CS698R: Deep Reinforcement Learning

Assignment #1

Name: Shiven Tripathi
Roll NO.: 190816

Solution to Problem 1: Multi-armed Bandits

1. 2 Armed Bernoulli Bandit:
   The environment was tested using test cases: (1,1) ; (0,1) ; (1,1) ; (0.5,0.5).
   On running the environment for large number of steps 100000 for each alpha and beta, the expected rewards were compared and found to be converging to the sum of the rewards under Bernoulli r.v. Below are the absolute differences between actual reward and expected reward.

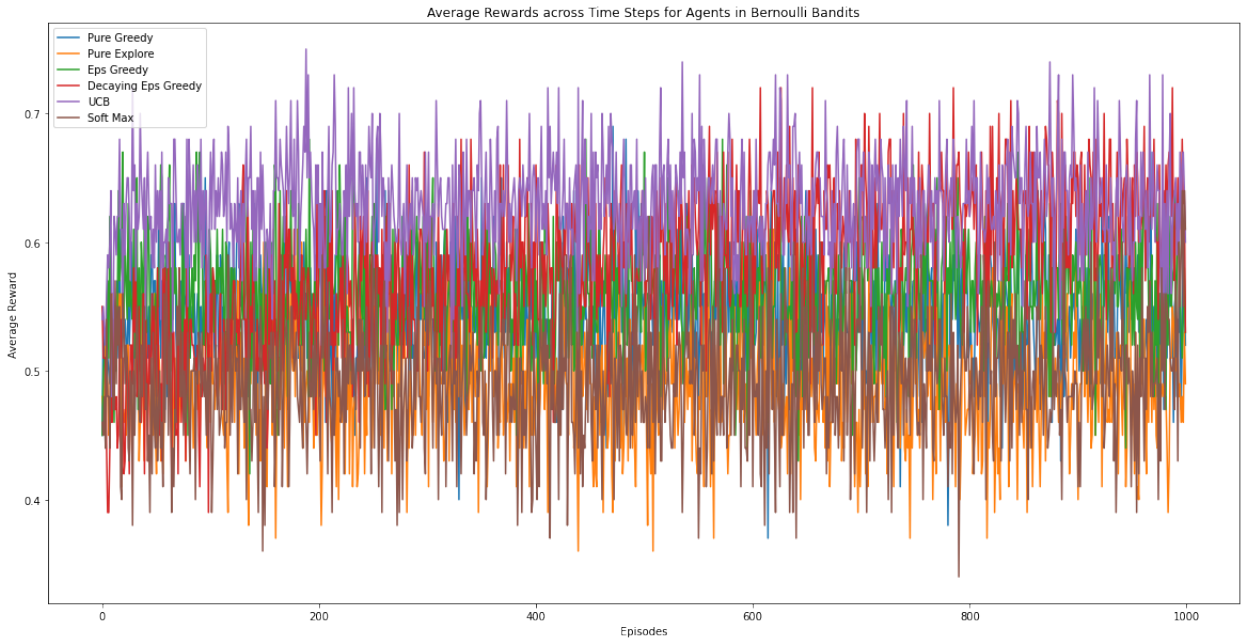
   ```
   Test: 0 [0.5, 0.5]
   0.000308
   0.000698
   Test: 1 [1, 0]
   0.0
   0.0
   Test: 2 [0, 1]
   0.0
   0.0
   Test: 3 [0, 0]
   0.0
   0.0
   Test: 4 [1, 1]
   0.0
   0.0
   ```

2. 10 Armed Gaussian Bandit

3. Code Notebook contains all the relevant plots alongwith parameters used for generating
   (a) Greedy Strategy
   (b) Explore Strategy
   (c) Epsilon Greedy Strategy
   (d) Decaying Epsilon Greedy Strategy
   (e) Soft Max Strategy
   (f) UCB Strategy

4. Rewards in 50 Bernoulli Bandit Testbed
   Observations:
   (a) UCB and Decaying Epsilon Greedy are found to be the best performing agents
   (b) Soft Max has a tendency to take unfavourable actions regularly, leading to a loss of reward
   (c) In the 2 Armed Situation, the differences between the Pure Greedy and the Pure Exploratory Strategy is not as apparent
   (d) Overall we can observe a trend of UCB performing well even for initial episodes, but decaying epsilon greedy catches up. This is possible only when epsilon is decayed exponentially

5. Rewards in 50 Gaussian Bandit Testbed
   Observations:



(a) Once again we can see UCB and Decaying Epsilon Greedy to be the top performing agents, but the difference is much clearer with 50 Armed Bandits

(b) Since a Decaying Epsilon Greedy starts with a higher epsilon than just Epsilon Greedy, we see an initial reward penalty, similar to what a pure exploratory agent suffers.

