Learning to Optimize under Non-Stationarity

Wang Chi Cheung
ISEM, NUS

David Simchi-Levi
IDSS, MIT

Ruihao Zhu
SDSC, MIT

Abstract

We introduce algorithms that achieve state-of-the-art *dynamic regret* bounds for non-stationary linear stochastic bandit setting. It captures natural applications such as dynamic pricing and ads allocation in a changing environment. We show how the difficulty posed by the non-stationarity can be overcome by a novel marriage between stochastic and adversarial bandits learning algorithms. Defining d, B_T, and T as the problem dimension, the *variation budget*, and the total time horizon, respectively, our main contributions are the tuned Sliding Window UCB (`SW-UCB`) algorithm with optimal $\widetilde{O}(d^{2/3}(B_T+1)^{1/3}T^{2/3})$ dynamic regret, and the tuning free bandit-over-bandit (`BOB`) framework built on top of the `SW-UCB` algorithm with best $\widetilde{O}(d^{2/3}(B_T+1)^{1/4}T^{3/4})$ dynamic regret.

1 Introduction

Multi-armed bandit (MAB) problems are online problems with partial feedback, when the learner is subject to uncertainty in his/her learning environment. Traditionally, most MAB problems are studied in the stochastic [6] and adversarial [7] environments. In the former, the model uncertainty is static and the partial feedback is corrupted by a mean zero random noise. The learner aims at estimating the latent static environment and converging to a static optimal decision. In the latter, the model is dynamically changed by an adversary. The learner strives to hedge against the changes, and compete favorably in comparison to certain benchmark policies.

While assuming a stochastic environment could be too simplistic in a changing world, sometimes the assumption of an adversarial environment could be too pessimistic. Recently, a stream of research works (see Related Works) focuses on MAB problems in a *drifting* environment, which is a hybrid of a stochastic and an adversarial environment. Although the environment can be dynamically and adversarially changed, the total change (quantified by a suitable metric) in a T step problem is upper bounded by B_T ($= \Theta(T^{\rho})$ for some $\rho \in (0,1)$), the *variation budget*. The feedback is corrupted by a mean zero random noise. The aim is to minimize the *dynamic regret*, which is the optimality gap compared to the sequence of (possibly dynamically changing) optimal decisions, by simultaneously estimating the current environment and hedging against future changes every time step. Most of the existing works for non-stationary bandits have focused on the the somewhat ideal case in which B_T is known. In practice, however, B_T is often not available ahead. Though some efforts have been made towards this direction [18, 21], how to design algorithms with low dynamic regret when B_T is unknown remains largely as a challenging problem.

In this paper, we design and analyze novel algorithms for the linear bandit problem in a drifting environment. Our main contributions are listed as follows.

- When the variation budget B_T is known, we characterize the lower bound of dynamic regret, and develop a tuned Sliding Window UCB (`SW-UCB`) algorithm with matched dynamic regret upper bound up to logarithmic factors.
- When B_T is unknown, we propose a novel Bandit-over-Bandit (`BOB`) framework that tunes `SW-UCB` adaptively. The application of `BOB` on `SW-UCB` algorithm achieves the best dependence on T compared to existing literature.

Related Works. MAB problems with stochastic and adversarial environments are extensively studied, as surveyed in [11, 20]. To model inter-dependence relationships among different arms, models for linear bandits in stochastic environments have been studied. In [5, 15, 23, 14, 1], UCB type algorithms for stochastic linear bandits were studied, and Abbasi-Yadkori et al. [1] possessed the state-of-art algorithm for the

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

problem. Thompson Sampling algorithms proposed in [24, 4, 2] are able to bypass the high computational complexities provided that one can efficiently sample from the posterior on the parameters and optimize the reward function accordingly. Unfortunately, achieving optimal regret bound via TS algorithms is possible only if the true prior over the reward vector is known.

Authors of [9, 8] considered the K-armed bandits in a drifting environment. They achieved the tight dynamic regret bound $\tilde{O}((KB_T)^{1/3}T^{2/3})$ when B_T is known. Wei et al. [25] provided refined regret bounds based on empirical variance estimation, assuming the knowledge of B_T. Subsequently, Karnin et al. [18] considered the setting without knowing B_T and $K = 2$, and achieved a dynamic regret bound of $\tilde{O}(B_T^{0.18}T^{0.82} + T^{0.77})$. In a recent work, [21] considered K-armed contextual bandits in drifting environments, and in particular demonstrated an improved bound $\tilde{O}(KB_T^{1/5}T^{4/5})$ for the K-armed bandit problem in drifting environments when B_T is not known, among other results. [19] considered a dynamic pricing problem in a drifting environment with linear demands. Assuming a known variation budget B_T, they proved an $\Omega(B_T^{1/3}T^{2/3})$ dynamic regret lower bound and proposed a matching algorithm. When B_T is not known, they designed an algorithm with $\tilde{O}(B_T T^{2/3})$ dynamic regret. In [10], a general problem of stochastic optimization under the known budgeted variation environment was studied. The authors presented various upper and lower bound in the full feedback settings. Finally, various online problems with full information feedback and drifting environments are studied in the literature [13, 17].

Apart from drifting environment, numerous research works consider the *switching environment*, where the time horizon is partitioned into at most S intervals, and it switches from one stochastic environment to another across different intervals. The partition is not known to the learner. Algorithms are designed for various bandits, assuming a known S [7, 16, 21], or assuming an unknown S [18, 21]. Notably, the Sliding Window UCB for the K-armed setting is first proposed by Garivier et al. [16], while it is only analyzed under switching environments.

