DBA: Dynamic Multi-Armed Bandit Algorithm

Sadegh Nobari
Rakuten, Inc.
sadegh@sqnco.com

Abstract

We introduce Dynamic Bandit Algorithm (DBA), a practical solution to improve the shortcoming of the pervasively employed reinforcement learning algorithm called Multi-Arm Bandit, aka Bandit. Bandit makes real-time decisions based on the prior observations. However, Bandit is heavily biased to the priors that it cannot quickly adapt itself to a trend that is interchanging. As a result, Bandit cannot, quickly enough, make profitable decisions when the trend is changing. Unlike Bandit, DBA focuses on quickly adapting itself to detect these trends early enough. Furthermore, DBA remains as almost as light as Bandit in terms of computations. Therefore, DBA can be easily deployed in production as a light process similar to The Bandit. We demonstrate how critical and beneficial is the main focus of DBA, i.e. the ability to quickly finding the most profitable option in real-time, over its state-of-the-art competitors. Our experiments are augmented with a visualization mechanism that explains the profitability of the decisions made by each algorithm in each step by animations. Finally we observe that DBA can substantially outperform the original Bandit by close to 3 times for a set Key Performance Indicator (KPI) in a case of having 3 arms.

Introduction

The multi-armed bandit problem (aka K- or N-armed bandit problem) maximizes the expected gain of resource allocation (arms) between competing choices in a setting that a fixed limited set of resources is available while the properties of the choices are partially known at the time of allocation.

Since the introduction of this problem as multi-armed bandit (Katehakis and Veinott Jr), several attempts have been done in expanding and improving solutions as well as the problem definition. In this paper we further generalize the problem definition to target more applications and amend the shortcomings of the original definition.

In practice choices that we are selecting can change their performance by time, we call it Dynamic Arms. For instance, imagine we are selecting the most attractive item for sale and our choices are pumpkin and moon cake. Given the time we make this selection the attractiveness of the items can drastically varies. In this example if we are close to Halloween or Chinese new year one item overcomes the other.

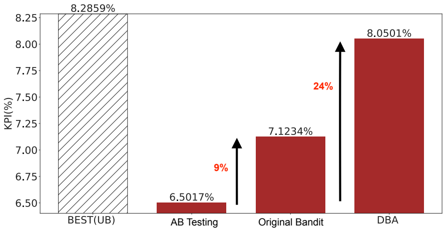


Figure 1: Overall performance of all the algorithms

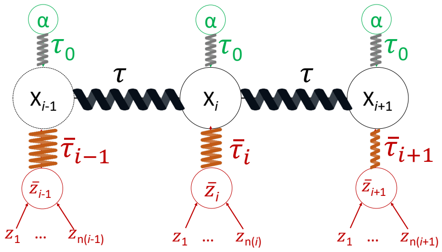


Figure 2: State diagram of DBA

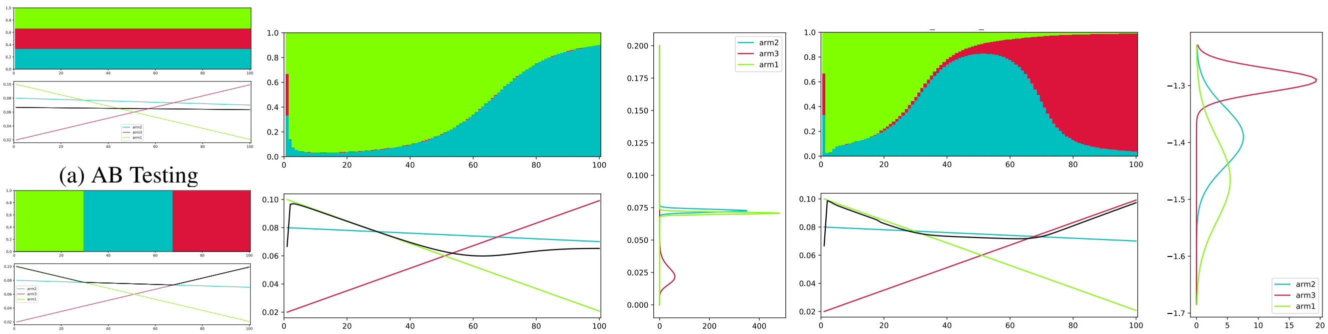
We propose Dynamic Bandit Algorithm (DBA) as a solution to the above generalization, i.e. Dynamic Arms. DBA finds applications in every situation that we want to pick the best candidate, given our set KPI, from multiple candidates. Candidates can be Layouts of website, Coupons, Products, Emails, Advertisements, Algorithms and etc.