(c) For later episodes, decaying epsilon greedy is able to match performance of UCB and consistently takes high reward actions

(d) We find that pure exploration is even worse a strategy than pure greedy for the multiple arm situation because for some seeds pure greedy can eliminate worse outcomes quickly, but pure explore always gets a random and therefore possibly much lower reward

6. Regret in 50 Bernoulli Bandit Testbed
   Regret has been defined to be the cumulative loss of maximum possible reward an agent could attain, when it takes a certain action over those episodes
   Observations:

   (a) We find that UCB is the best at minimising regret for an agent

   (b) For decaying epsilon greedy strategy and UCB the regret curve appears to be bounded logarithmically while for all other agents, regret increases linearly over the number of episodes

   (c) Once again since Decaying Epsilon Greedy picks sub optimal actions at the start, it has high regret, but over time it learns the optimal actions and has a log bounded regret like UCB

7. Regret in 50 Gaussian Bandit Testbed
   Observations:

   (a) Similar observations as above hold but in addition we can claim that the differences in regret minimisation are clearer for large number of arms

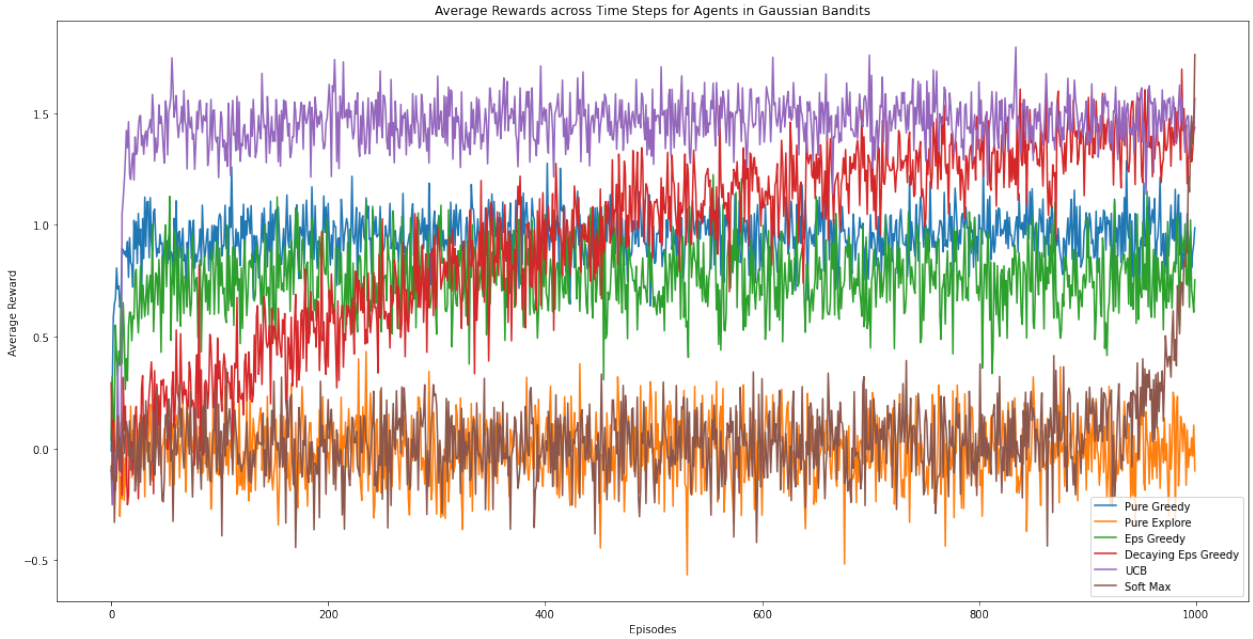
8. Optimal Action in 50 Bernoulli Bandit Testbed
   Observations:

   (a) This curve has similar interpretations to the reward curves but in addition observe how UCB has slight downward blips due to taking lesser optimal actions every now and then

   (b) Overall Decaying Epsilon Greedy is able to find the best action and consistently chooses it at the end of the episodes

   (c) The chances for Greedy Strategy to take optimal action are set in stone by the first few actions it takes, and therefore it is constant

