Adaptive ϵ-greedy for dynamic MAB

Itamar Golan Ira Rosenblum

March 2021

Abstract

In this work we examine a method to deal with non-stationary MAB environments, i.e. cases where the reward distribution per action is not constant, but neither is adversarial. We tackle the intermediary, but quite realistic scenario in which the change is well defined but not confrontational in nature. Our proposed method is empirically compared in various such cases to existing methods and evaluated using implementation and computer simulations. As we demonstrate, our method outperforms the classical approach in a significant majority of independently and pre-configured "drifting reward" scenarios.

1 Introduction

The Multi Armed Bandit problem is a statistical optimization problem, in which a fixed number of "Actions" (or "Arms") is available to choose from at discrete time frames, and after each action a reward from some initially unknown distribution is received. The problem gets its name from an image of a gambler at a casino, having a choice between several slot machines (also known as "one armed bandit"), but not knowing which one is best. The gambler can play any machine, possibly a large or even unbounded number of times, but he has to decide which on to play, when to play it, and at which order. Each "slot machine" awards the player with a different amount of reward, which is drawn from some potentially unknown distribution specific to that particular machine.

The MAB problem is a classical RL (Reinforcement Learning), in which an actor receives feedback from an environment following its choices. In an online continuous learning scenario, the problems can be viewed as an "exploration-exploitation" dilemma, in which the actor has to choose at each time step whether to continue his exploration in order to get more data and a better understanding of the underlying reward distributions, or to exploit his existing knowledge and understanding to choose the best arm and maximize its cumulative reward.

In classical MAB analysis, the environment from which the rewards and action results are drawn is considered to be static. That is, even though the learner does know - and may not know even after a long exploration run - the exact reward distributions of each action, the distributions are assumed to be constant and independent from the actor's actions. In this work, we propose an comparative empirical study to a different assumption - the environment in which the actor learns and operates is not static, but changes over time. It is therefore non-stationary, but dynamic - it evolves over time in ways which may be unknown beforehand.

In this study we look at existing models to address such dynamic environments, along with our own method, and put them to an empirical test - How do they fare in practical scenarios? We select the theoretical algorithms and methods, and implement them in code. Then, we run extensive and varying trials with real-life simulating scenarios to determine which methods produce the best results, measured by regret and reward.

2 Previous works

Existing research has proposed a few methods to extends the naive assumptions of the stationary MAB model. MAB problems with adversarial environments are extensively studied, as surveyed in (Bubeck and

Cesa-Bianchi 2012, Lattimore and Szepesv'ari 2018). The adversarial paradigm assumes the existence of a confronting adversary actively trying to counter the learner's actions and reduce its rewards. In this variant first introduced by Auer and Cesa-Bianchi (1998), at each iteration, an agent chooses an arm and an adversary simultaneously chooses the payoff structure for each arm. A specific, more limited type of Adversary is the oblivious one. We say that an adversary is oblivious if it is independent of the player's actions, i.e., if the reward at trial t is a function of t only.

However, the adversarial paradigm might be to strong for many real world use-cases, since it assumes the existent of an intelligence behind the environmental changes in the reward structure, when in many cases it is less organized and more simple in nature. This fact calls for a more lenient form of change in the reward structure. As an approach for this kind of scenarios, Besbes et al. (Besbes et al. 2014, 2015) proposed a framework for analyzing bandits in such "drifting environments", and considered the K-armed bandit setting with an assumption that the total change in a problem is upper bounded by B_T (which is $\Theta(T^\rho)$ for some $\rho \in (0, 1)$) known as the variation budget. They achieved the tight dynamic regret bound $O((KB_T)^{1/3}T^{2/3})$ by restarting the EXP3 algorithm (Auer et al. 2002a) periodically when B_T is known. Wei et al. (2016) provided refined regret bounds based on empirical variance estimation, assuming the knowledge of B_T, and Karnin and Anava (2016) considered the setting without knowing B_T and $K = 2$, and achieved a dynamic regret bound of $O(B_T^{9/50}T^{41/50} + T^{77/100})$ with a change point detection type technique.

In our work, however, we do not assume the existence of a variation budget or bound, and opt for a direct change in the exploration technique.

