Hyperparameter Tuning for the Contextual Bandit

Djallel Bouneffouf1, Emmanuelle Claeys2

1IBM Thomas J. Watson Research Center, Yorktown Heights, NY USA
2Strasbourg University, CNRS, IRMA, 7 Rue Rene Descartes, 67000 Strasbourg, France
{Djallel.bouneffouf}@ibm.com
claeys@unistra.fr

Abstract
We study here the problem of learning the exploration-exploitation trade-off in the contextual bandit problem with linear reward function setting. In the traditional algorithms that solve the contextual bandit problem, the exploration is a parameter that is tuned by the user. However, our proposed algorithm learn to choose the right exploration parameters in an online manner based on the observed context, and the immediate reward received for the chosen action. We have presented here two algorithms that uses a bandit to find the optimal exploration of the contextual bandit algorithm, which we hope is the first step toward the automation of the multi-armed bandit algorithm.

Introduction
In sequential decision problems such as clinical trials (Villar, Bowden, and Wason 2015) or recommender system (Mary, Gaudel, and Preux 2015), a decision-making algorithm must select among several actions at each given time-point. Each of these actions is associated with side information, or context (e.g., a user’s profile), and the reward feedback is limited to the chosen option. For example, in the clinical trials (Villar, Bowden, and Wason 2015), the actions correspond the treatment options being compared, the context is the patient’s medical record (e.g. health condition, family history, etc.) and the reward represents the outcome (successful or not) of the proposed treatment. In this setting, we are looking for a good trade-off between the exploration of the new drug and the exploitation of the known drug.

This inherent exploration-exploitation trade-off exists in many sequential decision problems, and is traditionally modeled as multi-armed bandit (MAB) problem, stated as follows: there are K “arms” (possible actions), each associated with a fixed but unknown reward probability distribution (Lai and Robbins 1985). At each step, an agent plays an arm (chooses an action) and receives a reward. This reward is drawn according to the selected arm’s law and is independent of the previous actions.

A particularly useful version of MAB is the contextual multi-armed bandit problem. In this problem, at each iteration, before choosing an arm, the agent observes a D-dimensional feature vector, or context, associated with each arm. The learner uses these contexts, along with the rewards of the arms played in the past, to choose which arm to play in the current iteration. Over time, the learner’s aim is to collect enough information about the relationship between the context vectors and rewards, so that it can predict the next best arm to play by looking at the corresponding contexts (feature vectors) (Agrawal and Goyal 2013). One smart solution for the contextual bandit is the LINUCB algorithm, which is based on online ridge regression, and takes the concept of upper-confidence bound (Li et al. 2010) to strategically balance between exploration and exploitation. The α parameter essentially controls exploration/exploitation. The problem is that it is difficult to decide in advance the optimal value of α. We introduce in this paper two algorithms, named “OPLINUCB” and “DOPLINUCB”, that computes the optimal value of α in both stationer and switching environment by adaptively balancing exp/exp according to the context.

The main contributions of this paper include proposing two new algorithms, for both stationary and non-stationary settings, which extend the existing bandit algorithms to the new setting, and (3) evaluating the algorithms empirically on a variety of datasets.

RELATED WORK
The multi-armed bandit problem is a model of exploration versus exploitation trade-off, where a player gets to pick within a finite set of decisions the one maximizing the cumulative reward. This problem has been extensively studied. Optimal solutions have been provided using a stochastic formulation (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002).

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
to the arm’s performance. In LINUCB (Li et al. 2010; Chu et al. 2011) and in Contextual Thompson Sampling (CTS) (Agrawal and Goyal 2013) and neural bandit (Allesiardo, Féraud, and Bouneffouf 2014), the authors assume a linear dependency between the expected reward of an action and its context; the representation space is modeled using a set of linear predictors. However, the exploration magnitude on these algorithms need to be given by the user. Authors in (Sharaf and Daumé III 2019), addresses the exploration trade-off by learning a good exploration strategy for offline tasks based on synthetic data, on which it can simulate the contextual bandit setting. Based on these simulations, the proposed algorithm uses an imitation learning strategy to learn a good exploration policy that can then be applied to true contextual bandit tasks at test time. The authors compare the algorithm to seven strong baseline algorithms that exploit current estimates without any exploration if there is sufficient randomness in the observed contexts. Absent this condition, we show that a greedy algorithm can be rate-optimal with positive probability.

As we can see, none of these previously proposed approaches involves learning dynamically the exploration trade-off in the contextual bandit setting, which is the main focus of this work.

Key Notion

This section focuses on introducing the key notions used in this paper.

The Contextual Bandit Problem.

