When and Whom to Collaborate with in a Changing Environment: A Collaborative Dynamic Bandit Solution

Chuanhao Li, Qingyun Wu, Hongning Wang
Department of Computer Science, University of Virginia
{cl5ev,qw2ky,hw5x}@virginia.edu

ABSTRACT

Collaborative bandit learning, i.e., bandit algorithms that utilize collaborative filtering techniques to improve sample efficiency in online interactive recommendation, has attracted much research attention as it enjoys the best of both worlds. However, all existing collaborative bandit learning solutions impose a stationary assumption about the environment, i.e., both user preferences and the dependency among users are assumed static over time. Unfortunately, this assumption hardly holds in practice due to users' ever-changing interests and dependency relations, which inevitably costs a recommender system sub-optimal performance in practice.

In this work, we develop a collaborative dynamic bandit solution to handle a changing environment for recommendation. We explicitly model the underlying changes in both user preferences and their dependency relation as a stochastic process. Individual user's preference is modeled by a mixture of globally shared contextual bandit models with a Dirichlet process prior. Collaboration among users is thus achieved via Bayesian inference over the global bandit models. To balance exploitation and exploration during the interactions, Thompson sampling is used for both model selection and arm selection. Our solution is proved to maintain a standard $\tilde{O}(\sqrt{T})$ Bayesian regret in this challenging environment. Extensive empirical evaluations on both synthetic and real-world datasets further confirmed the necessity of modeling a changing environment and our algorithm's practical advantages against several state-of-the-art online learning solutions.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Theory of computation** → **Online learning algorithms**; **Regret bounds**; • **Computing methodologies** → **Sequential decision making**.

KEYWORDS

non-stationary bandits, thompson sampling, bayesian non-parametric model, recommender systems

ACM Reference Format:
Chuanhao Li, Qingyun Wu, Hongning Wang. 2021. When and Whom to Collaborate with in a Changing Environment: A Collaborative Dynamic Bandit Solution. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462852

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462852

1 INTRODUCTION

Personalized recommendation is an essential component in most modern information service systems, as it helps alleviate information overload by tailoring the delivered content at a per-user basis [8, 36]. However, the content universe for most web services is usually large and undergoes frequent changes, which renders traditional methods, like collaborative filtering [25, 39] inappropriate, due to their offline training and online testing paradigm. Under this situation, the system needs to adaptively balance between the need of focusing on items that previously raised users' interest and the need of exploring new items for improving users' satisfaction in a long run. This exploration-exploitation dilemma is exemplified in a multi-armed bandit (MAB) problem [6], and classical algorithms like upper confidence bound [6, 27] and Thompson sampling [2, 4] have been proved to be optimal in striking a balance between these two conflicting needs. Therefore, bandit algorithms have become a reference solution to address this challenge. In particular, contextual bandit [27], an extension of MAB that incorporates contextual information, has been widely adopted in practice.

Moreover, as correlation between user preferences is common in many applications and contextual bandit cannot directly utilize it, various follow-up works seek to combine bandit algorithms with collaborative filtering in order to further improve sample efficiency via collaboration among user. For example, [18, 29] performed online clustering of users in a bandit learning setting, and [17, 31] further considered context/arm-dependent clustering of users. In [24, 41] online matrix factorization is studied with bandit feedback. When social relation among users is available, such as social networks, the inferred user dependency is introduced as structured regularization for user-specific bandit model learning [9, 43, 45].

We should note that all these collaborative bandit learning solutions impose a stationary assumption about the environment: both the user preferences and the dependency between users are static over time, which is a fundamental assumption in multi-armed bandit algorithms [1, 6]. This unfortunately is often violated in real-world situations where users' preferences may change dramatically over time due to various internal or external factors [19, 42], which in turn lead to shifts in user dependencies [40]. In some situations, non-stationarity may be alleviated to some extent by including contextual features describing external factors like season, topic and location, though it is usually difficult, if not impossible, to define such features ahead of time. But this does not work when the non-stationarity is caused by internal factors of the users, which

makes it necessary to design bandit algorithms that can adapt to such change in user preference. There have been numerous solutions proposed to address this challenge for MAB and contextual bandit problems [16, 32, 42], e.g., by detecting the change in user preference and then restarting the algorithm accordingly. However, despite being a natural extension enjoying the best of both worlds, the more challenging problem of collaborative bandit learning in a changing environment still remains open to the best of our knowledge. In this case, both user preferences and their dependency relation are dynamic, giving rise to new challenges in arm selection, user clustering, and change detection.

In this work, to address the aforementioned challenges in this new problem, we propose a bandit algorithm that enables collaborative model learning across users, while adapting to the changes in user preferences and user dependency. Specifically, motivated by the social psychology theories about social norms [15] that humans tend to form groups with others of similar minds and ability, we explicitly model the underlying changes in *both* user preferences and their dependency relation with a non-parametric stochastic process. Our solution does not assume an explicit network of users. Instead, we assume users share preference models in accordance of their underlying interest and dependency with others; and they switch models when their interest or received influence changes. To enable online learning of user preferences, we model the shared preference models as linear bandits. Collaboration among users is thus achieved via Bayesian inference over the globally shared models. To balance exploitation and exploration during the interactions, we use Thompson sampling [2, 4], with the main difference that our solution first samples at model level for each user before sampling at arm level, in order to efficiently explore if any existing global bandit model suits this user or a new bandit model needs to be created. Our solution maintains a standard $\tilde{O}(\sqrt{T})$ Bayesian regret in this challenging environment. Extensive empirical evaluations on both synthetic and real-world datasets for content recommendation confirmed the necessity of modeling a changing environment and our algorithm's practical advantages against several state-of-the-art online collaborative learning solutions.

2 RELATED WORK

Collaborative recommendation, including both traditional offline learning solutions such as collaborative filtering [25, 39], and interactive online learning solutions, such as collaborative bandit learning [9, 18, 41, 43], has shown great promise in personalized recommendation tasks. In particular, collaborative bandit learning, due to its ability of adapting to real-time user feedback, has received increasing attention in both industry and academia. Among them, there are several representative classes of solutions in modeling user dependency for collaborative recommendation. In the first type of solutions, when users' social relations are known (e.g., social network), the inferred dependency among users is encoded as a regularization for user-specific bandit model learning [9, 43, 45]. In the second type of solutions, where explicit user network is not assumed, the bandit parameters are estimated together with the dependency relation among users [17, 18, 29]. Typically, they cluster the user-specific bandit models via the learned model parameters during online updating. The third type of solutions appeal to latent factor models to capture the correlation between users and items in a lower dimensional space and estimate the latent factors with bandit feedback [24, 41]. We should note almost all existing collaborative learning solutions impose a stationary assumption about the environment, in which both user preferences and dependency are assumed to be static.

Non-stationarity appears in many real-world recommendation applications [34, 35], and has shown to cost stationary recommendation algorithms sub-optimal performance [42]. In standard bandit learning settings, a number of solutions have been proposed to deal with non-stationarity for multi-armed bandit [7, 16], contextual multi-armed bandit [11, 32], and contextual linear bandit [12, 37, 42, 47]. The main focus of these solutions is to eliminate the distortion from out-dated observations, which follow a different reward distribution than that of the current environment. To achieve this goal, common strategies include exponentially decaying the effect of past observations [37], discard past observations outside of a sliding window [12, 16], or adopt a change detector to actively detect the change point [42, 46] and then re-initialize the model.

