Preliminary Experiment

January 29, 2018

Simulation Details

Considered $K = 3$, $T = 6000$.
The Bandit priors that were considered:

- Uniform: Draw the mean rewards for the arms from [0.25, 0.75]
- "HeavyTail": We took the mean rewards to be randomly drawn from Beta($\alpha = 0.6, \beta = 0.6$). With this distribution it was likely to have arms that were at the extremes (close to 1 and close to 0) but also some of the arms with intermediate value means.
- Needle-in-haystack
  1. Low - 9 arms with mean 0.50, 1 arm with mean 0.01 (+ 0.01)
  2. Medium - 9 arms with mean 0.50, 1 arm with mean 0.55 (+ 0.05)
  3. High - 9 arms with mean 0.50, 1 arm with mean 0.70 (+ 0.20)

Algorithms considered:

1. ThompsonSampling with priors of $Beta(1, 1)$ for every arm.
2. DynamicGreedy with priors of $Beta(1, 1)$ for every arm
3. Bayesian Dynamic ϵ-greedy with priors of $Beta(1, 1)$ for every arm
   (a) $\epsilon = t^{-1/3}$
   (b) $\epsilon = T^{-1/3}$
   (c) $\epsilon = 0.05$
4. Non-Bayesian ϵ-greedy - the greedy decision was made based on empirical mean. When there were zero observations, assumed that the empirical mean was 0 (this seems questionable).
   (a) $\epsilon = t^{-1/3}$
   (b) $\epsilon = T^{-1/3}$
   (c) $\epsilon = 0.05$
5. UCB1
   (a) UCB1 with constant $\sqrt{2log(t)}$
   (b) UCB1 with constant 1

Simulation Procedure

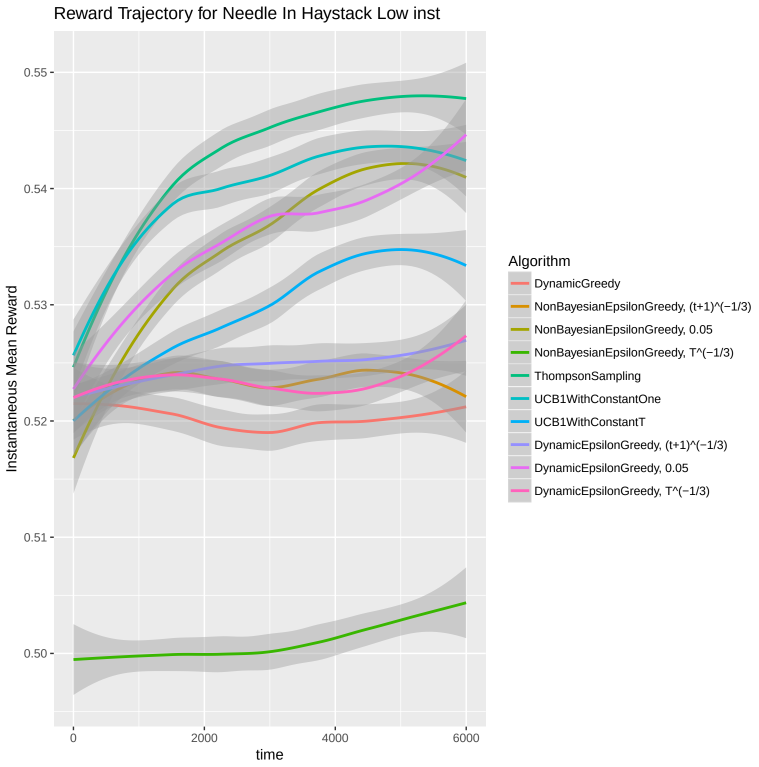
```
for Each prior $p$ do
    for Each experiment $i$ do
        Generate true distribution from $p$ (except for needle-in-haystack, just use $p$ itself)
        Generate realizations for each arm and round $t$ and instantiate bandit instance
        for Each algorithm $alg$ do
            Run simulation for $T$ periods
        end for
    end for
end for
```

One caveat to the reported results is that the Bayesian greedy results were run separately and thus have different true distributions and realizations than the rest of the algorithms. If we decide to include the Bayesian version instead of the non-Bayesian version we'll have to re-run these simulations.

