Reinforcement Learning : Theory and Practice - Programming Assignment 1

August 2016

Background

It is well known in Game Theory that the game of **Rock, Paper, Scissors** has one and only one Nash Equilibrium. It is optimal for both players to play *Rock*, *Paper*, and *Scissors* with a probability of $\frac{1}{3}$ for each action. Additionally, Rock, Paper, Scissors, is a zero-sum game, and in constant-sum games, fictitious play is guaranteed to converge to a Nash Equilibrium (Source: My Game Theory class). In the case of **Rock, Paper, Scissors**, fictitious play should lead to each player playing each action with a probability of $\frac{1}{3}$. We now provide a more intuitive explanation for fictitious play. In a repeated game setting, each player plays the optimal move given the frequency distribution over the other player's past actions. For example, suppose player *A* plays *Rock* and player *B* plays *Paper* in the first round of **Rock, Paper, Scissors**. After round 1, player *A* believes that player *B* will play *Paper* with probability 1, and so in round 2, player *A* plays *Scissors* to counter *Paper*. Similarly, player B believes that player *A* will play *Rock* with probability 1, and so player *B* plays *Paper* again in round 2. After round 2, player *B* has played *Paper* twice, so player A still believes that player *B* will play *Paper* with probability 1, and as a result player *A* will play *Scissors* again in round 3. However, *A* has played *Rock* once and *Scissors* once, so player *B* believes that player *A* will play *Rock* with probability $\frac{1}{2}$ and *Scissors* with probability $\frac{1}{2}$. This process continues infinitely, where each player chooses the optimal move based on the frequency with which the other player has chosen his actions. Eventually, both players will play *Rock*, *Paper*, and *Scissors* with a probability of $\frac{1}{3}$ for each action.

Problem Statement

We have a reinforcement learning agent that plays **Rock, Paper, Scissors** against a fictitious opponent. This fictitious opponent plays as described above, playing the optimal action based on the empirical distribution of the RL agent's actions. We formulate this as nonstationary 3-armed bandit problem. The agent can choose amongst three actions - *Rock*, *Paper*, and *Scissors*. The agent receives one of three rewards: 0 for a draw, -1 for a loss, and 1 for a win. This may not seem like a 3-armed bandit problem, but it is. The player can take one of three actions, and each action yields

a reward, that effectively, is drawn from a distribution. While for the rest of this report we refer to the fictitious player playing, we can think of this as a 3-armed bandit problem. The agent's reward is based upon the fictitious player's actions, but we can consider the fictitious player providing the k-armed bandit with a distribution over rewards at each time step. In this sense, our 3-armed bandit is highly nonstationary.

Motivation

This problem is very interesting because the agent's actions *predictably* changes the distribution over his reward. If the agent begins to favor one action for too long, then the fictitious player will play against that action, and the agent will begin losing. Thus, the agent may cycle forever trying to find the single most valuable action. It almost seems that the agent must learn the fictitious player's *behavior* to determine his best course of action, as opposed to learning a distribution that is in a sense, "rigged" against him. Additionally, while the agent may look for one optimal action, the key insight that the agent must discover is that there is no optimal action, but rather, an optimal strategy. The agent must choose his sequences of actions so as to maximize his rewards. One other motivation for this experiment is to evaluate the methods from Chapter 2 in an abnormal setting. This experiment shows that a variety of settings (like games) can be made to work within the general framework of k-armed bandits, and it can be useful to evaluate these methods against these settings.

Experiment

As stated in the **Background**, when two fictitious players play one another in **Rock, Paper, Scissors**, the converge to the Nash Equilbrium, that is, they take each action with probability $\frac{1}{3}$. However, while we have simulated two fictitious players and achieved the desired Nash Equilibrium, it can take quite long (in computation time) for the players to reach the Nash Equilibrium. As such, we do not focus on having our agent reach the Nash Equilibrium. Instead, we focus on the average reward of the agent, as that can easily be compared to the average reward of 0, that the fictitious players experience in the long run. So in a sense, by comparing our agent's average reward against that of a fictitious player, we are assessing whether our agent performs better than a fictitious player, when playing a fictitious player.
We run our experiments for 50 million time steps, and we test the average rewards obtained by various value estimation methods and action selection methods.

Hypothesis

I believe that greedy methods will not work well here, because as stated in the motivation, favoring one action too much causes loss. It is my personal opinion that none of the methods will work very well. Additionally, if the agent is successful.

