Unifying Clustered and Non-stationary Bandits

Chuanhao Li
University of Virginia
cl5ev@virginia.edu

Qingyun Wu
University of Virginia
qw2ky@virginia.edu

Hongning Wang
University of Virginia
hw5x@virginia.edu

Abstract

Non-stationary bandits and clustered bandits lift the restrictive assumptions in contextual bandits and provide solutions to many important real-world scenarios. Though they have been studied independently so far, we point out the essence in solving these two problems overlaps considerably. In this work, we connect these two strands of bandit research under the notion of test of homogeneity, which seamlessly addresses change detection for non-stationary bandit and cluster identification for clustered bandit in a unified solution framework. Rigorous regret analysis and extensive empirical evaluations demonstrate the value of our proposed solution, especially its flexibility in handling various environment assumptions, e.g., a clustered non-stationary environment.

1 Introduction

Most existing contextual bandit algorithms impose strong assumptions on the mapping between context and reward (Abbasi-Yadkori et al., 2011; Chu et al., 2011; Li et al., 2010): typically it is assumed that the expected reward associated with a particular action is determined by a *time-invariant function* of the context vector. This overly simplified assumption restricts the application of contextual bandits in many important real-world scenarios, where a learner has to serve a population of users with possible mutual dependence and changing interest. This directly motivates recent efforts that postulate more general reward assumptions (Wu et al., 2016; Filippi et al., 2010; Maillard and Mannor, 2014; Kleinberg et al., 2008); among them, *non-stationary bandits* (Wu et al., 2018; Slivkins and Upfal, 2008; Cao et al., 2019; Besson and Kaufmann, 2019; Russac et al., 2019; Chen et al., 2019) and *clustered bandits* (Gentile et al., 2014; Li et al., 2016; Gentile et al., 2017; Li et al., 2019) received much attention.

In non-stationary bandits, the reward mapping function becomes time-variant. A typical non-stationary setting is the abruptly changing environment, a.k.a, a piecewise stationary environment, in which the environment undergoes abrupt changes at unknown time points but remains stationary between two consecutive change points (Yu and Mannor, 2009; Garivier and Moulines, 2011). A working solution needs to either properly discount historical observations (Hartland et al., 2006; Garivier and Moulines, 2011; Russac et al., 2019) or detect the change points and reset the model estimation accordingly (Yu and Mannor, 2009; Cao et al., 2019; Wu et al., 2018). In clustered bandits, grouping structures of bandit models are assumed, e.g., users in a group share the same bandit model. But instead of assuming an explicit dependency structure, e.g., leveraging existing social network among users (Cesa-Bianchi et al., 2013; Wu et al., 2016), clustered bandit algorithms aim to simultaneously cluster and estimate the bandit models during the sequential interactions with users (Gentile et al., 2014; Li et al., 2016; Gentile et al., 2017; Li et al., 2019). Its essence is thus to measure the relatedness between different bandit models. Typically, confidence bound of model parameter estimation (Gentile et al., 2014) or reward estimation (Gentile et al., 2017) is used for this purpose.

So far these two problems have been studied in parallel; but the key principles to solve them overlap considerably. On the one hand, mainstream solutions for piecewise stationary bandits detect change points in the underlying reward distribution by comparing the observed rewards (Cao et al., 2019) or the quality of estimated rewards (Yu and Mannor, 2009; Wu et al., 2018) in a window of consecutive observations. If change happens in the window, the designed statistics

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

of interest would exceed a threshold with a high probability. This is essentially sequential hypothesis testing of a model's fitness (Siegmund, 2013). On the other hand, existing solutions for clustered bandits evaluate if two bandit models share the same set of parameters (Gentile et al., 2014; Li et al., 2016) or the same reward estimation on a particular arm (Gentile et al., 2017). This can also be understood as a goodness-of-fit test between models.

In this work, we take the first step to unify these two parallel strands of bandit research under the notion of *test of homogeneity*, and study non-stationarity in linear bandit with time-varying arm set, which distinguishes us from most existing work. We address both problems by testing whether the collection of observations in a bandit model follows the same distribution as that of new observations (i.e., change detection in non-stationary bandit algorithms) or of those in another bandit model (i.e., cluster identification in clustered bandit algorithms). Built upon our solution framework, bandit models can operate on individual users with much stronger flexibility, so that new bandit learning problems can be created and addressed. This enables us to study a new and challenging bandit problem in a *clustered non-stationary* environment, where the learner has to reset individual models when a change of reward distribution is detected, and merge them when they are determined as identical. This task of doing change detection while clustering is novel and important by itself (Mazhar et al., 2018), and has never been considered in bandit problem where the observations are non-IID in nature. Since our solution automatically detects changes and clustering structure, it has a much weaker assumption about the environment (e.g., it can be clustered, or non-stationary, or both). Furthermore, our solution enables data sharing across both users and time, when such structure exists in the environment, thus greatly reducing sample complexity in learning bandit models. Our rigorous regret analysis and extensive empirical evaluations demonstrate the value of this unified solution, especially its advantages in handling various environment assumptions.

2 Related work

Our work is closely related to the studies in non-stationary bandits and clustered bandits. In this section, we discuss the most representative solutions in each direction and highlight their connections.

Non-stationary bandits. Instead of assuming a time-invariant environment, the reward mapping is allowed to change over time in this problem setting. Commonly imposed assumptions include slowly-varying environment (Besbes et al., 2019; Cheung et al., 2019) and abruptly-changing environment (Moulines, 1985; Wu et al., 2018; Auer et al., 2019). We focus on the latter setting, which is also known as a piecewise stationary environment in literature (Yu and Mannor, 2009; Garivier and Moulines, 2011). In a non-stationary setting, the main focus is to eliminate the distortion from out-dated observations, which follow a different reward distribution than that of the current environment. Popular solutions for the piecewise stationary environment actively detect change points and reset bandit models accordingly (Yu and Mannor, 2009; Cao et al., 2019; Besson and Kaufmann, 2019; Wu et al., 2018; Hariri et al., 2015; Auer et al., 2019; Chen et al., 2019). It should be noted that this paper studies non-stationarity in linear bandit with time-varying arm set (Wu et al., 2018; Cheung et al., 2019; Russac et al., 2019; Zhao et al., 2020), which is different from the solutions for non-stationary MAB problem (Yu and Mannor, 2009; Cao et al., 2019; Besson and Kaufmann, 2019; Hariri et al., 2015; Auer et al., 2019) or the non-stationary contextual MAB (Agarwal et al., 2014; Luo et al., 2018; Chen et al., 2019). Therefore, their results do not apply to the setting considered in this paper. The closest work to our setting is Wu et al. (2018), which maintains a pool of base linear bandit models and adaptively adds or selects from them via a change detector, which monitors how well each base bandit model predicts the new observations. This in essence boils down to a likelihood-ratio test for change in the bandit parameter. To the best of our knowledge, all the other studies for non-stationary linear bandit assume a slowly-varying environment and adopts strategies like sliding window (Cheung et al., 2019), decaying weight (Russac et al., 2019) or periodical restart (Zhao et al., 2020) to eliminate the distortion from out-dated observations.

