Adaptive ϵ-greedy for non-stationary Multi Armed Bandits

Itamar Golan
203251152
golan.itamar@gmail.com

Ira Rosenblum
312508856
ira.rosenblum2@gmail.com

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

One of the basic MAB paradigms 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.

3.1 proposed algorithm

The classic ϵ-greedy selection strategy selects the expected optimal arm with probability $1 - \epsilon$, and one of the other $K - 1$ potential arms with probability $\frac{\epsilon}{K-1}$ for each arm.

Inspired by the epsilon-greedy algorithm, we designed a novel algorithm which is supposed to perform better than epsilon greedy in drifted environments. Our work is not less inspired by the concept of 'forgetting principal', meaning that we are weakening through time our learning dependency on the past. In drifting environments, mainly characterized by changing rewards distributions, the major failure points of algorithms such as epsilon-greedy is the equal dependency on recent and past observations. For example, in a frequent cosine reward stream, the last seen rewards are usually much more relevant for forecasting future rewards, than the average of all seen history. Therefore, we developed an algorithm which these basic characteristics into account. First, it is conducting estimation based on a shorter and more recent history of observations. Second, it calibrates the window size dynamically according to

the hereby defined "std moment", measuring the ratios between recent observations' std and less recent observations' std. Last, it estimates the reward distribution expectancy by a weighted average of the window's observations, with greater weight to the recent ones.

To put these ideas to pseudo-code for a MAB with K arms, with hyper-parameters ϵ for the exploration-exploitation ratio and α for the initial observation windows' size:

INITIATE $w_1 = w_2 = \ldots = w_K = \alpha$
INITIATE $\mu_1 = \mu_2 = \ldots = \mu_K = 0$ (MEAN ESTIMATORS PER ARM)
begin
for step t in (1 ... T)
  for arm i (1 ... K)
    recent_std = std({last w_i observations})
    less_recent_std = std({last $2w_i$ to last w_i observations})
    std_moment = $\frac{recent\_std}{less\_recent\_std}$
    if $std\_moment < 1$: (change is decreasing)
      $w_i := 2 \cdot w_i$
    else: (change is increasing)
      $w_i := 0.5 \cdot w_i$
  end for
end for
  $\mu_i = \frac{2}{w_i \cdot (w_i - 1)} \sum_{j=1}^{w_i} j \cdot \{observation\ (w_i - j)\ ago\}$ (weighted average with increased weight to recent observations)
  if $x \sim U[0,1] < \epsilon$
    return random action
  else:
    return $argmax(\mu_1, \ldots, \mu_K)$

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:

- linear drift: a reward distribution following a gaussian distribution with a linearly varying mean:

  $$R \sim N((a \cdot t + \mu_0), \sigma^2)$$

  Real life example - on an online retail platform, gradually but consistently changing costumer preferences.

- gau-sinusoidal: a reward distribution following a gaussian distribution with a sinusoidal varying mean:

  $$R \sim N(sin(a \cdot t + \mu_0), \sigma^2)$$

  Real life example - on an online retail platform, seasonally changing customer preferences.

- shifted gau-sigmoid: a reward distribution following a gaussian distribution with a shifted sigmoid varying mean:

  $$R \sim N(S(t - \mu_0), \sigma^2)$$

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

- 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.

The core idea behind simulating with different types of drifting reward distributions, is to show our model's robustness to any type of drifting, and to empirically demonstrate its superiority over the basic ϵ-greedy in such cases.

5 Results

5.1 elaboration on sino-gaussian test case

In order to examine our method, we initially tested several drifted environments. For simplicity's sake we'll specify here only the details of phase-shifted sinusoidal mean drift in reward distributions (visualized in figure 1)

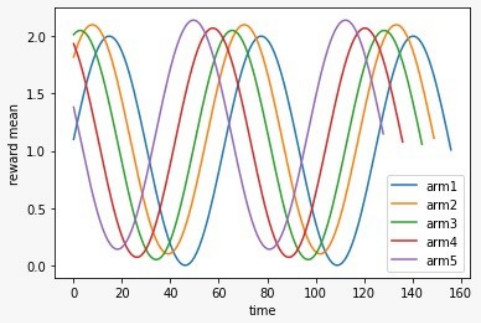

Figure 1: phase-shifted sinusoidal mean drift

We executed a simulation of ϵ-greedy versus our AEG, on several hyperparameters configurations and observed a distinct positive margin for the AEG in terms of cumulative reward, as figure 2 suggests