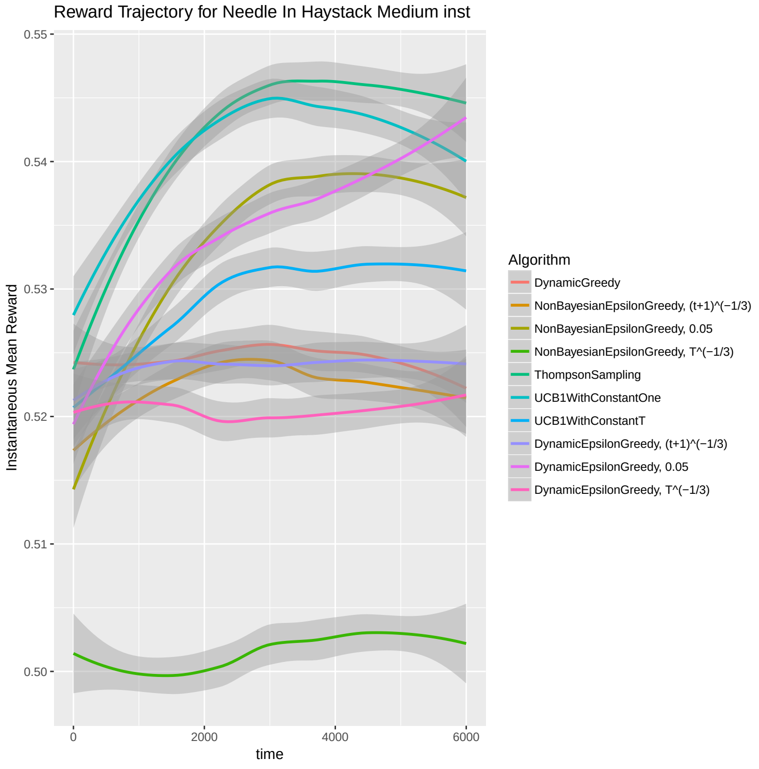
Results

The confidence bands come from the standard error of the loess regression used to generate the curves. The actual datapoints, when plotted, are incredibly volatile but this volatility decreases as we increase N. Part of this comes from the fact that what is plotted is the actual realized reward and not the mean reward *. We briefly discussed this last time and perhaps we should be reporting this instead (or in addition).