Clustered bandits. When serving a population of users, the vanilla linear bandit usually models the preference of each individual user in isolation, neglecting the correlation between users. In order to improve sample efficiency, such user correlation can be utilized to enable collaboration among each individual bandit models (Li, 2016; Gentile et al., 2014; Li et al., 2016; Gentile et al., 2017; Cesa-Bianchi et al., 2013; Wu et al., 2016). Besides leveraging explicit structure among users, such as social networks (Buccapatnam et al., 2013; Cesa-Bianchi et al., 2013; Wu et al., 2016; Yang et al., 2020), recent efforts focus on online clustering of bandits via the interactions with users (Gentile et al., 2014; Li et al., 2016; Gentile et al., 2017; Li et al., 2019). For example, Gentile et al. (2014) assumed that observations from different users in the same cluster share the same underlying bandit parameter. Thus, they estimate the clustering structure among users based on the difference between their

estimated bandit parameters. Li et al. (2016) used a similar idea to cluster items (arms) as well. Gentile et al. (2017) further studied arm-dependent clustering of users, by the projected difference between models on each arm. Li et al. (2019) proposed a phase-based algorithm to relax the uniform user frequency assumption in the analysis of Gentile et al. (2014). Essentially, these algorithms measure the relatedness between users by evaluating the homogeneity of observations associated with individual models, though they have used various measures for this purpose.

3 Methodology

In this section, we first formulate the problem setup studied in this paper. Then we describe two key components pertaining to non-stationary bandits and clustered bandits, and pinpoint the essential equivalence between them under the notion of homogeneity test, which becomes the cornerstone of our unified solution. Based on our construction of homogeneity test, we explain the proposed solution, followed by our theoretical analysis of the resulting upper regret bound of the proposed solution.

3.1 Problem formulation

To offer a unified approach that addresses the two target problems, we formulate a general bandit learning setting that encompasses both non-stationarity in individual models and existence of clustering structure.

Consider a learner that interacts with a set of n users, $\mathcal{U} = \{1, ..., n\}$. At each time $t = 1, 2, ..., T$, the learner receives an arbitrary user indexed by $i_t \in \mathcal{U}$, together with a set of available arms $C_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,K}\}$ to choose from, where $\mathbf{x}_{t,j} \in \mathbb{R}^d$ denotes the context vector associated with the arm indexed by j at time t (assume $\|\mathbf{x}_{t,j}\| \leq 1$ without loss of generality), and K denotes the size of arm pool C_t. After the learner chooses an arm $\mathbf{x}_t$, its reward $y_t \in \mathbb{R}$ is fed back from the user i_t. We follow the linear reward assumption (Abbasi-Yadkori et al., 2011; Chu et al., 2011; Li et al., 2010) and use $\theta_{i_t,t}$ to denote the parameter of the reward mapping function in user i_t at time t (assume $\|\theta_{i_t,t}\| \leq 1$). Under this assumption, the reward at time t is $y_t = \mathbf{x}_t^\top \theta_{i_t,t} + \eta_t$, where η_t is Gaussian noise drawn from $N(0, \sigma^2)$. Interaction between the learner and users repeats, and the learner's goal is to maximize the accumulated reward it receives from all users in $\mathcal{U}$ up to time T.

Denote the set of time steps when user $i \in \mathcal{U}$ is served up to time T as $\mathcal{N}_i(T) = \{1 \leq t \leq T : i_t = i\}$. Among time steps $t \in \mathcal{N}_i(T)$, user i's parameter $\theta_{i,t}$ changes abruptly at arbitrary time steps $\{c_{i,1}, ..., c_{i,\Gamma_i(T)-1}\}$, but remain constant between any two consecutive change points. $\Gamma_i(T)$ denotes the total number of stationary periods in $\mathcal{N}_i(T)$. The set of unique parameters that $\theta_{i,t}$ takes for any user at any time is denoted as $\{\phi_k\}_{k=1}^m$ and their frequency of occurrences in T is $\{p_k\}_{k=1}^m$. Note that we do not impose any assumption on the distribution over the user, nor on the distribution over the unique bandit parameter appearing in each round. Also note that the ground-truth linear parameters, the set of change points, the number and frequencies of unique parameters are unknown to the learner. Moreover, the number of users, i.e., n, and the number of unique bandit parameters across users, i.e., m, are finite but arbitrary.

Our problem setting defined above is general. The non-stationary and clustering structure of an environment can be specified by different associations between $\{\theta_{i,t}\}_{i=1}^n$ and $\{\phi_k\}_{k=1}^m$ across users over time $t = 1, 2, ..., T$. For instance, by setting $n > m$ and $\Gamma_i(T) = 1, \forall i \in \mathcal{U}$, the problem reduces to the clustered bandits problem, which assumes sharing of bandit models among users with stationary reward distributions. By setting $n = 1$, $m > 1$ and $\Gamma_i(T) > 1, \forall i \in \mathcal{U}$, it reduces to the piecewise stationary bandits problem, which only concerns users with non-stationary reward distributions in isolation.

To make our solution compatible with existing work in non-stationary bandits and clustered bandits, we also follow the three commonly made assumptions about the environment.

Assumption 1 (Change detectability) *For any user $i \in \mathcal{U}$ and any change point c in user i, there exists $\Delta > 0$ such that at least ρ portion of arms satisfy: $|\mathbf{x}^\top \theta_{i,c-1} - \mathbf{x}^\top \theta_{i,c}| > \Delta$ (Wu et al., 2018).*

Assumption 2 (Separateness among $\{\phi_k\}_{k=1}^m$) *For any two different unique parameters $\phi_i \neq \phi_j$, we have $\|\phi_i - \phi_j\| \geq \gamma > 0$ (Gentile et al., 2014, 2017; Li et al., 2019).*

Assumption 3 (Context regularity) *At each time t, arm set C_t is generated i.i.d. from a sub-Gaussian random vector $X \in \mathbb{R}^d$, such that $\mathbb{E}[XX^\top]$ is full-rank with minimum eigenvalue $\lambda' > 0$; and the variance ς^2 of the random vector satisfies $\varsigma^2 \leq \frac{\lambda'^2}{8 \ln 4K}$ (Gentile et al., 2014, 2017; Li et al., 2019).*

The first assumption establishes the detectability of change points in each individual bandit models over time. The second assumption ensures separation within the global unique parameter set shared by all users, and the third assumption specifies the property of context vectors. Based on these assumptions, we establish the problem setup in this work and illustrate

it on the left side of Figure 1.