However, these aforementioned solutions are not appropriate for collaborative recommendation in a non-stationary environment. First, none of the existing non-stationary bandit learning solutions model the possible dependency among users. This costs them the opportunity of leveraging the dependency among users to improve model estimation. Second, in online collaborative learning, not only individual users' preferences, but also the dependency among them, are subjected to unknown changes. Both factors have to be modeled for effective change detection and personalized recommendation. In addition, these solutions are not sample efficient in the sense that they simply discard outdated models and observations, without reusing or sharing them with other users to improve model estimation at current time.

3 METHODOLOGY

In this section, we first introduce how to perform personalized interactive recommendation with contextual bandits in a stationary environment, which is the building block of our proposed collaborative dynamic bandit solution. Then we describe our non-parametric stochastic process model for modeling the dynamics in user preferences and dependency in a non-stationary environment. Finally, we provide the details about the proposed collaborative dynamic bandit algorithm and the corresponding theoretical regret analysis.

3.1 Contextual bandit for interactive recommendation

In online interactive recommendation, the system sequentially chooses among a set of candidate items based on users' immediate feedback, such as click, ratings or dwell time [27, 44], in order to maximize the accumulated positive feedback in a finite period of time. This can be formulated as a contextual bandit problem [4, 27], where each candidate arm is associated with a d-dimensional vector x referred to as the context (assume $\|\mathrm{x}\|_2 \leq 1$ without loss of generality). Denote the candidate pool as $\mathcal{A}_t = \{\mathrm{x}_{t,1}, \mathrm{x}_{t,2}, \ldots, \mathrm{x}_{t,|\mathcal{A}_t|}\}$, which can be time-varying. The corresponding reward r_t is governed by the context vector x_t of the selected arm and an underlying

fixed but unknown bandit parameter $\boldsymbol{\theta}$ (assume $\|\boldsymbol{\theta}\|_2 \leq 1$). In practice, a recommender system maintains one bandit model $\boldsymbol{\theta}_u$ for each user u for personalization [9, 27, 43].

Thompson Sampling (TS) [2, 4] is a classic and popular bandit solution, which has been widely adopted in many real-world problems due to its flexibility and encouraging empirical performance. In TS, one needs to specify the prior distribution of the unknown bandit parameter $P(\boldsymbol{\theta}_u)$ and the likelihood function of the reward $P(r_i|\mathrm{x}_i, \boldsymbol{\theta}_u)$. Then with the set of observations $\{(\mathrm{x}_i, r_i)\}_{i=1}^t$ collected so far, the posterior of $\boldsymbol{\theta}_u$ is obtained by $P(\boldsymbol{\theta}_u|\{(\mathrm{x}_i, r_i)\}_{i=1}^t) \propto \prod_{i=1}^t P(r_i|\mathrm{x}_i, \boldsymbol{\theta}_u)P(\boldsymbol{\theta}_u)$. With a linear assumption $P(r_i|\mathrm{x}_i, \boldsymbol{\theta}_u) = \mathcal{N}(r_i|\mathrm{x}_i^\top \boldsymbol{\theta}_u, \sigma^2)$ and a conjugate prior $P(\boldsymbol{\theta}_u) = \mathcal{N}(\boldsymbol{\theta}_u|\mu_0, \Sigma_0)$, the posterior can be analytically computed as $P(\boldsymbol{\theta}_u|\{(\mathrm{x}_i, r_i)\}_{i=1}^t) = \mathcal{N}(\boldsymbol{\theta}_u|\mu_t, \Sigma_t)$, where μ_t and Σ_t denote the mean and covariance matrix respectively. In each round t, TS samples the bandit parameter $\tilde{\boldsymbol{\theta}}_{u,t}$ from the posterior distribution, i.e., $\tilde{\boldsymbol{\theta}}_{u,t} \sim \mathcal{N}(\boldsymbol{\theta}_u|\mu_{t-1}, \Sigma_{t-1})$, and then selects the arm with the highest reward under the sampled bandit parameter $\mathrm{x}_t = \arg\max_{\mathrm{x}\in\mathcal{A}_t} \mathrm{x}^\top \tilde{\boldsymbol{\theta}}_{u,t}$. In this work, we will restrict our attention to this linear reward setting.

3.2 Non-parametric modeling of an abruptly changing environment

In this work, we consider a typical but non-trivial non-stationary environment, an abruptly changing environment [16, 19, 21], for each user in a collection of N users, denoted as $\mathcal{U}$. In this environment, the ground-truth bandit model $\boldsymbol{\theta}_{u,t}$ for a particular user changes arbitrarily at unknown time points in an *asynchronous* manner, but remains constant between any two consecutive change points in this user. For example, in user u, we could have the following reward generation sequence,

$$\underbrace{r_0, r_1, \cdots, r_{c_{u,1}-1}}_{\text{governed by } \boldsymbol{\theta}_{u,c_{u,0}}}, \underbrace{r_{c_{u,1}}, r_{c_{u,1}+1}, \cdots, r_{c_{u,2}-1}}_{\text{governed by } \boldsymbol{\theta}_{u,c_{u,1}}}, \cdots, \underbrace{r_{c_{u,\Gamma_T^u}}, r_{c_{u,\Gamma_T^u}+1}, \cdots, r_T}_{\text{governed by } \boldsymbol{\theta}_{u,c_{u,\Gamma_T^u}}}$$

where $c_{u,i}$ denotes the time step for the i-th change point of user u (note that $c_{u,0} = 0, \forall u \in \mathcal{U}$). We should note that although the notations look verbose, the subscript u on the change points is necessary because the changes in different users are not necessarily synchronized. $\boldsymbol{\theta}_{u,c_{u,i}}$ is the ground-truth bandit parameter for user u between his/her i-th and the $(i+1)$-th change point. The change points $C_{u,T} = \{c_{u,i}\}_{i\in[0,\Gamma_T^u]}$ of the underlying reward distribution for user $u \in \mathcal{U}$ up to time T and the corresponding bandit parameters $\Theta_{u,T} = \{\boldsymbol{\theta}_{u,c}\}_{c\in C_{u,T}}$ are unknown to the learner. Γ_T^u denotes the number of change points for user u up to time T, which is also unknown. To reflect the nature of a collaborative learning environment, we further assume the bandit parameters $\Theta_{u,T}$ in each user overlap across the N users. Therefore, at a particular moment, users who share the same bandit parameters form clusters; and of course, this clustering structure is unknown to the learner as well. Due to the asynchronous changes of bandit parameters among users, the clustering of users is also evolving over time.

In such a non-stationary environment, existing contextual bandit solutions become incompetent, as the accumulated observations across different stationary periods damage their parameter estimation quality. Existing solutions [19, 42] concerning such an environment detect the changes in each user *independently* and re-build their parameter estimation from scratch after each detected change point. This unfortunately ignores the fact that users are related to each other in such a changing environment, e.g., the dynamically formed user clusters. In the rest of this section, we describe how we explicitly model the change in users as a stochastic process, which brings in the possibility of dynamic collaborative learning.

