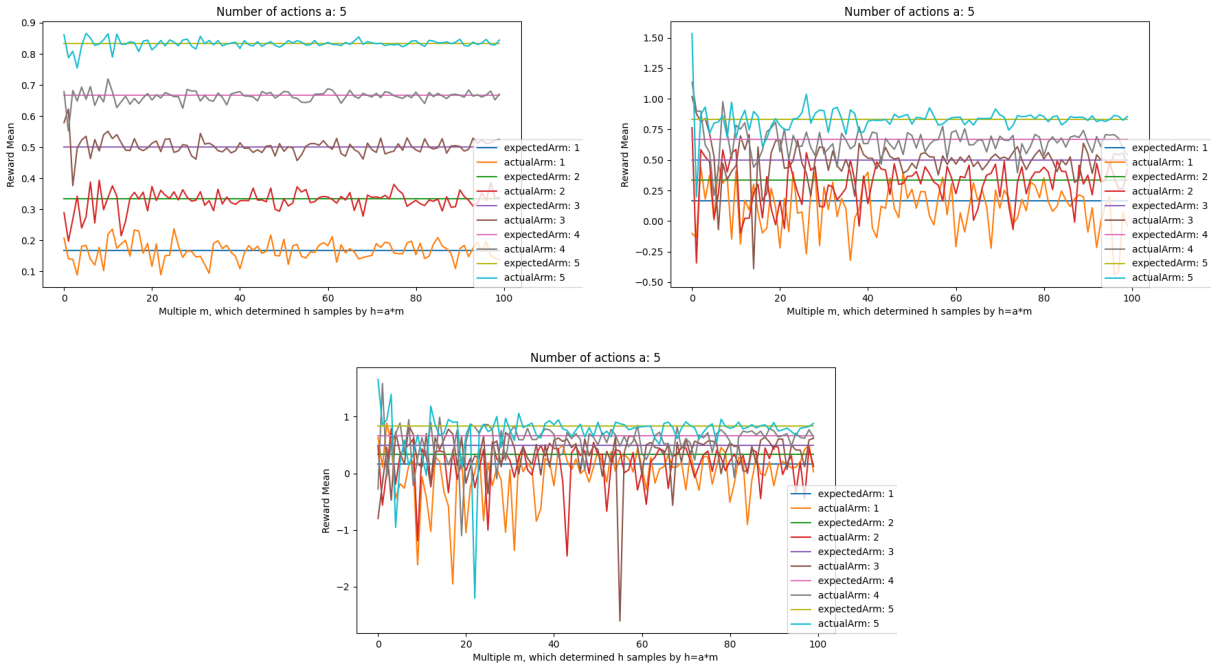

Figure 11: Comparison between the expected and actual reward means on UCB with 5 arms. Top left has a payout STD=0.1. Top right has a payout STD=0.5. Bottom has a payout STD=1

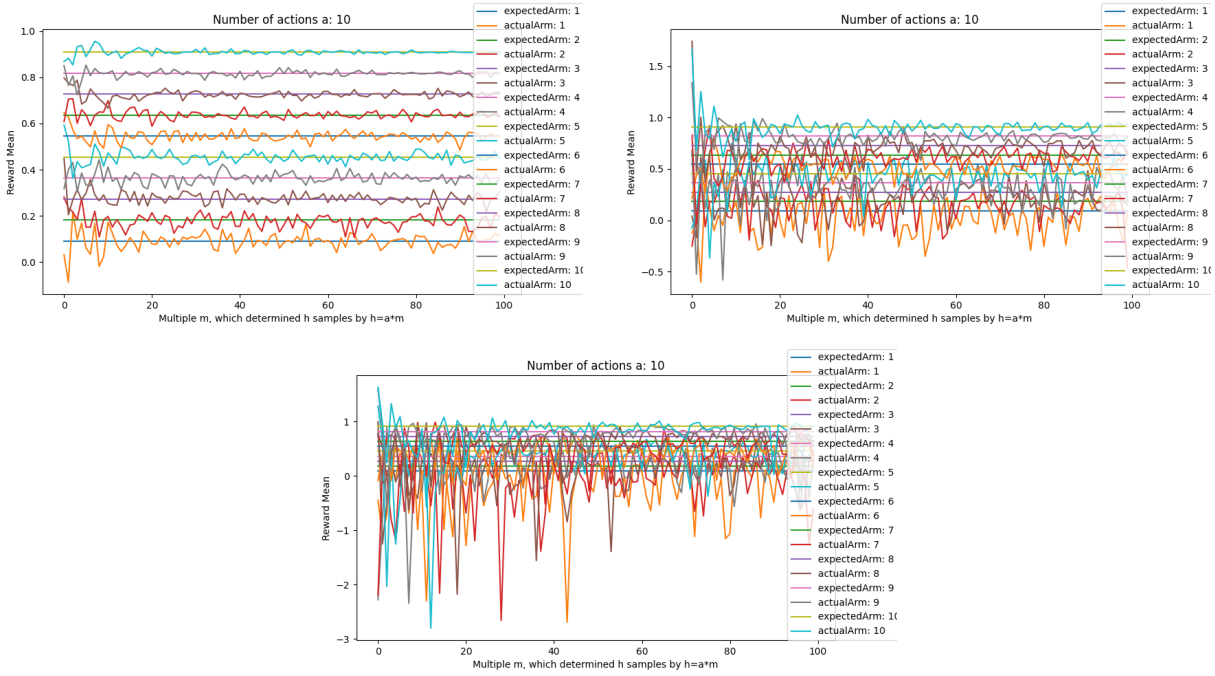

Figure 12: Comparison between the expected and actual reward means on UCB with 10 arms. Top left has a payout STD=0.1. Top right has a payout STD=0.5. Bottom has a payout STD=1