3.2 Test statistic for homogeneity

As discussed in Section 2, the key problem in non-stationary bandits is to detect changes in the underlying reward distribution, and the key problem in clustered bandits is to measure the relatedness between different models. We view both problems as testing homogeneity between two sets of observations to unify these two seemingly distinct problems. For change detection, we test homogeneity between recent and past observations to evaluate whether there has been a change in the underlying bandit parameters for these two consecutive sets of observations. For cluster identification, we test homogeneity between observations of two different users to verify whether they share the same bandit parameter. On top of the test results, we operate the bandit models accordingly for model selection, model aggregation, arm selection, and model update.

We use $\mathcal{H}_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t_1}$ and $\mathcal{H}_2 = \{(\mathbf{x}_j, y_j)\}_{j=1}^{t_2}$ to denote two sets of observations, where $t_1, t_2 \geq 1$ are their cardinalities. $(\mathbf{X}_1, \mathbf{y}_1)$ and $(\mathbf{X}_2, \mathbf{y}_2)$ denote design matrices and feedback vectors of $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively, where each row of $\mathbf{X}$ is the context vector of a selected arm and the corresponding element in $\mathbf{y}$ is the observed reward for this arm. Under linear reward assumption, $\forall(\mathbf{x}_i, y_i) \in \mathcal{H}_1$, $y_i \sim N(\mathbf{x}_i^\top \theta_1, \sigma^2)$, and $\forall(\mathbf{x}_j, y_j) \in \mathcal{H}_2$, $y_j \sim N(\mathbf{x}_j^\top \theta_2, \sigma^2)$. The test of homogeneity between $\mathcal{H}_1$ and $\mathcal{H}_2$ can thus be formally defined as testing whether $\theta_1 = \theta_2$.

Because θ_1 and θ_2 are not observable, the test has to be performed on their estimates, for which maximum likelihood estimator (MLE) is a typical choice. Denote MLE for θ on a dataset $\mathcal{H}$ as $\vartheta = (\mathbf{X}^\top \mathbf{X})^- \mathbf{X}^\top \mathbf{y}$, where $(\cdot)^-$ stands for generalized matrix inverse. A straightforward approach to test homogeneity between $\mathcal{H}_1$ and $\mathcal{H}_2$ is to compare $\|\vartheta_1 - \vartheta_2\|$ against the estimation confidence on ϑ_1 and ϑ_2. The clustering methods by Gentile et al. (2014, 2017) essentially followed this idea. However, theoretical guarantee on the false negative probability of this method only exists when the minimum eigenvalues of $\mathbf{X}_1^\top \mathbf{X}_1$ and $\mathbf{X}_2^\top \mathbf{X}_2$ are lower bounded by a predefined threshold. In other words, when one does not have sufficient observations in either $\mathcal{H}_1$ or $\mathcal{H}_2$, this test will not be effective.

To address this limitation, we choose the test statistic that has been proved to be *uniformly most powerful* for this type of problems (Chow, 1960; Cantrell et al., 1991; Wilson, 1978):

$$s(\mathcal{H}_1, \mathcal{H}_2) = \frac{||\mathbf{X}_1(\vartheta_1 - \vartheta_{1,2})||^2 + ||\mathbf{X}_2(\vartheta_2 - \vartheta_{1,2})||^2}{\sigma^2} \quad (1)$$

where $\vartheta_{1,2}$ denotes the estimator using data from both $\mathcal{H}_1$ and $\mathcal{H}_2$. The knowledge about σ^2 can be relaxed by replacing it with empirical estimate, which leads to Chow test that has an F-distribution (Chow, 1960).

When $s(\mathcal{H}_1, \mathcal{H}_2)$ is above a threshold υ, it suggests the pooled estimator deviates considerably from the individual estimators on two datasets. Thus, we conclude $\theta_1 \neq \theta_2$; otherwise, we conclude $\mathcal{H}_1$ and $\mathcal{H}_2$ are homogeneous. The choice of υ is critical, as it determines the type-I and type-II error probabilities of the test. Upper bounds of these two error probabilities are given below and their proofs are provided in the appendix.

Theorem 3.1 *The test statistic $s(\mathcal{H}_1, \mathcal{H}_2)$ follows a non-central χ^2 distribution $s(\mathcal{H}_1, \mathcal{H}_2) \sim \chi^2(df, \psi)$, where the degree of freedom $df = rank(\mathbf{X}_1) + rank(\mathbf{X}_2) - rank(\begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix})$, and the non-centrality parameter $\psi = \frac{\begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix}^\top \left[\mathbf{I}_{t_1+t_2} - \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}(\mathbf{x}_1^\top \mathbf{x}_1 + \mathbf{x}_2^\top \mathbf{x}_2)^- \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top\end{bmatrix}\right] \begin{bmatrix}\mathbf{X}_1\theta_1 \\ \mathbf{X}_2\theta_2\end{bmatrix}}{\sigma^2}$.*

Lemma 3.2 *When $\theta_1 = \theta_2$, $\psi = 0$; the type-I error probability can be upper bounded by:*

$$P\big(s(\mathcal{H}_1, \mathcal{H}_2) > \upsilon | \theta_1 = \theta_2\big) \leq 1 - F(\upsilon; df, 0),$$

where $F(\upsilon; df, 0)$ denotes the cumulative density function of distribution $\chi^2(df, 0)$ evaluated at υ.

This lemma states that given two datasets $\mathcal{H}_1$ and $\mathcal{H}_2$ (hence the degree-of-freedom *df* is determined), the type-I error probability of this test only depends on the specified threshold υ.

Lemma 3.3 *When $\theta_1 \neq \theta_2$, $\psi \geq 0$; the type-II error probability can be upper bounded by,*

$$P\big(s(\mathcal{H}_1, \mathcal{H}_2) \leq \upsilon | \theta_1 \neq \theta_2\big) \leq \begin{cases} F\Big(\upsilon; d, \psi\Big), & \text{if } \mathbf{X}_1 \text{ and } \mathbf{X}_2 \text{ are full-rank.} \\ F(\upsilon; df, 0), & \text{otherwise.} \end{cases}$$

where $\psi = \frac{||\theta_1 - \theta_2||^2/\sigma^2}{1/\lambda_{min}(\mathbf{X}_1^\top \mathbf{X}_1) + 1/\lambda_{min}(\mathbf{X}_2^\top \mathbf{X}_2)}$.