Motivated by the social psychology theories about social norms [15], in this work instead of considering the preferences of each user as fixed but unknown, we treat them as stochastic by assuming each user's model parameter $\boldsymbol{\theta}_{u,c}$ is drawn from a Dirichlet Process (DP) [5, 14]. Specifically, a Dirichlet Process, $\mathrm{DP}(\alpha_0, G_0)$ with a base distribution G_0 and a scaling parameter α_0, is a distribution over distributions. An important property of DP is that samples from it often share some common values, and therefore naturally form clusters. The number of unique draws, i.e., the number of clusters, varies with respect to the data and thus is random, instead of being pre-specified. This process can be formally described as follows,

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha_0, G_0) \\ \boldsymbol{\theta}_{u,c_{u,i}}|G &\sim G, \forall u \in \mathcal{U}, c_{u,i} \in C_u \\ r_t|\boldsymbol{\theta}_{u,c_{u,i}}, \mathrm{x}_t &\sim \mathcal{N}(r_t|\mathrm{x}_t^\top \boldsymbol{\theta}_{u,c_{u,i}}, \sigma^2), \forall t \in [c_{u,i}, c_{u,i+1}-1] \end{aligned} \tag{1}$$

where the hyper-parameter α_0 controls the concentration of unique draws from the DP prior, the base distribution G_0 specifies the prior distribution of the bandit parameters in each individual model, and G represents the mixing distribution of the sampled results of $\boldsymbol{\theta}_{u,c}$. To enable efficient posterior inference, conjugate priors are expected in G_0. Due to our linear reward assumption, we impose a zero-mean isotropic Gaussian prior governed by a single precision parameter λ on $\boldsymbol{\theta}_{u,c}$ as $G_0 = N(0, \lambda^{-1}\mathrm{I})$. With the DP prior defined above, when a new user arrives or an existing user changes his/her preference at time t, the distribution of this user's new bandit parameter $\boldsymbol{\theta}_{u,t}$ conditioned on all existing bandit parameters $\Theta_{t-1} = \{\boldsymbol{\theta}_{u,c}\}_{u\in\mathcal{U}, c\in C_{u,t-1}}$ can be analytically derived by integrating out G in Eq (1):

$$P(\boldsymbol{\theta}_{u,t}|\Theta_{t-1}, \alpha_0, G_0) = \frac{\alpha_0 G_0}{|\Theta_{t-1}| + \alpha_0} + \frac{\sum_{\boldsymbol{\theta}\in\Theta_{t-1}} \delta_{\boldsymbol{\theta}_{u,t}}(\boldsymbol{\theta})}{|\Theta_{t-1}| + \alpha_0} \tag{2}$$

where $\delta_{\boldsymbol{\theta}_{u,t}}(\cdot)$ is a delta function concentrated at $\boldsymbol{\theta}_{u,t}$. This conditional distribution well captures the idea of social psychology theories about social norms [15]: when a user's preference changes or a new user comes, the prior distribution over the new model that he/she tends to choose is proportional to the popularity of existing models in overall user population at the moment.

To facilitate our discussion about this clustering property, we denote the set of unique draws in Θ_{t-1} as $\{\phi_z\}_{z=1}^{K_{t-1}}$, where K_{t-1} is the total number of unique draws from DP so far. Then we introduce an indicator variable $z_{u,t}$ such that $\boldsymbol{\theta}_{u,t} = \phi_{z_{u,t}}$, i.e., $z_{u,t}$ is the model index in this globally shared unique bandit parameter set. Denote $\mathcal{Z}_t = \{z_{u,c}\}_{u\in\mathcal{U}, c\in C_{u,t}}$, and an equivalent form of Eq (2) is:

$$P(z_{u,t} = k|\alpha_0, \mathcal{Z}_{t-1}\}) \propto \begin{cases} n_{k,t-1} & \text{if } k \in [K_{t-1}] \\ \alpha_0 & \text{if } k = K_{t-1}+1 \end{cases} \tag{3}$$

where $n_{k,t-1} = \sum_{z\in\mathcal{Z}_{t-1}} 1\{z = k\}$ is the number of times elements in Θ_{t-1} takes value ϕ_k.

As a result, the imposed DP prior encourages users to form shared groups at any particular moment of time, which makes

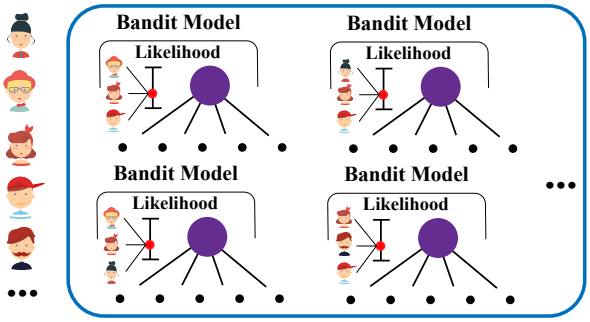


Figure 1: Illustration of CoDBand. An adaptively maintained pool of contextual bandit models is shared among all the users with respect to the underlying clustering structure of them. Bandit models are assigned to users based on fitness with user history data.

online collaborative learning feasible. We should emphasize that our collaborative bandit solution does not require any knowledge about K_T or $\{\phi_z\}_{z=1}^{K_T}$, but adaptively learns them via Bayesian inference with the observations obtained during its interaction with users.

3.3 Collaborative Dynamic Bandit

In the non-stationary environment specified above, to make personalized recommendations in real-time, several challenges have to be addressed: 1) as the changes in a user are unknown to the learner, how to detect the potential changes in each user's bandit parameters; 2) how to estimate the globally shared bandit parameters with the observations obtained from different users.

As our solution, an adaptively maintained pool of contextual bandit models is shared among all the users (as shown in Figure 1). To address the challenges above, a change point detector is used to detect the changes in each user's bandit parameter, and a collapsed Gibbs sampler is used to select a suitable bandit model to serve the user. This sampling procedure selects a global bandit model for a user by taking into consideration both how well the model fits the user's recent history data as well as the model's popularity among all the users. This captures the intuition that when there is limited knowledge about a user (e.g., cold start), it is better to explore whether the well-established popular models fit the user, compared with directly starting from scratch (as in [42]). Global bandit models are created, updated or removed from the pool in an adpative manner as the algorithm interacts with the users. We name the resulting bandit algorithm as Collaborative Dynamic Bandit, or CoDBand in short, and illustrate the details of it in Algorithm 1.

Before presenting the detailed description of the two core components of CoDBand, i.e., change detection and collapsed Gibbs sampling, we first introduce how observations are managed in it:

- CoDBand maintains a set $\mathcal{D}_t^u$ for each user $u \in \mathcal{U}$ that is updated by each new observation from u (line 19 in Algorithm 1), and is reset to $\mathcal{D}_t^u = \emptyset$ when a change point in u is detected (line 25-26 in Algorithm 1). As a result, $\mathcal{D}_t^u$ reflects the target user u's recent preferences, as it contains only observations in the current stationary period of u with a high probability.

Algorithm 1: Collaborative Dynamic Bandit (CoDBand)

Input : σ, a, b, λ, δ_1, δ_2, τ

Initialize: Construct user set $\mathcal{U}$ and initialize $\mathcal{U} = \emptyset$. Construct global bandit model set $\mathcal{G}$ and initialize $\mathcal{G} = \emptyset$. Sample $\alpha_0 \sim \Gamma(a, b)$.