3.3.2 Regret Performance between UCB and ϵ Greedy

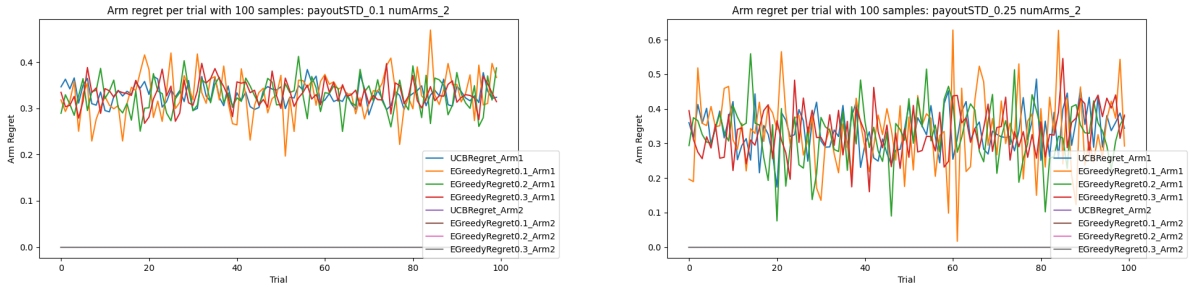

Figure 13: Comparison of regret on a 2 arm simulation between UCB and ϵ Greedy, where $\epsilon = 0.1, 0.2, 0.3$. Left figure has a payout STD=0.1. Right figure has a payout STD=0.25

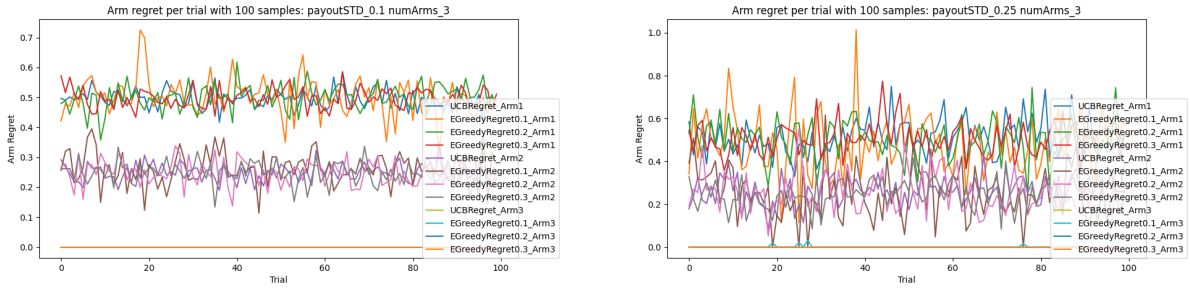

Figure 14: Comparison of regret on a 3 arm simulation between UCB and ϵ Greedy, where $\epsilon = 0.1, 0.2, 0.3$. Left figure has a payout STD=0.1. Right figure has a payout STD=0.25

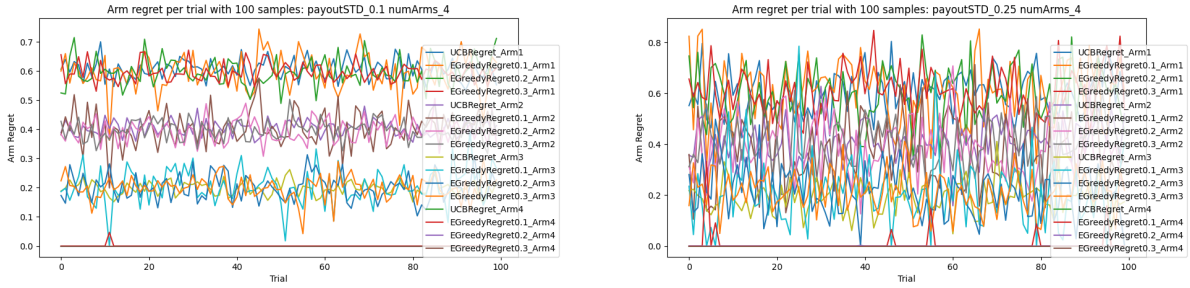

Figure 15: Comparison of regret on a 4 arm simulation between UCB and ϵ Greedy, where $\epsilon = 0.1, 0.2, 0.3$. Left figure has a payout STD=0.1. Right figure has a payout STD=0.25

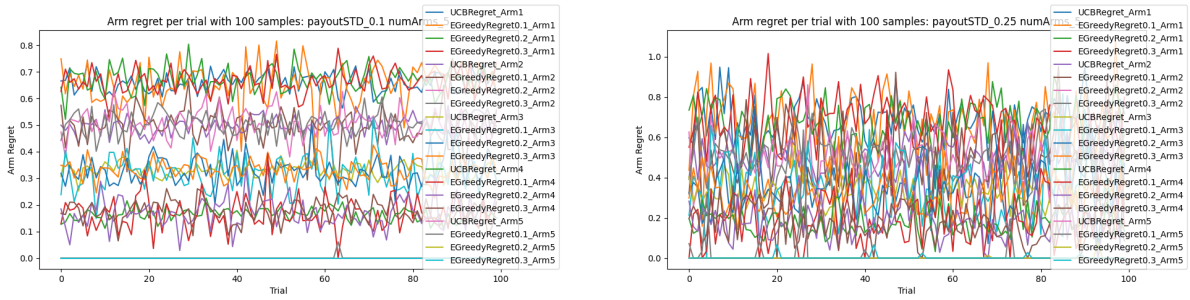

Figure 16: Comparison of regret on a 5 arm simulation between UCB and ϵ Greedy, where $\epsilon = 0.1, 0.2, 0.3$. Left figure has a payout STD=0.1. Right figure has a payout STD=0.25

3.4 Analysis

3.4.1 Known Worlds

It's clear from the utility sum/avg tables and the sample policies that both value iteration and policy iteration work well for determining a policy that can lead the agent towards rewards. Since, γ is a discount factor, and ϵ is an error factor, it makes sense why the best summations and averages occur at $\gamma = 0.9$ and $\epsilon = 0.01$.

