AutoBandit: A Meta Bandit Online Learning System

Miao Xie[1] *, **Wotao Yin** and **Huan Xu**
Alibaba Group, Hangzhou, China
[1]0520shui@163.com,{wotao.yin, huan.xu}@alibaba-inc.com

Abstract

Recently online multi-armed bandit (MAB) is growing rapidly, as novel problem settings and algorithms motivated by various practical applications are being studied, building on the top of the classic bandit problem. However, identifying the best bandit algorithm from many potential candidates for a given application is not only time-consuming but also relying on human expertise, which hinders the practicality of MAB. To alleviate this problem, this paper outlines an intelligent system called AUTOBANDIT, equipped with many out-of-the-box MAB algorithms, for automatically and adaptively choosing the best with suitable hyper parameters online. It is effective to help a growing application for continuously maximizing cumulative rewards of its whole life-cycle. With a flexible architecture and user-friendly web-based interfaces, it is very convenient for the user to integrate and monitor online bandits in a business system. At the time of publication, AUTOBANDIT has been deployed for various industrial applications.

1 Introduction & Motivation

In a sequential decision-making problem, an agent must learn to choose the best action out of several candidates (called arms) to play by balancing exploration and exploitation for receiving reward so that the cumulative rewards over time is maximized. This problem is formulated as the multi-armed bandit (MAB) problem, which is frequently encountered in various practical applications, from clinical trials to recommendation systems and even machine learning methods [Bouneffouf and Rish, 2019].

Although many algorithms for MAB have been introduced in recent decades, identifying an algorithm, which is best suitable for a given application, remains important and challenging. Existing systems like Vowpal Wabbit and SageMaker RL implement many practical bandit algorithms but lack an effective online strategy for adaptively identifying the best one from them. Solving this problem manually is very time-consuming and usually relies on human expertise, for the following reasons. Firstly, MAB algorithms are often associated with different assumptions, leading to various application scope. Stochastic bandit algorithms [Auer *et al.*, 2002a; Gopalan *et al.*, 2014], like UCB1 and Thompson sampling (TS) [Agrawal and Goyal, 2012], treat all arms independently and each associated with a fixed but unknown reward probability distribution. They can make an unbiased prediction with sufficient online data but they may need a long time to converge for a large number of arms. Contextual bandit algorithms [Chu *et al.*, 2011; Qin *et al.*, 2014; Ghosh *et al.*, 2017; Li *et al.*, 2016; Liu *et al.*, 2018] assume arms to share a linear parametric function of contexts for achieving a better convergence, but they may suffer from a large regret due to the bias of reward estimation with finite dimensional features. To avoid learning from the scratch, many algorithms [Shivaswamy and Joachims, 2012; Zhang *et al.*, 2019] incorporate historical data and cluster structures [Bouneffouf *et al.*, 2019] into contextual bandits. Motivated by various practical problems, many other kinds of bandit algorithms have been proposed recently, such as Non-Stationary Bandit [Zhou *et al.*, 2020b], Neural Bandit [Zhou *et al.*, 2020a], Adversarial Bandit [Bistritz *et al.*, 2019], Semi-parametric Bandit [Peng *et al.*, 2019], etc. Secondly, there are usually many hyper-parameters, which have strong effects on cumulative rewards and even the order of gap-dependent regret upper bound [Auer *et al.*, 2002a]. Thirdly, the best algorithm will vary as the time in the whole life-cycle of a given application. For example, the performance of different bandit algorithms are significant distinct for solving the problem

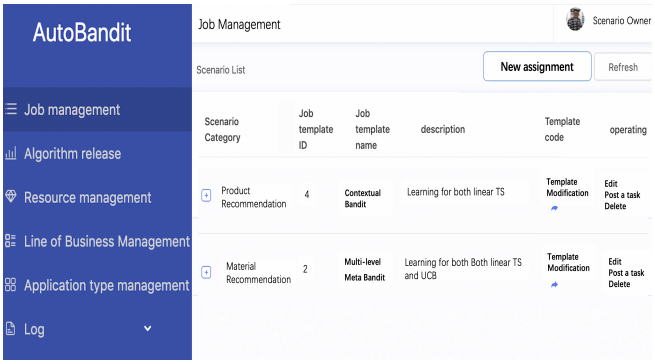


Figure 1: AUTOBANDIT Demo System.