Related works

DBA is a novel generalization the original Bandit (Katehakis and Veinott Jr). Below are other state-of-the-art variations:

Cascading Bandits imitates the popular model of user behaviour in web search (Craswell et al.). Given a list of K links, the user examines this list with an order and selects one. Cascade Bandit redefines the reward function to take into account this order (Kveton et al. ; Zong et al.).

Contextual Bandits: The bandit algorithm selects a candidate without using any contextual information. For example, if we know that a user is a kid or adult we can select or avoid selecting some candidates. The contextual bandit extends the original model by making the decision conditional on the state of the environment (Joseph et al.).



(a) AB Testing

(b) Upper Bound (BEST)

(c) Original Bandit

(d) DBA

Figure 3: Running all the algorithms in case of 3 arms with linear performance.

Collaborative Filtering (CF) Bandits: Given a training data, classical CF and content-based filtering methods are learning a *static* recommendation model. To overcome this static model, CF Bandit takes into account the collaborative effects of the users' interactions. The resulting algorithm thus takes advantage of preference patterns in the data in a way akin to the CF methods (Li, Karatzoglou, and Gentile).

DBA

Assumes each candidate has a Dynamic performance over the time. Hence, given the past and current observations for each item, DBA relies on a multivariate normal distribution that is a normal distribution such that its mean follows another normal distribution. Internally, DBA follows a wiener process (aka standard Brownian motion process or Lévy processes) with a Gaussian increments between the states with independent increments. Figure 2 illustrates the state diagram of DBA. In this figure past, present and future states are indicated as X_{i-1}, X_i and X_{i+1} with a normal distribution having a precision of τ between the states. Similarly, Observations are indicated as Z_{i-1}, Z_i and Z_{i+1} that are following a normal distribution with precisions of $\bar{\tau}_{i-1}$, $\bar{\tau}_i$ and $\bar{\tau}_{i+1}$, accordingly. Finally we apply a set of prior normal distributions $\mathcal{N}(\alpha, \frac{1}{\sqrt{\tau}})$ for each state to avoid over-fitting.

Experiment and Result

We conducted several experiments on synthetic (Nobari et al.) and real datasets. Synthetic datasets are carefully generated arms with interchanging trends such that each arm has a chance to be the best performing arm. Furthermore, multiple arms of linear and non-linear (polynomial) performance are examined. Our experiments include AB testing (equal chance for selecting each arm), Upper Bound (given the future performance of each arm, selects the best arm), Original Bandit and DBA. In every selection we consider distributing the selection over 500 participants for 100 trials[1].

Figure 3 illustrates the result of running 100 trials over 3 arms, i.e. arm1 (green), arm2 (red) and arm3 (blue), all with a linear performance. In this Figure a bar chart contains the proportion of the selection and the bottom chart shows the performance of each arm as well as a black line that indicates the obtained performance after the selection by each algorithm in every trial. For Bandit (c) and DBA (d) the right chart shows the predicted final distribution for each arm.

Figure 1 shows the performance of all the algorithms in which DBA outperforms the AB test by 24% lift of the set KPI, i.e. 3 times than the 9% lift of the Original Bandit. This proves the importance of learning the interchanging trends and being able to quickly switch to the best performing arm.

Conclusion and Future Work

This paper redefines the multi-arm bandit definition by assuming an interchanging trend for the performance of each arm. Given this generalization a new algorithm, DBA, is proposed. Finally an exhaustive comparison of the state-of-the-art algorithms is demonstrated through animations that shows the effect of each decision in every step and how this decision is made based on the distributions that are predicted. In future, we are planning to expand DBA in various directions, namely by augmenting contextual knowledge, cascading models and in collaborative filtering.

References

Craswell, N.; Zoeter, O.; Taylor, M.; and Ramsey, B. An experimental comparison of click position-bias models. In *WSDM'08*.

Joseph, M.; Kearns, M.; Morgenstern, J. H.; and Roth, A. Fairness in learning: Classic and contextual bandits. In *NIPS'16*.

Katehakis, M. N., and Veinott Jr, A. F. The multi-armed bandit problem: decomposition and computation. In *MOOR'87*.

Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. Cascading bandits: Learning to rank in the cascade model. In *ICML'15*.

Li, S.; Karatzoglou, A.; and Gentile, C. Collaborative filtering bandits. In *SIGIR'16*.

Nobari, S.; Lu, X.; Karras, P.; and Bressan, S. Fast random graph generation. In *EDBT'11*.

Zong, S.; Ni, H.; Sung, K.; Ke, N. R.; Wen, Z.; and Kveton, B. Cascading bandits for large-scale recommendation problems. In *UAI'16*.

[1] Full demonstration of DBA: https://nobari.github.io/DBA