9. Optimal Action in 50 Gaussian Bandit Testbed



Solution to Problem 2: MC Estimates and TD Learning

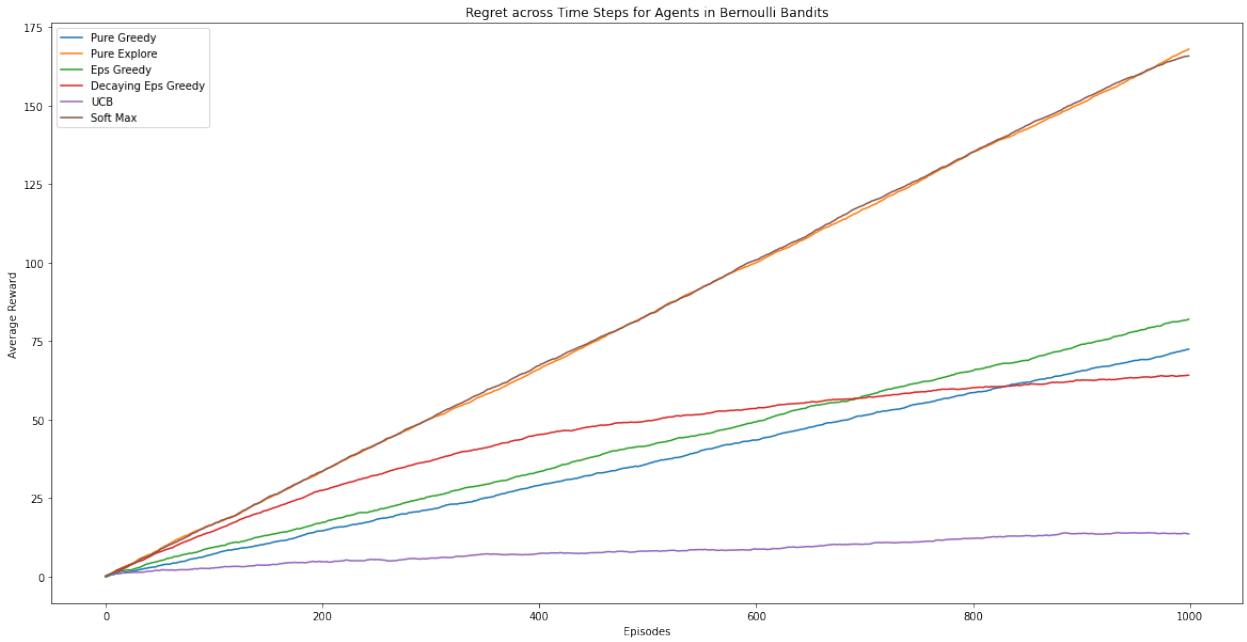
1. Generate trajectory for a policy
   Trajectory generation for an arbitrary soft policy was completed using the following test cases

```
Test: 0 [7, 0.5]
Step: 0 State: 2 Reward: 0
Step: 1 State: 3 Reward: 0
Step: 2 State: 2 Reward: 0
Step: 3 State: 1 Reward: 0
Step: 4 State: 2 Reward: 0
Step: 5 State: 1 Reward: 0
Step: 6 State: 2 Reward: 0
Step: 7 State: 3 Reward: 0
Step: 8 State: 4 Reward: 0
Step: 9 State: 5 Reward: 0
Step: 10 State: 6 Reward: 1
Test: 1 [10, 1]
Step: 0 State: 6 Reward: 0
Step: 1 State: 7 Reward: 0
Step: 2 State: 8 Reward: 0
Step: 3 State: 9 Reward: 1
Test: 2 [3, 1]
Step: 0 State: 2 Reward: 1
```

2. Decay Alpha step size parameter
   Observe the decay from 1 to 0.001 in 1000 time steps

3. Monte Carlo Prediction in RWE
   Tested using arbitrary policy and error in estimates compared

```
True Estimates: [0.0, 0.2, 0.4, 0.6, 0.8]
Testing FVMC
FVMC Estimates:  [2.92646921e-05 7.26306968e-04 6.00297168e-03 6.29028278e-02
 5.61109616e-01]
```