Following (Langford and Zhang 2008), this problem is defined as follows. At each time point (iteration) $t \in \{1, ..., T\}$, a player is presented with a context (feature vector) $c(t) \in \mathbb{R}^K$ before choosing an arm $i \in A = \{1, ..., K\}$. We will denote by $C = \{C_1, ..., C_N\}$ the set of features (variables) defining the context. Let $r = (r_1(t), ..., r_K(t))$ denote a reward vector, where $r_i(t) \in [0, 1]$ is a reward at time $t$ associated with the arm $i \in A$. Herein, we will primarily focus on the Bernoulli bandit with binary reward, i.e. $r_i(t) \in \{0, 1\}$. Let $\pi : C \rightarrow A$ denote a policy. Also, $D_{c,r}$ denotes a joint distribution $(c,r)$. We will assume that the expected reward is a linear function of the context, i.e. $E[r_k(t)|c(t)] = \mu_k^T c(t)$, where $\mu_k$ is an unknown weight vector (to be learned from the data) associated with the arm $k$.

Thompson Sampling (TS).

The TS (Thompson 1933a), also known as Basyesian posterior sampling, is a classical approach to multi-arm bandit problem, where the reward $r_k(t)$ for choosing an arm $k$ at time $t$ is assumed to follow a distribution $Pr(r_k(t) = \mu_k)$ with the parameter $\mu_k$. Given a prior $Pr(\mu_k)$ on these parameters, their posterior distribution is given by the Bayes rule, $Pr(\mu_k|r_k) \propto Pr(r_k|\mu_k)Pr(\mu_k)$. A particular case of the Thomson Sampling approach assumes a Bernoulli bandit problem, with rewards being 0 or 1, and the parameters following the Beta prior. TS initially assumes arm $k$ to have prior Beta$(1,1)$ on $\mu_k$ (the probability of success). At time $t$, having observed $S_k(t)$ successes (reward = 1) and $F_k(t)$ failures (reward = 0), the algorithm updates the distribution on $\mu_k$ as Beta$(S_k(t), F_k(t))$. The algorithm then generates independent samples $\theta_k(t)$ from these posterior distributions of the $\mu_k$, and selects the arm with the largest sample value. For more details, see, for example, (Agrawal and Goyal 2012).

Regression Tree

In total generality unsupervised learning is an ill posed problem. Nevertheless in some situations it is possible to back the clustering problem by a supervised one. In our setting we can use the reward estimation as a supervision for the group creation. Thus we just need a supervised learning technique which explicitly creates groups that we could reuse. See (Kotsiantis 2007) for a survey on such techniques. Among all existing approaches, the use of a tree built by recursive partitioning is a popular approach. It allows to estimate different means under specifics explanatory variables and have an interpretation as a regression analysis (Strasser and Weber 1999). Moreover some efficient implementations as “CART” (Breiman et al. 1984) and “C4.5” are available.

For a regression models describing the conditional distribution of a response variable $X$ given the status of $N_F$ covariates by means of tree-structured recursive partitioning. The $d$-dimensional covariate vector $f_X = (f_X^1, ..., f_X^N)$ is taken from a sample space $\mathcal{F}$. Both response variables and covariates may be measured at arbitrary scale. If we assume that the conditional distribution $D(X|f_X)$ of the respond $X$ variable given the covariate $f_X$ depend of a function $z$ of the covariate:

$$D(X|f_X) = D(X|f_X^1, ..., f_X^N) = D(X|z(f_X^1, ..., f_X^N))$$

With $m \leq N_F$. Let’s $\mathcal{L}$ be a learning sample for training the model of regression relationship on a n random sample of independent and identically distributed observation, possibly with some covariates observation is missing:

$$\mathcal{L}_n = \{X_i, f_X^1, ..., f_X^N; i = 1, ..., n\}$$

CTree (conditional inference Tree) is an algorithm derived from C.A.R.T : CTREE proposed by (Hothorn, Hornik, and Zeileis) (2006) which is a non parametric class of regression trees embedding tree-structured regression models into a well-defined theory of conditional...
We describe here two algorithm that learn the exploration of the contextual bandit algorithm.

**OPLINUCB**

The proposed algorithm is named "OPLINUCB" for Non-Parametric LINUCB. This algorithm has to solve two levels multi-armed bandit problems. The first level is the classical multi-armed bandit problem applied to find the parameters of the algorithm. The second level problem is a contextual bandit problem that use the parameters find in the first level to find the optimal arm to play. Let \( n_{\alpha_i}(t) \) be the number of times the \( i \)-th exploration value has been selected so far, let \( r_{\alpha_i}^j(t) \) be the cumulative reward associated with the exploration value \( i \), and let \( r_k(t) \) be the reward associated with the arm \( k \) at time \( t \). The algorithm takes as an input the candidate values for \( \alpha \), as well as the initial values of the Beta distribution parameters in TS. At each iteration \( t \), we update the values of those parameters, \( S_{\alpha_i} \) and \( F_{\alpha_i} \), to represent the current total number of successes and failures, respectively, and then sample the "probability of success" parameter \( \theta_{\alpha_i} \) from the corresponding Beta distribution, separately for each exploration value \( \alpha_i \) to estimate \( \mu_i \), which is the mean reward conditioned to the use of the variable \( i : \mu^i = \frac{1}{K} \sum_k E[r_k, 1\{i \in C^k\}] \) (step 7).