Compared with the type-I error probability, this lemma shows that the type-II error probability also depends on the ground-truth parameters (θ_1, θ_2) and the variance of noise σ^2.

These error probabilities are the key concerns in our problem: in change detection, they correspond to the early and late detection of change points (Wu et al., 2018); and in cluster identification, they correspond to missing a user model in the neighborhood and placing a wrong user model in the neighborhood (Gentile et al., 2014). Given it is impossible to completely eliminate these two types of errors in a non-deterministic

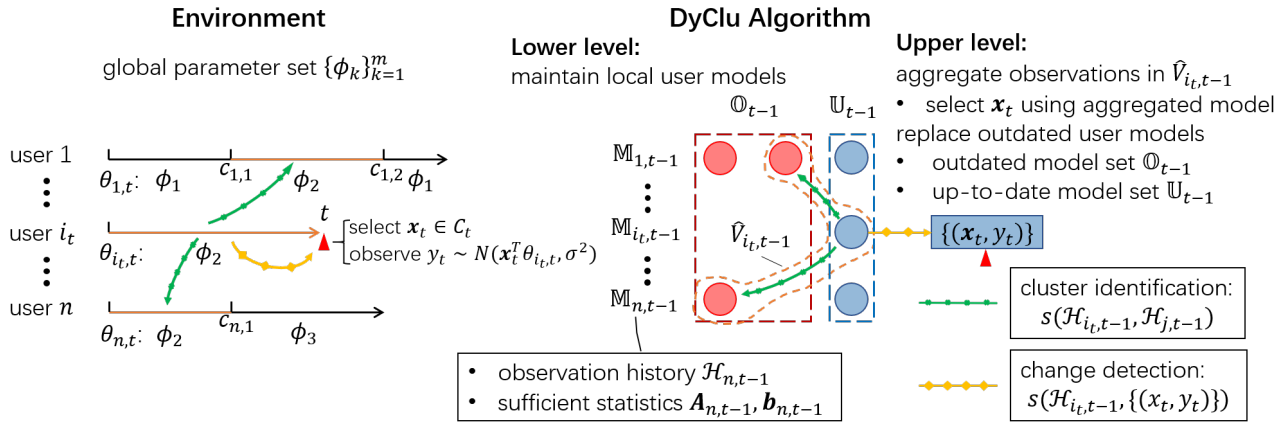

Figure 1: Online bandit learning in a non-stationary and clustered environment. The environment setting is shown on the left side of the figure, where each user's reward mapping function undergoes a piecewise stationary process; and the reward mapping functions are globally shared across users. The proposed DyClu algorithm is illustrated on the right side of the figure. The model has a two-level hierarchy: at the lower level, individual users' bandit models are dynamically maintained; and at the upper level, a unified test of homogeneity is performed for the purpose of change detection and cluster identification among the lower-level user models.

algorithm, the uniformly most powerful property of the test defined in Eq (1) guarantees its sensitivity is optimal at any level of specificity.

3.3 Algorithm

In the environment specified in Section 3.1, the user's reward mapping function is piecewise stationary (e.g., the line segments on each user's interaction trace in Figure 1), which calls for the learner to actively detect changes and re-initialize the estimator to avoid distortion from outdated observations (Yu and Mannor, 2009; Cao et al., 2019; Besson and Kaufmann, 2019; Wu et al., 2018). A limitation of these methods is that they do not attempt to reuse outdated observations because they implicitly assume each stationary period has an unique parameter. Our setting relaxes this by allowing existence of identical reward mappings across users and time (e.g., the orange line segments in Figure 1), which urges the learner to take advantage of this situation by identifying and aggregating observations with the same parameter to obtain a more accurate reward estimation.

Since neither the change points nor the grouping structure is known, in order to reuse past observations while avoiding distortion, the learner needs to accurately detect change points, stores observations in the interval between two consecutive detections together, and then correctly identify intervals with the same parameter as the current one. In this paper, we propose to unify these two operations using the test in Section 3.2, which leads to our algorithm Dynamic Clustering of Bandits, or DyClu in short. DyClu forms a two-level hierarchy as shown in Figure 1: at the lower level, it stores observations in each interval and their sufficient statistics in a user model; at the upper level, it detects change in user's reward function to decide when to create new user models and clusters individual user models for arm selection. Detailed steps of DyClu are explained in Algorithm 1.

The lower level of DyClu manages observations associated with each user $i \in \mathcal{U}$ in user models, denoted by $\mathbb{M}_{i,t}$. Each user model $\mathbb{M}_{i,t} = (\mathbf{A}_{i,t}, \mathbf{b}_{i,t}, \mathcal{H}_{i,t})$ stores:

1. $\mathcal{H}_{i,t}$: a set of observations associated with user i since the initialization of $\mathbb{M}_{i,t}$ up to time t, where each element is a context vector and reward pair $(\mathbf{x}_k, y_k)$.
2. Sufficient statistics: $\mathbf{A}_{i,t} = \sum_{(\mathbf{x}_k,\cdot)\in\mathcal{H}_{i,t}} \mathbf{x}_k\mathbf{x}_k^\top$ and $\mathbf{b}_{i,t} = \sum_{(\mathbf{x}_k,y_k)\in\mathcal{H}_{i,t}} \mathbf{x}_k y_k$.

Every time DyClu detects change in a user's reward mapping function, a new user model is created to replace the previous one (line 15 in Algorithm 1). We refer to the replaced user models as outdated models and the others up-to-date ones. We denote the set of all outdated user models at time t as $\mathbb{O}_t$ and the up-to-date ones as $\mathbb{U}_t$. In Figure 1, the row of circles next to $\mathbb{M}_{1,t-1}$ represents all the user models for user 1, red ones denote outdated models and the blue one denotes up-to-date model.