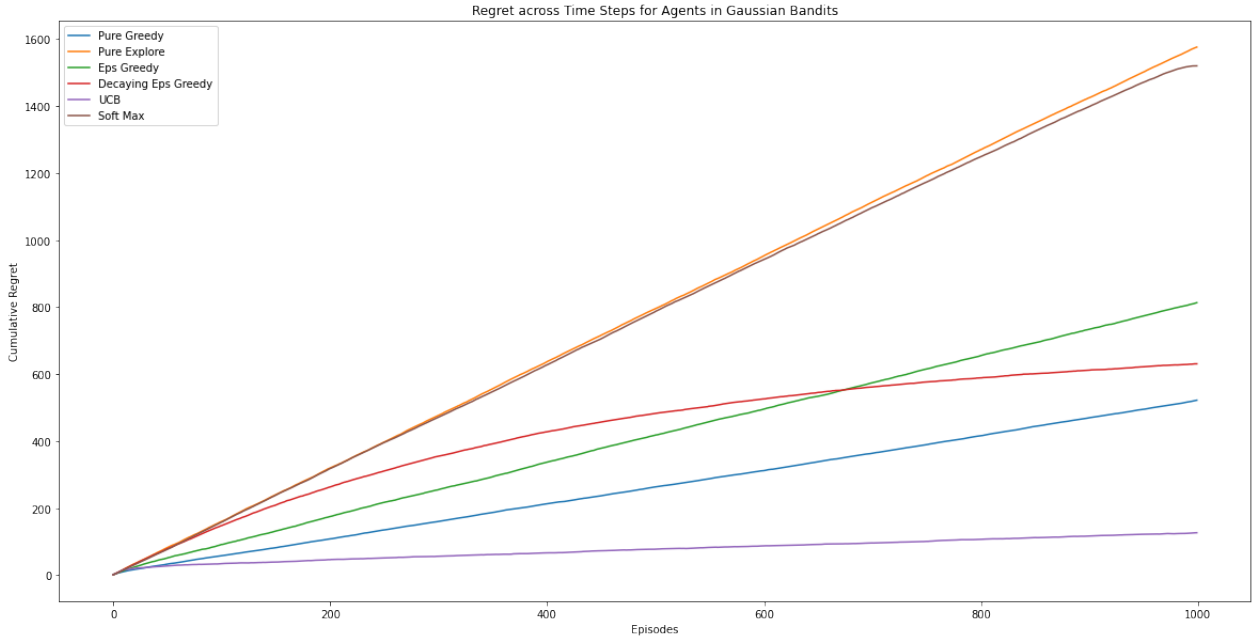


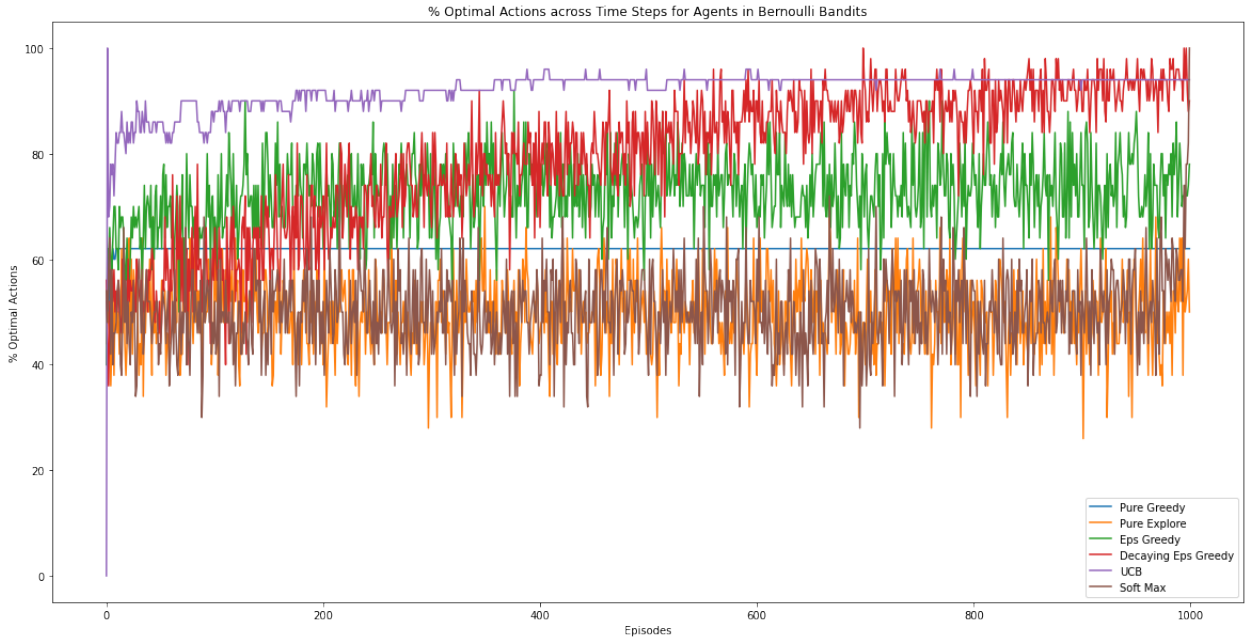

```
Error: [2.92646921e-05 1.99273693e-01 3.93997028e-01 5.37097172e-01
 2.38890384e-01]
Testing EVMC
EVMC Estimates: [6.60513173e-05 7.87438711e-04 7.18339151e-03 6.74012127e-02
 5.79506579e-01]
Error: [6.60513173e-05 1.99212561e-01 3.92816608e-01 5.32598787e-01
 2.20493421e-01]
```