Finally, it is worth pointing out that our Bandits-over-Bandits framework has connections with algorithms for online model selection and bandit corralling, see *e.g.*, [3] and references therein. This and similar techniques have been investigated under the context of non-stationary bandits in [21, 8]. Notwithstanding, existing works either have no theoretical guarantee or can only obtain sub-optimal dynamic regret bounds.

2 Problem Formulation

In this section, we introduce the notations to be used throughout the discussions and the model formulation.

2.1 Notation

Throughout the paper, all vectors are column vectors, unless specified otherwise. We define $[n]$ to be the set $\{1, 2, \ldots, n\}$ for any positive integer n. The notation $a : b$ is the abbreviation of consecutive indexes $a, a + 1, \ldots, b$. We use $\|\boldsymbol{x}\|$ to denote the Euclidean norm of a vector $\boldsymbol{x} \in \Re^d$. For a positive definite matrix $A \in \Re^{d\times d}$, we use $\|\boldsymbol{x}\|_A$ to denote the matrix norm $\sqrt{\boldsymbol{x}^\top A\boldsymbol{x}}$ of a vector $\boldsymbol{x} \in \Re^d$. We also denote $x \vee y$ and $x \wedge y$ as the maximum and minimum between $x, y \in \Re$, respectively. When logarithmic factors are omitted, we use $\widetilde{O}(\cdot)$ to denote function growth.

2.2 Learning Model

In each round $t \in [T]$, a decision set $D_t \subseteq \Re^d$ is presented to the learner, and it has to choose an action $X_t \in D_t$. Afterwards, the reward

$$Y_t = \langle X_t, \theta_t\rangle + \eta_t$$

is revealed. Here, we allow D_t to be chosen by an *oblivious adversary* whose actions are independent of those of the learner, and can be determined before the protocol starts [12]. $\theta_t \in \Re^d$ is an unknown d-dimensional vector, and η_t is a random noise drawn i.i.d. from an unknown sub-Gaussian distribution with variance proxy R. This implies $\mathbf{E}[\eta_t] = 0$, and $\forall\lambda \in \Re$ we have $\mathbf{E}[\exp(\lambda\eta_t)] \le \exp\left(\frac{\lambda^2R^2}{2}\right)$. Following the convention of existing bandits literature [1, 4], we assume there exist positive constants L and S, such that $\|X\| \le L$ and $\|\theta_t\| \le S$ holds for all $X \in D_t$ and all $t \in [T]$, and the problem instance is normalized so that $|\langle X, \theta_t\rangle| \le 1$ for all $X \in D_t$ and $t \in [T]$.

Instead of assuming the stochastic environment, where reward function remains stationary across the time horizon, we allow it to change over time. Specifically, we consider the general *drifting environment:* the sum of ℓ_2 differences of consecutive θ_t's should be bounded by some variation budget $B_T = \Theta(T^\rho)$ for some $\rho \in (0, 1)$, *i.e.*,

$$\sum_{t=1}^{T-1} \|\theta_{t+1} - \theta_t\| \le B_T. \tag{1}$$

We again allow the θ_t's to be chosen adversarially by an oblivious adversary. We also denote the set of all possible obliviously selected sequences of θ_t's that satisfies inequality (1) as $\Theta(B_T)$.

The learner's goal is to design a policy π to maximize the cumulative reward, or equivalently to minimize the worst case cumulative regret against the optimal policy π^*, that has full knowledge of θ_t's. Denoting $x_t^* = \text{argmax}_{x \in D_t} \langle x, \theta_t \rangle$, the dynamic regret of a given policy π is defined as

$$\mathcal{R}_T(\pi) = \sup_{\theta_{1:T} \in \Theta(B_T)} \mathbf{E}\left[\sum_{t=1}^{T} \langle x_t^* - X_t, \theta_t \rangle\right],$$

where the expectation is taken with respect to the (possible) randomness of the policy.

3 Lower Bound

We first provide a lower bound on the the regret to characterize the best achievable regret.

Theorem 1. *For any $T \geq d$, the dynamic regret of any policy π satisfies $\mathcal{R}_T(\pi) = \Omega\left(d^{\frac{2}{3}} B_T^{\frac{1}{3}} T^{\frac{2}{3}}\right)$.*

Sketch Proof. The construction of the lower bound instance is similar to the approach of [9]: nature divides the whole time horizon into $\lceil T/H \rceil$ blocks of equal length H rounds (the last block can possibly have less than H rounds). In each block, the nature initiates a new stationary linear bandit instance with parameters from the set $\{\pm\sqrt{d/4H}\}^d$. Nature also chooses the parameter for a block in a way that depends only on the learner's policy, and the worst case regret is $\Omega(d\sqrt{H})$. Since there is at least $\lfloor T/H \rfloor$ number of blocks, the total regret is $\Omega(dT/\sqrt{H})$. By examining the variation budget constraint, we have that the smallest possible H one can take is $\lceil (dT)^{\frac{2}{3}} B_T^{-\frac{2}{3}} \rceil$. The statement then follows. Please refer to Section A for the complete proof. □

4 Sliding Window Regularized Least Squares Estimator

As a preliminary, we introduce the sliding window regularized least squares estimator, which is the key tool in estimating the unknown parameters $\{\theta_t\}_{t=1}^T$. Despite the underlying non-stationarity, we show that the estimation error of this estimator can gracefully adapt to the parameter changes.