3.4.2 Unknown Worlds

The performance of Q agents within unknown worlds is quite interesting.

For table representations, the agents were able to solve the worlds, or produce policies that lead agents to the reward locations. It's also clear that, besides world 2, exploitation using SARSA $\epsilon = 0.75$ or Q learning was preferred over exploration with SARSA $\epsilon = 0.5$. Also, with the configurations of $\gamma = 0.9$, $R^+ = 0.1$, and $N_e = 10$, it's quite clear that exploitation strategies won over exploration strategies. Due to the relatively small discrete state spaces of these worlds, this makes sense, as the agents should be able to find the reward locations given enough episodes.

For function approximation representations, the collective performance of the agents were somewhat disappointing. There were cases with heavy exploration bias with SARSA $\epsilon = 0.5$, low γs and higher N_e where the agents were able to learen some information about the world. However, the agents for the most part tended to struggling to learn anything about the world, especially for world 2. This lackluster performance most likely showcases the flaws with the utility function described in section 2.2.3.

Since the starting locations were located near (0, 0) and the reward locations were located northeast of the starting locations, this utility function was sufficient enough to make some sense of the world. However, if the distance between the starting location and reward location is somewhat large, such as with world 2, then this utility function is limiting in its performance. Also, with the reward location was located in southwest of the starting location, then it's performance could also struggle. Nevertheless, function approximation

still shows promise for produce good agents, since this lackluster utility function did deliver a couple of configurations that could learn significant information about the worlds.

3.4.3 Multi-Arm Bandit Problems

Based on the results from the decision accuracy trends, it's clear that an upper confidence bound is quite accurate for predicting the mean rewards, given that there is low standard deviation with the payouts. Interestingly, UCB does converge better for arms that have higher reward means. This is most likely explained by the fact that higher reward arms will be pulled more often, so this phenomena is reasonable to observe.

In terms of regret comparison between UCB and ϵ greedy bounds, UCB has much less deviation compared to ϵ greedy variants, given low payout standard deviations. Lower deviation in regret can provide better sense of the arm rewards, so UCB can be a better option if the payout standard deviation is low.

4 Conclusion

Reinforcement learning is a powerful tool for solving Markov Decision Processes. As shown in this paper, if the transition and reward models are known beforehand, then determinstic techniques such as value and policy iteration are sufficient for computing the optimal utility function and policy. However, if these models are not known in advance, then variations of Q learning are options for learning the policies. If the world model has a relatively small and discrete state space, then a table representation can be sufficient. However, if the model model is large and continuous, then a function approximation has opportunity to be more scalable, given a good utility function that models the world well.

In terms of solving Multi-arm bandit problems, UCB is a good technique for modeling the expected reward means, and is a better technique for modelling the regret of arms compared to ϵ greedy bounds.

References

[1] Chad Crawford. Cs7313-mdp. *https://github.com/TUmasters/cs7313-mdp*, 2018.

[2] Robert Geraghty. Bandit-sim. *https://github.com/robger98/Bandit_Sim*, 2021.

[3] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearsons, 4 edition, 2021.

5 Appendix

5.1 4x3 Discrete grid world with a table agent

Discrete 4x3 grid world results with a tabular agent									
γ	R^+	N_e	$\#Episodes$	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	1	10	500	0.054	0.100	0.066	0.088	0.054	0.101
0.1	1	50	500	0.089	0.086	0.090	0.075	0.080	0.082
0.1	1	100	500	0.070	0.095	0.060	0.100	0.033	0.120
0.1	2.5	10	500	0.113	0.051	0.112	0.053	0.108	0.054
0.1	2.5	50	500	0.081	0.085	0.073	0.071	0.088	0.086
0.1	2.5	100	500	0.066	0.099	0.064	0.094	0.069	0.092
0.1	5	10	500	0.112	0.059	0.098	0.055	0.115	0.056
0.1	5	50	500	0.089	0.083	0.079	0.087	0.096	0.078
0.1	5	100	500	0.064	0.098	0.059	0.095	0.057	0.105
0.5	1	10	500	0.121	0.059	0.037	0.067	0.115	0.062
0.5	1	50	500	0.081	0.101	0.086	0.089	0.083	0.102
0.5	1	100	500	0.060	0.102	0.067	0.096	0.046	0.108
0.5	2.5	10	500	0.127	0.045	0.118	0.058	0.102	0.097
0.5	2.5	50	500	0.057	0.125	0.094	0.083	0.089	0.096
0.5	2.5	100	500	0.057	0.104	0.062	0.100	0.061	0.103
0.5	5	10	500	0.094	0.106	0.107	0.080	0.114	0.057
0.5	5	50	500	0.080	0.099	0.095	0.088	0.096	0.081
0.5	5	100	500	0.055	0.102	0.059	0.108	0.066	0.099
0.9	1	10	500	0.120	0.065	0.123	0.046	0.122	0.055
0.9	1	50	500	0.089	0.091	0.097	0.085	0.062	0.121
0.9	1	100	500	0.070	0.095	0.060	0.100	0.033	0.119
0.9	2.5	10	500	0.116	0.071	0.048	0.062	0.115	0.071
0.9	2.5	50	500	0.094	0.086	0.092	0.085	0.095	0.086
0.9	2.5	100	500	0.059	0.101	0.054	0.107	0.028	0.123
0.9	5	10	500	0.119	0.051	0.047	0.072	0.061	0.063
0.9	5	50	500	0.088	0.094	0.091	0.089	0.082	0.098
0.9	5	100	500	0.078	0.093	0.065	0.096	0.037	0.119