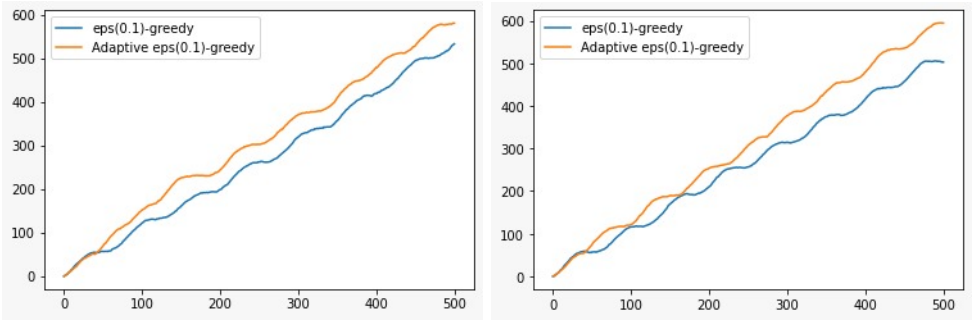

Figure 2: cumulative reward, 4 arms, $\sigma = 0.5$. on the left $\epsilon = 0.1$, on the right $\epsilon = 0.01$

Furthermore, we studied the relative performance boost AEG grants the cumulative reward and regret on top of the naïve ϵ-greedy baseline:

Environment	AEG Regret	AEG Reward
sinusoidal	−10.11%	+8.85%
cosinusoidal	−7.8%	+6.43%
sinusoidal squared	−8.32%	+2.96%
Sigmoid	−20.11%	+3.12%

5.2 General empirical testing framework

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 the specific (and pre-configured) drifting scenario or environment parameters, our AEG method seems like a generic drifting-environment successful solution.

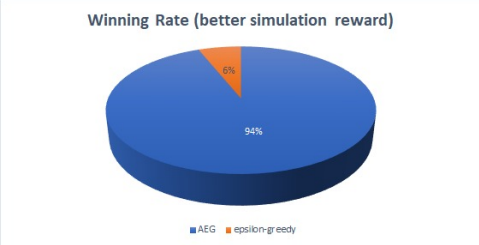

Figure 3: the best method on all parameters configurations

6 Discussion

In this work we were aiming to suggest a novel and practical solution to a common obstacle of multi arm bandits' algorithms known as non-stationary environments. MAB classical models are prone to fail when the rewards distribution suddenly change in certain ways. In simple words, we developed a promising method to deal with generic drifts in the arms' reward distributions, regardless of their specific natures. Our algorithm, AKA adaptive ϵ-greedy model, seems to bridge this issue, by dynamically and adaptively treating the history of reward observations in a more flexible way, letting older history be gradually forgotten, and allowing fresh data generated by the drifted effect, to rapidly adapt the model. Of course, this is an initial door opening investigation, and a wider more in-depth research is needed to derive even better methodologies to deal with the forgetting principle, maybe in a sense to be inspired by forecasting or time series ideas like auto retrogressive models or recurrent neural network. Our contribution though is that according to our experiments, not many observations are needed in order to achieve superiority over classical models in terms of accumulated regret and reward.

6.1 Code repository

All of our code is accessible in the following GitHub repository. Feel free to contact us for further code classes or documentation if needed for further exploration in this field.

References

[1] Omar Besbes, Yonatan Gur, and Assaf J. Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. CoRR, abs/1405.3316, 2014.

[2] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, COLT, volume 23 of JMLR Proceedings, pages 39.1–39.26. JMLR.org, 2012.

[3] Arthur Flajolet and Patrick Jaillet. Real-time bidding with side information. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5162–5172. Curran Associates, Inc., 2017.

[4] Aurlien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. In Algorithmic Learning Theory, pages 174–188, Oct. 2011.

[5] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. CoRR, abs/1402.6028, 2014.