1 **for** $t = 1$ *to* T **do**
2 Observe current user u_t, and candidate arm pool $\mathcal{A}_t$;
3 **if** $u_t \notin \mathcal{U}$ **then**
4 $\mathcal{U} = \mathcal{U} \cup u_t$;
5 Initialize an observation set for u_t: $\mathcal{D}_{t-1}^{u_t} = \emptyset$;
6 **end**
7 **ARM SELECTION**;
8 **if** $\mathcal{D}_{t-1}^{u_t} = \emptyset$ **then**
9 Sample a model index $\tilde{z}_{u_t}$ for user u_t using Eq (3) ;
10 **if** $\tilde{z}_{u_t} = |\mathcal{G}| + 1$ **then**
11 Initialize a new global model $\mathcal{M}_{\tilde{z}_{u_t}}$: $n_{\tilde{z}_{u_t}} = 0$, $\Sigma_{\tilde{z}_{u_t}}^{-1} = \lambda I \in \mathbb{R}^{d \times d}$, $\mathrm{b}_{\tilde{z}_{u_t}} = \mathbf{0} \in \mathbb{R}^d$, $\mu_{\tilde{z}_{u_t}} = \Sigma_{\tilde{z}_{u_t}} \mathrm{b}_{\tilde{z}_{u_t}}$;
12 Add it to the global model set $\mathcal{G} = \mathcal{G} \cup \mathcal{M}_{\tilde{z}_{u_t}}$;
13 **end**
14 $n_{\tilde{z}_{u_t}} = n_{\tilde{z}_{u_t}} + 1$;
15 **end**
16 Sample $\tilde{\theta}_t \sim \mathcal{N}(\mu_{\tilde{z}_{u_t}}, \Sigma_{\tilde{z}_{u_t}})$;
17 Select $x_t = \arg\max_{x \in \mathcal{A}_t} \mathrm{x}^\top \tilde{\theta}_t$, and observe reward r_t;
18 **MODEL UPDATE**;
19 Compute $e_{u_t,t}$ according to Eq (4), and update $\hat{e}_{u_t,t}$;
20 Update the observation set: $\mathcal{D}_t^{u_t} = \mathcal{D}_{t-1}^{u_t} \cup \{(\mathrm{x}_t, r_t)\}$;
21 Update global model $\mathcal{M}_{\tilde{z}_{u_t}}$: $\Sigma_{\tilde{z}_{u_t}}^{-1} = \Sigma_{\tilde{z}_{u_t}}^{-1} + \frac{1}{\sigma^2} \mathrm{x}_t \mathrm{x}_t^\top$, $\mathrm{b}_{\tilde{z}_{u_t}} = \mathrm{b}_{\tilde{z}_{u_t}} + \frac{1}{\sigma^2} \mathrm{x}_t r_t$, $\mu_{\tilde{z}_{u_t}} = \Sigma_{\tilde{z}_{u_t}} \mathrm{b}_{\tilde{z}_{u_t}}$;
22 $\tilde{z}_{u_t}, \mathcal{G}$ = Collapsed Gibbs Sampler$(\tilde{z}_{u_t}, \mathcal{D}_t^{u_t}, \mathcal{G})$;
23 α_0 = Update Parameter$(\alpha_0, |\mathcal{G}|, a, b, \sum_{k=1}^{|\mathcal{G}|} n_k)$ [13];
24 **CHANGE DETECTION**;
25 **if** $\hat{e}_{u_t,t} > \delta_1 + \sqrt{\frac{\log 1/\delta_2}{\tau}}$ **then** Set $\mathcal{D}_t^{u_t} = \emptyset$, $\hat{e}_{u_t,t} = 0$;
26 **end**

- CoDBand also maintains a pool of globally shared bandit models denoted as $\mathcal{G}_t$, and each bandit model $\mathcal{M}_{k,t} \in \mathcal{G}_t$ maintains a posterior distribution $\mathcal{N}(\mu_{k,t}, \Sigma_{k,t})$ of the unknown bandit parameter and a counter $n_{k,t}$ recording the number of times $\mathcal{M}_{k,t}$ is assigned to a user (line 14 in Algorithm 1). It is obvious from the context that $\mathcal{G}_t$, $\mathcal{M}_{k,t}$, $\mu_{k,t}$, $\Sigma_{k,t}$ and $n_{k,t}$ are all updated over time, so the subscript t is omitted for simplicity in the following discussions.

Intuitively, each bandit model $\mathcal{M}_k \in \mathcal{G}$ represents a typical type of user behaviors that are learned from the system's interaction history with all users. The set $\mathcal{D}_t^u$ serves as an anchor to decide which bandit model $\mathcal{M}_k$ best fits user u's recent preferences. In the rest of this section, we will introduce details about how we perform change detection to maintain $\mathcal{D}_t^u$, and how we use collapsed Gibbs sampling to update and select $\mathcal{M}_k$ in individual users.

3.3.1 Change Detection. Since we assume change points are arbitrary and unknown to the learner, the change point detector from [42] can be adopted to detect the changes in a user's bandit parameter. This is done by constructing the test variable

$$e_{u_t,t} = \mathbf{1}\big\{|\hat{r}_t - r_t| > \mathrm{CB}_{u_t,t-1}(\mathrm{x}_t) + \epsilon\big\}. \tag{4}$$

$e_{u_t,t}$ indicates whether the received reward r_t deviates too much from the estimated reward $\hat{r}_t = \mathrm{x}^\top \hat{\theta}_{u_t,t-1}$, where $\hat{\theta}_{u_t,t-1} = \big(\lambda I + \sum_{(\mathrm{x}_i,r_i)\in\mathcal{D}^{u_t}_{t-1}} \mathrm{x}_i\mathrm{x}_i^\top\big)^{-1}\big(\sum_{(\mathrm{x}_i,r_i)\in\mathcal{D}^{u_t}_{t-1}} r_i\mathrm{x}_i\big)$ is the Ridge regression estimator using observations in $\mathcal{D}^{u_t}_{t-1}$. $\mathrm{CB}_{u_t,t-1}(\mathrm{x})$ denotes the high probability confidence bound from [1], which is defined as $\mathrm{CB}_{u_t,t-1}(\mathrm{x}) = \alpha_{u_t,t-1}\sqrt{\mathrm{x}^\top\Big(\lambda I + \sum_{(\mathrm{x}_i,r_i)\in\mathcal{D}^{u_t}_{t-1}} \mathrm{x}_i\mathrm{x}_i^\top\Big)^{-1}\mathrm{x}}$, where $\alpha_{u_t,t-1} = \sigma\sqrt{d\log\Big(1 + \frac{|\mathcal{D}^{u_t}_{t-1}|}{d\lambda}\Big) + 2\log\frac{1}{\delta_1}} + \sqrt{\lambda}$. And in Eq (4), $\epsilon = \sqrt{2}\sigma\mathrm{erf}^{-1}(\delta_1 - 1)$ represents the high probability bound of Gaussian noise in the received feedback and $\mathrm{erf}^{-1}(\cdot)$ is the inverse of Gaussian error function.