*Contact Author

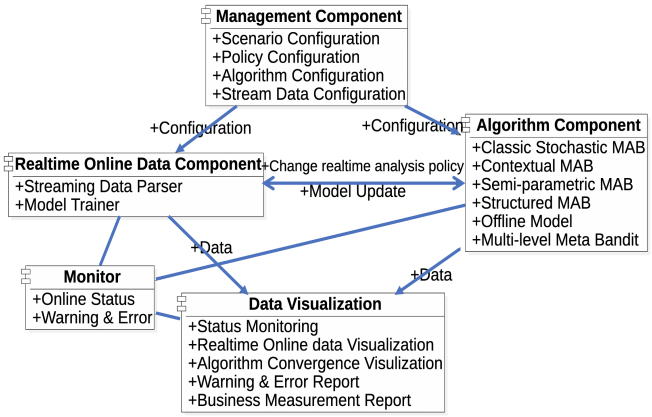


Figure 2: System Architecture.

of different stages, from the cold-start (with scarce data) to the mature stage (with rich data) in a recommendation system. Finally, implementing a bandit algorithm in an industrial system needs a long real-time calculation chain from online decision process to online data analysis.

In this paper, we design a novel multi-level meta bandit strategy for adaptively choosing the best candidate algorithm with its suitable hyper-parameters in an online fashion. Based on it, we propose a meta bandit online learning system called AUTOBANDIT [i], equipped with many out-of-the-box MAB algorithms and friendly user interfaces, which can be easily integrated by any online system in only four steps. We have evaluated this system in various industrial applications including product recommendation in the app Idlefish, cover picture selection for videos in the app Youku and advertisement allocation strategy selection in the app "1688" . Online A/B testing results show its superiority and effectiveness, especially for finding the best bandit algorithm over the time.

2 The AUTOBANDIT System

2.1 System Architecture

Figure 2 shows the system architecture of AUTOBANDIT. It contains the following important components. Online decision algorithm component implements many kinds of bandit algorithms, including classic stochastic MABs [Auer *et al.*, 2002a; Gopalan *et al.*, 2014; Thompson, 1933], contextual bandits [Chu *et al.*, 2011; Li *et al.*, 2016], semi-parametric bandits [Peng *et al.*, 2019], prior knowledge based bandits [Bouneffouf *et al.*, 2019; Yue *et al.*, 2012], bandits with history data [Shivaswamy and Joachims, 2012; Bouneffouf *et al.*, 2019; Zhang *et al.*, 2019] and Neural bandits [Zhou *et al.*, 2020a]. All of them follow the same calculation pipeline from loading arms information to making online decision. It supplies a web-service function for being called by other industrial system to get online results. A real-time online data component is deployed on Flink [Carbone *et al.*, 2015], for obtaining action reward and learning model by streaming data analysis. Management component with web-based user interfaces is designed for custom configuration. The components store and share data in HDFS on the cloud.

The system can be integrated with an online industrial system in only four steps, which enable the users to leverage bandit algorithms at a very low cost: (1) uploading arm information to database. (2) selecting a model learner and implementing a data source for parsing rewards in Flink. (3) making system configurations. (4) invoking the web-service function by HTTP commands. Data visualization and monitors make this system easier to be used and operated. Once there are something wrong, emergency information will be delivered as intent messages to operators. This feature is significant for online algorithms because their behavior will evolve dramatically. Besides, the real-time status of the system like algorithm convergence, business metrics (e.g. Click-Through-Rate, CTR) and other important information can be visualized easily.

It is very interesting to state that the system provides an "auto-pilot" mode, where the system will automatically deploy several potential candidate bandit algorithms and adaptively choosing the best policy with suitable hyper-parameters from them online using the Multi-level Meta Bandit Strategy.

2.2 Multi-level Meta Bandit Strategy

The problem of identifying the best bandit algorithm adaptively online for maximizing the cumulated rewards for a given scenario can be divided into two online decision subproblems: algorithm selection and hyper-parameter optimization. Firstly, we formulate the algorithm selection problem by the bandit problem in a non-stationary environment [Auer *et al.*, 2002b], where we treat candidate algorithms as arms. We call it as Algorithm Machine. During the online decision, these candidate algorithms will continuously learn from online streaming data for better performance with different convergence rates, so the reward distribution behind them is not stationary. Secondly, the hyper-parameter optimization problem is modelled by a series of rule-based constrained bandit problems. Specifically, as for discrete hyper-parameters, each value of the discrete space can be seen as an independent arm. In the case of continuous variables, continuum armed bandit algorithms [Auer *et al.*, 2007] can be used for selecting in a continuous space. We call the problem of hyper-parameter optimization as Parameter Machines. To reduce the search space of hyper-parameters, we introduce some rule-based constraints based on expert experience for pruning obvious unsuitable parameter settings. The system will converge if all above machines have been converged.

