Competing Bandits

Guy Aridor, Kevin Liu

December 6, 2017

Introduction

- Focused on a “Competing Bandits” model similar to Mansour, Slivkins, Wu (2017) where we have two principals that compete for agents
- Principals can only learn if the agents select them and aim to maximize the number of agents which select them
- We look at this model from a simulation perspective and run several experiments asking how certain variants of the setup change the results

Model Setup

1. Agents choose between two competing principals (who can be thought of as firms). Agents seek to maximize their reward. In our implementation, an agent's beliefs are based on the realized rewards that prior agents received.
2. Before the beginning of the game, each principal selects a bandit algorithm to use, and starts with some prior beliefs on the distributions of the K arms. Each time the principal is chosen by an agent, the principal runs its bandit algorithm and selects an arm. Each principal seeks to maximize its market share, and can do so by choosing arms with high rewards so that agents will choose them.
3. At each $t = 1, ..., T$, a single agent enters (and lives only for one period) and selects a principal according to the Agent algorithm and the agent's beliefs.
4. The chosen principal runs its bandit algorithm to select an arm. The reward generated by the arm is given to the agent. The principal not chosen is not aware of the $(arm, reward)$ pair.

Algorithm Preliminaries

- Principals:
  1. Adaptive Exploration
     - UCB1
     - Thompson Sampling
  2. Non-Adaptive Exploration
     - Dynamic ϵ-greedy
     - Explore Then Exploit
  3. Greedy
     - Bayesian Greedy
     - Static Greedy
- Agents:
  1. HardMax - Perfect expected utility maximizers based on beliefs over principals
  2. HardMaxWithRandom - With probability ϵ play random actions, otherwise play HardMax chosen actions
  3. SoftMax - The principals are chosen randomly according to a logistic function (higher scores $\implies$ higher probability of being chosen)

Simulation Details

- Considered only Bernoulli rewards with Beta Priors
- Simulations have $K = 10$
- Reputation score generated by a sliding window average (fix memory sufficiently high except for limited memory experiment)
- Unless specified otherwise, reported results found by averaging results from 25 simulations

Experiments - Overview

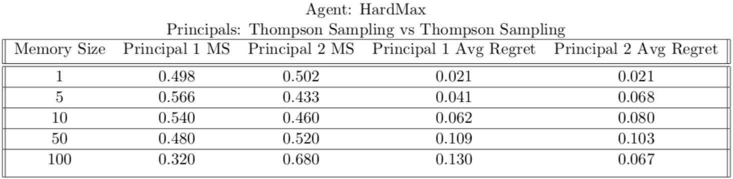
- In our simulations we considered the following five experiments:
    1. Competing algorithms
    2. Limited memory
    3. Tuning SoftMax
    4. Prior or Algorithm?
    5. Effects of “Warm start”

Experiment 1 - Competing Algorithms

- In this experiment, we ran several competing algorithms against each other.
- Several things we learned from this experiment:
  1. Regardless of the behavioral assumption, Thompson Sampling seems to get a larger market share regardless of what the competing algorithm is.
  2. In general, non-adaptive exploration algorithms seem to "beat" greedy algorithms.
  3. UCB does surprisingly poorly against every algorithm except for StaticGreedy
  4. There does not appear to be a large advantage to using an adaptive exploration algorithm compared to a non-adaptive exploration algorithm (as previously noted, Thompson Sampling does slightly better but UCB does worse).
  5. The results do not seem to qualitatively vary across different behavioral assumptions.

Experiment 2 - Limited Memory

- In this experiment we consider what happens when agents are perfect expected utility maximizers, but have limited memory.
- The driving question is can you get sufficient exploration in any given simulation simply by limiting how much agents remember?



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	0.566	0.433	0.041	0.068
10	0.540	0.460	0.062	0.080
50	0.480	0.520	0.109	0.103
100	0.320	0.680	0.130	0.067

Experiment 3 - Tuning SoftMax

- The logistic function takes the form

$$f(x) = \epsilon + \frac{1 - 2\epsilon}{1 + e^{-\alpha(x)}}$$

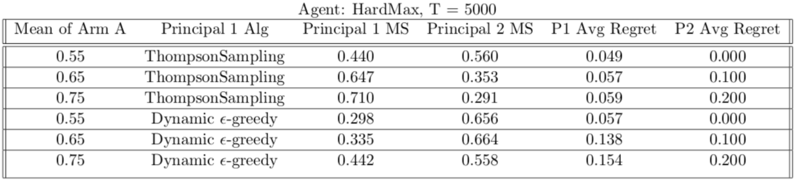
where ϵ controls the baseline probability each principal is chosen, and α controls the steepness of the curve. x is the difference in score between the two principals.

- We set up a situation where a "smarter" algorithm starts at a disadvantage to a "dumber" algorithm due to different prior beliefs, and tune values of ϵ and α that result in the "smarter" algorithm winning
- We settled on $\epsilon = 0.05$ and $\alpha = 10$.

Experiment 4 - Prior or Algorithm

- The purpose of this experiment was to attempt to figure out if it's better to use a better learning algorithm matters more or have better initial information.
- Ran simulations competing StaticGreedy (with almost correct beliefs) vs Thompson Sampling (with perverse beliefs) with $K = 10$
- StaticGreedy had highest expected mean on the 2nd best arm, but Thompson Sampling had priors that thought the worst arms

  were the best and that the best arms were very bad. Who wins?

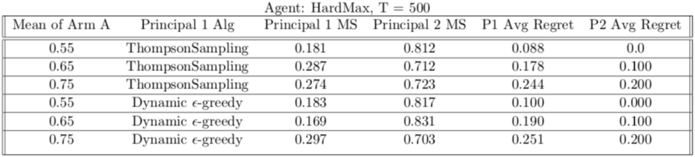


Agent: HardMax, T = 5000

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	P1 Avg Regret	P2 Avg Regret
0.55	ThompsonSampling	0.440	0.560	0.049	0.000
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

Experiment 4 - Prior or Algorithm Cont'd

- What if we drastically reduce the number of periods in the simulation?



Agent: HardMax, T = 500

Mean of Arm A	Principal 1 Alg	Principal 1 MS	Principal 2 MS	P1 Avg Regret	P2 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

- Given enough time, the algorithms that explore will catch up, but if we have sufficiently few periods then better initial information seems to win (according to our parameterization)

Additional Considerations

- What if we gave the principals free information at the start by giving them N agents? Some simulations showed that there was no difference.