Instantaneous Reward Plots

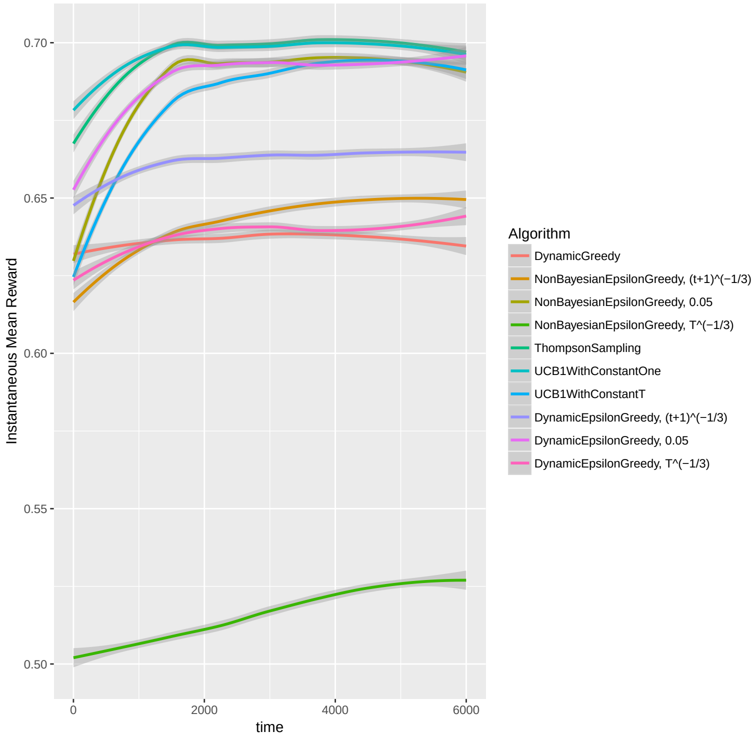


Reward Trajectory for Needle In Haystack Low inst

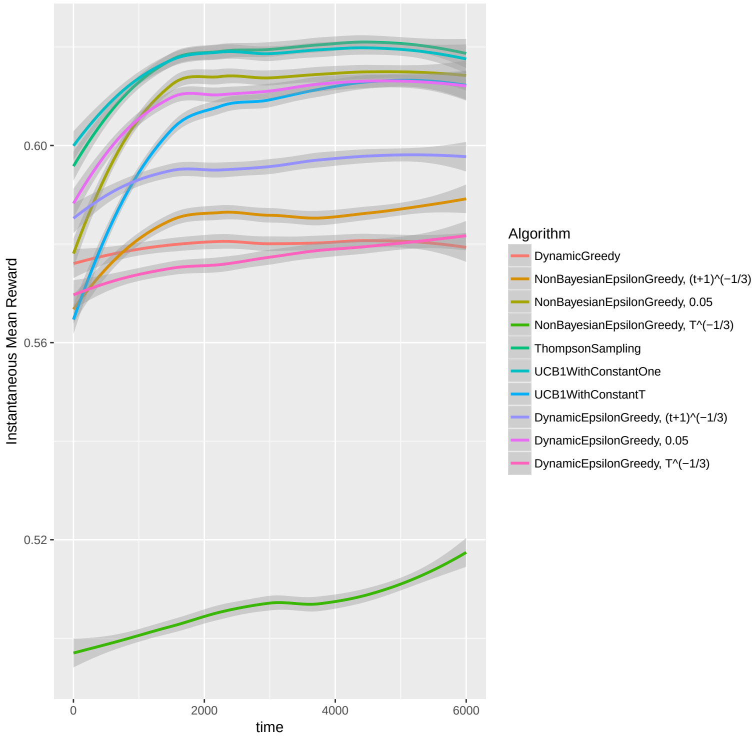


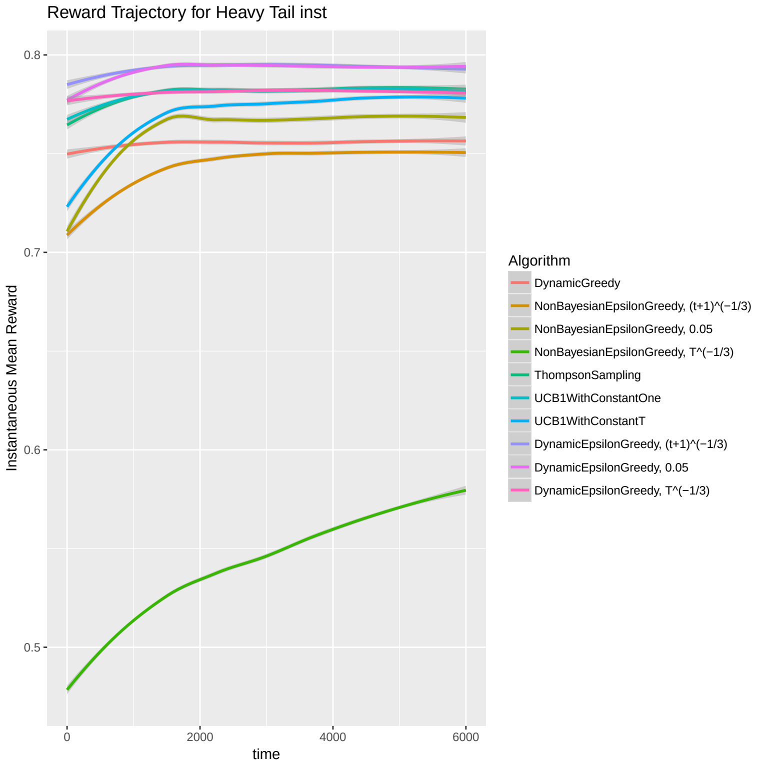
Reward Trajectory for Needle In Haystack Medium inst

