Competing Bandits

Guy Aridor, Kevin Liu

December 3, 2017

Introduction / Setup

In this project we mainly look at the following sequence of actions:

1. Two principals compete against each other for agents and each can only get reward if an agent picks them over the other. Before the beginning of the game, each principal selects a bandit algorithm to use.

2. At each $t = 1, ..., T$, a single agent enters (and lives only for one period) and selects a principal based on some belief they have about the expected payoff from picking each principal.

3. The chosen principal runs its bandit algorithm to select an arm. The reward generated by the arm is given to the agent.

This is based off the setup of the model in Mansour, Slivkins, and Wu (2017). We wrote a variety of simulations to look to see if we could reproduce the original results in the paper as well as a few experiments testing certain pertubations of the model.

The first step was to take the model in the paper and make it amenable to simulation as the methodology in the theoretical analysis of the paper made it hard to simulate the beliefs of the agent. To tackle this we assumed that the agent had access to a "reputation" score that came from a sliding window average of the past n agents that had picked this principal.

Then, we implemented the following four behavioral algorithms that describe the decision rule of the agent given the reputation score:

1. HardMax where the agent chooses the firm with a larger score, breaking ties uniformly

2. HardMaxRandom: Each agent uses HardMax with probability $1 - \epsilon$, and chooses between the principals uniformly at random with probability ϵ

3. SoftMax: Each firm is chosen with probability $e^{\alpha score}$ for some given α

4. SoftMaxRandom: Each agent uses SoftMax with probability $1 - \epsilon$ and chooses between the principals uniformly at random with probability ϵ

5. Uniform: Each agent chooses between the two principals uniformly at random

In the simulations we maintained the assumption in the paper that the principals commit to a learning algorithm in the first period and thus we implemented standard bandit learning algorithms for the principals. In general, we were interested in thinking about the differences that would arise when the principals were running greedy algorithms, non-adaptive exploration algorithms, or adaptive exploration algorithms. Bandit algorithms that we implemented include:

1. ExploreThenExploit

2. StaticGreedy

3. DynamicGreedy

4. DynamicEpsilonGreedy

5. ThompsonSampling

6. UCB

Experiment 1 - Finite Memory

Recall that in our simulations we encoded agents' beliefs about the principals as coming from a sliding window average. One set of simulations that we ran considered the consequences of changing the size of this window. This can be interpreted as agents having finite memory so that "bad" actions by the principals in the past will stop mattering at some point. Intuitively one would expect this to increase exploration even if we fixed agents as playing $HardMax$ and thus being expected utility maximizers. In the paper the main result around $HardMax$ was that ANY deviation from dynamic greedy would cause a principal to lose all remaining agents. However, if there is finite memory we don't necessarily expect this to be the case.

There are two possible ways to define finite memory and we report results from both. The results we report are both the market share that the principals get as well as the regret that they incur since we want to determine both if there is more exploration and if there are differences in the resulting share of agents they get (i.e. if they can recover from "bad"' choices).

The two definitions of finite memory we explore are:

1. Agents remember only the last n periods, no matter who was picked. If a principal wasn't picked in the last n periods, then we default back to the original prior.

2. Agents form their score for a principal based on the last n observations they have from that principal.

Agent: HardMax
Principals: Thompson Sampling vs Thompson Sampling

Memory Size	Principal 1 MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
1	0.498	0.502	0.021	0.021
5	0.5219	0.478	0.119	0.132
10	0.407	0.593	0.165	0.113
50	0.420	0.580	0.168	0.131
100	0.460	0.540	0.161	0.160

Agent: HardMax
Principals: Thompson Sampling vs Dynamic ϵ-Greedy

Memory Size	Thompson Sampling MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
1	0.508	0.491	0.020	0.029
5	0.464	0.536	0.128	0.109
10	0.248	0.751	0.218	0.062
50	0.160	0.839	0.259	0.0338
100	0.161192	0.834	n/a	n/a

For the case of 100, there were many cases when the principal was never chosen!

Experiment 2 - Which Algorithm Wins?

This experiment mainly looked at what happens if we fix the priors of the principals to be identical over the arms and the agents to be identical over the principals at the beginning of the simulation.

We looked at each behavioral assumption and asked, fixing that behavioral assumption, is there any case where playing a better learning algorithm significantly helps the principal get market share?

[insert plots]

Experiment 3 - Tuning SoftMax

This experiment mainly looked at what happens if we do a grid search over alpha parameters in tuning SoftMax.

[insert plots]

Experiment 4 - Prior or Algorithm

A natural question to ask in this context is whether the correctness of the initial information that the principals have matters more than the learning algorithm they employ. Specifically, we want to test what happens if we fix one principal as having a better prior than the other but that principal plays *StaticGreedy* and thus only uses their original priors to decide on their actions. Suppose the other principal has a more "incorrect" set of initial beliefs but employs a more sophisticated, adaptive exploration algorithm such as Thompson Sampling. Will the principal playing a smarter learning algorithm catch up eventually (for some behavioral assumptions) or is the original prior more important?

We simulate this in the following scenario. We set $K = 10$ where 8 of the arms have approximately identical means ($Bernoulli(0.45)$), one of the arms (denote it as arm B) has a higher mean ($Bernoulli(0.55)$), and we vary the mean of the "best" arm (denote it as arm A) in increments of 0.1. Thus, we consider a simulation where $Bernoulli(0.55), Bernoulli(0.65), Bernoulli(0.75)$.