Sample-Average Method

We first consider the case that our agent chooses actions greedily. The fictitious player will always play against the *action that our agent has played most frequently*. As a result, the more our agent plays an action, that action lessens in value over time, and our agent will switch actions. However, since our agent is greedy, it will repeat this new action and eventually the new action will lose value as well. As a result, the fictitious player always eventually exploits our greedy agent's "greediness" by playing against that action. We hypothesize that the greedy agent will perform poorly. The case where our agent is ε-greedy is unclear. As the fictitious player slowly begins to play against our agent's actions, it is beneficial for our agent to randomly switch actions. However, it is unintuitive to estimate the success of such behavior, and as such, We hypothesize that an ε-greedy agent will perform poorly as well, though better than a greedy agent. Further, we believe that as ε increases, the agent will improve its average reward, and that when $\varepsilon = 1$ (i.e. the agent chooses actions uniformly randomly), the Nash Equilibrium will emerge, and the agent will have an average reward of 0.

Recency-Weighted Average

The recency-weighted average method of estimating values of actions places more weight on recent rewards. If our agent is playing greedily, this should prove useful for detecting when the fictitious player is targeting the agent's current action (e.g. playing *Rock* when our agent plays *Scissors*). As α increases, we hypothesize that our agent will perform better. A high α, coupled with ε-greedy action selection should yield positive rewards.

UCB with Recency Weighted Average Estimation

When our agent uses UCB action selection, we have him use a recency weight average value estimation. The parameter we vary for UCB is the constant c. During exploration, actions estimated to be "better" are more likely to be explored. In a sense, this is greedy-like, and as such this property will, in our opinion, cause the agent to perform poorly. On the other hand, the second term will benefit the agent, as it forces the agent to take actions that it has not taken in many time steps. This creates a balancing effect, but in our opinion, unless c is extremely high, is insufficient to offset the general issues facing UCB.

Softmax Action Selection

Softmax action selection, in our opinion, should perform better than any of the above methods. This is because when updating preferences, the average reward from all time steps is used. If an action is significantly worse than the average, then likely it is an action that has been performed many times, because the fictitious player plays against that action. The update rule will lower the preference of this action, and the agent will taken other actions, that perhaps it has not taken many times as of yet. That

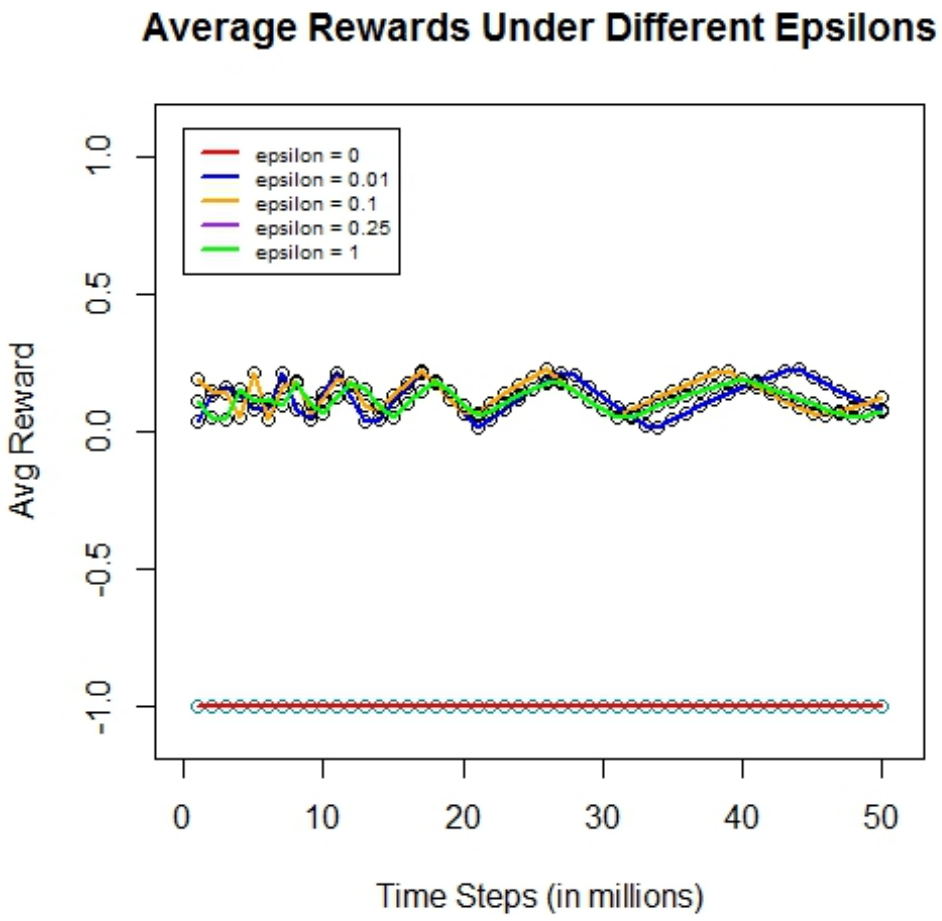
is, since the fictitious player plays against the agent's most frequent action, and softmax action selection provides an opportunity to detect this action quickly, the agent can obtain more reward by avoiding that frequently played action.