Consider a sliding window of length w, and consider the observation history $\{(X_s, Y_s)\}_{s=1\vee(t-w)}^{t-1}$ during the time window $(1 \vee (t-w)) : (t-1)$. The ridge regression problem with regularization parameter λ (> 0) is stated below:

$$\min_{\theta \in \Re^d} \lambda \|\theta\|^2 + \sum_{s=1\vee(t-w)}^{t-1} (X_s^\top \theta - Y_s)^2. \tag{2}$$

Denote $\hat{\theta}_t$ as a solution to the regularized ridge regression problem, and define matrix $V_{t-1} := \lambda I + \sum_{s=1\vee(t-w)}^{t-1} X_s X_s^\top$. The solution $\hat{\theta}_t$ has the following explicit expression:

$$\begin{aligned}\hat{\theta}_t &= V_{t-1}^{-1}\left(\sum_{s=1\vee(t-w)}^{t-1} X_s Y_s\right) \\ &= V_{t-1}^{-1}\left(\sum_{s=1\vee(t-w)}^{t-1} X_s X_s^\top \theta_s + \sum_{s=1\vee(t-w)}^{t-1} \eta_s X_s\right).\end{aligned} \tag{3}$$

The difference $\hat{\theta}_t - \theta_t =$ has the following expression:

$$\begin{aligned}&V_{t-1}^{-1}\left(\sum_{s=1\vee(t-w)}^{t-1} X_s X_s^\top \theta_s + \sum_{s=1\vee(t-w)}^{t-1} \eta_s X_s\right) - \theta_t \\ =&V_{t-1}^{-1} \sum_{s=1\vee(t-w)}^{t-1} X_s X_s^\top (\theta_s - \theta_t) + V_{t-1}^{-1} \sum_{s=1\vee(t-w)}^{t-1} \eta_s X_s \\ &- \lambda\theta_t,\end{aligned} \tag{4}$$

The first term on the right hand side of eq. (4) is the estimation inaccuracy due to the non-stationarity; while the second term is the estimation error due to random noise. We now upper bound the two terms separately. We upper bound the first term in the ℓ_2 sense.

Lemma 1. *For any $t \in [T]$, we have*

$$\begin{aligned}&\left\| V_{t-1}^{-1} \sum_{s=1\vee(t-w)}^{t-1} X_s X_s^\top (\theta_s - \theta_t) \right\| \\ \leq& \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\|.\end{aligned}$$

Sketch Proof. Our analysis relies on bounding the maximum eigenvalue of $V_{t-1}^{-1} \sum_{s=1\vee(t-w)}^{p} X_s X_s^\top$ for each $p \in \{1\vee(t-w), \ldots, t-1\}$. Please refer to Section B of appendix for the complete proof. □

Adopting the analysis in [1], we upper bound the second term in the matrix norm sense.

Lemma 2 ([1])**.** *For any $t \in [T]$ and any $\delta \in [0,1]$, we have with probability at least $1-\delta$,*

$$\begin{aligned}\left\| \sum_{s=1\vee(t-w)}^{t-1} \eta_s X_s - \lambda\theta_t \right\|_{V_{t-1}^{-1}} \leq& R\sqrt{d \ln\left(\frac{1 + wL^2/\lambda}{\delta}\right)} \\ &+ \sqrt{\lambda} S.\end{aligned}$$

From now on, we shall denote

$$\beta := R\sqrt{d \ln\left(\frac{1 + wL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S \tag{5}$$

for the ease of presentation. With these two lemmas, we have the following deviation inequality type bound for the latent expected reward of any action $x \in D_t$ in any round t.

Theorem 2. *For any $t \in [T]$ and any $\delta \in [0, 1]$, with probability at least $1 - \delta$, it holds for all $x \in D_t$ that*

$$\left|x^\top(\hat{\theta}_t - \theta_t)\right| \le L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|x\|_{V_{t-1}^{-1}}$$

Sketch Proof. The proof is a direct application of Lemmas 1 and 2. Please refer to Section C of the appendix for the complete proof. □

5 Sliding Window-Upper Confidence Bound (SW-UCB) Algorithm: A First Order Optimal Strategy

In this section, we describe the Sliding Window Upper Confidence Bound (`SW-UCB`) algorithm. When the variation budget B_T is known, we show that `SW-UCB` algorithm with a tuned window size achieves a dynamic regret bound which is optimal up to a multiplicative logarithmic factor. When the variation budget B_T is unknown, we show that `SW-UCB` algorithm can still be implemented with a suitably chosen window size so that the regret dependency on T is optimal, which still results in first order optimality in this case [19].

5.1 Design Intuition

In the stochastic environment where the linear reward function is stationary, the well known UCB algorithm follows the principle of optimism in face of uncertainty. Under this principle, the learner selects the action that maximizes the UCB, or the value of "mean plus confidence radius" [6]. We follow the principle by choosing in each round the action X_t with the highest UCB, *i.e.*,

$$\begin{aligned} X_t =& \text{argmax}_{x\in D_t} \Big\{ \langle x, \hat{\theta}_t\rangle \\ &+ L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|x\|_{V_{t-1}^{-1}} \Big\} \\ =& \text{argmax}_{x\in D_t} \Big\{ \langle x, \hat{\theta}_t\rangle + \beta \|x\|_{V_{t-1}^{-1}} \Big\}. \end{aligned} \tag{6}$$

When the number of actions is moderate, the optimization problem (6) can be solved by an enumeration over all $x \in D_t$. Upon selecting X_t, we have

$$\begin{aligned} &\langle x_t^*, \hat{\theta}_t\rangle + L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|x_t^*\|_{V_{t-1}^{-1}} \\ \le& \langle X_t, \hat{\theta}_t\rangle + L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|X_t\|_{V_{t-1}^{-1}}, \end{aligned} \tag{7}$$

by virtue of UCB. From Theorem 2, we further have with probability at least $1 - \delta$,

$$\langle x_t^*, \theta_t - \hat{\theta}_t\rangle \le L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|x_t^*\|_{V_{t-1}^{-1}}, \tag{8}$$

and

$$\begin{aligned} &\langle X_t, \hat{\theta}_t\rangle + L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|X_t\|_{V_{t-1}^{-1}} \\ \le& \langle X_t, \theta_t\rangle + 2L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + 2\beta \|X_t\|_{V_{t-1}^{-1}}. \end{aligned} \tag{9}$$