We suppose that the "dumb" principal 2 has relatively correct beliefs in that she has a prior such that arm B is the best arm. We suppose that the "smart" principal 1 has perverse beliefs such that she has a prior that the 8 worst arms have the best mean reward ($Beta(0.45, 0.55)$), arm B has slightly worse mean reward ($Beta(0.4, 0.6)$, and arm A has substantially worse mean reward ($Beta(0.1, 0.9)$).

We run simulations under this setup to see if the "smart" principal's learning algorithm can let her overcome the bad beliefs.

Agent: HardMax, T = 5000

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
0.55	ThompsonSampling	0.440	0.560	0.049	0.00
0.65	ThompsonSampling	0.647	0.353	0.057	0.100
0.75	ThompsonSampling	0.710	0.291	0.059	0.200
0.55	Dynamic ϵ-greedy	0.298	0.656	0.057	0.000
0.65	Dynamic ϵ-greedy	0.335	0.664	0.138	0.100
0.75	Dynamic ϵ-greedy	0.442	0.558	0.154	0.200

Agent: HardMaxWithRandom, T = 5000

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
0.55	ThompsonSampling	0.317	0.683	0.031	0.00
0.65	ThompsonSampling	0.630	0.369	0.028	0.100
0.75	ThompsonSampling	0.787	0.291	0.015	0.200
0.55	Dynamic ϵ-greedy	0.343	0.702	0.038	0.000
0.65	Dynamic ϵ-greedy	0.400	0.600	0.076	0.100
0.75	Dynamic ϵ-greedy	0.615	0.385	0.074	0.200

Agent: SoftMax, T = 5000

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
0.55	ThompsonSampling	0.442	0.558	0.027	0.00
0.65	ThompsonSampling	0.600	0.400	0.025	0.100
0.75	ThompsonSampling	0.721	0.280	0.026	0.200
0.55	Dynamic ϵ-greedy	0.442	0.556	0.038	0.000
0.65	Dynamic ϵ-greedy	0.400	0.600	0.021	0.100
0.75	Dynamic ϵ-greedy	0.666	0.334	0.050	0.200

Regardless of the behavioral model of the agent, we see that the "dumb" principal 2 wins a higher share of the market than principal 1 when she happens to have correct prior beliefs about the best arm. This happens whether principal 1 plays an adaptive or a non-adaptive exploration algorithm. Intuitively this makes sense as the "dumb" principal will end up getting regret 0 simply because of her correct beliefs and it is incredibly hard for a smarter learning algorithm to make up for this.

However, as we increase the reward of arm A such that principal 2 no longer has completely correct beliefs (she believes the second-best arm is the best arm), we see that when principal 1 plays an adaptive exploration algorithm she catches up very quickly and takes most of the market. However, when principal 1 plays a non-adaptive exploration algorithm we see that the "dumb" algorithm with better initial information still wins out, especially when the agent is HardMax. As we make the agent "more behavioral", the non-adaptive exploration algorithm becomes progressively better.

The previous results show that, especially when playing an adaptive exploration algorithm, it is possible that a better learning algorithm helps overcome initial bad information even if the opponent has almost correct information. However, one expects this takes time and that perhaps if we set T to be lower, the "dumb" principal would still win. To test this, we re-run the same simulations as above but set $T = 500$

Agent: HardMax, T = 500

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	Principal 1 Avg Regret	Principal 2 Avg Regret
0.55	ThompsonSampling	0.181	0.812	0.088	0.0
0.65	ThompsonSampling	0.287	0.712	0.178	0.100
0.75	ThompsonSampling	0.274	0.723	0.244	0.200
0.55	Dynamic ϵ-greedy	0.183	0.817	0.100	0.000
0.65	Dynamic ϵ-greedy	0.169	0.831	0.190	0.100
0.75	Dynamic ϵ-greedy	0.297	0.703	0.251	0.200

This demonstrates that part of the results from above were driven by the large T and that with a low T of 500, the "smart" learning algorithm does not have enough time to overcome the bad initial information. Thus we conclude that the "smart" learning algorithm will matter if the number of rounds is sufficiently high but for a low number of rounds, better initial information may be more important than a better learning algorithm.

Experiment 5 - Warm Start and HardMax

What is the effect of a "warm start" on the performance of HardMax? In this case we define a "warm start" as giving the principal free observations (but do not change the information set of the agent). Our simulations show that with HardMax the initial rounds matter a lot in a particular simulation and whoever wins in the first few rounds takes the entire market. We explore if having a warm start makes it so that the market is more evenly split in a given simulation.

We experimented with giving a warm start of 0, 5, 25, 50, and 100 observations to each principal. We ran the following competing algorithms: UCB vs UCB, UCB vs DynamicGreedy, StaticGreedy vs UCB and observed little effect. Regardless of the algorithm that was played or the number of free observations we gave to the principals, the sequence of simulations lead to market shares in each simulation looking roughly the same. Namely, in a given simulation one of the principals would take the entire market but since the principals had the same priors over the arms and the agents had the same priors over the principals, it was random who would take the entire market. Thus, it seems that free observations for the principal do not impact the resulting market share.

Conclusion

conclude