When the reward distribution remains stationary (e.g., observations in $\mathcal{D}^{u_t}_{t-1}$ and (x_t, r_t) are homogeneous), with a probability at least $1-\delta_1$, the test variable $e_{u_t,t} = 0$ [42]. To account for the noise in one individual observation, an empirical mean of $e_{u_t,t}$ over the τ most recent interactions with user u_t is maintained, which is denoted as $\hat{e}_{u_t,t} = \frac{1}{\min(|\mathcal{D}_{u_t,t-1}|,\tau)}\sum_i e_{u_t,i}$ (line 19 in Algorithm 1). When $\hat{e}_{u_t,t} > \delta_1 + \sqrt{\frac{\ln 1/\delta_2}{2\tau}}$ (obtained by Hoeffding inequality), a change is said to be detected in user u_t's bandit parameter and the value of $\hat{e}_{u_t,t}$ is reset (line 25-26 in Algorithm 1).

3.3.2 Collapsed Gibbs Sampling. As mentioned earlier, a collapsed Gibbs sampler is used to select the bandit model $\mathcal{M}_k \in \mathcal{G}$ for user u, by sampling a model index $\tilde{z}_u$ from its posterior distribution conditioned on $\mathcal{D}^u_t$, as illustrated in Algorithm 2. The conditional posterior of $\tilde{z}_u$ consists of two parts: the conditional prior distribution of $\tilde{z}_u$ in Eq (3), e.g., popularity of the bandit model among all users, and the marginalized likelihood on $\mathcal{D}^u_t$, e.g., fitness with the user's recent history. With the conjugate Gaussian prior we introduced in Eq (1), the marginalized likelihood $P(r_i|\mathrm{x}_i, \tilde{z}_u = k, \mathcal{G}) = \int \mathcal{N}(r_i|\mathrm{x}_i^\top\phi, \sigma^2)\mathcal{N}(\phi|\mu_k, \Sigma_k^{-1})d\phi = \mathcal{N}\big(r_i|\mathrm{x}_i^\top\mu_k, \sigma^2 + \mathrm{x}_i^\top\Sigma_k^{-1}\mathrm{x}_i\big)$ can be analytically computed. Therefore, the conditional posterior distribution of $\tilde{z}_u$ can be computed as,

$$\begin{aligned} &P(\tilde{z}_u = k|\alpha_0, \{n_k\}_{k=1}^K, \mathcal{D}^u, \mathcal{G}) \\ &\propto P(\tilde{z}_u = k|\alpha_0, \{n_k\}_{k=1}^K) P(\mathcal{D}^u|\tilde{z}_u = k, \mathcal{G}) \\ &\propto \begin{cases} n_k \prod_{(\mathrm{x}_i,r_i)\in\mathcal{D}^u} \mathcal{N}(r_i|\mathrm{x}_i^\top\mu_k, \sigma^2 + \mathrm{x}_i^\top\Sigma_k^{-1}\mathrm{x}_i) & \text{if } k \in [K] \\ \alpha_0 \prod_{(\mathrm{x}_i,r_i)\in\mathcal{D}^u} \mathcal{N}(r_i|0, \sigma^2 + \lambda^{-1}\mathrm{x}_i^\top\mathrm{x}_i) & \text{if } k = K+1 \end{cases} \end{aligned} \tag{5}$$

Note that the concentration parameter α_0 affects the number of global models learnt by CoDBand, which is unknown in practice and may requires manual tuning. To alleviate this, we introduce a Gamma prior, i.e. $\alpha_0 \sim \Gamma(a, b)$, and update it with Gibbs sampling as well (line 23 in Algorithm 1). The sampling procedure for α_0 is the same as Section 6 of [13].

In the model update stage of each iteration (line 22 in Algorithm 1), the collapsed Gibbs sampler is executed to re-assign the model index for the user u_t given this user's $\mathcal{D}^{u_t}_t$, and the bandit models involved in this procedure will be updated accordingly (line 1 and 6 in Algorithm 2). Intuitively, as we have more observations about the user, we can select a better suited global model for him/her with an increasing confidence. It is worth noting that when the target user is new or with newly detected changes, CoDBand tends to choose a currently popular model for him/her to start with (line 8-9 in Algorithm 1), rather than to always create a new model, due to our underlying DP modeling assumption of user preferences. This choice is arguably preferred, especially when a large population of users are presented. As the sufficient statistics are maintained at model-level, e.g., the globally shared models in $\mathcal{G}$, instead of at user-level, collaborative learning is achieved. When a user switches to an existing model, the system can take advantage of the already accumulated statistics to make more accurate recommendations.

After a model is sampled for the user, arm selection is conducted using Thompson sampling (line 16-17 in Algorithm 1). Compared with standard Thompson sampling [4, 10], we are introducing another layer of exploration in the model space. This is because CoDBand first samples a model index from the posterior over all possible bandit models and then samples a bandit parameter conditioning on the sampled model.

Algorithm 2: Collapsed Gibbs Sampler

Input : model index $\tilde{z}_u$, observation set $\mathcal{D}^u$, global bandit model set $\mathcal{G}$

Output : new model index $\tilde{z}'_u$, updated global bandit model set $\mathcal{G}$

1 Remove $\mathcal{D}^u$ from global model $\mathcal{M}_{\tilde{z}_u}$: $n_{\tilde{z}_u} = n_{\tilde{z}_u} - 1$,
$\Sigma^{-1}_{\tilde{z}_u} = \Sigma^{-1}_{\tilde{z}_u} - \frac{1}{\sigma^2}\sum_{(x_i,r_i)\in\mathcal{D}^u} \mathrm{x}_i\mathrm{x}_i^\top$,
$\mathrm{b}_{\tilde{z}_u} = \mathrm{b}_{\tilde{z}_u} - \frac{1}{\sigma^2}\sum_{(x_i,r_i)\in\mathcal{D}^u} \mathrm{x}_i r_i$, $\mu_{\tilde{z}_u} = \Sigma_{\tilde{z}_u}\mathrm{b}_{\tilde{z}_u}$;
2 **if** $n_{\tilde{z}_u} = 0$ **then** Remove $\mathcal{M}_{\tilde{z}_u}$: $\mathcal{G} = \mathcal{G} \setminus \mathcal{M}_{\tilde{z}_u}$;
3 Sample new model index $\tilde{z}'_u$ for $\mathcal{D}^u$ according to Eq (5);
4 Update global model $\mathcal{M}_{\tilde{z}'_u}$ with $\mathcal{D}^u$: $n_{\tilde{z}'_u} = n_{\tilde{z}'_u} + 1$,
$\Sigma^{-1}_{\tilde{z}'_u} = \Sigma^{-1}_{\tilde{z}'_u} + \frac{1}{\sigma^2}\sum_{(x_i,r_i)\in\mathcal{D}^u} \mathrm{x}_i\mathrm{x}_i^\top$,
$\mathrm{b}_{\tilde{z}'_u} = \mathrm{b}_{\tilde{z}'_u} + \frac{1}{\sigma^2}\sum_{(x_i,r_i)\in\mathcal{D}^u} \mathrm{x}_i r_i$, $\mu_{\tilde{z}'_u} = \Sigma_{\tilde{z}'_u}\mathrm{b}_{\tilde{z}'_u}$;