3 Proposed method

The basic MAB paradigm is the ϵ-greedy approach, in which a small parameter (ϵ) is selected, from which the exploration-exploitation ratio is derived. For each time step, an exploration is performed in $p = \epsilon$, and an exploitation in $p = 1 - \epsilon$. The exploitation is basically choosing the best arm according to the accumulated knowledge from the exploration, for example the arm with the highest mean reward so far.

In stationary environments this (and similar) approach might suffice since the "historical" accumulated data is still relevant to decision making even many time steps later. However, in non-stationary and dynamic frameworks this paradigm will collapse and even worse - it might still be mathematically valid, but with very suboptimal results and a high regret. ϵ-first and ϵ-decreasing strategies suffer from the same problem, even more so - the main assumption of these methods being that initially gathered statistics are of a greater value to future decisions than data collected later. In a dynamically changing environment, these ideas collapse and would perform poorly.
Our suggestions to counter this problem is the proposed adaptive ϵ-greedy method.

4 Simulated scenarios

To put our (and others') ideas to the test, we devised a set of real-life simulating scenarios that create an environment in which we can check the performance of different approaches. The main metrics for a model's success are of course regret and cumulative reward. The scenarios are the following:

1. gau-sinusoidal: a reward distribution following a gaussian distribution with a sinusoidal varying mean:
$$R \sim N(sin(a \cdot t + t_0), \sigma^2)$$
Real life example - on an online retail platform, seasonally changing customer preferences.

2. shifted gau-sigmoid: a reward distribution following a gaussian distribution with a shifted sigmoid varying mean:
$$R \sim N(S(x - t_0), \sigma^2)$$

Real life example - on an online retail platform, a sudden change in customer preferences due to some external event.

3. steepening gaussian: a reward distribution following a gaussian distribution with a decreasing variance:

$$R \sim N(\mu, \sigma^2/t)$$

Real life example - on an online retail platform, new and unstable products become veteran products with more constant pricing and customer acceptance.

5 Results

We designed a meta proof based on brute force search of different configurations built from the following items:

- Type of drifting: linear, sinusoidal, sigmoid etc.
- number of steps: 250, 25000, etc.
- number of arms: 2, 25, 50, etc.
- ϵ values: 0, 0.1, 0.2, etc.
- Gaussian noise of reward expectancy
- Algorithm – eps-greedy vs. ours AEG

Then, we used each permutation of those parameters to run an independent simulation. Using both our AEG algorithm and naïve epsilon greedy. Finally, calculating their corresponding reward and regret values. Altogether, we executed 10,000 simulations. In 93.9% of simulations our AEG outperformed. That is, regardless to specific (and pre-configured) drifting scenario or environment parameter, our AEG method seems like a generic drifting-environment successful solution.

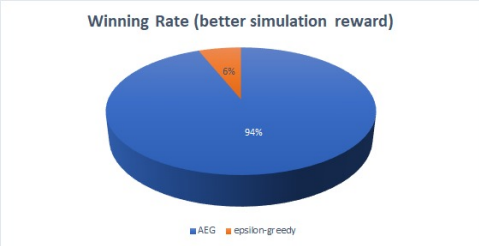

Figure 1: the best method on all parameters configurations

A few of our resulting computation are illustrated in the figures hereafter.

6 Discussion

7 References

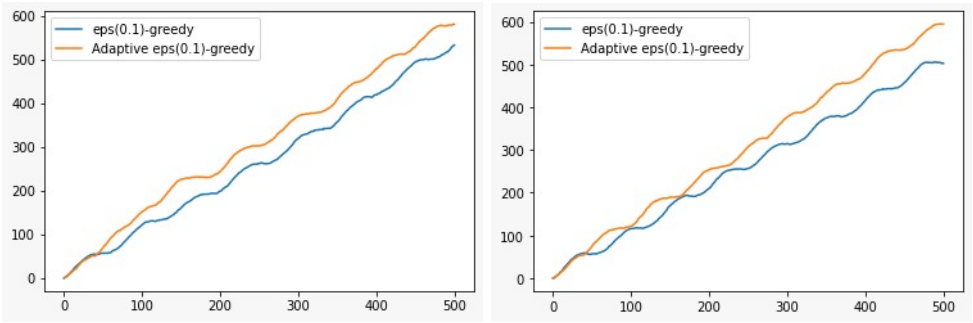

Figure 2: cumulative reward, 4 arms, $\sigma = 0.5$. on the left $\epsilon = 0.1$, on the right $\epsilon = 0.01$

We now delve right into the proof.

