Epsilon-greediness in competing and non-competing bandit problem

Brian O. [1] Kamorudeen Taiwo [1] Suleimana Mohammed [1] Asenath Odondi [1]

[1]Department of Mathematical Sciences, Purdue University Fort Wayne Supervisor: Prof. Alessandro Maria Selvitella

Bandits Problem

Bandits problems are basic abstraction of decision making under uncertainty with limited information. At each step. the agent chooses an **arm** or action $\mathcal{A}_i$ from a set actions $\mathcal{A}$, and afterwards gets a feedback signal, called reward ($\mathcal{R}_t$), that encodes the success of the action selection. The agent task is to learn the action selections that maximises expected reward (figs 1 & 2).

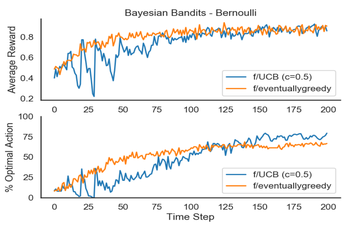

Figure 1. UCB(0.5)-Eventually greedy

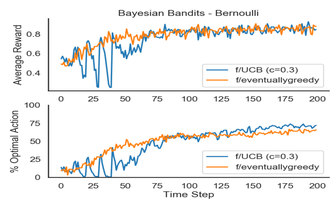

Figure 2. UCB(0.5)-Eventually greedy

Example applications include online advertising, adaptive routing and financial portfolio design. We introduce the eventually ϵ-greedy approach that performs as good as the state of the art, the upper-confidence bound algorithm (UCB) (figs 1 & 2).

In the competitive bandit setup, agents have preferences for actions and "arms" or actions are modelled to have preference for agents too. The preferences are driven by the available data which impacts the statistical uncertainty of the problem. Example applications include online marketing, crowd sourcing, medical resource allocation etc. The goal is to have a stable "agent-arm" pair matching.

Competing Bernoulli Bandit

Define the set of agents $\mathcal{P}$ and arms $\mathcal{A}$ respectively:

$$\mathcal{P} = \{p_1, p_2 \cdots p_N\}, \quad \mathcal{A} = \{a_1, a_2 \cdots a_K\} \quad \text{where} \quad N \leq K$$

Each arm $a_j \in \mathcal{A}$ has a known fixed ranking $\pi_j(i)$ of agents p_i. This could perhaps be from a recommendation system or model. Each agent also has a ranking of arms based on the reward, R_i for each a_i.

We define a stable matching as one in which no agent-arm pair prefer each other over their respective pair assignments. This is determined by the stability of the reward matrix from the bandit algorithm.

We run the bandit algorithm for a fixed ranking $\pi_j(i)$ of agents and return the reward matrix. Thereafter, given the reward matrix (of each arm by agents) and the fixed ranking (of each agent by arms), we use the Gale-Shapeley stable matching algorithm .

Balancing Exploitation-Exploration trade-offs

The bandit problem introduces the dichotomy between exploration ("of new actions") and exploitation ("of good actions"). Our approach balances exploitation and exploration by starting out with an exploration phase and eventually transitioning to the exploitation phase. The $\boldsymbol{\epsilon}$ is the probability that an agent takes a random action (explore) or not (exploit). By using a variable but decaying $\boldsymbol{\epsilon}$ we get an eventually greedy agent.

Reference

[1] Liu, L. T., Mania, H., and Jordan, M. I. Competing bandits in matching markets, 2020.

Experimental Results

Eventually Greedy And Epsilon-greedy Bernoulli Bandits

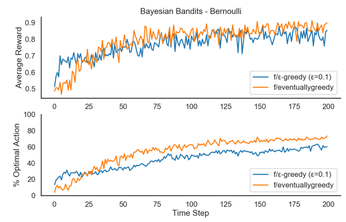

Figure 3. Epsilon greedy(0.1)-Eventually greedy

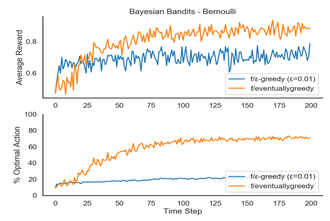

Figure 4. Epsilon greedy(0.01)-Eventually greedy

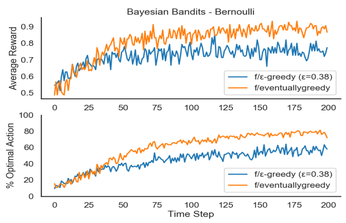

Figure 5. Epsilon greedy(0.38)-Eventually greedy

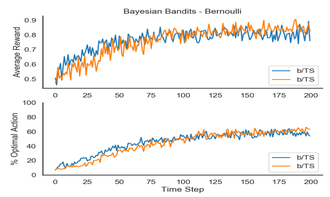

Figure 6. Under Thompson Sampling

From figs. 3, 4 and 5 above, it is obvious that the eventually epsilon greedy algorithm always performs better than ϵ-greedy algorithm. Under Thompson sampling and with appropriate tweaking of parameters both algorithms are comparable (figure 6).