4 REGRET ANALYSIS

We analyze the accumulative Bayesian regret of CoDBand, which is defined as:

$$\mathbb{E}[R_T] = \mathbb{E}\Big[\sum_{t=1}^T r_t\Big] = \mathbb{E}\Big[\sum_{t=1}^T \mathrm{x}_t^{*\top}\theta_{u_t,t} - \mathrm{x}_t^\top\theta_{u_t,t}\Big], \tag{6}$$

where the expectation is taken with respect to the prior distribution of the bandit parameter $\theta_{u_t,t}$. x^*_t is the best arm in hindsight and x_t is the selected arm at time t. To analyze Bayesian regret, we define the upper confidence bound function $U_t : [K_t] \times \mathcal{A}_t \to \mathbb{R}$ and the lower confidence function $L_t : [K_t] \times \mathcal{A}_t \to \mathbb{R}$ by

$$\begin{aligned} U_t(k, \mathrm{x}) &= f_{\hat{\theta}_{k,t-1}}(\mathrm{x}) + \alpha_{k,t-1}||\mathrm{x}||_{V^{-1}_{k,t-1}} \\ L_t(k, \mathrm{x}) &= f_{\hat{\theta}_{k,t-1}}(\mathrm{x}) - \alpha_{k,t-1}||\mathrm{x}||_{V^{-1}_{k,t-1}} \end{aligned}$$

where $V_{k,t} = \lambda I + \sum_{s\in\mathcal{I}(k)} \mathrm{x}_i\mathrm{x}_i^\top$, and $\mathcal{I}(k)$ denotes the set of time steps where the bandit parameter $\theta_{u_t,t}$ takes value ϕ_k.

Denote $\mathcal{H}_t = \sigma(\mathrm{x}_1, r_1, \ldots, \mathrm{x}_t, r_t)$ as the σ-algebra generated by the interaction sequence at time step t. Our regret analysis draws its key insight from [38] that for Thompson sampling method we have

$\mathbb{P}(\tilde{z}_t|\mathcal{H}_{t-1}) = \mathbb{P}(z_t^*|\mathcal{H}_{t-1})$ and $\mathbb{P}(\mathrm{x}_t|\tilde{z}_t = k, \mathcal{H}_{t-1}) = \mathbb{P}(\mathrm{x}_t^*|z_t^* = k, \mathcal{H}_{t-1})$. Therefore, $\mathbb{P}[(\mathrm{x}_t = x) \cap (\tilde{z}_t = k)|\mathcal{H}_{t-1}] = \mathbb{P}[(\mathrm{x}_t^* = x) \cap (z_t^* = k)|\mathcal{H}_{t-1}]$. In addition, since $U_t(k, \mathrm{x})$ and $L_t(k, \mathrm{x})$ are deterministic functions, $\mathbb{E}[U_t(\tilde{z}_t, \mathrm{x}_t)|\mathcal{H}_{t-1}] = \mathbb{E}[U_t(z_t^*, \mathrm{x}_t^*)|\mathcal{H}_{t-1}]$ and $\mathbb{E}[L_t(\tilde{z}_t, \mathrm{x}_t)|\mathcal{H}_{t-1}] = \mathbb{E}[L_t(z_t^*, \mathrm{x}_t^*)|\mathcal{H}_{t-1}]$. Based on a similar decomposition as in [26, 38], we obtain the following result.

Lemma 4.1. *The accumulated Bayesian regret defined in Eq* (6) *can be decomposed into the following three terms:*

$$\begin{aligned}\mathbb{E}[R_T] \leq 2\sum_{t=1}^{T}\mathbb{P}\{\big[\mathrm{x}_t^\top\theta_{u_t,t} < L_t(z_t^*, \mathrm{x}_t)\big] \cup \big[\mathrm{x}_t^{*\top}\theta_{u_t,t} > U_t(z_t^*, \mathrm{x}_t^*)\big]\}\\ + \sum_{t=1}^{T}\mathbb{E}[U_t(z_t^*, \mathrm{x}_t) - L_t(z_t^*, \mathrm{x}_t)]\\ + \sum_{t=1}^{T}\mathbb{E}[\big[U_t(z_t^*, \mathrm{x}_t^*) - U_t(z_t^*, \mathrm{x}_t)\big] \cdot 1\{\tilde{z}_t \neq z_t^*\}]\end{aligned}$$

It is worth noting that the first two terms can be found in the Bayesian regret for linear Thompson sampling (Section 6.2.1 in [38]) as well: the first term is related to the case when reward estimation error exceeds its high confidence bound, which is bounded by the constant 4 based on Theorem 2 in [1]; the second term corresponds to the rate of convergence of the confidence interval. and by rewriting the summation over each model, and then applying Theorem 3 in [1], it is bounded by $O\Big(\sigma d\sqrt{T}\log T(\sum_{k=1}^{K_T}\sqrt{p_k})\Big)$ where $p_k = \frac{|\mathcal{I}(k)|}{T}$ for $k \in [K_T]$ denotes the portion of time steps that the bandit parameter takes value ϕ_k.

The key difference between our regret analysis and that of linear Thompson sampling is the additional third term, which corresponds to the regret due to sampling a wrong model. This is unique to our problem because compared with linear Thompson sampling, CoDBand addresses exploration and exploitation not only on arm level, but also on model level. To bound this term, we further decompose it based on whether late detection has happened. Denote $\mathcal{L}_t$ as the late detection event at time t that the change detector defined in Section 3.3.1 fails to detect the most recent change point so far, and the complement of $\mathcal{L}_t$ is denoted as $\mathcal{L}_t^C$. Then we can further decompose the third term as:

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}[\big[U_t(z_t^*, \mathrm{x}_t^*) - U_t(z_t^*, \mathrm{x}_t)\big] \cdot 1\{\tilde{z}_t \neq z_t^*\}] \leq C_0\sum_{t=1}^{T}\mathbb{E}[1\{\tilde{z}_t \neq z_t^*\}]\\ &= C_0\sum_{t=1}^{T}P(\tilde{z}_t \neq z_t^*|\mathcal{L}_t^C)P(\mathcal{L}_t^C) + C_0\sum_{t=1}^{T}P(\tilde{z}_t \neq z_t^*|\mathcal{L}_t)P(\mathcal{L}_t)\\ &\leq \underbrace{C_0\sum_{t=1}^{T}P(\mathcal{L}_t)}_{A_1} + \underbrace{C_0\sum_{t=1}^{T}P(\tilde{z}_t \neq z_t^*|\mathcal{L}_t^C)}_{A_2}\end{aligned}$$

where $C_0 = 2+\sigma\sqrt{\frac{2}{\lambda}\log\frac{1}{\delta_1}}$ is the constant upper bound of $U_t(z_t^*, \mathrm{x}_t^*)$ obtained by setting $t = 0$. The term A_1 represents the regret penalty due to late detection, which can be upper bounded by Lemma 4.2.