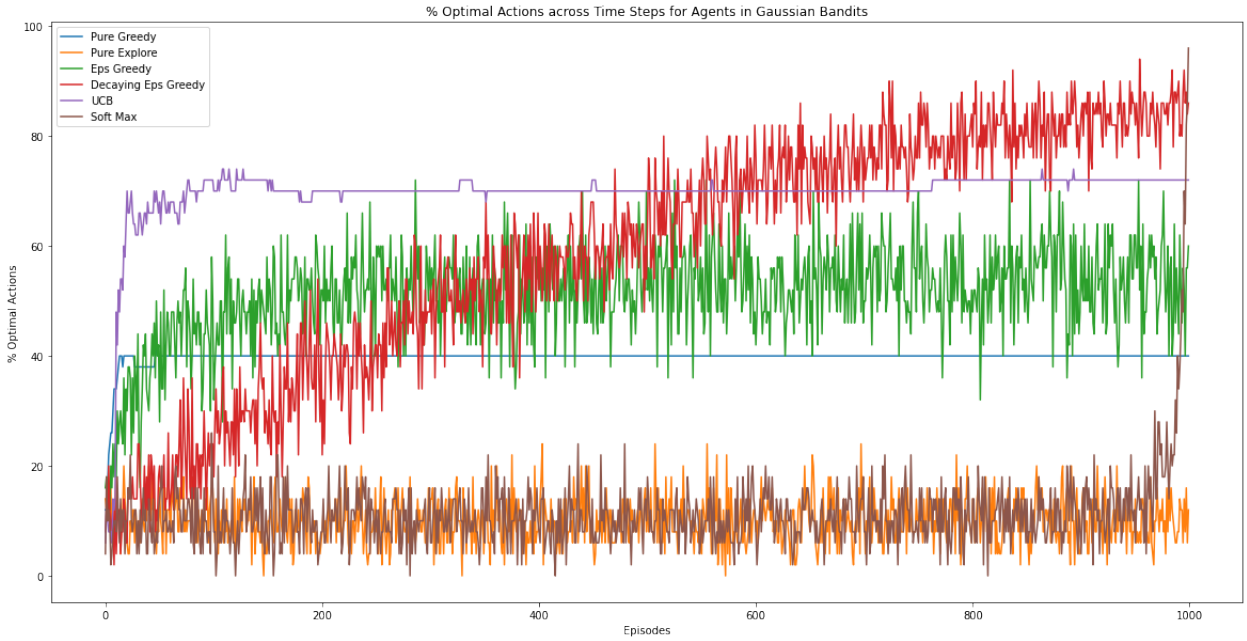
4. TD Prediction in RWE
   Tested using arbitrary policy and error in estimates compared

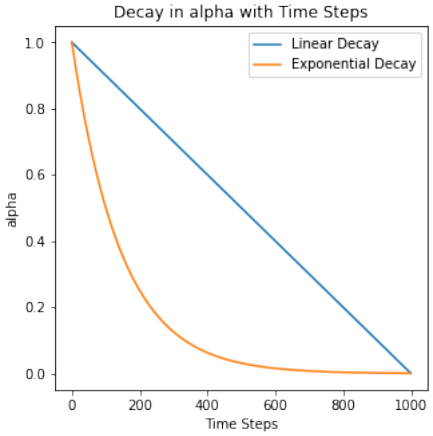
```
True Estimates: [0.0, 0.2, 0.4, 0.6, 0.8]
Testing TD
TD Estimates:  [6.60157032e-05 6.35057921e-04 5.50742886e-03 5.05953864e-02
 4.69675318e-01]
Error: [6.60157032e-05 1.99364942e-01 3.94492571e-01 5.49404614e-01
 3.30324682e-01]
```

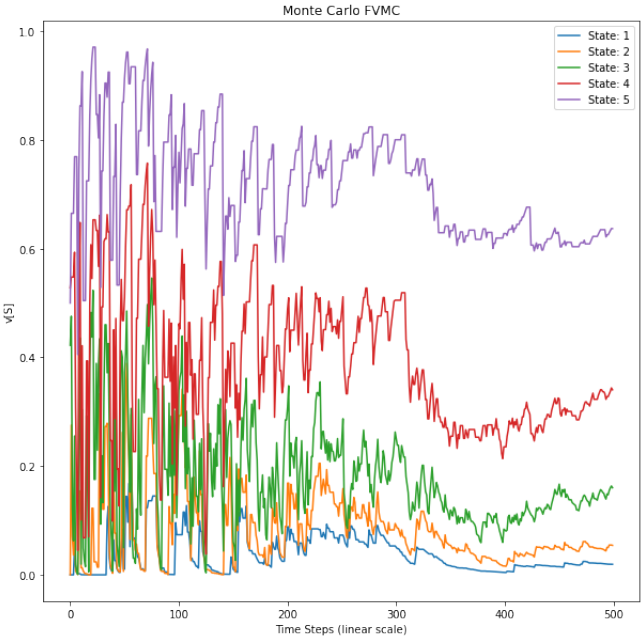
5. MC-FVMC Estimates in RWE (linear scale)
6. MC-EVMC Estimates in RWE (linear scale)
7. TD Estimates in RWE (linear scale)
8. MC-FVMC Estimates in RWE (log scale)
9. MC-EVMC Estimates in RWE (log scale)
10. TD Estimates in RWE (log scale)
11. Comparison for RWE
12. MC-FVMC Estimate of Target Value in RWE
13. MC-EVMC Estimate of Target Value in RWE
14. TD Estimate of Target Value in RWE
15. Comparison for RWE