The upper level of DyClu is responsible for managing the user models via change detection and model clustering. It replaces outdated models in each user

Algorithm 1 Dynamic Clustering of Bandits (DyClu)

1: **Input:** sliding window size τ, $\delta, \delta_e \in (0,1)$, threshold for change detection and neighbor identification υ^e and υ^c, and regularization parameter λ
2: **Initialization:** for each user model $\mathbb{M}_{i,0}, \forall i \in \mathcal{U}$: $\mathbf{A}_{i,0} = \mathbf{0} \in \mathbb{R}^{d\times d}$, $\mathbf{b}_{i,0} = \mathbf{0} \in \mathbb{R}^d$, $\mathcal{H}_{i,0} = \emptyset$, $\hat{e}_{i,0} = 0$; the set of outdated user models $\mathbb{O}_0 = \emptyset$, and up-to-date user models $\mathbb{U}_0 = \{\mathbb{M}_{i,0}\}_{i\in\mathcal{U}}$
3: **for** $t = 1, 2, ..., T$ **do**
4: Observe user $i_t \in \mathcal{U}$, and set of available arms $C_t = \{x_{t,1}, ..., x_{t,K}\}$
5: Choose arm $\mathbf{x}_t \in C_t$ by Eq 2:

$$\arg\max_{\mathbf{x}\in C_t} \mathbf{x}^\top \hat{\theta}_{\hat{V}_{i_t,t-1}} + CB_{\hat{V}_{i_t,t-1}}(x)$$

6: Observe reward y_t from user i_t
7: Compute $e_{i_t,t} = \mathbf{1}\left\{S(\mathcal{H}_{i_t,t-1}, (\mathbf{x}_t^\top, y_t)) > \upsilon^e\right\}$
8: Update $\hat{e}_{i_t,t} = \sum_{\tilde{t}_{i_t}(\tau)<j\le t: i_j = i_t} e_{i_t,j}$
9: **if** $\hat{e}_{i_t,t} \le 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}$ **then**
10: **if** $e_{i_t,t} = 0$ **then**
11: $\mathbb{M}_{i_t,t}$: $\mathcal{H}_{i_t,t} = \mathcal{H}_{i_t,t-1} \cup (\mathbf{x}_t, y_t)$, $\mathbf{A}_{i_t,t} = \mathbf{A}_{i_t,t-1} + \mathbf{x}_t\mathbf{x}_t^\top$, $\mathbf{b}_{i_t,t} = \mathbf{b}_{i_t,t-1} + \mathbf{x}_t y_t$
12: **end if**
13: **else**
14: $\mathbb{O}_t = \mathbb{O}_{t-1} \cup \mathbb{M}_{i_t,t-1}$, $\hat{e}_{i_t,t} = 0$
15: Replace $\mathbb{M}_{i_t,t-1}$ with $\mathbb{M}_{i_t,t} = (A_{i_t,t} = \mathbf{0}, b_{i_t,t} = \mathbf{0}, \mathcal{H}_{i,t} = \emptyset)$ in $\mathbb{U}_t$
16: **end if**
17: Compute $\hat{V}_{i_t,t} = \{\mathbb{M} \in \mathbb{U}_t \cup \mathbb{O}_t : S(\mathcal{H}_{i_t,t}, \mathcal{H}) \le \upsilon^c\}$ and update $\hat{V}_{i,t}$ for $i \ne i_t$ accordingly.
18: **end for**

and aggregates models across users and time for arm selection.

• **Change detection.** A one-sample homogeneity test is used to construct a test variable $e_{i_t,t} = \mathbf{1}\{s(\mathcal{H}_{i_t,t-1}, \{(\mathbf{x}_t, y_t)\}) > \upsilon^e\}$ to measure whether the user model $\mathbb{M}_{i_t,t-1}$ is 'admissible' to the new observation $(\mathbf{x}_t, y_t)$. υ^e is a chosen threshold for change detection. To make more reliable change detection, we use the empirical mean of $e_{i_t,t}$ in a sliding window of size $\min(|\mathcal{H}_{i_t,t-1}|, \tau)$ as the test statistic, denoted as $\hat{e}_{i_t,t} = \frac{1}{\min(|\mathcal{H}_{i_t,t-1}|,\tau)} \sum_k e_{i_t,k}$. Lemma 3.4 specifies the upper bound of early detection probability using $\hat{e}_{i,t}$, which is used for selecting threshold for it.

Lemma 3.4 *From Lemma 3.2, type-1 error probability $P(e_{i,t} = 1) = 1 - F(\upsilon^e; 1, 0)$, and thus $\mathbb{E}[e_{i,t}] = 1 - F(\upsilon^e; 1, 0)$. Applying Hoeffding inequality gives,*

$$P\Big(\hat{e}_{i,t} > 1 - F(\upsilon^e; 1, 0) + \sqrt{\frac{\log 1/\delta_e}{2\tau}}\Big) \le \delta_e$$

At any time step t, DyClu only updates $\mathbb{M}_{i_t,t-1}$ when $e_{i_t,t} = 0$ (line 10-12 in Algorithm 1). This guarantees that if the underlying reward distribution has changed, with a high probability we have $e_{i_t,t} = 1$, and thus the user model $\mathbb{M}_{i_t,t-1}$ will not be updated. This prevents any distortion in $\mathcal{H}_{i_t,t}$ by observations from different reward distributions.

When $\hat{e}_{i_t,t}$ exceeds the threshold specified by Lemma 3.4, DyClu will inform the lower level to move $\mathbb{M}_{i_t,t-1}$ to the outdated model set $\mathbb{O}_t = \mathbb{O}_{t-1} \cup \{\mathbb{M}_{i_t,t-1}\}$; and then create a new model $\mathbb{M}_{i_t,t} = (A_{i_t,t} = \mathbf{0}, b_{i_t,t} = \mathbf{0}, \mathcal{H}_{i,t} = \emptyset)$ for user i_t as shown in line 13-16 in Algorithm 1.

• **Clustering of user models.** In this step, DyClu finds the set of "neighborhood" user models $\hat{V}_{i_t,t}$ of current user model $\mathbb{M}_{i_t}, t$, where $\hat{V}_{i_t,t-1} = \{\mathbb{M} = (\mathbf{A}, \mathbf{b}, \mathcal{H}) \in \mathbb{U}_t \cup \mathbb{O}_t : s(\mathcal{H}_{i_t,t}, \mathcal{H}) \le \upsilon^c\}$. Basically, DyClu executes homogeneity test between $\mathbb{M}_{i_t,t}$ and all other user models $\mathbb{M} \in \mathbb{U}_t \cup \mathbb{O}_t$ (both outdated and up-to-date) with a given threshold υ^c (line 17 in Algorithm 1). Lemma 3.2 and 3.3 again specify error probabilities of each decision.