Lemma 4.2. *Let $S_{u,c}$ denote the length of stationary period after the c'th change point of user u. According to Lemma 3.4 in [42], assume at least ρ portion of arms in $\mathcal{A}_t, \forall t$ satisfy $|\mathrm{x}^\top\theta_{u,c_{u,i}} - \mathrm{x}^\top\theta_{u,c_{u,i+1}}| \geq \Delta$, and by setting $\delta_1 \leq 1 - \frac{1}{\rho}\big(1 - \frac{\sqrt{\lambda}}{2\min(S_{u,c})\rho}(\Delta - 2\sqrt{\lambda} - 2\epsilon)\big)$ and $\tau \geq \frac{2\ln\frac{2}{\delta_2}}{(\rho(1-\delta_1)-\delta_1)^2}$, the probability of detection when change has happened is $p_d \geq 1 - \delta_2$. This leads to the following upper bound of the term A_1:*

$$\begin{aligned}A_1 = C_0\sum_{u\in\mathcal{U}}\sum_{c\in C_{u,T}}\sum_{t\in S_{u,c}}P(\mathcal{L}_t) \leq C_0\sum_{u\in\mathcal{U}}\sum_{c\in C_{u,T}}\frac{1-\delta_2^{|S_{u,c}|}}{1-\delta_2}\\ \leq \frac{C_0}{1-\delta_2}\sum_{u\in\mathcal{U}_T}\Gamma_T^u\end{aligned}$$

The term A_2 corresponds to the regret penalty caused by sampling a wrong model for arm selection when there is no late detection. It is related to the reward gap Δ between different bandit parameters as well as the model's confidence in the estimation. We bound it by Lemma 4.3.

Lemma 4.3. *Adopting the same assumption as in [17, 18], at each time t, arm set $\mathcal{A}_t$ 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 vectors satisfies $\varsigma^2 \leq \frac{\lambda'^2}{8\ln 4K}$. Then the term A_2 can be upper bounded by:*

$$A_2 = C_0\sum_{t=1}^{T}P(\tilde{z}_t \neq z_t^*|\mathcal{L}_t^C) = O(K_T\big[(\sum_{u\in\mathcal{U}}\Gamma_T^u) + C_1\big])$$

with probability at least $1 - \delta'$, where $C_1 = \frac{\psi_L\sigma^2}{\Delta^2\lambda'^2}\log\frac{d}{\delta'}$ and ψ_L is a constant that depends on d, σ, λ.

Combining all the components together we obtain the final regret upper bound $\mathbb{E}[R_T] = O\Big(\sigma d\sqrt{T}\log T(\sum_{k=1}^{K_T}\sqrt{p_k}) + K_T\sum_{u\in\mathcal{U}}\Gamma_T^u\Big)$. CoDBand achieves a standard $\tilde{O}(\sqrt{T})$ regret bound with respect to time horizon T, and the added regret only depends on the underlying grouping structure among users $\sum_{k=1}^{K_T}\sqrt{p_k}$ and the total number of stationary periods among all users $\sum_{u\in\mathcal{U}}\Gamma_T^u$, which are independent from the recommendations of the system.

5 EVALUATIONS

We performed extensive empirical evaluations of CoDBand against several related baseline bandit algorithms, which can be summarized into the following three categories. First, contextual bandits that do not consider collaboration effects or the non-stationarity of the environment: we include **LinUCB** [27], which has been shown to be effective in providing interactive personalized recommendations in a stationary environment. Second, collaborative bandits: **CLUB** [18], which assumes the existence of underlying stationary user clusters and learns the user clusters and cluster-wise bandit models on the fly. **SCLUB** [29], which is a recent extension of **CLUB** for non-uniform distribution of the clusters. Third, contextual bandits that account for a non-stationary environment in a per-user basis, including **AdTS** [19]: which is an adaptive Thompson Sampling algorithm with a cumulative sum test based change detection module; and **dLinUCB**, which is a state-of-the-art non-stationary contextual bandit algorithm [42]. These two non-stationary solutions have shown to be the most competitive among the other non-stationary bandit solutions according to [42]. We compared all the algorithms in both simulations and large-scale real-world datasets to compare their effectiveness in handling a changing environment for collaborative recommendation. Our code for conducting these experiments will be available online. In

(a) Simulation setting 1 (b) Simulation setting 2 (c) Simulation setting 3

Figure 2: Performance comparison on synthetic datasets.

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

	N	K	S_{min}	S_{max}	T	σ	oracle.	LinUCB	adTS	dLinUCB	CLUB	SCLUB	CoDBand
1	100	10	500	3000	3000	0.1	124	24050	9183	3030	24602	24602	**1193**
2	100	50	500	3000	3000	0.1	575	24352	19433	2858	24762	24980	**2252**
3	100	100	500	3000	3000	0.1	922	28108	20828	3388	28424	28585	**2688**
4	100	10	200	500	3000	0.1	128	54791	52282	17475	55098	55268	**5143**
5	100	10	500	800	3000	0.1	131	51095	40538	8401	51440	51604	**2423**
6	100	10	800	1100	3000	0.1	128	39035	26851	6549	39395	39477	**2342**
7	100	10	500	3000	3000	0.13	175	27101	20555	3742	27163	27633	**3043**
8	100	10	500	3000	3000	0.16	280	23949	21320	4833	23693	24436	**3629**

simulation-based experiments, we also include **oracle-LinUCB** for comparison, which runs an instance of LinUCB for each unique global bandit model in the corresponding stationary period in each user. Comparing with it helps us understand the added regret from errors in change detection and model clustering.

5.1 Experiments on synthetic dataset

Simulation settings: In simulation, we generate a set of users $\mathcal{U}$ ($|\mathcal{U}| = N$) with an arm pool $\mathcal{A}$ of size 1000, in which each arm a is associated with a d-dimensional feature vector $\mathrm{x} \in \mathbb{R}^d$ with $\|\mathrm{x}\|_2 \leq 1$. To simulate an abruptly changing environment, for each user we sample a sequence of time intervals from (S_{min}, S_{max}) uniformly. Each time interval is considered as a stationary period such that we can naturally get the change points in each user. Note that since the stationary periods for different users are drawn independently, it is highly *unlikely* for the users to change synchronously. At the change point c of each user u, we experimented with three different settings to decide the ground-truth bandit parameters: 1) $\boldsymbol{\theta}_{u,c}$ is generated according to the DP model described in Eq (3); 2) $\boldsymbol{\theta}_{u,c}$ is sampled from a fixed set of unique bandit parameters $\{\phi_k\}_{k=1}^K$ with a predefined mixture weight; 3) a stationary environment is also included for comparison, where $\boldsymbol{\theta}_{u,c}$ remains the same over time. Note that neither the users' change points, nor the ground-truth bandit parameters are disclosed to the learners. At each time step $t \in [T]$, all users in $\mathcal{U}$ gets served one by one, and a subset of arms $\mathcal{A}_t \subset \mathcal{A}$ are randomly chosen and disclosed to the learner. The ground-truth reward r_t is corrupted by Gaussian noise $\eta_t \sim N(0, \sigma^2)$ before giving back to the learners.

Empirical regret comparison on synthetic dataset: We set the number of user $N = 100$, the total number of time steps $T = 3000$, and the range of stationary period length for Settings 1 and 2 as $(S_{min} = 500, S_{max} = 3000)$. Setting 1 and 2 are initialized with the same set of unique bandit parameters of size $K = 10$. We set $N = 500$, $T = 600$ and $K = 2$ for setting 3. We report the accumulated regret of all algorithms under the three simulation settings in Figure 2. We can observe that LinUCB, CLUB and SCLUB all suffer linear regret after the first change point in Setting 1 and 2 because of their strong but unrealistic stationary assumption. Both AdTS and dLinUCB can react to the environment changes, but they are slow and less accurate in doing so, and thus accumulate faster increasing regret. In addition, AdTS has a large probability of making false change detections and incurs fast increasing regret in the stationary Setting 3, where the underlying bandit model in each user does not change. The proposed CoDBand can not only quickly identify the changes in each user, but also properly recognize which existing model to reuse, which brings further reduction of regret comparing to those non-collaborative or non-stationary baselines. It is worth noting that in Setting 2, DP prior is mis-specified in CoDBand as the underlying bandit parameter generation does not follow this stochastic process, but CoDBand can still quickly identify the correct bandit model to use, and obtain better performance than all the baselines. In the three settings, the oracle-LinUCB baseline performed the best, as it knows exactly when the change happens and how the different users are related to each other. But the added regret from CoDBand is acceptable, given the algorithm needs to both detect the change and cluster the models on the fly without any prior knowledge about the environment.