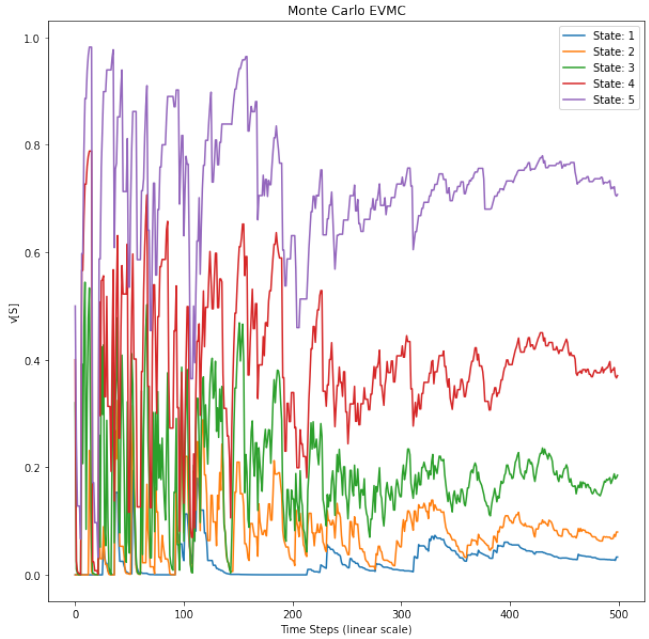

Regret across Time Steps for Agents in Gaussian Bandits
Pure Greedy
Pure Explore
Eps Greedy
Decaying Eps Greedy
UCB
Soft Max
Cumulative Regret
Episodes
0
200
400
600
800
1000
1200
1400
1600

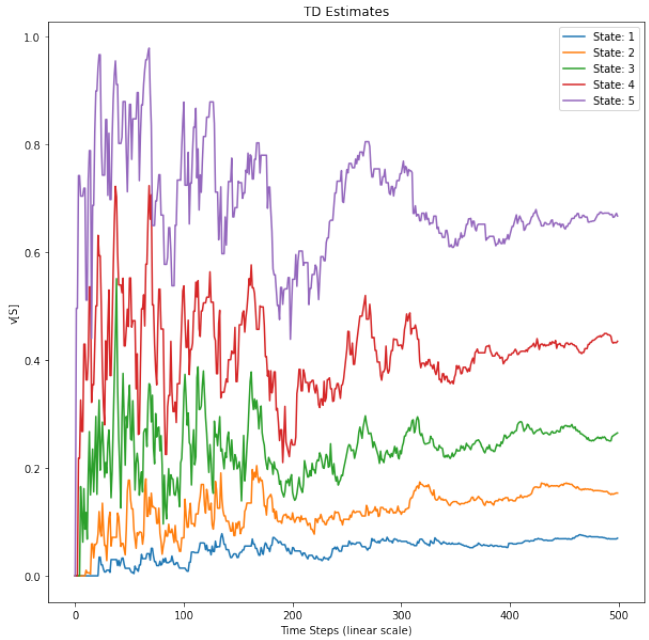

% Optimal Actions across Time Steps for Agents in Bernoulli Bandits
% Optimal Actions
Episodes
Pure Greedy
Pure Explore
Eps Greedy
Decaying Eps Greedy
UCB
Soft Max
0
20
40
60
80
100
200
400
600
800
1000