Results

Sample Average Method

Below are the experimental results we obtained by having our agent use the sample average method of estimating action values. We graphed our agent's average reward as a function of the number of time steps passed. We do this for multiple levels of greediness.

Figure 1: Average Rewards using Sample Average Value Estimation



Our hypothesis that a greedy agent would perform poorly is correct. Interestingly, a greedy agent *loses* the game of **Rock, Paper, Scissors** every time. We did not anticipate this level of performance. What we learn from this is that fictitious play performs very effectively against agents who solely exploit. We hypothesized that ε-greedy agents would perform better than greedy agents, and our hypothesis was correct. However, we expected ε-greedy agents would perform poorly. However, these ε-greedy agents average a positive reward! Further, the ε-greedy agents' average reward functions are oscillating, with different wavelengths, but they all are within the same neighborhood of values. This is especially intriguing, since this suggests that even a small amount of exploration is sufficient to obtain positive reward against a fictitious player. This also suggest that there is an upper bound on the average reward, since

when $\varepsilon = 0.01$ and when $\varepsilon = 1$, we still get roughly the same values.

Recency-Weighted Average Method

Below are the experimental results we obtained by having our agent use the Recency-Weighted Average Method of value estimation. We run experiments across different levels of greediness, and we vary the step-size parameter. Each graph corresponds to a different step-size parameter (see captions).

Figure 2: Average Rewards using Recency-Weighted Average Value Estimation with $\alpha = 0$

When $\alpha = 0$, the estimates do not change. In each case of ε in the above graph, the agent had the same initial estimate of his actions, so the action value remained constant across all ε's we tested. Thus, we can conclude that given the differing average rewards in the graph, that simply the act of exploring more yields a greater average

reward. That said, all the plots on the graph have a negative average reward, so the agent definitely performed poorly in this case.

Figure 3: Average Rewards using Recency-Weighted Average Value Estimation with $\alpha = 0.33$