When selecting an arm for user i_t at time t, DyClu aggregates the sufficient statistics of user models in neighborhood $\hat{V}_{i_t,t-1}$. Then it adopts the popular UCB strategy by Auer (2002); Li et al. (2010) to balance exploitation and exploration. Specifically, DyClu selects arm $\mathbf{x}_t$ that maximizes the UCB score computed by aggregated sufficient statistics as follows (line 5 in Algorithm 1),

$$\mathbf{x}_t = \arg\max_{\mathbf{x}\in C_t} \mathbf{x}^\top \hat{\theta}_{\hat{V}_{i_t,t-1}} + CB_{\hat{V}_{i_t,t-1}}(\mathbf{x}) \qquad (2)$$

In Eq (2), $\hat{\theta}_{\hat{V}_{i_t,t-1}} = \mathbf{A}^{-1}_{\hat{V}_{i_t,t-1}} \mathbf{b}_{\hat{V}_{i_t,t-1}}$ is the ridge regression estimator using aggregated statistics $\mathbf{A}_{\hat{V}_{i_t,t-1}} = \lambda \mathbf{I}_d + \sum_{(\mathbf{A}_j, \mathbf{b}_j, \mathcal{H}_j)\in\hat{V}_{i_t,t-1}} \mathbf{A}_j$ and $\mathbf{b}_{\hat{V}_{i_t,t-1}} = \sum_{(\mathbf{A}_j, \mathbf{b}_j, \mathcal{H}_j)\in\hat{V}_{i_t,t-1}} \mathbf{b}_j$; the confidence bound of reward estimation for arm $\mathbf{x}$ is $CB_{\hat{V}_{i_t,t-1}}(\mathbf{x}) = \alpha_{\hat{V}_{i_t,t-1}} \sqrt{\mathbf{x}^\top \mathbf{A}^{-1}_{\hat{V}_{i_t,t-1}} \mathbf{x}}$, where $\alpha_{\hat{V}_{i_t,t-1}} = \sigma\sqrt{d \log\left(1 + \frac{\sum_{(A_j, b_j, \mathcal{H}_j)\in\hat{V}_{i_t,t-1}} |\mathcal{H}_j|}{d\lambda}\right) + 2\log\frac{1}{\delta}} + \sqrt{\lambda}$.

3.4 Regret analysis

Denote $R_T = \sum_{t=1}^T \theta_{i_t}^\top \mathbf{x}_t^* - \theta_{i_t}^\top \mathbf{x}_t$ as the accumulative regret, where $\mathbf{x}_t^* = \arg\max_{x_{t,j}\in C_t} \theta_{i_t}^\top \mathbf{x}_{t,j}$ is the optimal arm at time t. Our regret analysis relies on the high probability results by Abbasi-Yadkori et al. (2011) and decomposition of "good" and "bad" events according to change detection and clustering results. The full proof, along with ancillary results and discussions, are given in the appendix.

Theorem 3.5 *Under Assumptions 1, 2 and 3, the regret of DyClu is upper bounded by:*

$$R_T = O\Big(\sigma d\sqrt{T \log^2 T}\Big(\sum_{k=1}^m \sqrt{p_k}\Big) + \sum_{i\in\mathcal{U}} \Gamma_i(T) \cdot C\Big)$$

where $C = \frac{1}{1-\delta^e} + \frac{\sigma^2}{\gamma^2\lambda'^2}\log\frac{d}{\delta'}$, *with a probability at least* $(1-\delta)(1-\frac{\delta_e}{1-\delta_e})(1-\delta^{'})$.

Note that the first term matches the regret of the ideal case that the learner knows the exact change points and clustering structure of each user and time step, while the second term corresponds to the additional regret due to the interplay between errors in change detection and clustering, which is unique to our problem. To better understand this result, we discuss in the following paragraph how it compares with state-of-the-art bandit solutions in settings like non-stationary environment only or clustered environment only.

Case 1: Setting $m = 1$, $n = 1$ and $\Gamma_1(T) = 1$ reduces the problem to the basic linear bandit setting, because the environment consists of only one user with a stationary reward distribution for the entire time of interaction. With only one user who has a stationary reward distribution, we have $\sum_{k=1}^{1}\sqrt{p_k} = 1$ where p_k is frequency of occurrences of ϕ_k in T as defined in Section 3.1. In addition, since there is only one stationary period, the added regret caused by late detection does not exist; and the added regret due to the failure in clustering can be bounded by a constant, which only depends on environment variables (see Lemma **??** in appendix for details). The upper regret bound of DyClu then becomes $O\big(\sigma d\sqrt{T\log^2 T}\big)$, which achieves the same order of regret as that in LinUCB (Abbasi-Yadkori et al., 2011). **Case 2**: Setting $\Gamma_i(T) = 1, \forall i \in \mathcal{U}$ reduces the problem to the clustered bandit setting (Gentile et al., 2014), because all users in the environment have a stationary reward distribution of their own. Similar to Case 1, the added regret caused by late detection becomes zero and the added regret due to the failure in clustering is bounded by a constant, which leads to the upper regret bound of $O\big(\sigma d\sqrt{T\log^2 T}(\sum_{k=1}^{m}\sqrt{p_k})\big)$. DyClu achieves the same order of regret as that in CLUB (Gentile et al., 2014). **Case 3**: Setting $n = 1$ reduces the problem to a piecewise stationary bandit setting, because the environment consists of only one user with piecewise stationary reward distributions. For the convenience of comparison, we can rewrite the upper regret bound of DyClu in the form of $O\big(\sum_{k\in[m]} R_{Lin}(|N_k^{\phi}(T)|) + \Gamma_1(T)\big)$, where $R_{Lin}(t) = O\Big(d\sqrt{t\log^2 t}\Big)$ (Abbasi-Yadkori et al., 2011) and $N_k^{\phi}(T) = \Big\{1 \le t^{'} \le T : \theta_{i_{t'},t'} = \phi_k\Big\}$ is the set of time steps up to time T when the user being served has the bandit parameter equal to ϕ_k. Detailed derivation of this is given in appendix (Section **??**). Note that the upper regret bound of dLinUCB (Wu et al., 2018) for this setting is $O\big(\Gamma_1(T)R_{Lin}(S_{max}) + \Gamma_1(T)\big)$, where S_{max} denotes the maximum length of stationary periods. The regret of DyClu depends on the number of unique bandit parameters in the environment, instead of the number of stationary periods as in dLinUCB, because DyClu can reuse observations from previous stationary periods. This suggests DyClu has a tighter regret bound if different stationary periods share the same unique bandit parameters; for example, in situations where a future reward mapping function switches back to a previous one.

4 Experiments

We investigate the empirical performance of DyClu by comparing with a list of state-of-the-art baselines for both non-stationary bandits and clustered bandits on synthetic and real-world recommendation datasets.

4.1 Experiment setup and baselines

• **Synthetic dataset.** We create a set of unique bandit parameters $\{\phi_k\}_{k=1}^{m}$ and arm pool $\{\mathbf{x}_j\}_{j=1}^{K}$ ($K = 1000$), where ϕ_k and $\mathbf{x}_j$ are first sampled from $N(\mathbf{0}_d, \mathbf{I}_d)$ with $d = 25$ and then normalized so that $\forall k, j, \|\phi_k\| = 1$ and $\|\mathbf{x}_j\| = 1$. When sampling $\{\phi_k\}_{k=1}^{m}$, the separation margin γ is set to 0.9 and enforced via rejection sampling. n users are simulated. In each user, we sample a series of time intervals from (S_{min}, S_{max}) uniformly; and for each time interval, we sample a unique parameter from $\{\phi_k\}_{k=1}^{m}$ as the ground-truth bandit parameter for this period. This creates asynchronous changes and clustering structure in users' reward functions. The users are served in a round-robin fashion. At time step $t = 1, 2, \ldots, T$, a subset of arms are randomly chosen and disclosed to the learner. Reward of the selected arm is generated by the linear function governed by the corresponding bandit parameter and context vector, with additional Gaussian noise sampled from $N(0, \sigma^2)$.