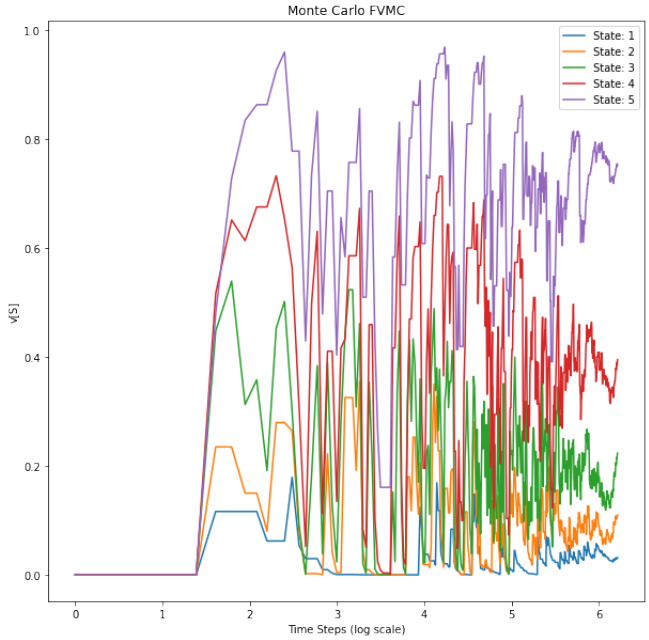

% Optimal Actions across Time Steps for Agents in Gaussian Bandits
Pure Greedy
Pure Explore
Eps Greedy
Decaying Eps Greedy
UCB
Soft Max
% Optimal Actions
Episodes

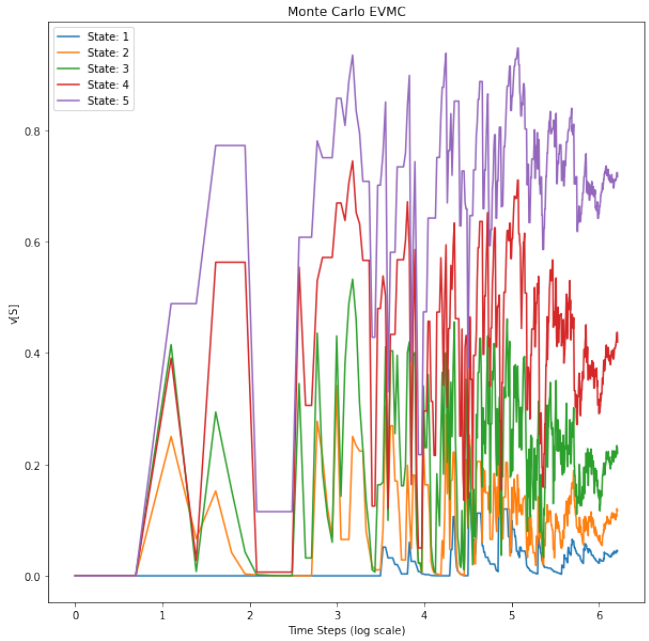

Decay in alpha with Time Steps
Linear Decay
Exponential Decay
alpha
Time Steps

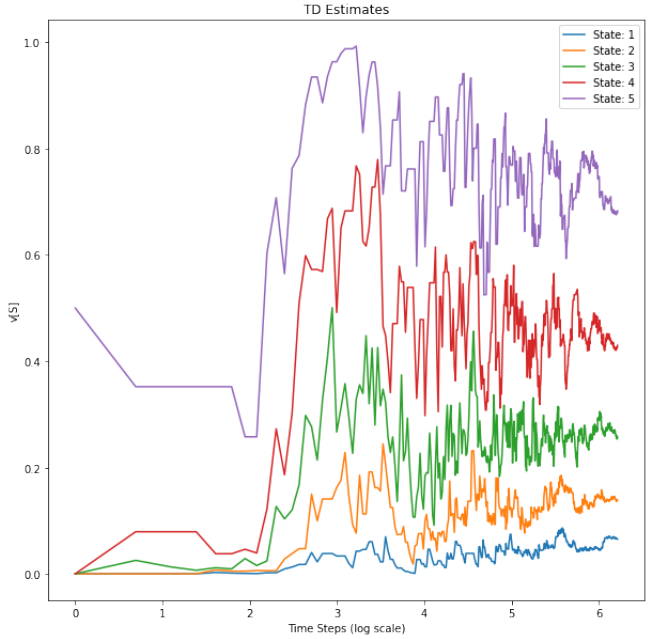

Monte Carlo FVMC
State: 1
State: 2
State: 3
State: 4
State: 5
v[S]
Time Steps (linear scale)