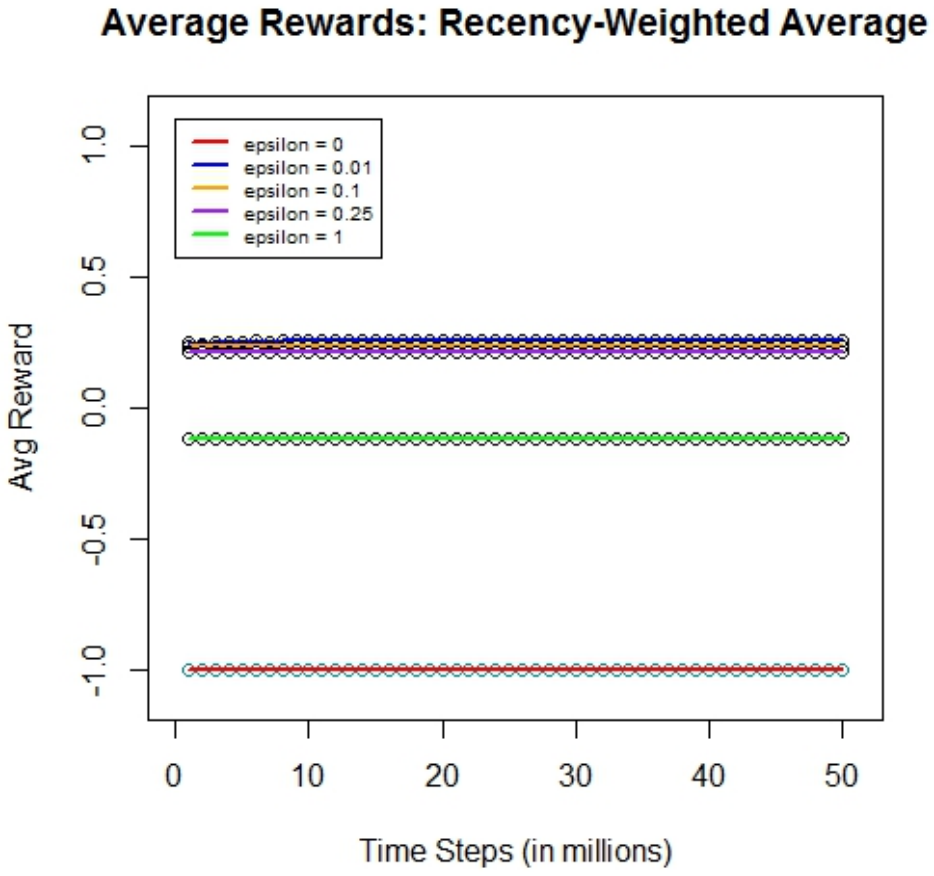


Figure 4: Average Rewards using Recency-Weighted Average Value Estimation with $\alpha = 0.67$