Combining inequalities (7), (8), and (9), we establish the following high probability upper bound for the expected per round regret, *i.e.*, with probability $1 - \delta$,

$$\langle x_t^* - X_t, \theta_t\rangle \le 2L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + 2\beta \|X_t\|_{V_{t-1}^{-1}}. \tag{10}$$

The regret upper bound of the `SW-UCB` algorithm (to be formalized in Theorem 3) is thus

$$\begin{aligned} &2 \sum_{t\in[T]} L \sum_{s=1\vee(t-w)}^{t-1} \|\theta_s - \theta_{s+1}\| + \beta \|X_t\|_{V_{t-1}^{-1}} \\ =& \widetilde{O}\left(w B_T + \frac{dT}{\sqrt{w}} \right). \end{aligned} \tag{11}$$

If B_T is known, the learner can set $w = \lfloor d^{2/3} T^{2/3} B_T^{-2/3} \rfloor$ and achieve a regret upper bound $\widetilde{O}(d^{2/3} B_T^{1/3} T^{2/3})$. If B_T is not known, which is often the case in practice, the learner can set $w = \lfloor (dT)^{2/3} \rfloor$ to obtain a regret upper bound $\widetilde{O}(d^{2/3}(B_T + 1)T^{2/3})$.

5.2 Design Details

In this section, we describe the details of the `SW-UCB` algorithm. Following its design guideline, the `SW-UCB` algorithm selects a positive regularization parameter $\lambda\ (> 0)$, and initializes $V_0 = \lambda I$. In each round t, the `SW-UCB` algorithm first computes the estimate $\hat{\theta}_t$ for θ_t according to eq. 3, and then finds the action X_t with largest UCB by solving the optimization problem (6). Afterwards, the corresponding reward Y_t is observed. The pseudo-code of the `SW-UCB` algorithm is shown in Algorithm 1.

Algorithm 1 SW-UCB algorithm

Input: Sliding window size w, dimension d, variance proxy of the noise terms R, upper bound of all the actions' ℓ_2 norms L, upper bound of all the θ_t's ℓ_2 norms S, and regularization constant λ.
Initialization: $V_0 \leftarrow \lambda I$.
for $t = 1, \ldots, T$ **do**
  $\hat{\theta}_t \leftarrow V_{t-1}^{-1}\left(\sum_{s=1\vee(t-w)}^{t-1} X_s Y_s\right)$.
  $X_t \leftarrow \text{argmax}_{x\in D_t}\left\{x^\top \hat{\theta}_t\right.$
    $\left. + \|x\|_{V_{t-1}^{-1}}\left[R\sqrt{d\ln\left(\frac{1+wL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S\right]\right\}$.
  $Y_t \leftarrow \langle X_t, \theta_t\rangle + \eta_t$.
  $V_t \leftarrow \lambda I + \sum_{s=1\vee(t-w+1)}^{t} X_s X_s^\top$.
end for

5.3 Regret Analysis

We are now ready to formally state a regret upper bound of the SW-UCB algorithm.

Theorem 3. *The dynamic regret of the SW-UCB algorithm is upper bounded as*

$$\mathcal{R}_T\left(\textit{SW-UCB algorithm}\right) = \widetilde{O}\left(wB_T + \frac{dT}{\sqrt{w}}\right).$$

When B_t (>0) is known, by taking $w = O((dT)^{2/3}B_T^{-2/3})$, the dynamic regret of the SW-UCB algorithm is

$$\mathcal{R}_T\left(\textit{SW-UCB algorithm}\right) = \widetilde{O}\left(d^{\frac{2}{3}}B_T^{\frac{1}{3}}T^{\frac{2}{3}}\right).$$

When B_t is unknown, by taking $w = O((dT)^{2/3})$, the dynamic regret of the SW-UCB algorithm is

$$\mathcal{R}_T\left(\textit{SW-UCB algorithm}\right) = \widetilde{O}\left(d^{\frac{2}{3}}\left(B_T+1\right)T^{\frac{2}{3}}\right).$$

Sketch Proof. The proof utilizes the fact that the per round regret of the SW-UCB algorithm is upper bounded by the UCB of the chosen action, and decomposes the UCB into two separated terms according to Lemmas 1 and 2, *i.e.*,

regret in round t = regret due to non-stationarity in round t + regret due to estimation error in round t.