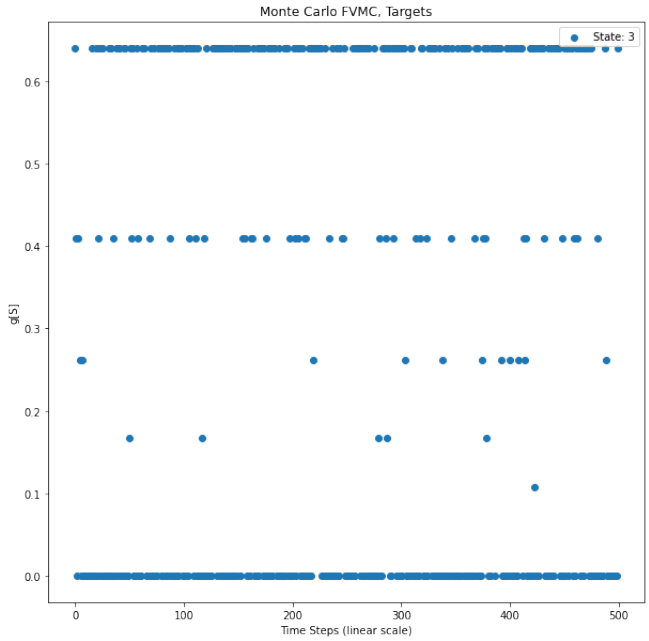

Monte Carlo EVMC
State: 1
State: 2
State: 3
State: 4
State: 5
V[S]
Time Steps (linear scale)

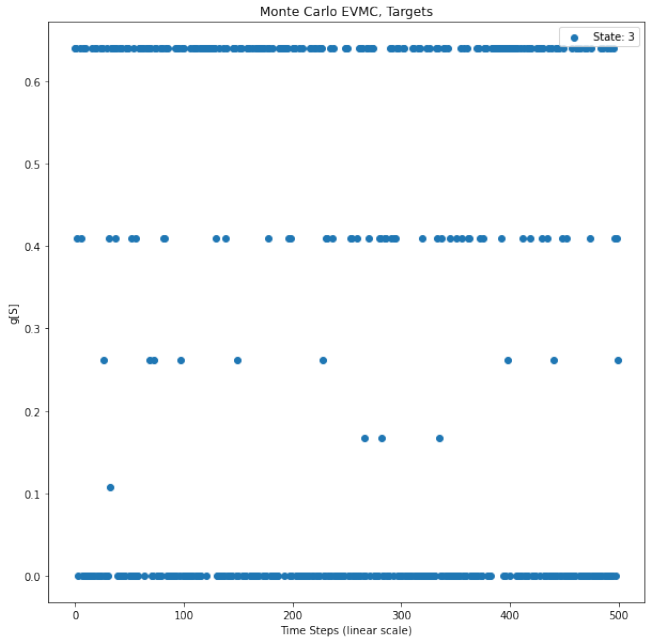

TD Estimates
State: 1
State: 2
State: 3
State: 4
State: 5
V[S]
Time Steps (linear scale)

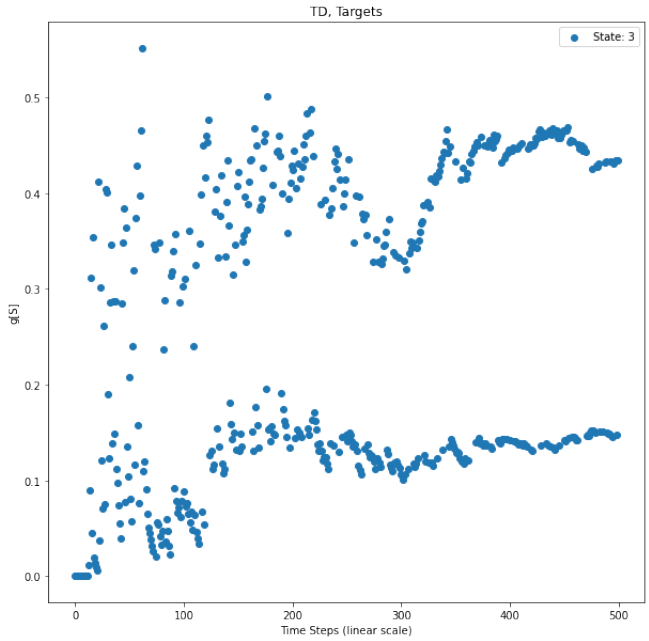

Monte Carlo FVMC
State: 1
State: 2
State: 3
State: 4
State: 5
V[S]
Time Steps (log scale)


Monte Carlo EVMC
State: 1
State: 2
State: 3
State: 4
State: 5
V[S]
Time Steps (log scale)


TD Estimates
State: 1
State: 2
State: 3
State: 4
State: 5
v[S]
Time Steps (log scale)


Monte Carlo FVMC, Targets
State: 3
g[S]
Time Steps (linear scale)


Monte Carlo EVMC, Targets
State: 3
g[S]
Time Steps (linear scale)


TD, Targets
State: 3
g[S]
Time Steps (linear scale)