5.2 10x10 Discrete grid world with a +1 reward at (10, 10) and a table agent

Discrete 10x10 with a +1 reward at (10, 10) results with a tabular agent									
γ	R^+	N_e	$\#Episodes$	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	1	10	500	-0.000	0.006	0.001	0.006	0.002	0.007
0.1	1	50	500	0.001	0.006	-0.000	0.006	0.001	0.007
0.1	1	100	500	0.003	0.008	-0.000	0.007	-0.001	0.007
0.1	2.5	10	500	0.001	0.006	0.003	0.007	0.003	0.008
0.1	2.5	50	500	0.002	0.007	-0.002	0.006	0.001	0.006
0.1	2.5	100	500	-0.000	0.007	0.002	0.007	-0.000	0.007
0.1	5	10	500	0.002	0.007	0.000	0.007	-0.000	0.006
0.1	5	50	500	0.000	0.007	-0.000	0.007	-0.001	0.007
0.1	5	100	500	0.002	0.008	0.002	0.008	0.002	0.008
0.5	1	10	500	0.008	0.009	0.009	0.009	0.010	0.009
0.5	1	50	500	0.008	0.009	0.005	0.008	0.005	0.008
0.5	1	100	500	0.006	0.010	0.005	0.009	0.007	0.010
0.5	2.5	10	500	0.012	0.009	0.008	0.009	0.010	0.009
0.5	2.5	50	500	0.006	0.008	0.007	0.008	0.006	0.008
0.5	2.5	100	500	0.006	0.009	0.006	0.009	0.006	0.009
0.5	5	10	500	0.011	0.009	0.008	0.009	0.010	0.009
0.5	5	50	500	0.006	0.008	0.006	0.009	0.006	0.009
0.5	5	100	500	0.006	0.009	0.005	0.009	0.006	0.009
0.9	1	10	500	0.018	0.009	0.013	0.010	0.014	0.010
0.9	1	50	500	0.016	0.010	0.011	0.010	0.012	0.010
0.9	1	100	500	0.013	0.012	0.010	0.011	0.009	0.011
0.9	2.5	10	500	0.017	0.010	0.013	0.009	0.013	0.010
0.9	2.5	50	500	0.012	0.010	0.012	0.10	0.012	0.010
0.9	2.5	100	500	0.012	0.012	0.006	0.010	0.012	0.012
0.9	5	10	500	0.015	0.010	0.014	0.009	0.017	0.010
0.9	5	50	500	0.012	0.010	0.009	0.009	0.008	0.009
0.9	5	100	500	0.013	0.011	0.007	0.010	0.012	0.012

5.3 10x10 Discrete grid world with a +1 reward at (5, 5) and a table agent

Discrete 10x10 with a +1 reward at (5, 5) results with a tabular agent									
γ	R^+	N_e	$\#Episodes$	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	1	10	500	0.054	0.024	0.052	0.025	0.056	0.024
0.1	1	50	500	0.045	0.031	0.053	0.032	0.051	0.033
0.1	1	100	500	0.037	0.034	0.038	0.034	0.033	0.032
0.1	2.5	10	500	0.048	0.023	0.055	0.025	0.058	0.024
0.1	2.5	50	500	0.047	0.031	0.044	0.031	0.046	0.030
0.1	2.5	100	500	0.036	0.034	0.034	0.032	0.032	0.032
0.1	5	10	500	0.054	0.025	0.056	0.026	0.057	0.025
0.1	5	50	500	0.047	0.033	0.042	0.029	0.045	0.030
0.1	5	100	500	0.038	0.034	0.034	0.032	0.039	0.035
0.5	1	10	500	0.074	0.026	0.073	0.025	0.077	0.026
0.5	1	50	500	0.062	0.037	0.059	0.037	0.062	0.036
0.5	1	100	500	0.046	0.040	0.045	0.040	0.045	0.040
0.5	2.5	10	500	0.074	0.026	0.073	0.025	0.077	0.025
0.5	2.5	50	500	0.061	0.038	0.059	0.037	0.061	0.037
0.5	2.5	100	500	0.045	0.041	0.042	0.039	0.041	0.037
0.5	5	10	500	0.074	0.026	0.073	0.025	0.077	0.026
0.5	5	50	500	0.062	0.038	0.058	0.036	0.058	0.037
0.5	5	100	500	0.046	0.040	0.043	0.038	0.046	0.041
0.9	1	10	500	0.077	0.025	0.076	0.026	0.075	0.025
0.9	1	50	500	0.061	0.036	0.061	0.038	0.061	0.038
0.9	1	100	500	0.047	0.041	0.042	0.040	0.048	0.040
0.9	2.5	10	500	0.017	0.010	0.013	0.009	0.013	0.010
0.9	2.5	50	500	0.060	0.037	0.063	0.038	0.065	0.034
0.9	2.5	100	500	0.044	0.039	0.046	0.040	0.045	0.041
0.9	5	10	500	0.080	0.025	0.077	0.026	0.078	0.026
0.9	5	50	500	0.063	0.037	0.064	0.038	0.061	0.037
0.9	5	100	500	0.047	0.039	0.046	0.040	0.048	0.041