The first term can be upper bounded by a intuitive telescoping sum; while for the second term, although a similar quantity is analyzed by the authors of [1] using a (beautiful) matrix telescoping technique under the stationary environment, we note that due to the "forgetting principle" of the SW-UCB algorithm, we cannot directly adopt the technique. Our proof thus makes a novel use of the Sherman-Morrison formula to overcome the barrier. Please refer to Section D of appendix for the complete proof. □

6 Bandit-over-Bandit (BOB) Algorithm: Automatically Adapting to the Unknown Variation Budget

In Section 5, we have seen that, by properly tuning w, the learner can achieve a first order optimal $\widetilde{O}\left(d^{2/3}(B_T+1)T^{2/3}\right)$ regret bound even if the knowledge of B_T is not available. However, in the case of an unknown and large B_T, *i.e.*, $B_T = \Omega(T^{1/3})$, the bound becomes meaningless as it is linear in T. To handle this case, we wish to design an online algorithm that incurs a dynamic regret of order $\widetilde{O}\left(d^{\nu}B_T^{1-\sigma}T^{\sigma}\right)$ for some $\nu \in [0,1]$ and $\sigma \in (0,1)$, without knowing B_T. Note from Theorem 1, no algorithm can achieve a dynamic regret of order $o(d^{2/3}B_T^{1/3}T^{2/3})$, so we must have $\sigma \geq \frac{2}{3}$. In this section, we develop a novel Bandit-over-Bandit (BOB) algorithm that achieves a regret of $\tilde{O}(d^{2/3}B_T^{1/4}T^{3/4})$. Hence, (BOB) still has a dynamic regret sublinear in T when $B_T = \Theta(T^{\rho})$ for any $\rho \in (0,1)$ and B_T is not known, unlike the SW-UCB algorithm.

6.1 Design Challenges

Reviewing Theorem 3, we know that setting the window length w to a fixed value

$$w^* = \left\lfloor (dT)^{2/3}(B_T+1)^{-2/3}\right\rfloor \tag{12}$$

can give us a $\widetilde{O}\left(d^{2/3}(B_T+1)^{1/3}T^{2/3}\right)$ regret bound. But when B_T is not provided a priori, we need to also "learn" the unknown B_T in order to properly tune w. In a more restrictive setting in which the differences between consecutive θ_t's follow some underlying stochastic process, one possible approach is applying a suitable machine learning technique to learn the underlying stochastic process at the beginning, and tune the parameter w accordingly. In the more general setting, however, this strategy cannot work as the change between consecutive θ_t's can be arbitrary (or even adversarially) as long as the total variation is bounded by B_T.

6.2 Design Intuition

The above mentioned observations as well as the established results motivate us to make use of the SW-UCB algorithm as a sub-routine, and "hedge" against the changes of θ_t's to identify a reasonable fixed window length [7]. To this end, we describe the main idea of the Bandit-over-Bandit (BOB) algorithm. The BOB algorithm divides the whole time horizon into $\lceil T/H\rceil$ blocks of equal length H rounds (the last block can possibly have less than H rounds), and specifies a set J ($\subseteq [H]$) from which each w_i is drawn from. For each block $i \in [\lceil T/H\rceil]$, the BOB algorithm first selects a window length w_i ($\in J$), and initiates a new copy of the SW-UCB algorithm with the selected window length as a sub-routine to choose actions for this block. On top of this, the BOB algorithm also maintains a separate algorithm for adversarial multi-armed bandits, *e.g.*, the EXP3 algorithm, to govern the selection of window length for each block, and thus the name Bandit-over-Bandit. Here, the total reward of each block is used as feedback for the EXP3 algorithm.

To determine H and J, we first consider the regret of the BOB algorithm. Since the window length is constrained to be in J, and is less than or equal to H, w^* is not necessarily the optimal window length in this case, and we hence denote the optimally tuned window length as $w^\dagger$. By design of the BOB algorithm, its regret can be decomposed as the regret of an algorithm that optimally tunes the window length $w_i = w^\dagger$ for each block i plus the loss due to learning the value $w^\dagger$ with the EXP3 algorithm,

$$\begin{aligned}
&\mathbf{E}\left[\text{Regret}_T(\texttt{BOB algorithm})\right]\\
=&\mathbf{E}\left[\sum_{t=1}^{T}\langle x_t^*,\theta_t\rangle-\sum_{t=1}^{T}\langle X_t,\theta_t\rangle\right]\\
=&\mathbf{E}\left[\sum_{t=1}^{T}\langle x_t^*,\theta_t\rangle-\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w^\dagger\right),\theta_t\right\rangle\right]\\
&+\mathbf{E}\left[\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w^\dagger\right),\theta_t\right\rangle\right.\\
&\left.-\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w_i\right),\theta_t\right\rangle\right].
\end{aligned}\tag{13}$$

Here for a round t in block i, $X_t(w)$ refers to the action selected in round t by the SW-UCB algorithm with window length $w\wedge(t-(i-1)H-1)$ initiated at the beginning of block i.

By Theorem 3, the first expectation in eq. (13) can be upper bounded as

$$\begin{aligned}
&\mathbf{E}\left[\sum_{t=1}^{T}\langle x_t^*,\theta_t\rangle-\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w^\dagger\right),\theta_t\right\rangle\right]\\
=&\mathbf{E}\left[\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle x_t^*-X_t\left(w^\dagger\right),\theta_t\right\rangle\right]\\
=&\sum_{i=1}^{\lceil T/H\rceil}\widetilde{O}\left(w^\dagger B_T(i)+\frac{dH}{\sqrt{w^\dagger}}\right)\\
=&\widetilde{O}\left(w^\dagger B_T+\frac{dT}{\sqrt{w^\dagger}}\right),
\end{aligned}\tag{14}$$

where $B_T(i)=\sum_{t=(i-1)H+1}^{(i\cdot H\wedge t)-1}\|\theta_t-\theta_{t+1}\|$ is the total variation in block i.

We then turn to the second expectation in eq. (13). We can easily see that the number of rounds for the EXP3 algorithm is $\lceil T/H\rceil$ and the number of possible values of w_i's is $|J|$. Denoting the maximum absolute sum of rewards of any block as random variable Q, the authors of [7] gives the following regret bound.

$$\begin{aligned}
&\mathbf{E}\left[\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w^\dagger\right),\theta_t\right\rangle\right.\\
&\left.-\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w_i\right),\theta_t\right\rangle\right]\\
\leq&\mathbf{E}\left[\widetilde{O}\left(Q\sqrt{\frac{|J|T}{H}}\right)\right].
\end{aligned}\tag{15}$$

To proceed, we have to give a high probability upper bound for Q.