The pseudo-code of OPLINUCB is sketched in Algorithm 2.

**DOPLINUCB**

The proposed algorithm is named "DOPLINUCB" for Dynamic Non-Parametric LINUCB. This algorithm has to solve two levels contextual bandit problems. The first level where the algorithm is using the context to decide on the ex-

---

### Algorithm 1: CTree

1: **Require:** A learning sample of data \( \mathcal{L}_n \)
2: for \( w=w_1, ..., w_N \) do
3: For case weights \( w \) test the global null hypothesis of independence between \( m \) covariates and the response. Stop if hypothesis cannot be rejected. Otherwise select the covariate \( j \) with the strongest association to \( X \).
4: Choose a set \( b^* \subset B^*_j \) in order to split \( B^*_j \) into two disjoint sets: \( b^* \) and \( b^c \). Change the case weights for determine the two subgroups (left and right).
5: Modify case weight of left and right subset respectively.
6: end for

---

### Algorithm 2: OPLINUCB

**Input:** \((\alpha_1, ..., \alpha_N)\): candidate values for \( \alpha \), \( A_t \), the initial values \( S_{\alpha_i}(0) \) and \( F_{\alpha_i}(0) \) of the Beta distribution parameters.

for \( t=1 \) to \( T \) do

Observe the context \( x_t \in \mathbb{R}^d \) for all \( \alpha_i \) do

\[ S_{\alpha_i}(t) = S_{\alpha_i}(0) + r_{\alpha_i}^j(t-1) \]
\[ F_{\alpha_i}(t) = F_{\alpha_i}(0) + n_{\alpha_i}(t-1) - r_{\alpha_i}^j(t-1) \]

Sample \( \theta_{\alpha_i} \) from \( \text{Beta}(S_{\alpha_i}(t), F_{\alpha_i}(t)) \) distribution end for

Choose exploration \( \alpha_t = \arg\max_{\alpha \in A_t} p_t, \alpha \) for all \( \alpha \in A_t \) do

if \( \alpha \) is new then

\[ A_{\alpha} \leftarrow I_d \] (d-dimensional identity matrix)
\[ b_{\alpha} \leftarrow 0_{d \times 1} \] (d-dimensional zero vector)
end if

\[ \Theta_{\alpha} \leftarrow \Theta_{\alpha}^{-1} \ast \Theta_{\alpha} \]
\[ p_{t, \alpha} \leftarrow \Theta_{\alpha}^{-\top} x_{t, \alpha} + \alpha \sqrt{x_{t, \alpha}^\top \Theta_{\alpha}^{-1} x_{t, \alpha}} \]
end for

Choose arm \( \alpha_t = \arg\max_{\alpha \in A_t} p_{t, \alpha} \) with ties broken arbitrarily, and observe a real-valued payoff \( r_t \)

\[ A_{\alpha_t} \leftarrow A_{\alpha_t} + x_{t, \alpha_t} x_{t, \alpha_t}^\top \]
\[ b_{\alpha_t} \leftarrow b_{\alpha_t} + r_t x_{t, \alpha_t} \]
end for

---

In Algorithm 2, \( A_t \) is the set of arms at iteration \( t \), where \( x_{a, t} \) is the feature vector of arms \( a \) with \( d \)-dimension, \( \Theta_{\alpha} \) is the unknown coefficient vector of the feature \( x_{a, t} \), \( \alpha \) is a constant and \( A_{\alpha} = D_{\alpha}^\top D_{\alpha} + I_d \). \( D_{\alpha} \) is a design matrix of \( m \times d \) at trial \( t \), whose rows correspond to \( m \) training inputs (e.g., \( m \) contexts that are observed previously for arm \( a \)), and \( b_{\alpha} \in \mathbb{R}^m \) is the corresponding response vector (e.g., the corresponding \( m \) click/no-click user feedback). Applying ridge regression to the training data \((D_{\alpha}, c_{\alpha})\) gives an estimate of the coefficients: \( \Theta_{\alpha} = (D_{\alpha}^\top D_{\alpha} + I_d)^{-1}D_{\alpha} c_{\alpha} \), where \( I_d \) is the \( d \times d \) identity matrix and \( c_{\alpha} \) are independent conditioned by corresponding rows in \( D_{\alpha} \).
We show in the following two scenarios, one where we evaluate the algorithm described in Algorithm 1. The second level problem is also contextual bandit problem, where the algorithm uses the context and the exploration provided by the first level to find the optimal arm to play. The pseudo-code of DOPLINUCB is sketched in Algorithm 3.