To improve the convergence rate, we design a computation strategy called Multi-level Meta Bandit Strategy, which contains two interleaving computation processes: cascading decision process (in decision direction) and reward back propagation process (in learning direction). Figure 3 illustrates its computation graph. Its structure looks like a tree. The root is a meta controller, which manages the whole decision and learning process as follows. The direction from the left to right shows the process of the cascading decision. At each decision time, the meta controller first uses Algorithm Machine to select the best algorithm. Then, for the selected algorithm, meta controller will invoke several Parameter Machines to de-

[i]Demo Video Link: https://youtu.be/S2dtafSulak

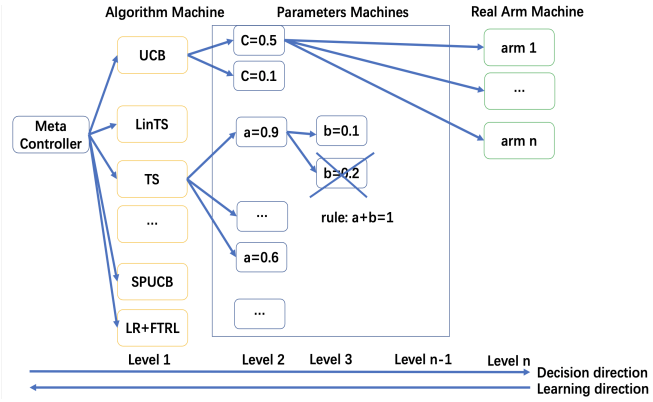
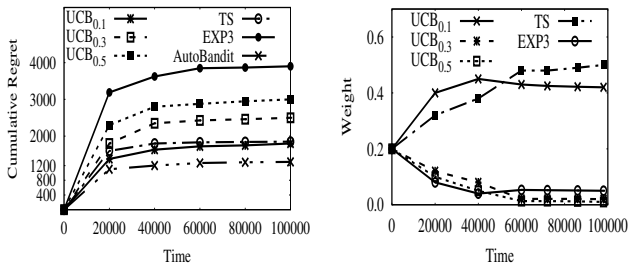


Figure 3: Computation Graph of Multi-level Meta Bandit



(a) (b)

Figure 4: Experimental average results on synthetic dataset

cide all significant hyper-parameters one by one sequentially. Each hyper-parameter associated with a Parameter Machine would be seen as a level of the tree. Certainly, in this step, rule-based constraints should be used to prune candidates if have. After all hyper-parameters are selected, meta controller will use the decided bandit algorithm for solving the given real bandit problem, that is Real Arm Machine, which can be seen as leaves. Thus different arms of Algorithm Machine may have different levelled decision paths. In the learning direction, the obtained reward propagates in the direction from leaves to root for all paths parallelly to make all bandit algorithms learn from it. This manner can work because all candidate algorithms with hyper-parameters are targeting at solving the same given real problem. By this way, the learning process will be accelerated largely. Experimental results on both synthetic and real datasets show that the convergence rate is even faster than individual candidate algorithms and a better cumulative regret can be achieved.

In practice, once the accumulated online data is relatively sufficient, it is possible that some complicated training models like Logistic Regression/Neural Network/Random Forest with Follow-The-Regular-Leader would be better than bandits for a period. For this reason, we recommend to treat these algorithm as arms in Algorithm Machine to explore these models together with candidate bandit algorithms in the designed strategy.

3 The Prototype System

According to the designed system architecture with the multi-level meta bandit strategy, we deployed AUTOBANDIT system in the private real business cloud environment in Alibaba. Figure 1 shows an example webpage. The system can offer bandit algorithm services for two types of users, bandit developers for implementing new algorithms and scenario owners for using these algorithms in an application scenario. Firstly, a bandit developer can develop new bandit algorithms and design rules for a meta bandit by implementing abstract functions following the same pipeline. They are able to monitor the running status of their algorithms in all scenarios. Secondly, the system provides user-friendly web interfaces for scenario owners to select scenario configurations and algorithm settings. They get the service online if and only if configurations are ready without any code effort. If the multi-level meta bandit strategy is enabled, the evolving process of identifying the best algorithm can be observed.