Lemma 1 *This is the first lemma of the lecture.*

Proof: The proof is by induction on ... □

Theorem 2 *This is the first theorem.*

Proof: This is the proof of the first theorem theorem. □

7.1 A few items of note

Here is an itemized list:

- this is the first item
- this is the second item

7.2 A few more items

Here is an enumerated list:

1. this is the first item
2. this is the second item

8 Next topic

We are now ready for a major definition.

Definition This is the definition of *myword*.

Corollary 3 *This is a corollary following from the definition of myword.*

Sometimes we define terms in the middle of a paragraph. This is a *different term* being defined. Wasn't that easy?

On to the next page:

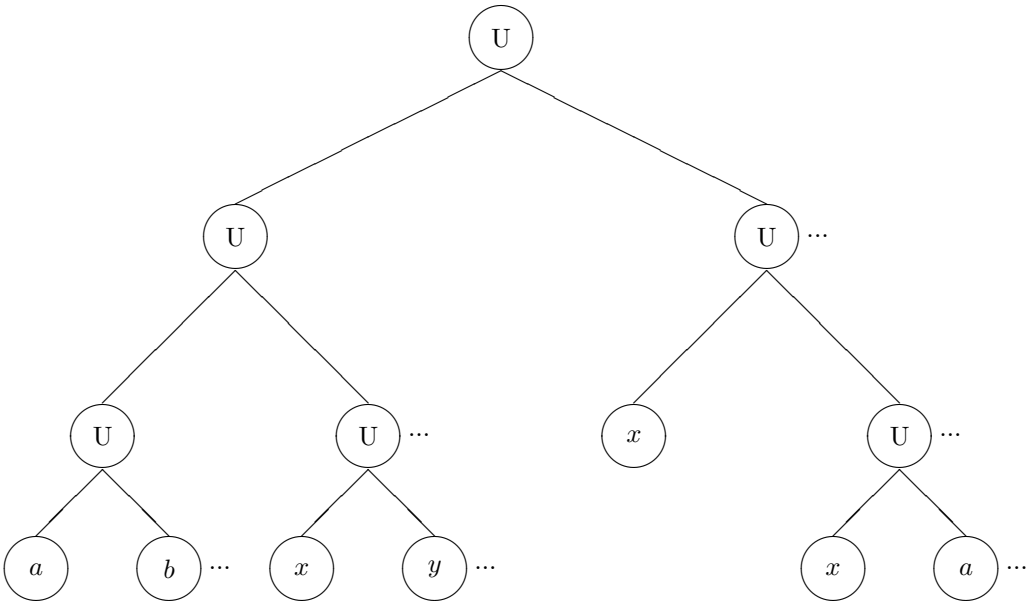

Figure 3: This is my picture.

Figure 4: This is a new picture.

This can be seen in Figure 3. Note that latex actually places this text *before* the figure, even though it appears after the figure in the .tex file.

```
FULL_i(h), h ∈ {1...n−1}
  begin
  if NUMV_i(ℓ_max, h) ≥ n − h
    then return (true)
         &lt;else &gt;return (false)
  end FULL

MAKELABEL_i
  begin
  if i ≠ i_max
    then h' := minimum h such that FULL(h) = true
         x_i := NEXTLABEL(ℓ_max, h')
  end MAKELABEL_i
```

Figure 5: Code for MAKELABEL$_i$ of BCTSS

.

9 Exercises

1. Kama-kama yatzaa Hapoel Beer-Sheva mul Makabee Tel-Aviv be-onat 82?

*2. Tanin hu yoter aroch o yoter yarok?

*3. Ma shem hamishpacha shel ha-denni sh-amar: "$2B \vee \neg 2B$".