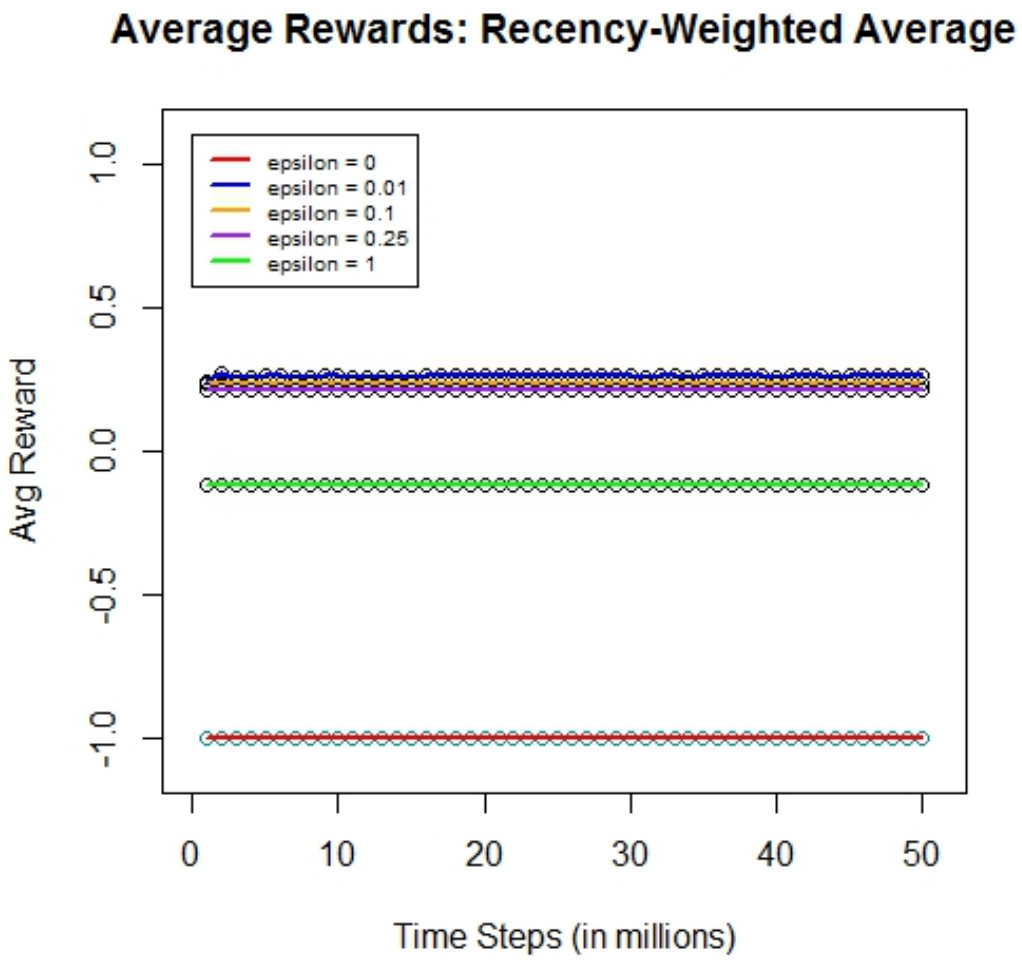
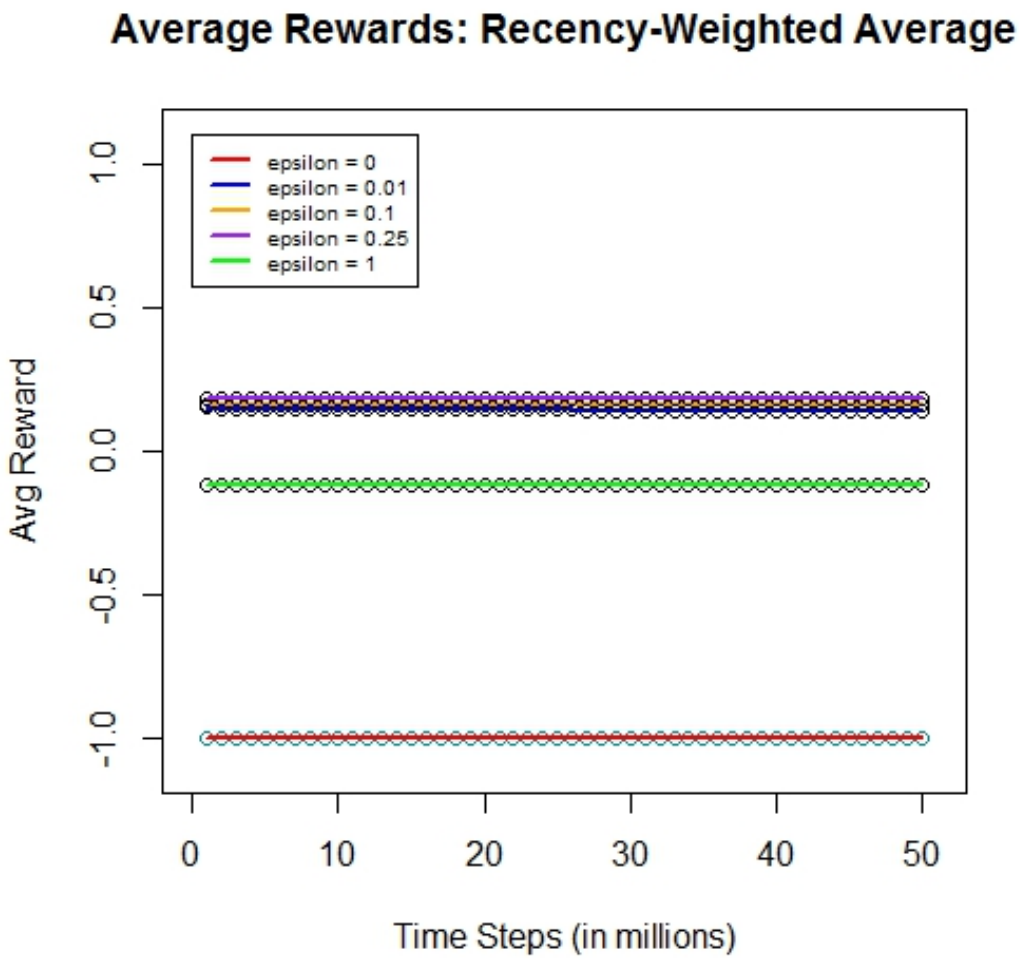


Figure 5: Average Rewards using Recency-Weighted Average Value Estimation with $\alpha = 1$