### Experimentation

We show in the following two scenarios, one where we evaluate the algorithms in stationary environments, and the second with a non-stationary environment.

### Stationary Environment

In this scenario we consider the case of two Bernoulli arms, our dataset is Adult dataset (Kohavi and Becker 1994). Reward is a prediction task to determine whether a person makes over 50K a year. Each person is defined by some categorical and continuous information (age, work class, etc). We fix the interval of possible $\alpha$ in $[0.01, 1]$ with step of 0.01. Mean, median, min and max of empirical average cumulative regret of each LinUCB with different $\alpha$ are provided in Table 1. Rewards function of each arm are stationary. We ran LinUCB with a hundred different values of $\alpha$ and compare empirical average cumulative regret with DOPLINUCB and OPLINUCB. Without train set, it’s not possible to use DOPLINUCB. Nevertheless, OPLINUCB choose an $\alpha$ in 100 value of alpha. DOPLINUCB and OPLINUCB can’t beats the best $\alpha$ value (min on table ) but OPLINUCB beats every time the median/mean.

### Non-Stationary Environment

In this experiment, we consider a challenging setting: reward function of each arm changes at a fixed number of iteration. We keep a training-set of 5000 items. Table 2 shows cumulative regrets when reward function of each arm changes at 100/1000/10000 iteration.

We can observe that in this scenario the proposed algorithm outperform beats the best $\alpha$ value (min on table). This is explainable by the fact that the proposed algorithm is learning a dependency between the context and the right exploration value, which is not the case of the min approach. This results show that when we are facing a non-stationary environment, having a dynamic exploration exploitation trade-off is useful, to learn this context/exploration dependency, which is beneficial for the contextual bandit algorithm.

### Conclusion

We have studied the problem of learning the exploration exploitation trade-off in the contextual bandit problem with linear reward function setting. In the traditional algorithms that solve the contextual bandit problem, the exploration is a parameter that is tuned by the user. However, our proposed algorithm learn to choose the right exploration parameters in an online manner based on the observed context, and the immediate reward received for the chosen action. We have presented here two algorithms that uses a bandit to find the optimal exploration of the contextual bandit algorithm. The Evaluation showed that the two proposed algorithms gives better results then which we hope is the first step toward an automated multi-armed bandit algorithm.

### References

[Agrawal and Goyal 2012] Agrawal, S., and Goyal, N. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In COLT 2012 - The 25th Annual Conference on
R. E. 2010. A contextual-bandit approach to personalized news article recommendation. *CoRR*.

[Lin et al. 2018a] Lin, B.; Bouneffouf, D.; Cecchi, G. A.; and Rish, I. 2018a. Contextual bandit with adaptive feature extraction. In 2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018, 937–944.

[Lin et al. 2018b] Lin, B.; Bouneffouf, D.; Cecchi, G. A.; and Rish, I. 2018b. Contextual bandit with adaptive feature extraction. In Tong, H.; Li, Z. J.; Zhu, F.; and Yu, J., eds., 2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018, 937–944. IEEE.

[Mary, Gaudel, and Preux 2015] Mary, J.; Gaudel, R.; and Preux, P. 2015. Bandits and recommender systems. In *Machine Learning, Optimization, and Big Data - First International Workshop, MOD 2015, Taormina, Sicily, Italy, July 21-23, 2015, Revised Selected Papers*, 325–336.

[Noothigattu et al. 2018] Noothigattu, R.; Bouneffouf, D.; Mattei, N.; Chandra, R.; Madan, P.; Varshney, K. R.; Campbell, M.; Singh, M.; and Rossi, F. 2018. Interpretable multi-objective reinforcement learning through policy orchestration. *CoRR* abs/1809.08343.

[Sharaf and Daumé III 2019] Sharaf, A., and Daumé III, H. 2019. Meta-learning for contextual bandit exploration. *arXiv preprint arXiv:1901.08159*.

[Shih 2004] Shih, Y.-S. 2004. A note on split selection bias in classification trees. *Computational Statistics & Data Analysis* 45(3):457–466.

[Strasser and Weber 1999] Strasser, H., and Weber, C. 1999. The asymptotic theory of permutation statistics. *Mathematical Methods of Statistics* 8:220–250.

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

[Thompson 1933b] Thompson, W. 1933b. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25:285–294.

[Villar, Bowden, and Wason 2015] Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics* 30(2):199.