To show the effectiveness of AUTOBANDIT, we conducted experiments on a synthetic data of 300 arms, which were generated following the assumption of stochastic multi-armed bandit. Three classic bandit algorithms were selected as baselines. We considered the hyper-parameter of UCB1, $c \in \{0.1, 0.3, 0.5\}$. We fixed $\gamma = 0.007$ for EXP3. In the experiment, we used EXP4 for solving Algorithm Machine. As shown in Figure 4(a), we can see UCB1 with $c = 0.1$ has the best cumulative regret compared to other baselines but the gap with TS is gradually narrowed down. Figure 4(b) plots the weights of these algorithms in AUTOBANDIT. It is clear that the weight of TS will increase and eventually exceed that of UCB1 with $c = 0.1$. It means that AUTOBANDIT has successfully identified $\text{UCB}_{0.1}$ as the best algorithm early and switched from it to TS eventually. In this experiment, AUTOBANDIT achieves the best cumulative regret compared to others. Although AUTOBANDIT cannot always beat the best candidate in all cases, it is able to approach them.

Currently, many applications at Alibaba have integrated AutoBandit and obtained good performance. Using online A/B testing for comparing AUTOBANDIT to Random or manual policy, we can always get significant improvement for business metrics automatically. For example, Youku used it for selecting cover-pictures for videos and achieved 5%-40% improvement in CTR. Idlefish improved 5%-20% CTR under the condition of achieving the same Click Value Rate (CVR) for product recommendation. The advertisement system of the app "1688" obtained 3%-10% improvement in Cost Per View of advertisement for advertisement allocation strategy. In these cases, we can also clearly observe the ability of adaptively identifying the best algorithm over the time.

4 Conclusions

Equipped with a novel multi-level meta bandit strategy, AUTOBANDIT can adaptively choose the best candidate algorithm with its suitable hyper-parameters online. Experimental results of various industrial applications show its superior performance for helping an application from its birth to maturity. We believe that there are huge potential market opportunities because thousands of new apps with potential application scenarios are uploaded to app markets every day.

References

[Agrawal and Goyal, 2012] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

[Auer *et al.*, 2002a] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-Time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2-3):235–256, 2002.

[Auer *et al.*, 2002b] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[Auer *et al.*, 2007] Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *International Conference on Computational Learning Theory*, pages 454–468. Springer, 2007.

[Bistritz *et al.*, 2019] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Exp3 learning in adversarial bandits with delayed feedback. *Advances in neural information processing systems*, 2019.

[Bouneffouf and Rish, 2019] Djallel Bouneffouf and Irina Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*, 2019.

[Bouneffouf *et al.*, 2019] Djallel Bouneffouf, Srinivasan Parthasarathy, Horst Samulowitz, and Martin Wistuba. Optimal exploitation of clustering and history information in multi-armed bandit. *IJCAI. IJCAI*, 2019.

[Carbone *et al.*, 2015] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4), 2015.

[Chu *et al.*, 2011] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[Ghosh *et al.*, 2017] Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3761–3767, 2017.

[Gopalan *et al.*, 2014] Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *International Conference on Machine Learning*, pages 100–108, 2014.

[Li *et al.*, 2016] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1245–1253. JMLR.org, 2016.

[Liu *et al.*, 2018] Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. Transferable contextual bandit for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[Peng *et al.*, 2019] Yi Peng, Miao Xie, Jiahao Liu, Xuying Meng, Nan Li, Cheng Yang, Tao Yao, and Rong Jin. A practical semi-parametric contextual bandit. *IJCAI*, 2019.

[Qin *et al.*, 2014] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In Mohammed Javeed Zaki, Zoran Obradovic, Pang-Ning Tan, Arindam Banerjee, Chandrika Kamath, and Srinivasan Parthasarathy, editors, *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 461–469. SIAM, 2014.

[Shivaswamy and Joachims, 2012] Pannagadatta Shivaswamy and Thorsten Joachims. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*, pages 1046–1054, 2012.

[Thompson, 1933] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[Yue *et al.*, 2012] Yisong Yue, Sue Ann Hong, and Carlos Guestrin. Hierarchical exploration for accelerating contextual bandits. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 979–986, 2012.

[Zhang *et al.*, 2019] Chicheng Zhang, Alekh Agarwal, Hal Daumé Iii, John Langford, and Sahand Negahban. Warm-starting contextual bandits: Robustly combining supervised and bandit feedback. In *International Conference on Machine Learning*, pages 7335–7344. PMLR, 2019.

[Zhou *et al.*, 2020a] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[Zhou *et al.*, 2020b] Huozhi Zhou, Lingda Wang, Lav Varshney, and Ee-Peng Lim. A near-optimal change-detection based algorithm for piecewise-stationary combinatorial semi-bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6933–6940, 2020.