5.4 4x3 Continuous grid world with a function approximation agent

Continuous 4x3 world with a function approximation agent									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	1	100	2.5	0.243	0.314	0.289	0.337	0.260	0.336
0.1	1	50	2.5	0.202	0.310	0.191	0.340	0.206	0.330
0.1	1	100	1	0.167	0.327	0.205	0.322	0.183	0.323
0.1	1	100	5	0.168	0.351	0.184	0.381	0.236	0.282
0.1	1	50	5	0.171	0.384	0.290	0.339	0.230	0.308
0.1	1	50	1	0.181	0.368	0.198	0.322	0.241	0.336
0.1	2.5	50	1	0.206	0.369	0.187	0.347	0.205	0.349
0.1	1	10	5	0.249	0.338	0.157	0.387	0.126	0.332
0.1	2.5	50	2.5	0.227	0.285	0.121	0.336	0.152	0.369
0.1	1	10	2.5	0.248	0.357	0.144	0.386	0.129	0.339
0.1	2.5	10	1	0.116	0.357	0.123	0.351	0.085	0.324
0.1	2.5	50	5	0.229	0.328	0.109	0.367	0.307	0.330
0.1	2.5	10	5	0.067	0.328	0.134	0.375	0.261	0.316
0.1	1	10	1	0.195	0.390	0.115	0.355	0.180	0.363
0.1	2.5	10	2.5	0.106	0.343	0.084	0.333	0.154	0.368
0.1	5	100	1	0.194	0.308	0.242	0.354	0.181	0.331
0.1	2.5	100	2.5	0.192	0.330	0.183	0.312	0.230	0.322
0.1	5	50	1	0.166	0.345	0.153	0.341	0.204	0.361
0.1	2.5	100	1	0.214	0.348	0.210	0.367	0.248	0.347
0.1	2.5	100	5	0.149	0.335	0.232	0.329	0.142	0.333
0.1	5	100	2.5	0.159	0.328	0.182	0.343	0.156	0.336
0.1	5	100	5	0.298	0.304	0.168	0.342	0.186	0.368
0.1	5	50	2.5	0.135	0.332	0.221	0.358	0.197	0.325
0.1	5	10	5	0.175	0.391	0.168	0.342	0.123	0.360
0.1	5	50	5	0.110	0.360	0.121	0.348	0.312	0.286
0.5	1	10	2.5	0.119	0.353	0.187	0.335	0.160	0.384
0.1	5	10	2.5	0.219	0.296	0.201	0.332	0.228	0.370
0.5	1	10	5	0.111	0.362	0.133	0.387	0.158	0.378
0.1	5	10	1	0.198	0.366	0.128	0.330	0.158	0.380

Continuous 4x3 world with a function approximation agent. Cont. 1									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.5	1	10	1	0.160	0.357	0.124	0.366	0.159	0.388
0.5	1	100	2.5	0.261	0.334	0.217	0.342	0.237	0.333
0.5	1	100	1	0.206	0.345	0.212	0.339	0.205	0.359
0.5	2.5	100	1	0.193	0.324	0.196	0.318	0.173	0.331
0.5	2.5	50	1	0.215	0.346	0.190	0.355	0.209	0.271
0.5	1	50	1	0.197	0.360	0.178	0.309	0.214	0.339
0.5	2.5	100	5	0.139	0.325	0.159	0.347	0.288	0.329
0.5	2.5	100	2.5	0.240	0.327	0.185	0.345	0.230	0.308
0.5	2.5	50	5	0.210	0.361	0.231	0.347	0.162	0.349
0.5	1	100	5	0.313	0.317	0.134	0.336	0.246	0.328
0.5	1	50	2.5	0.257	0.366	0.221	0.330	0.161	0.361
0.5	2.5	10	5	0.213	0.360	0.086	0.338	0.178	0.381
0.5	2.5	50	2.5	0.209	0.350	0.133	0.359	0.158	0.377
0.5	2.5	10	1	0.108	0.322	0.190	0.350	0.137	0.349
0.5	2.5	10	2.5	0.072	0.319	0.116	0.350	0.137	0.381
0.5	1	50	5	0.202	0.330	0.045	0.310	0.188	0.314
0.5	5	50	5	0.232	0.293	0.238	0.285	0.272	0.309
0.5	5	50	1	0.201	0.345	0.214	0.329	0.206	0.348
0.5	5	100	1	0.215	0.351	0.231	0.355	0.213	0.332
0.9	1	10	5	0.231	0.307	0.247	0.342	0.131	0.331
0.5	5	100	5	0.182	0.350	0.271	0.323	0.137	0.342
0.9	1	50	2.5	0.142	0.356	0.271	0.308	0.185	0.352
0.5	5	100	2.5	0.175	0.301	0.176	0.326	0.165	0.302
0.5	5	50	2.5	0.175	0.375	0.148	0.343	0.162	0.376
0.9	1	10	2.5	0.284	0.340	0.271	0.293	0.162	0.388
0.5	5	10	2.5	0.161	0.349	0.165	0.339	0.114	0.337
0.9	1	10	1	0.173	0.343	0.190	0.316	0.168	0.364
0.5	5	10	5	0.088	0.336	0.221	0.357	0.156	0.357

Continuous 4x3 world with a function approximation agent. Cont. 2									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.9	1	50	5	0.088	0.347	0.129	0.347	0.132	0.332
0.5	5	10	1	0.129	0.372	0.130	0.339	0.147	0.392
0.9	1	50	1	0.126	0.343	0.170	0.326	0.121	0.348
0.9	2.5	100	1	0.208	0.353	0.229	0.339	0.199	0.335
0.9	2.5	100	2.5	0.218	0.302	0.216	0.318	0.243	0.325
0.9	2.5	50	1	0.184	0.350	0.196	0.353	0.168	0.308
0.9	2.5	10	5	0.120	0.316	0.188	0.332	0.152	0.365
0.9	2.5	100	5	0.158	0.340	0.218	0.333	0.154	0.340
0.9	2.5	50	5	0.158	0.355	0.166	0.323	0.358	0.287
0.9	2.5	50	2.5	0.252	0.361	0.251	0.348	0.183	0.349
0.9	2.5	10	2.5	0.145	0.357	0.157	0.369	0.123	0.332
0.9	5	10	2.5	0.096	0.354	0.180	0.351	0.126	0.296
0.9	2.5	10	1	0.175	0.333	0.095	0.299	0.145	0.341
0.9	1	100	1	0.167	0.335	0.285	0.368	0.230	0.365
0.9	5	10	1	0.122	0.337	0.167	0.368	0.177	0.359
0.9	5	10	5	0.071	0.330	0.077	0.299	0.125	0.354
0.9	1	100	2.5	0.102	0.348	0.229	0.353	0.162	0.356
0.9	1	100	5	0.174	0.334	0.075	0.337	0.099	0.336
0.9	5	100	1	0.190	0.322	0.168	0.290	0.233	0.346
0.9	5	50	2.5	0.189	0.285	0.233	0.365	0.248	0.325
0.9	5	100	5	0.158	0.356	0.151	0.339	0.197	0.361
0.9	5	100	2.5	0.127	0.321	0.227	0.363	0.180	0.312
0.9	5	50	1	0.157	0.317	0.189	0.321	0.191	0.321
0.9	5	50	5	0.092	0.319	0.128	0.351	0.124	0.347