Lemma 3.

$$\Pr\left(Q\leq H+2R\sqrt{H\ln\frac{T}{\sqrt{H}}}\right)\geq 1-\frac{2}{T}.$$

Sketch Proof. The proof makes use of the R-sub-Gaussian property of the noise terms as well as the union bound over all the blocks. Please refer to Section E of the appendix for the complete proof. □

Note that the regret of our problem is at most T, eq. (15) can be further upper bounded as

$$\begin{aligned}
&\mathbf{E}\left[\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w^\dagger\right),\theta_t\right\rangle\right.\\
&\left.-\sum_{i=1}^{\lceil T/H\rceil}\sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\left\langle X_t\left(w_i\right),\theta_t\right\rangle\right]\\
\leq&\mathbf{E}\left[\widetilde{O}\left(Q\sqrt{\frac{|J|T}{H}}\right)\middle|Q\leq H+2HR\sqrt{\ln T}\right]\\
&\times\Pr\left(Q\leq H+2HR\sqrt{\ln T}\right)\\
&+\mathbf{E}\left[\widetilde{O}\left(Q\sqrt{\frac{|J|T}{H}}\right)\middle|Q\geq H+2HR\sqrt{\ln T}\right]\\
&\times\Pr\left(Q\geq H+2HR\sqrt{\ln T}\right)\\
=&\widetilde{O}\left(\sqrt{H|J|T}\right)+T\cdot\frac{2}{T}\\
=&\widetilde{O}\left(\sqrt{H|J|T}\right).
\end{aligned}\tag{16}$$

Combining eq. (13), (14), and (16), the regret of the BOB algorithm is

$$\mathcal{R}_T(\texttt{BOB algorithm})=\widetilde{O}\left(w^\dagger B_T+\frac{dT}{\sqrt{w^\dagger}}+\sqrt{H|J|T}\right).\tag{17}$$

Eq. (17) exhibits a similar structure to the regret of the SW-UCB algorithm as stated in Theorem 3, and this immediately indicates a clear trade-off in the design of the block length H. On one hand, H should be small to control the regret incurred by the EXP3 algorithm in identifying $w^\dagger$, *i.e.*, the third term in eq. (17); on the other hand, H should also be large enough so that $w^\dagger$ can get close to $w^* = \lfloor (dT)^{2/3}(B_T+1)^{-2/3}\rfloor$ so that the sum of the first two terms in eq. (17) is minimized. A more careful inspection also reveals the tension in the design of J. Obviously, we hope that $|J|$ is small, but we also wish J to be dense enough so that it forms a cover to the set H. Otherwise, even if H is large and $w^\dagger$ can approach w^*, approximating w^* with any element in J can cause a major loss.

These observations suggest the following choice of J.

$$J = \left\{H^0, \left\lfloor H^{\frac{1}{\Delta}}\right\rfloor, \ldots, H\right\} \tag{18}$$

for some positive integer Δ. For the purpose of analysis, suppose the (unknown) parameter $w^\dagger$ can be expressed as $\text{clip}_J\left(\lfloor d^\epsilon T^\alpha (B_T+1)^{-\alpha}\rfloor\right)$ with some $\alpha \in [0,1]$ and $\epsilon > 0$ to be determined, where $\text{clip}_J(x)$ finds the largest element in J that does not exceed x. Notice that $|J| = \Delta + 1$, the regret of the BOB algorithm then becomes

$$\begin{aligned}
&\mathcal{R}_T(\texttt{BOB}\text{ algorithm})\\
=&\widetilde{O}\left(d^\epsilon (B_T+1)^{1-\alpha} T^\alpha H^{\frac{2}{\Delta}}\right.\\
&\left.+d^{1-\frac{\epsilon}{2}}(B_T+1)^{\frac{\alpha}{2}} T^{1-\frac{\alpha}{2}} H^{\frac{2}{\Delta}} + \sqrt{HT\Delta}\right)\\
=&\widetilde{O}\left(d^\epsilon (B_T+1)^{1-\alpha} T^\alpha\right.\\
&\left.+d^{1-\frac{\epsilon}{2}}(B_T+1)^{\frac{\alpha}{2}} T^{1-\frac{\alpha}{2}} + \sqrt{HT}\right),
\end{aligned} \tag{19}$$

where we have set $\Delta = \lceil \ln H\rceil$ in eq. (19); Since $w^\dagger \in J$ (or $w^\dagger \le H$), and H should not depend on B_T, we can set

$$H = \lfloor d^\epsilon T^\alpha \rfloor, \tag{20}$$

and the regret of the BOB algorithm (to be formalized in Theorem 4) is upper bounded as

$$\begin{aligned}
&\mathcal{R}_T(\texttt{BOB}\text{ algorithm})\\
=&\widetilde{O}\left(d^\epsilon (B_T+1)^{1-\alpha} T^\alpha + d^{1-\frac{\epsilon}{2}}(B_T+1)^{\frac{\alpha}{2}} T^{1-\frac{\alpha}{2}}\right.\\
&\left.+d^{\frac{\epsilon}{2}} T^{\frac{1}{2}+\frac{\alpha}{2}}\right)\\
=&\widetilde{O}\left(d^{\frac{2}{3}}(B_T+1)^{\frac{1}{4}} T^{\frac{3}{4}}\right).
\end{aligned} \tag{21}$$

Here, we have taken $\alpha = 1/2$ and $\epsilon = 2/3$, but we have to emphasize that the choice of $w^\dagger, \alpha,$ and ϵ are purely for an analysis purpose. The only parameters that we need to design are

$$H = \left\lfloor d^{\frac{2}{3}} T^{\frac{1}{2}}\right\rfloor, \Delta = \lceil \ln H\rceil, J = \left\{1, \left\lfloor H^{\frac{1}{\Delta}}\right\rfloor, \ldots, H\right\}, \tag{22}$$

which clearly do not depend on B_T.