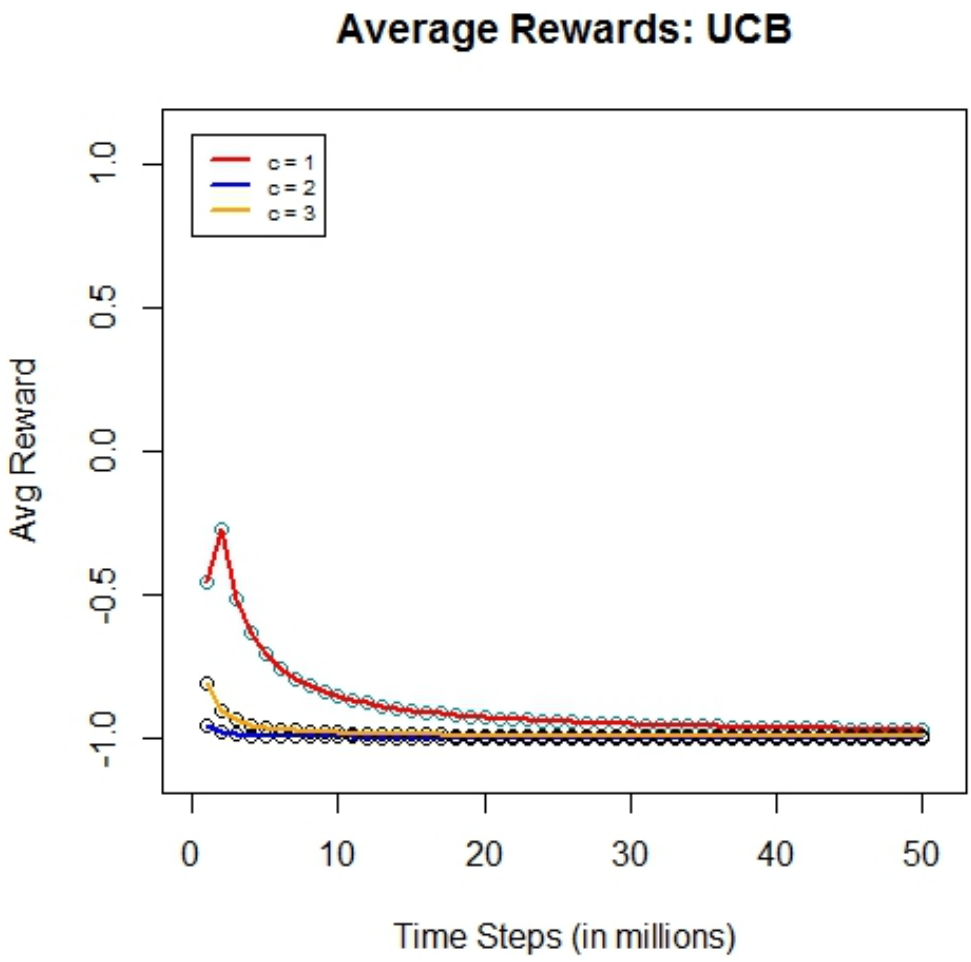
The above three graphs partially support our hypothesis. Our hypothesis was that as α and ε both increased, the average reward would increase. The graphs do support our hypothesis in that a high α and ε yield a positive average reward. However, given the clusters in the graph, we can infer that we do not *need* a high ε or a high α. In fact, it seems that a positive α and a positive ε suffice to ensure a positive average reward. However, there is one notable exception from all three graphs: the case where $\varepsilon = 1$, where we place a very high weight on the most recent reward and no weight on previous rewards. Intuitively, to completely disregard the past should surely lead to a lesser reward, especially since the fictitious player's actions are purely dependent

upon our agent's history.

UCB

The following graph depicts our experimental results from having our agent use UCB action selection. We had the agent use Recency-Weighted Value estimate, with $\alpha = 0.5$ for all of these UCB experiments.

Figure 6: Average rewards using UCB action selection and Recency-Weighted Average Value Estimation with $\alpha = 0.5$