5.5 10x10 Continuous grid world with a +1 reward at (10, 10) and a function approxmation agent

Continuous 10x10 world with a reward +1 at (10,10) and a function approximation agent									
γ	R^+	N_e	$radius$	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	1	100	5	-0.099	0.005	-0.099	0.005	-0.099	0.005
0.1	1	50	5	-0.099	0.004	-0.099	0.004	-0.099	0.006
0.1	1	10	5	-0.099	0.005	-0.100	0.003	-0.100	0.002
0.1	1	100	2.5	-0.099	0.004	-0.099	0.005	-0.098	0.007
0.1	1	50	2.5	-0.099	0.003	-0.099	0.006	-0.098	0.005
0.1	1	10	2.5	-0.099	0.004	-0.100	0.002	-0.099	0.005
0.1	1	100	1	-0.096	0.009	-0.097	0.007	-0.096	0.006
0.1	1	50	1	-0.098	0.007	-0.098	0.005	-0.098	0.007
0.1	1	10	1	-0.100	0.002	-0.099	0.007	-0.098	0.008
0.1	2.5	100	5	-0.099	0.004	-0.059	0.027	-0.100	0.002
0.1	2.5	50	5	-0.099	0.004	-0.099	0.003	-0.099	0.006
0.1	5	50	5	-0.100	0.002	-0.099	0.006	-0.099	0.005
0.1	2.5	10	5	-0.100	0.001	-0.099	0.006	-0.098	0.010
0.1	5	10	5	-0.100	0.000	-0.099	0.003	-0.100	0.005
0.1	2.5	100	2.5	-0.098	0.006	-0.098	0.006	-0.098	0.005
0.1	2.5	10	2.5	-0.098	0.011	-0.100	0.003	-0.100	0.003
0.1	5	10	2.5	-0.100	0.002	-0.099	0.005	-0.095	0.015
0.1	5	50	2.5	-0.099	0.005	-0.099	0.006	-0.099	0.004
0.1	2.5	50	2.5	-0.098	0.008	-0.099	0.007	-0.099	0.004
0.1	2.5	50	1	-0.098	0.008	-0.098	0.006	-0.098	0.006
0.1	2.5	100	1	-0.096	0.008	-0.096	0.009	-0.096	0.009
0.1	5	50	1	-0.098	0.006	-0.098	0.005	-0.098	0.005
0.1	5	10	1	-0.099	0.003	-0.099	0.003	-0.099	0.004
0.1	2.5	10	1	-0.099	0.005	-0.099	0.004	-0.099	0.004
0.1	5	100	5	-0.099	0.003	-0.099	0.004	-0.099	0.005

Continuous 10x10 world with a reward +1 at (10,10) and a function approximation agent. Cont 1.									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.5	1	100	5	-0.099	0.003	-0.099	0.003	-0.099	0.005
0.5	1	10	5	-0.100	0.002	-0.100	0.002	-0.100	0.001
0.5	1	50	5	-0.099	0.004	-0.100	0.002	-0.100	0.002
0.5	2.5	10	5	-0.100	0.001	-0.100	0.001	-0.099	0.005
0.1	5	100	2.5	-0.099	0.005	-0.099	0.004	-0.098	0.006
0.5	1	100	2.5	-0.099	0.004	-0.099	0.003	-0.098	0.006
0.5	1	10	2.5	-0.099	0.004	-0.100	0.001	-0.100	0.002
0.5	1	50	2.5	-0.099	0.007	-0.099	0.004	-0.099	0.003
0.5	2.5	10	2.5	-0.099	0.004	-0.096	0.012	-0.098	0.008
0.1	5	100	1	-0.096	0.008	-0.096	0.007	-0.096	0.008
0.5	1	100	1	-0.096	0.009	-0.096	0.007	-0.097	0.007
0.5	1	50	1	-0.098	0.005	-0.099	0.004	-0.098	0.006
0.5	1	10	1	-0.099	0.006	-0.099	0.006	-0.098	0.007
0.5	2.5	10	1	-0.099	0.003	-0.100	0.001	-0.099	0.005
0.5	2.5	50	5	-0.100	0.003	-0.100	0.001	-0.100	0.002
0.5	5	100	5	-0.099	0.005	-0.100	0.002	-0.099	0.004
0.5	2.5	100	5	-0.099	0.003	-0.099	0.003	-0.099	0.003
0.5	5	10	5	-0.096	0.012	-0.100	0.002	-0.099	0.004
0.5	5	50	5	-0.100	0.003	-0.099	0.003	-0.100	0.001
0.5	5	50	2.5	-0.099	0.003	-0.099	0.003	-0.099	0.006
0.5	5	100	2.5	-0.099	0.005	-0.099	0.005	-0.099	0.005
0.5	2.5	100	2.5	-0.099	0.004	-0.099	0.004	-0.098	0.006
0.5	2.5	50	2.5	-0.099	0.005	-0.099	0.003	-0.099	0.003
0.5	5	10	2.5	-0.100	0.004	-0.099	0.004	-0.100	0.004
0.5	2.5	100	1	-0.098	0.004	-0.097	0.007	-0.096	0.008
0.5	5	100	1	-0.098	0.004	-0.097	0.006	-0.097	0.006
0.5	5	50	1	-0.098	0.005	-0.098	0.006	-0.099	0.004
0.5	5	10	1	-0.099	0.003	-0.099	0.006	-0.099	0.004
0.5	2.5	50	1	-0.098	0.004	-0.097	0.008	-0.098	0.007