6.3 Design Details

We are now ready to describe the details of the BOB algorithm. With H, Δ and J defined as eq. (22), the BOB algorithm additionally initiates the parameter

$$\begin{aligned}
&\gamma = \min\left\{1, \sqrt{\frac{(\Delta+1)\ln(\Delta+1)}{(e-1)\lceil T/H\rceil}}\right\},\\
&s_{j,1} = 1 \quad \forall j = 0, 1, \ldots, \Delta.
\end{aligned} \tag{23}$$

for the EXP3 algorithm [7]. The BOB algorithm then divides the time horizon T into $\lceil T/H\rceil$ blocks of length H rounds (except for the last block, which can be less than H rounds). At the beginning of each block $i \in [\lceil T/H\rceil]$, the BOB algorithm first sets

$$p_{j,i} = (1-\gamma)\frac{s_{j,i}}{\sum_{u=0}^{\Delta} s_{u,i}} + \frac{\gamma}{\Delta+1} \quad \forall j = 0, 1, \ldots, \Delta, \tag{24}$$

and then sets $j_i = j$ with probability $p_{j,i}$ for all $j = 0, \ldots, \Delta$. The selected window length is thus $w_i = \left\lfloor H^{j_i/\Delta}\right\rfloor$. Afterwards, the BOB algorithm selects actions X_t by running the SW-UCB algorithm with window length w_i for each round t in block i, and the total collected reward is $\sum_{t=(i-1)H+1}^{i\cdot H\wedge T} Y_t = \sum_{t=(i-1)H+1}^{i\cdot H\wedge T}\langle X_t, \theta_t\rangle + \eta_t$. Finally, the rewards are rescaled by dividing $2H + 4R\sqrt{H\ln(T/\sqrt{H})}$, and then added by $1/2$ so that it lies within $[0,1]$ with high probability, and the parameter $s_{j_i,i+1}$ is set to

$$s_{j_i,i}\cdot\exp\left(\frac{\gamma}{(\Delta+1)p_{j_i,i}}\left(\frac{1}{2} + \frac{\sum_{t=(i-1)H+1}^{i\cdot H\wedge T} Y_t}{2H + 4R\sqrt{H\ln\frac{T}{\sqrt{H}}})}\right)\right); \tag{25}$$

while $s_{u,i+1}$ is the same as $s_{u,i}$ for all $u \neq j_i$. The pseudo-code of the BOB algorithm is shown in Algorithm 2.

6.4 Regret Analysis

We are now ready to present the regret analysis of the BOB algorithm.

Theorem 4. *The dynamic regret of the BOB algorithm with the SW-UCB algorithm as a sub-routine is*

$$\mathcal{R}_T(\textit{BOB algorithm}) = \widetilde{O}\left(d^{\frac{2}{3}}(B_T+1)^{\frac{1}{4}} T^{\frac{3}{4}}\right).$$

Algorithm 2 `BOB` algorithm

Input: Time horizon T, the dimension d, variance proxy of the noise terms R, upper bound of all the actions' ℓ_2 norms L, upper bound of all the θ_t's ℓ_2 norms S, and a constant λ.
Initialize H, Δ, J by eq. (22), $\gamma, \{s_{j,1}\}_{j=0}^{\Delta}$ by eq. (23).
for $i = 1, 2, \ldots, \lceil T/H \rceil$ **do**
  Define distribution $(p_{j,i})_{j=0}^{\Delta}$ by eq. (24).
  Set $j_t \leftarrow j$ with probability $p_{j,i}$.
  $w_i \leftarrow \lfloor H^{j_t/\Delta} \rfloor$.
  $V_{(i-1)H} = \lambda I$.
  for $t = (i-1)H + 1, \ldots, i \cdot H \wedge T$ **do**
    $\hat{\theta}_t \leftarrow V_{t-1}^{-1} \left(\sum_{s=[(i-1)H+1] \vee (t-w_i)}^{t-1} X_s Y_s \right)$.
    Pull arm $X_t \leftarrow \text{argmax}_{x \in D_t} \Big\{ x^\top \hat{\theta}_t + \|x\|_{V_{t-1}^{-1}} \left[R\sqrt{d \ln \left(T\left(1 + w_i L^2/\lambda\right)\right)} + \sqrt{\lambda} S \right] \Big\}$.
    Observe $Y_t = \langle X_t, \theta_t \rangle + \eta_t$.
    $V_t \leftarrow \lambda I + \sum_{s=[(i-1)H+1] \vee (t+1-w_i)}^{t} X_s X_s^\top$.
  end for
  Define $s_{j_i, i+1}$ according to eq. (25)
  Define $s_{u,i+1} \leftarrow s_{u,i} \ \forall u \neq j_i$
end for

Sketch Proof. The proof of the theorem essentially follows Section 6.2, and we thus omit it. ☐

7 Numerical Experiments

As a complement to our theoretical results, we conduct numerical experiments on synthetic data to compare the regret performances of the `SW-UCB` algorithm and the `BOB` algorithm with a modified EXP3.S algorithm analyzed in [8]. Note that the algorithms in [8] are designed for the stochastic MAB setting, a special case of us, we follow the setup of [8] for fair comparisons. Specifically, we consider a 2-armed bandit setting, and we vary T from 3×10^4 to 2.4×10^5 with a step size of 3×10^4. We set θ_t to be the following sinusoidal process, *i.e.*, $\forall t \in [T]$,

$$\theta_t = \begin{pmatrix} 0.5 + 0.3 \sin\left(5 B_T \pi t / T\right) \\ 0.5 + 0.3 \sin\left(\pi + 5 B_T \pi t / T\right) \end{pmatrix}. \tag{26}$$

The total variation of the θ_t's across the whole time horizon is upper bounded by $\sqrt{2} B_T = O(B_T)$. We also use i.i.d. normal distribution with $R = 0.1$ for the noise terms.