Reward Trajectory for Needle In Haystack High inst



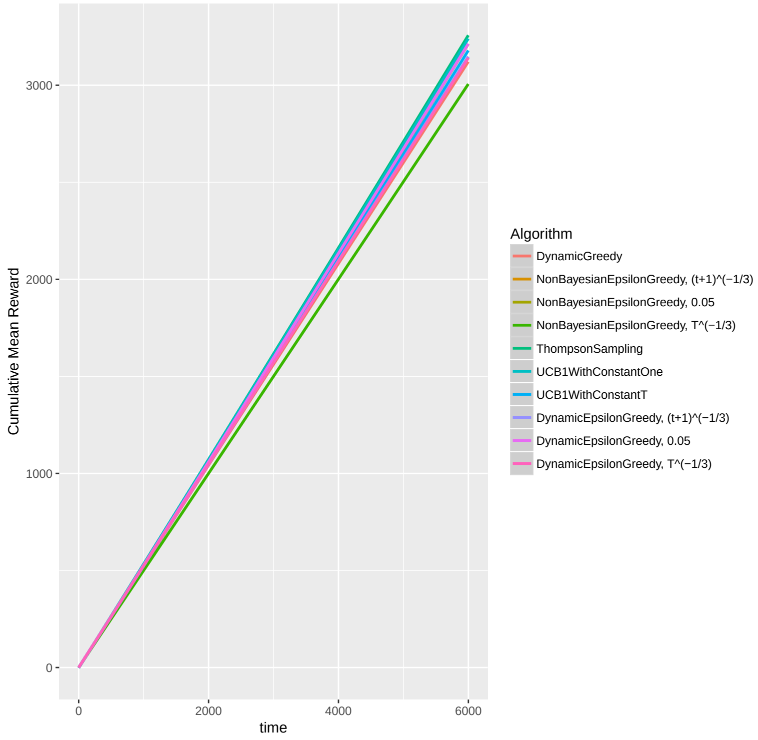
Reward Trajectory for Uniform inst



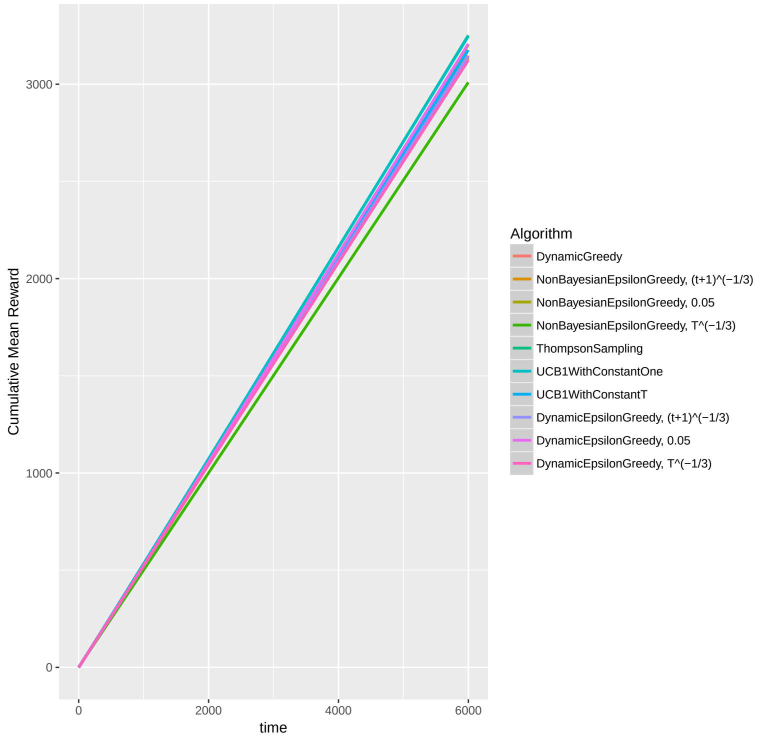

Reward Trajectory for Heavy Tail inst
Instantaneous Mean Reward
0.8
0.7
0.6
0.5
0
2000
4000
6000
time
Algorithm
DynamicGreedy
NonBayesianEpsilonGreedy, (t+1)^(−1/3)
NonBayesianEpsilonGreedy, 0.05
NonBayesianEpsilonGreedy, T^(−1/3)
ThompsonSampling
UCB1WithConstantOne
UCB1WithConstantT
DynamicEpsilonGreedy, (t+1)^(−1/3)
DynamicEpsilonGreedy, 0.05
DynamicEpsilonGreedy, T^(−1/3)