• **LastFM dataset.** The LastFM dataset is extracted from the music streaming service Last.fm (Cesa-Bianchi et al., 2013), which contains 1892 users and 17632 items (artists). "Listened artists" of each user are treated as positive feedback. We followed Wu et al. (2018) to preprocess the dataset and simulate a clustered non-stationary environment by creating 20 "hybrid users". We first discard users with less than 800 observations and then use PCA to reduce the dimension of TF-IDF feature vector to $d = 25$. We create hybrid users by sampling three real users uniformly and then concatenating their associated data points together. Hence, data points of the same real user would appear in different hybrid users, which is analogous to stationary periods that share the same unique bandit parameters across different users and time.

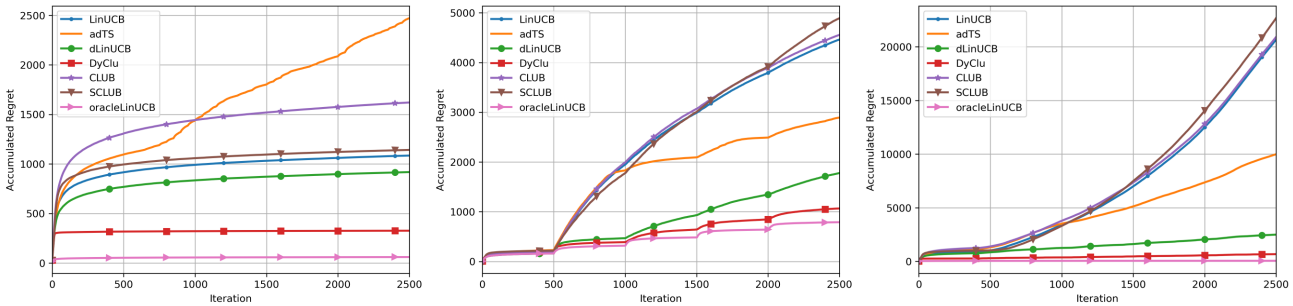

Figure 2: Accumulated regret on synthetic datasets with three different environment settings. Environment 1: $n = 100$ users share a global set of $m = 5$ unique bandit parameters, and each user remains stationary all the time. Environment 2: $n = 20$ user with fixed stationary period length 500; each period sample a unique bandit parameter. Environment 3: $n = 100$ users share a global set of $m = 5$ unique bandit parameters, and each user changes in a asynchronous manner.

• **Baselines.** We compare DyClu with a set of state-of-the-art bandit algorithms: linear bandit LinUCB by Abbasi-Yadkori et al. (2011), non-stationary bandit dLinUCB by Wu et al. (2018) and adTS by Slivkins and Upfal (2008), as well as clustered bandit CLUB by Gentile et al. (2014) and SCLUB by Li et al. (2019). For experiments on synthetic dataset, we also include oracle-LinUCB for comparison, which runs an instance of LinUCB for each unique bandit parameter. Comparing with it helps us understand the added regret due to errors in change detection and clustering.

• **Hyper-parameters.** We set the same regularization parameter $\lambda = 0.1$ for all the algorithms, and set the same sliding window size $\tau = 20$ for both dLinUCB and DyClu on synthetic dataset and $\tau = 50$ on LastFM dataset. The thresholds υ^e and υ^c for DyClu are essentially the upper-tail critical values of chi-square distributions $\chi^2(1)$ and $\chi^2(d)$, which directly control the type-I error probability for change detection and clustering, i.e. $1 - F(\upsilon^e; 1)$ and $1 - F(\upsilon^c; d)$ respectively. Their values affect the second term in the regret upper bound given in Theorem 3.5 (see Lemma D.1 and Lemma D.2 in appendix for details). In all our experiments, υ^e and υ^c are selected such that the corresponding significance level equals to 0.01, e.g., to make $F(\upsilon^c; 25) = 0.01$, we set $\upsilon^c = 44.314$.

4.2 Experiment results

• **Empirical comparisons on synthetic dataset.** We compare accumulated regret of all bandit algorithms under three environment settings, and the results are reported in Figure 2. Environment 1 simulates the clustered bandit setting in Gentile et al. (2014), where *no* change in the reward function is introduced. DyClu outperformed other baselines, including CLUB and SCLUB, demonstrating the quality of its identified clustering structure. Specifically, compared with adTS that incurs high regret as a result of too many false detections, the change detection in DyClu has much less false positives, as there is no change in each user's reward distribution. Environment 2 simulates the piecewise stationary bandit setting in Wu et al. (2018). Algorithms designed for stationary environment, e.g., CLUB, SCLUB, and LinUCB suffer from a linear regret after the first change point. DyClu achieved the best performance, with a wide margin from the second best, dLinUCB, which is designed for this environment. It shows the power of our change detection method against dLinUCB's. Environment 3 combines previous two settings with both non-stationarity and clustering structure. DyClu again outperformed others. It is worth noting that regret of all algorithms increased compared with Environment 1 due to the nonstationarity, but the increase in DyClu is the smallest. And in all settings, DyClu's performance is closest to the oracle-LinUCB's, which shows that DyClu can correctly cluster and aggregate observations from the dynamically changing users.

• **Sensitivity to environment settings.** According to our regret analysis, the performance of DyClu depends on environment parameters like the number of unique bandit parameters m, the number of stationary periods $\Gamma_i(T)$ for $i \in \mathcal{U}$, and variance of Gaussian noise σ^2. We investigate their influence on DyClu and baselines, by varying these parameters while keeping the others fixed. The accumulated regret under different settings are reported in Table 1. DyClu outperformed other baselines in all 9 different settings, and the changes of its regret align with our theoretical analysis. A larger number of unique parameters m leads to higher regret of DyClu as shown in setting 1, 2 and 3, since observations are split into more clusters with smaller size each. In addition, larger number of stationary periods incurs more errors in change detec-

Table 1: Comparison of accumulated regret under different environment settings.

	n	m	S_{min}	S_{max}	T	σ	oracle.	LinU.	adTS	dLinU.	CLUB	SCLUB	DyClu
1	100	10	400	2500	2500	0.09	115	19954	9872	2432	20274	19989	853
2	100	50	400	2500	2500	0.09	489	20952	9563	2420	21205	21573	1363
3	100	100	400	2500	2500	0.09	873	21950	10961	2549	22280	22262	1958
4	100	10	200	400	2500	0.09	112	39249	36301	10831	39436	43836	3025
5	100	10	800	1000	2500	0.09	113	34385	13788	3265	34441	33514	1139
6	100	10	1200	1400	2500	0.09	112	24769	8124	2144	24980	23437	778
7	100	10	400	2500	2500	0.12	166	22453	10567	3301	22756	22683	1140
8	100	10	400	2500	2500	0.15	232	19082	10000	5872	19427	20664	1487
9	100	10	400	2500	2500	0.18	307	23918	11255	9848	24050	23677	1956

tion, leading to an increased regret. This is confirmed by results in setting 4, 5 and 6. Lastly, as shown in setting 7, 8 and 9, larger Gaussian noise leads to higher regret, as it slows down convergence of reward estimation and change detection.