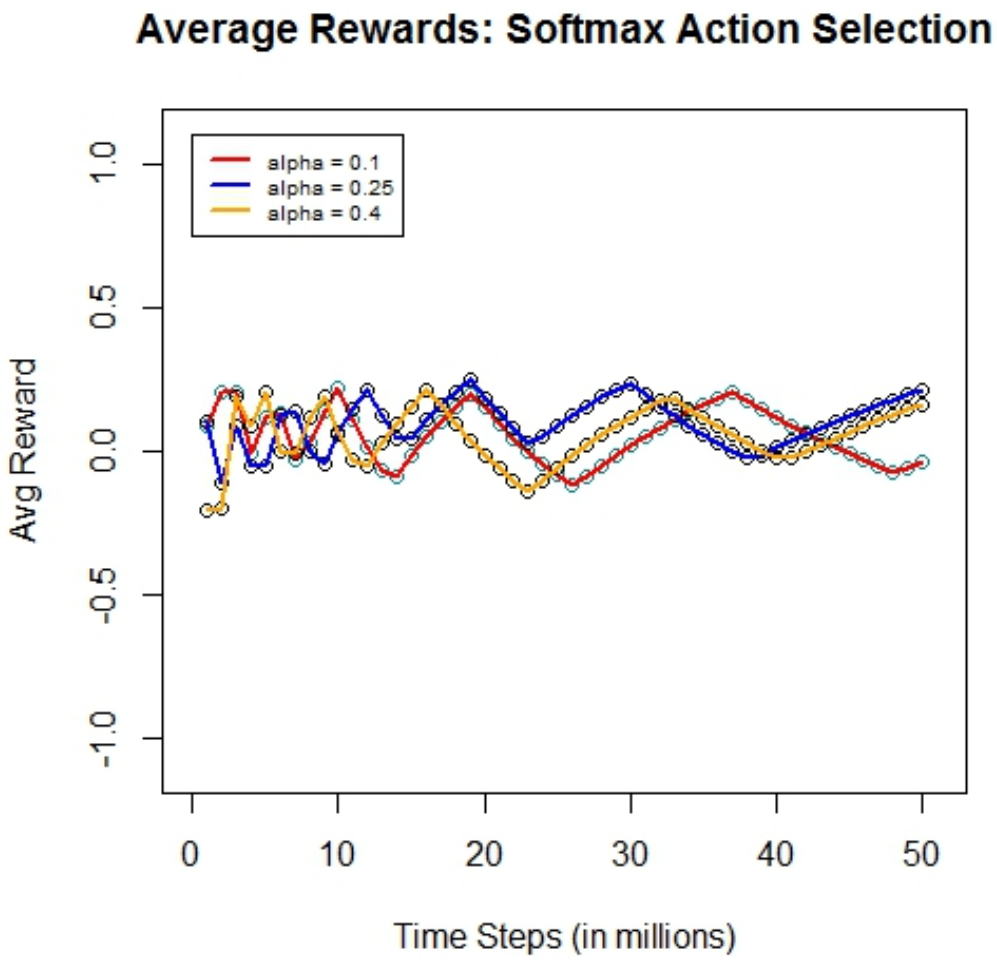
The above graph supports our hypothesis in that we expected UCB to perform poorly.

However, we believed that increasing c would improve the performance, which is not supported by the experimental data. The only conclusion we can draw is that because UCB emphasizes exploring actions that are more likely to be optimal, increasing c would make these actions be chosen more often. However, if the agent believes certain actions to have a higher likelihood of being optimal, then that would suggest that the agent has exploited these actions in the past. As a result, the agent may have selected these actions with a high frequency. We already know that fictitious play is quick to counter high frequency actions. Because fictitious play can counter these actions quickly, it effectively impedes the agent's UCB exploration process.

Softmax Action Selection

The graph below shows our agent's experimental results using a softmax action selection with varying αs.

Figure 7: Average rewards using Softmax action selection while varying α



We hypothesized that softmax action selection would perform better than all other value estimation or action selection methods. It seems that at best softmax action selection performs on par with some of the other methods, and at worst yields a negative reward. Similar to the sample average methods, the average reward is an oscillating function. These curves suggest that the agent experiences sequences where he wins the majority of his games, and then sequences where he loses the majority of his games. I believe that the softmax method takes a significant amount of time steps to lower the preference of an action. Suppose our agent is on a losing streak. Then it is possible that the agent is playing a high frequency action, against which the fictitious player is countering. If it takes a significant number of time steps to lower the pref-

erence of such a high frequency action, the agent would experience a losing streak, because he must continue playing high frequency actions. Similar, after sufficiently lowering his preference of the high frequency action, the fictitious player counters against the high frequency action, but the agent no longer plays that high frequency action (he has lowered its preference). From this point forward, the agent experiences a winning streak (until another losing streak).

Extensions

In this report, we have only explored one learning method from game theory, fictitious play. There are many other potential methods worthy of exploration. IF we can model other methods as bandit problems, we can perform this same analysis and experiments on those game-theoretic learning methods. Furthermore, we have many unanswered questions here. Why do the sample average and the softmax methods yield oscillating curves? There are still underlying mathematical properties to be explored, amongst other things.

Conclusion

Ultimately, we are trying to answer the question: What constitutes a satisfactory value estimation/ action selection method? Furthermore, if we have a definition of what constitutes a satisfactory value estimation/action selection method, can we produce a series of tests that test whether a certain method is satisfactory? There are infinitely many scenarios to consider, and it may be impossible to have one method that works under all types of nonstationarity. However, if we consider unique scenarios such as this, we may build insight into why these estimation methods succeed (or fail), and what properties they hold, and then we can extend these to other scenarios.