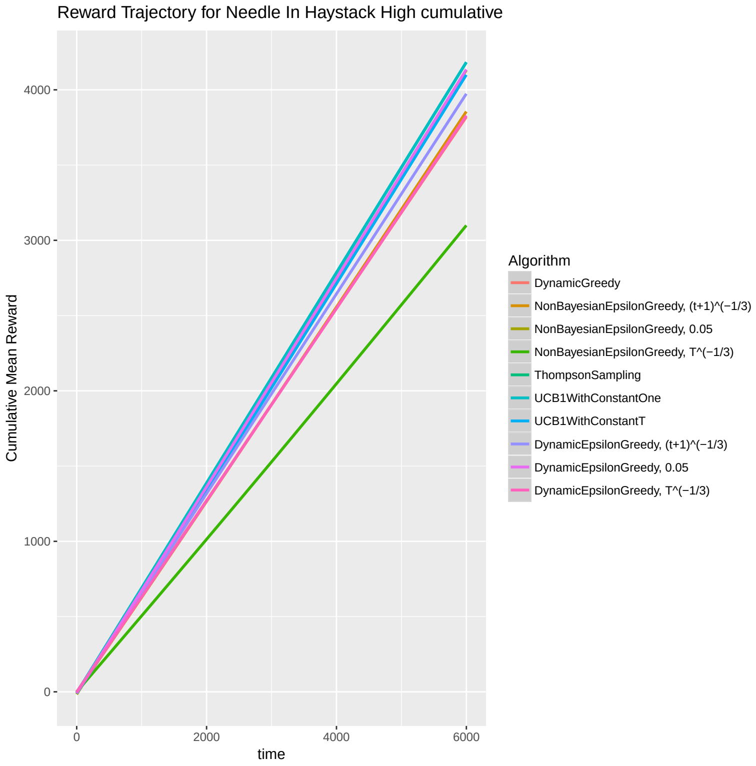
Cumulative Reward Plots

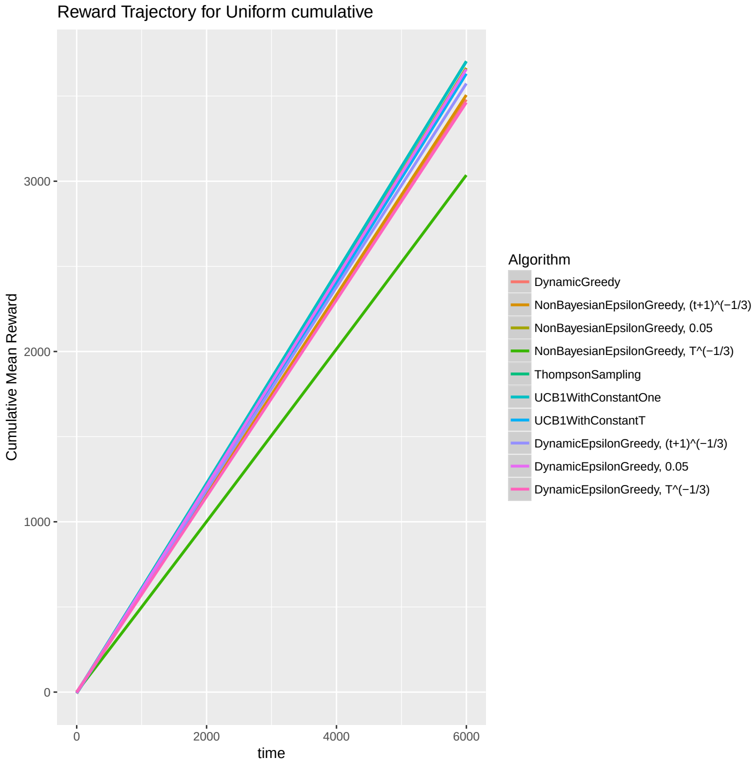
Reward Trajectory for Needle In Haystack Low cumulative