• **Empirical comparisons on LastFM.** We report normalized accumulative reward (ratio between baselines and uniformly random arm selection strategy (Wu et al., 2019)) on LastFM in Figure 3. In this environment, realizing both non-stationarity and clustering structure is important for an online learning algorithm to perform well. DyClu's improvement over other baselines confirms its quality in partitioning and aggregating relevant data points across users. The advantage of DyClu is more apparent at the early stage of learning, where each local user model has not collected sufficient amount of observations for individualized reward estimation; and thus change detection and clustering are more difficult there.

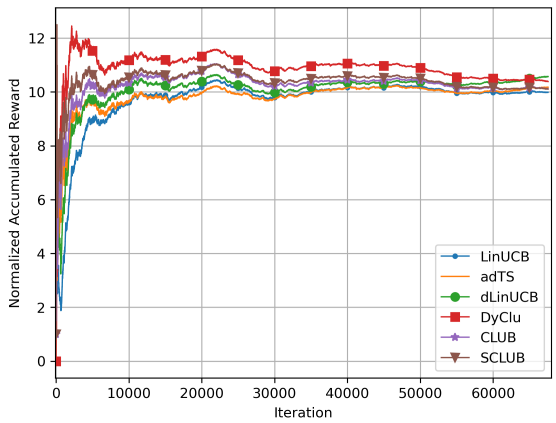

Figure 3: Comparison of accumulated reward normalized by a random policy on LastFM dataset.

5 Conclusion

In this work, we unify the efforts in non-stationary bandits and clustered bandits via homogeneity test, and also propose a new bandit problem setting that generalizes both. Our solution adaptively detects changes in the underlying reward distribution and clusters bandit models for aggregated arm selection. The resulting upper regret bound matches with the ideal algorithm's only up to a constant; and extensive empirical evaluations validate its effectiveness in a non-stationary and clustered environment.

There are still several directions left open in this research: 1) Assumption 1 and 2 are mainly needed for controlling the type-II error of change detection and clustering, to ensure the estimator used for arm selection have no heterogeneous observations (i.e., contamination). These assumptions are arguably rigid, considering if the difference in bandit parameters is small or negligible, aggregating such heterogeneous observations may not be detrimental. To the best of our knowledge, no existing work for linear bandit addressed this issue (Gentile et al., 2014; Li et al., 2019; Wu et al., 2018). Pursuing this direction requires balancing the quantity vs. quality trade-off of contaminated observations. 2) Our current analysis for the test statistic assumes Gaussian reward noise, and it would be interesting to consider the more general sub-Gaussian noise that is commonly assumed in bandit literature. 3) Despite the existence of multiple users, all computations are done in a centralized manner; to make it more practical, asynchronous and distributed model update is more desired.

Acknowledgements

We thank Huazheng Wang and Yiling Jia for their helpful suggestions, and the anonymous reviewers for their insightful and constructive comments. This work is supported by by National Science Foundation under grant IIS-1553568, IIS-1838615 and IIS-1904183.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 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.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 529–538. ACM, 2016.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *International Conference on Machine Learning*, pages 136–144, 2014.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

Qingyun Wu, Naveen Iyer, and Hongning Wang. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 495–504. ACM, 2018.

Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the brownian restless bandits. In *COLT*, pages 343–354, 2008.

Yang Cao, Wen Zheng, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure for piecewise-stationary bandit: a change-point detection approach. *AISTATS,(Okinawa, Japan)*, 2019.

Lilian Besson and Emilie Kaufmann. The generalized likelihood ratio test meets klucb: an improved algorithm for piece-wise non-stationary bandits. *arXiv preprint arXiv:1902.01575*, 2019.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, pages 12017–12026, 2019.

Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal, and parameter-free. *arXiv preprint arXiv:1902.00980*, 2019.

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765, 2014.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548. ACM, 2016.

Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1253–1262. JMLR. org, 2017.

Shuai Li, Wei Chen, and Kwong-Sak Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

Jia Yuan Yu and Shie Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1177–1184. ACM, 2009.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, ALT'11, pages 174–188, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642244117.

Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michéle Sebag. Multi-armed bandit, dynamic environments and meta-bandits. 2006.

Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

David Siegmund. *Sequential analysis: tests and confidence intervals*. Springer Science & Business Media, 2013.

Othmane Mazhar, Cristian Rojas, Carlo Fischione, and Mohammad Reza Hesamzadeh. Bayesian model selection for change point detection and clustering. In *International Conference on Machine Learning*, pages 3433–3442. PMLR, 2018.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration–exploitation in a multi-armed bandit problem with non-stationary rewards. *Stochastic Systems*, 9(4):319–337, 2019.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1079–1087, 2019.

Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. 1985.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158, 2019.

Negar Hariri, Bamshad Mobasher, and Robin Burke. Adapting to user preference changes in interactive recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. A simple approach for non-stationary linear bandits. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 2020, 2020.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

Haipeng Luo, Chen-Yu Wei, Alekh Agarwal, and John Langford. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*, pages 1739–1776, 2018.

Shuai Li. *The art of clustering bandits.* PhD thesis, Università degli Studi dell'Insubria, 2016.

Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Multi-armed bandits in the presence of side observations in social networks. In *52nd IEEE Conference on Decision and Control*, pages 7309–7314. IEEE, 2013.

Kaige Yang, Laura Toni, and Xiaowen Dong. Laplacian-regularized graph bandits: Algorithms and theoretical analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 3133–3143, 2020.

Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, pages 591–605, 1960.

R Stephen Cantrell, Peter M Burrows, and Quang H Vuong. Interpretation and use of generalized chow tests. *International Economic Review*, pages 725–741, 1991.

AL Wilson. When is the chow test ump? *The American Statistician*, 32(2):66–68, 1978.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, pages 2080–2090, 2019.