Known Constant Variation Budget. We start from the known constant variation budget case, *i.e.*, $B_T = 1$, to measure the regret growth of the two optimal algorithms, *i.e.*, the `SW-UCB` algorithm and the modified EXP3.S algorithm, with respect to the total number of rounds. The log-log plot is shown in Fig. 1. From the plot, we can see that the regret of `SW-UCB` algorithm is only about 20% of the regret of EXP3.S algorithm.

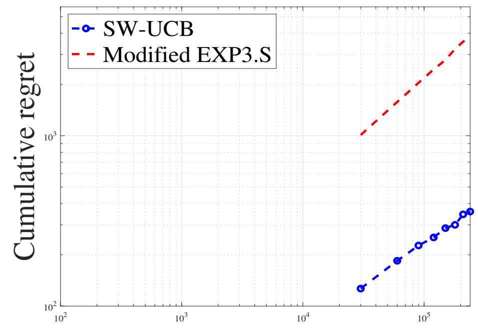

Figure 1: Log-log plot for $B_T = O(1)$.

Unknown Time-Dependent Variation Budget. We then turn to the more realistic time-dependent variation budget case, *i.e.*, $B_T = T^{1/3}$. As the modified EXP3.S algorithm does not apply to this setting, we compare the performances of the `SW-UCB` algorithm and the `BOB` algorithm. The log-log plot is shown in Fig. 2. From the results, we verify that the slope of the regret growth of both algorithms roughly match the established results, and the regret of `BOB` algorithm's is much smaller than that of the `SW-UCB` algorithm's.

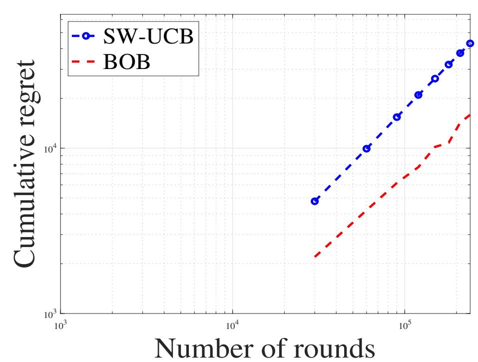

Figure 2: Log-log plot for $B_T = O(T^{1/3})$.

References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, 2011.

[2] M. Abeille and A. Lazaric. Linear thompson sampling revisited. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[3] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire. Corralling a band of bandit algorithms.

In *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*, 2017.

[4] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[5] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. In *Journal of Machine Learning Research, 3:397–422, 2002.*, 2002.

[6] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning, 47, 235–256*, 2002.

[7] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. In *SIAM Journal on Computing, 2002, Vol. 32, No. 1 : pp. 48–77*, 2002.

[8] O. Besbes, Y. Gur, and A. Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. In *Available at: https://ssrn.com/abstract=2436629, year = 2018,*.

[9] O. Besbes, Y. Gur, and A. Zeevi. Stochastic multi-armed bandit with non-stationary rewards. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[10] O. Besbes, Y. Gur, and A. Zeevi. Non-stationary stochastic optimization. In *Operations Research, 2015, 63 (5), 1227–1244*, 2015.

[11] S. Bubeck and N. Cesa-Bianchi. *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. Foundations and Trends in Machine Learning, 2012, Vol. 5, No. 1: pp. 1–122, 2012.

[12] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[13] C. Chiang, T. Yang, C. Lee, M. Mahdavi, C. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Conference on Learning Theory (COLT)*, 2012.

[14] W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[15] V. Dani, T. Hayes, and S. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Conference on Learning Theory (COLT)*, 2008.

[16] A. Garivier and E. Moulines. On upper-confidence bound policies for switching bandit problems. In *The 22nd International Conferenc on Algorithmic Learning Theory (ALT)*, 2011.

[17] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online optimization : Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[18] Z. Karnin and O. Anava. Multi-armed bandits: Competing with optimal sequences. In *Procedding of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[19] N. Keskin and A. Zeevi. Chasing demand: Learning and earning in a changing environments. In *Mathematics of Operations Research, 2016, 42(2), 277–307*, 2016.

[20] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press 2018, 2018.

[21] H. Luo, C. Wei, A. Agarwal, and J. Langford. Efficient contextual bandits in non-stationary worlds. In *Proceedings of the 31st Conference on Learning Theory (COLT)*, 2018.

[22] R. Rigollet and J. Hütter. *High Dimensional Statistics*. Lecture Notes, 2018, 2018.

[23] P. Rusmevichientong and J. Tsitsiklis. Linearly parameterized bandits. In *Mathematics of Operations Research, 35(2):395–411, 2010*, 2010.

[24] D. Russo and B. V. Roy. Learning to optimize via posterior sampling. In *Mathematics of Operations Research*, 2014.

[25] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. Tracking the best expert in non-stationary stochastic environments. In *Proceedings of the 29th Annual Conference on Neural Information Processing (NIPS)*, 2016.