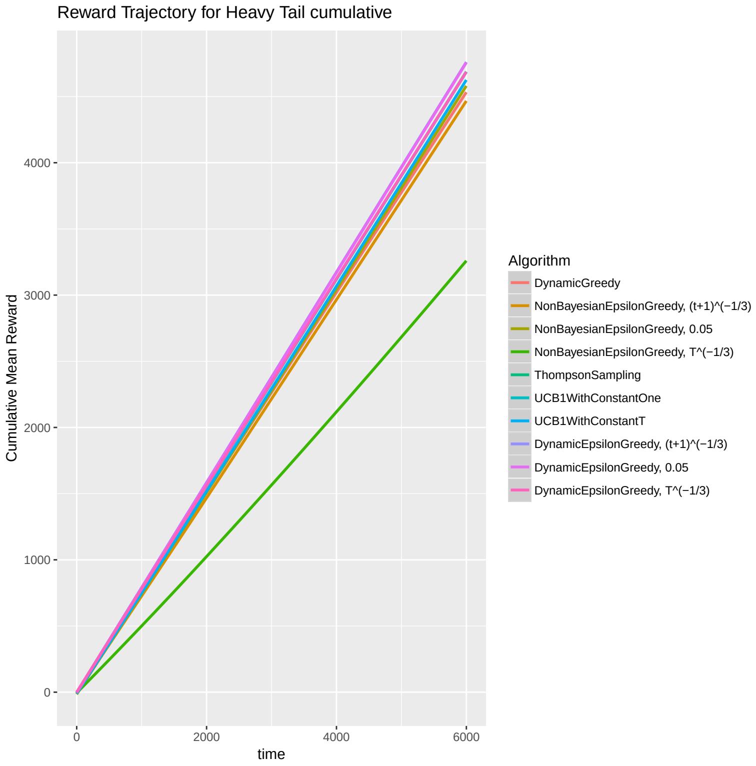


Reward Trajectory for Needle In Haystack Medium cumulative




Reward Trajectory for Needle In Haystack High cumulative
Cumulative Mean Reward
4000
3000
2000
1000
0
0
2000
4000
6000
time
Algorithm
DynamicGreedy
NonBayesianEpsilonGreedy, (t+1)^(−1/3)
NonBayesianEpsilonGreedy, 0.05
NonBayesianEpsilonGreedy, T^(−1/3)
ThompsonSampling
UCB1WithConstantOne
UCB1WithConstantT
DynamicEpsilonGreedy, (t+1)^(−1/3)
DynamicEpsilonGreedy, 0.05
DynamicEpsilonGreedy, T^(−1/3)


Reward Trajectory for Uniform cumulative
Cumulative Mean Reward
3000
2000
1000
0
0
2000
4000
6000
time
Algorithm
DynamicGreedy
NonBayesianEpsilonGreedy, (t+1)^(−1/3)
NonBayesianEpsilonGreedy, 0.05
NonBayesianEpsilonGreedy, T^(−1/3)
ThompsonSampling
UCB1WithConstantOne
UCB1WithConstantT
DynamicEpsilonGreedy, (t+1)^(−1/3)
DynamicEpsilonGreedy, 0.05
DynamicEpsilonGreedy, T^(−1/3)



Empirical Standard Deviation

Another thing we discussed previously was reporting and plotting the empirical standard deviation of the realized rewards. These tend to be quite large. Across almost every distribution and algorithm, the empirical standard deviation of the vector of realized rewards was on average 0.4-0.5. This was calculated, for each algorithm and prior, as follows: we have a vector of N realized rewards for each t and we can find the empirical standard deviation of this vector. Averaging across each t for each algorithm and prior, all of them have an average empirical standard deviation between 0.4-0.5. An alternative statistic of potential interest would be to look at the empirical standard of the mean rewards of the selected arms which would likely be smaller. Not sure which makes more sense.