Continuous 10x10 world with a reward +1 at (10,10) and a function approximation agent. Cont 2.									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.9	2.5	10	5	-0.100	0.003	-0.100	0.000	-0.100	0.000
0.9	1	100	5	-0.098	0.006	-0.099	0.003	-0.099	0.004
0.9	1	10	5	-0.100	0.001	-0.100	0.003	-0.100	0.002
0.9	2.5	10	2.5	-0.100	0.001	-0.100	0.002	-0.100	0.002
0.9	2.5	50	5	-0.099	0.004	-0.100	0.002	-0.099	0.006
0.9	1	10	2.5	-0.098	0.009	-0.100	0.002	-0.100	0.001
0.9	1	50	5	-0.093	0.014	-0.099	0.004	-0.100	0.002
0.9	2.5	100	2.5	-0.098	0.007	-0.099	0.003	-0.099	0.004
0.9	2.5	50	2.5	-0.099	0.003	-0.099	0.004	-0.099	0.005
0.9	2.5	100	5	-0.099	0.004	-0.099	0.005	-0.099	0.003
0.9	1	100	2.5	-0.096	0.008	-0.098	0.007	-0.099	0.003
0.9	1	50	2.5	-0.099	0.003	-0.100	0.003	-0.099	0.005
0.9	2.5	100	1	-0.097	0.007	-0.097	0.006	-0.097	0.007
0.9	1	50	1	-0.098	0.005	-0.098	0.005	-0.098	0.005
0.9	2.5	50	1	-0.098	0.006	-0.099	0.004	-0.098	0.007
0.9	1	10	1	-0.099	0.003	-0.099	0.007	-0.100	0.002
0.9	1	100	1	-0.096	0.007	-0.096	0.008	-0.096	0.007
0.9	2.5	10	1	-0.100	0.003	-0.099	0.003	-0.099	0.005
0.9	5	50	5	-0.100	0.001	-0.099	0.004	-0.100	0.002
0.9	5	100	5	-0.099	0.005	-0.099	0.003	-0.099	0.004
0.9	5	10	5	-0.100	0.003	-0.100	0.002	-0.100	0.001
0.9	5	50	2.5	-0.099	0.003	-0.100	0.002	-0.099	0.006
0.9	5	100	2.5	-0.099	0.004	-0.098	0.006	-0.098	0.006
0.9	5	10	2.5	-0.100	0.001	-0.099	0.005	-0.100	0.003
0.9	5	100	1	-0.098	0.005	-0.098	0.006	-0.096	0.008
0.9	5	10	1	-0.099	0.007	-0.099	0.004	-0.100	0.003
0.9	5	50	1	-0.098	0.005	-0.098	0.006	-0.098	0.005

5.6 10x10 Continuous grid world with a +1 reward at (5, 5) and a function approxmation agent

Continuous 10x10 world with a reward +1 at (5,5) and a function approximation agent.									
γ	R^+	N_e	$radius$	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.1	2.5	100	2.5	-0.075	0.044	-0.048	0.082	-0.060	0.067
0.1	2.5	50	2.5	-0.083	0.043	-0.041	0.092	-0.078	0.060
0.1	2.5	100	1	-0.056	0.052	-0.058	0.055	-0.063	0.042
0.1	1	100	2.5	-0.079	0.041	-0.070	0.053	-0.051	0.069
0.1	1	100	1	-0.048	0.070	-0.052	0.062	-0.051	0.062
0.1	1	50	1	-0.062	0.060	-0.073	0.042	-0.077	0.042
0.1	1	100	5	-0.046	0.077	-0.085	0.026	-0.091	0.033
0.1	1	50	2.5	-0.049	0.084	-0.086	0.037	-0.086	0.045
0.1	2.5	50	5	-0.096	0.018	-0.068	0.077	-0.080	0.059
0.1	2.5	100	5	-0.061	0.073	-0.083	0.060	-0.046	0.086
0.1	1	50	5	-0.049	0.093	-0.092	0.025	-0.084	0.058
0.1	2.5	50	1	-0.081	0.041	-0.069	0.053	-0.061	0.063
0.1	1	10	5	-0.094	0.033	-0.092	0.039	-0.098	0.014
0.1	2.5	10	2.5	-0.093	0.037	-0.095	0.029	-0.096	0.032
0.1	1	10	2.5	-0.095	0.026	-0.096	0.019	-0.095	0.029
0.1	2.5	10	5	-0.096	0.019	-0.057	0.089	-0.097	0.012
0.1	1	10	1	-0.093	0.024	-0.080	0.058	-0.082	0.049
0.1	2.5	10	1	-0.094	0.026	-0.085	0.050	-0.090	0.049
0.1	5	100	2.5	-0.054	0.072	-0.061	0.064	-0.046	0.076
0.1	5	100	5	-0.086	0.042	-0.057	0.074	-0.050	0.084
0.1	5	100	1	-0.052	0.053	-0.048	0.071	-0.052	0.063
0.5	1	100	5	-0.087	0.047	-0.077	0.050	-0.086	0.036
0.5	1	100	1	-0.052	0.073	-0.054	0.071	-0.066	0.043
0.1	5	50	5	-0.087	0.051	-0.093	0.033	-0.091	0.033