Competing Bernoulli Bandits

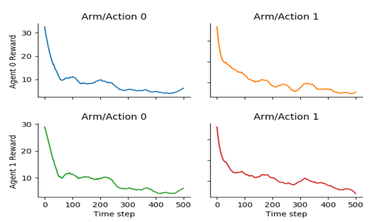

Figure 7. UCB-Eventually greedy

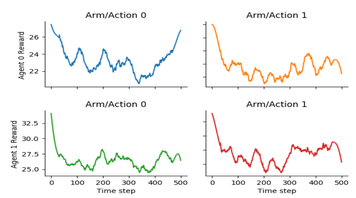

Figure 8. Epsilon greedy

Fig. 7 is the case where the reward matrix for the competing agents eventually converges. In this case, we fix a stopping criteria and apply the Gale-Shapeley matching algorithm. However, as shown in fig. 8, we can obtain unstable reward matrix when certain pair of agents are combined (in this case both agents are ϵ-greedy) resulting in unstable matching.

Further Remarks

The various agent algorithm combination and stability remark is as shown below:

Agents	UCB	Greedy	ϵ-greedy	Event.-greedy
UCB	✓	✓	✗	✓
greedy	✓	✗	✓	✓
Event.-greedy	✓	✓	✗	✗
ϵ-greedy	✗	✓	✗	✗

Table 1. Agent-pair combinations and stability of reward matrix in competing bandits

Model Setup

Given k arms (or actions), we define by $R_i(t), i \in \{1, 2 \cdots k\}$, the reward associated with playing arm i at step t. The reward is assumed to be sampled from an i.i.d Bernoulli distribution, **Bern(θ_i)** , where i represents the particular ith arm.

The true values of $(\theta_1, \theta_2, \cdots \theta_k)$ for all the arms are unknown and the agent has to learn this over time.

Let $\mathcal{F}(t) \in \{1, 2 \cdots k\}$ represent the arm played at time step t. The number of times a particular arm i has been played in the first t rounds is given as:

$$N_i(t) = \sum_{k=i}^{t} (\mathcal{F}(k) = i)$$

The associated reward, $R_i(t)$ and the corresponding mean reward $\theta_{i,N_i(t)}$ for arm i are given as:

$$R_i(t) = \sum_{k=1}^{t} \theta_i \mathbf{1}(\mathcal{F}(k) = i), \quad \theta_{i,N_i(t)} = \frac{R_i(t)}{N_i(t)}$$

A greedy algorithm would continue to play the arm that maximises $\theta_{i,N_i(t)}$ ($i_{optimal}$) while an ϵ-greedy would explore random arm sometimes with probability ϵ. An eventually greedy would start off being non-greedy taking random actions then eventually converges on selecting the arm that maximises $\theta_{i,N_i(t)}$.

Eventually Epsilon Strategy

To achieve eventually ϵ-greedy strategy, observe that:

$$\sum_{k=n}^{N} \frac{1}{2^k} = \frac{1}{2^{n-1}} - \frac{1}{2^N}, \quad N > n \quad (A)$$

In particular, observe that $\lim_{n \to N} \left(\frac{1}{2^{n-1}} - \frac{1}{2^N} \right) = 0$.

We compute ϵ iteratively from (A) by sampling n and N from a uniform random distribution such that $N > n$ and increment the sampling range at each step. This causes the $\epsilon \to 0$. The result of this strategy is that the agent eventually transitions to taking $i_{optimal}$ and not any other i.

Major conclusions

We introduce the **eventually ϵ-greedy** approach to solving the **Bernoulli multi-arm bandit** problem. This approach performs better than the ϵ-greedy approach and as good as the **upper-confidence bound algorithm (UCB)**. We also considered the competing bandit set-up and compare the performances of our algorithm and other other algorithms.

- **Reward and action selection** Eventually epsilon greedy approach performs better than ϵ-greedy approach under these two criteria. It has comparable performance to the Upper confidence interval algorithm.
- **Competing bandits and stability of results** The general observable pattern is that, any cross-combination of UCB, eventually greedy, and greedy agents result in agents' average reward that eventually converges. When both agents are either epsilon (or eventually) epsilon greedy or greedy then the plot of the average reward fluctuate across different runs.

These results are also reproducible under Thompson-sampling.