Continuous 10x10 world with a reward +1 at (5,5) and a function approximation agent. Cont 1.									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.5	1	100	2.5	-0.059	0.058	-0.080	0.052	-0.086	0.036
0.1	5	50	2.5	-0.049	0.086	-0.049	0.086	-0.082	0.043
0.5	1	50	1	-0.076	0.049	-0.075	0.060	-0.069	0.062
0.5	1	10	5	-0.093	0.032	-0.094	0.031	-0.092	0.039
0.5	1	50	5	-0.094	0.028	-0.095	0.023	-0.095	0.017
0.5	1	50	2.5	-0.061	0.071	-0.093	0.024	-0.086	0.039
0.5	1	10	2.5	-0.094	0.022	-0.095	0.025	-0.090	0.051
0.1	5	10	5	-0.096	0.029	-0.098	0.010	-0.096	0.024
0.1	5	10	2.5	-0.092	0.028	-0.096	0.027	-0.068	0.081
0.1	5	50	1	-0.066	0.058	-0.052	0.078	-0.061	0.062
0.5	1	10	1	-0.078	0.062	-0.088	0.044	-0.087	0.042
0.1	5	10	1	-0.090	0.033	-0.091	0.038	-0.084	0.053
0.5	2.5	100	5	-0.076	0.062	-0.055	0.078	-0.051	0.079
0.5	5	100	5	-0.086	0.035	-0.081	0.048	-0.060	0.076
0.5	5	100	2.5	-0.076	0.060	-0.042	0.073	-0.079	0.036
0.5	5	100	1	-0.062	0.055	-0.066	0.058	-0.060	0.069
0.5	2.5	50	1	-0.072	0.049	-0.075	0.055	-0.061	0.070
0.5	5	50	1	-0.075	0.061	-0.070	0.059	-0.075	0.048
0.5	2.5	100	2.5	-0.073	0.060	-0.071	0.055	-0.077	0.041
0.5	2.5	50	5	-0.032	0.095	-0.089	0.041	-0.088	0.042
0.5	2.5	100	1	-0.063	0.056	-0.057	0.062	-0.049	0.072
0.5	5	50	2.5	-0.088	0.036	-0.058	0.072	-0.079	0.059
0.5	2.5	50	2.5	-0.090	0.031	-0.081	0.051	-0.077	0.062
0.5	5	10	5	-0.098	0.012	-0.085	0.060	-0.096	0.026
0.5	2.5	10	5	-0.094	0.038	-0.094	0.038	-0.089	0.044
0.5	5	50	5	-0.095	0.029	-0.095	0.020	-0.085	0.054
0.5	5	10	2.5	-0.075	0.063	-0.099	0.005	-0.094	0.033
0.5	2.5	10	2.5	-0.095	0.021	-0.095	0.023	-0.095	0.028
0.5	5	10	1	-0.084	0.044	-0.094	0.025	-0.092	0.027
0.5	2.5	10	1	-0.087	0.043	-0.091	0.028	-0.088	0.038

Continuous 10x10 world with a reward +1 at (5,5) and a function approximation agent. Cont 2.									
γ	R^+	N_e	*radius*	QLearnμ	QLearnσ	SARSA0.50μ	SARSA0.50σ	SARSA0.75μ	SARSA0.75σ
0.9	1	100	5	-0.050	0.073	-0.083	0.044	-0.036	0.100
0.9	2.5	100	1	-0.058	0.067	-0.062	0.059	-0.055	0.067
0.9	2.5	100	2.5	-0.068	0.060	-0.079	0.042	-0.086	0.034
0.9	1	50	5	-0.093	0.035	-0.091	0.038	-0.091	0.035
0.9	1	100	2.5	-0.062	0.059	-0.078	0.043	-0.056	0.063
0.9	1	100	1	-0.055	0.071	-0.061	0.053	-0.057	0.073
0.9	1	10	5	-0.092	0.033	-0.099	0.007	-0.096	0.030
0.9	2.5	10	5	-0.096	0.023	-0.096	0.020	-0.098	0.024
0.9	2.5	50	2.5	-0.064	0.056	-0.085	0.033	-0.072	0.069
0.9	1	50	2.5	-0.070	0.050	-0.088	0.036	-0.090	0.025
0.9	2.5	100	5	-0.070	0.048	-0.085	0.044	-0.079	0.047
0.9	1	50	1	-0.077	0.052	-0.073	0.054	-0.062	0.065
0.9	2.5	50	1	-0.070	0.062	-0.077	0.052	-0.071	0.068
0.9	2.5	50	5	-0.093	0.030	-0.092	0.042	-0.092	0.041
0.9	2.5	10	2.5	-0.084	0.045	-0.096	0.022	-0.095	0.026
0.9	1	10	2.5	-0.081	0.041	-0.097	0.014	-0.095	0.021
0.9	1	10	1	-0.083	0.042	-0.091	0.032	-0.088	0.038
0.9	2.5	10	1	-0.091	0.036	-0.091	0.038	-0.084	0.054
0.9	5	100	1	-0.065	0.046	-0.057	0.063	-0.063	0.059
0.9	5	100	5	-0.088	0.035	-0.085	0.052	-0.084	0.044
0.9	5	50	5	-0.094	0.037	-0.060	0.092	-0.092	0.037
0.9	5	100	2.5	-0.072	0.052	-0.083	0.038	-0.076	0.055
0.9	5	10	5	0.024	0.103	-0.098	0.018	-0.098	0.009
0.9	5	10	2.5	-0.078	0.060	-0.095	0.022	-0.096	0.017
0.9	5	50	2.5	-0.091	0.029	-0.090	0.033	-0.087	0.044
0.9	5	50	1	-0.069	0.056	-0.077	0.049	-0.079	0.047
0.9	5	10	1	-0.093	0.024	-0.089	0.037	-0.091	0.029