To further verify the robustness of CoDBand under different simulation settings, we varied the parameters in Setting 2, e.g., the number of unique bandit parameters K, the minimum and maximum length for stationary periods S_{min} and S_{max}, the variance of noise σ^2, and report the algorithms' corresponding regrets in Table 1. The results show that CoDBand can successfully cope with different environment settings and outperform the baselines. In addition, the trends of how regret changes with different parameters align with our regret analysis. For example, with the increase of the number of unique bandit parameter K in the same number of N users, the regret increases, because less observations can be shared among users. The regret also increases substantially with shorter stationary periods, as more errors would occur in change detection. In addition, larger amount of noise in the reward not only slows down CoDBand's bandit parameter estimation but also affects its change detection accuracy, and therefore leads to higher regret.

5.2 Experiments on real-world datasets

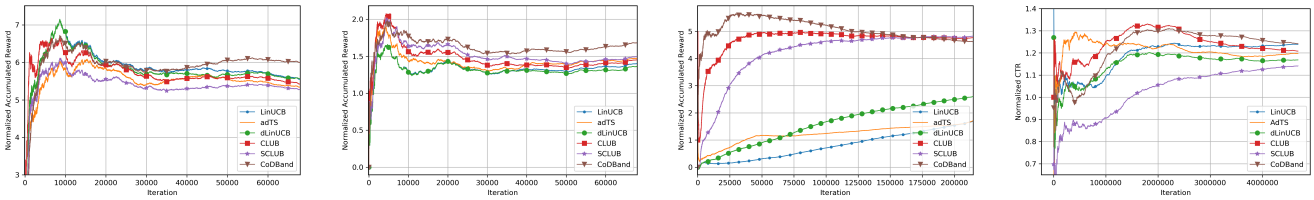
LastFM and Delicious: The LastFM dataset is extracted from the music streaming service Last.fm, and the Delicious dataset is extracted from the social bookmark sharing service Delicious. They were made availalbe by the HetRec 2011 workshop. The LastFM dataset contains 1892 users and 17632 items (artists). We consider the "*listened artists*" in each user as positive feedback. The Delicious dataset contains 1861 users and 69226 items (URLs). We treat the bookmarked URLs in each user as positive feedback. Both datasets provide social network information about the users. Following the settings in [9], we pre-processed these two datasets in order to fit them into a contextual bandit setting. Firstly, we used all tags associated with an item to create a TF-IDF feature vector to represent each item. Then we used PCA to reduce the dimensionality of the feature vectors and retained the first 25 principle components to construct the context vectors, i.e., $d = 25$. We fixed the size of candidate arm pool to $|\mathcal{A}_t| = 25$; for a particular user u, we randomly picked one item from his/her nonzero reward items, and randomly picked the other 24 from those zero reward items. On these two datasets, since each individual user's observations are sparse and mostly collected from a short period of time, it is hard to directly observe non-stationarity. Previous studies [23, 42] introduce non-stationarity in the following way: create 10 user groups (so-called super-user) via spectral clustering base on user social network. Users in the same user group are considered to have similar result preferences. Then the super-users are stacked together chronologically to create a hybrid user, i.e., non-stationarity. The boundaries between super-users are considered as preference change points of the hybrid user. In this work, to highlight the effectiveness of collaboration, we further make this non-stationary environment more challenging by splitting each super-user into 3 parts, and refer to them as mini-super users. We randomize the order of 3×10 mini-super users. In this case, collaborative bandit solutions should identify the overlap between mini-super users from the same super user and take advantage of observation sharing, while failing to detect such collaborative effects will cost an algorithm sub-optimal performance in such a setting. To clarify, in the rest of the discussions, when we mention "user" concerning LastFM and Delicious datasets, we are referring to the mini-super users.

We report normalized reward, e.g., the ratio between accumulative reward collected from the bandit algorithms and that from a random selection policy on LastFM and Delicious datasets in Figure 3 (a) and (b) respectively. We can observe that on both datasets, CoDBand outperforms the baselines. The advantage of CoDBand is more apparent at the later stage of learning, where it accumulated enough observations to accurately estimate a set of global bandit models that were representative to predict result preferences of users in the population. These global bandit models can be used to provide high quality recommendations for new users or users that have recently switched their preferences, whereas the other baselines either got distracted by the outdated observations in their model estimation, or discarded the outdated observations and completely restart from scratch.

To further investigate what kind of preferences in the user population that CoDBand has captured, we visualized its learnt global models on the LastFM dataset. In this dataset, each artist is associated with a list of tags provided by the users. The tags are usually descriptive and reflect music genres or artist styles. For each learnt global model, we use the tags associated with the top-100 artists scored by this model to generate a word cloud. Figure 4 demonstrates four representative groups (based on their inferred popularity) CoDBand has learnt on LastFM, which clearly correspond to four different music genres –"*J-pop*", "*blues rock*", "*new wave*", and "*industrial metal*". This qualitative result demonstrates CoDBand's capability in recognizing the potential clustering structure of users' preferences solely from their click feedback.

MovieLens: We also evaluated the algorithms with data extracted from the MovieLens 20M dataset that contains 20 million ratings with 27,000 movies and 138,000 users [20]. We followed a similar procedure in [30] to pre-process the data to fit a contextual bandit setting. First, we extracted TF-IDF feature vectors using information like movie titles, genres, and tags provided by users. We then applied PCA to the resulting TF-IDF feature vectors, and retained the first 25 principle components as the context vectors, i.e., $d = 25$. Then we normalized all features to have a zero mean and unit variance. We converted ratings to binary reward by mapping non-zero ratings to 1, and zero ratings to 0. The event sequence is generated by first filtering out users with less than 3000 observations, and then at each time when a particular user u is served, the candidate arm pool for user u is generated by keeping the movie with nonzero reward at this time stamp and sampling another 24 zero-reward movies rated by this user, i.e., $|\mathcal{A}_t| = 25$.

We report the normalized accumulated reward of all algorithms in Figure 3 (c). It is worth noticing that all the bandit algorithms with collaborative learning, e.g. CLUB, SCLUB and CoDBand perform substantially better than the other baselines. This indicates that users in the MovieLens dataset share much interests in common, and therefore data sharing is of vital importance for improving the performance. We can observe that CoDBand accumulated reward much faster than CLUB and SCLUB in the early stage. This suggests CoDBand is capable of estimating a good clustering structure over users with limited number of observations available and as a result starting to benefit from the shared observations earlier than CLUB and SCLUB. We attribute this advantage to its DP model based model selection solution, which leverages the concentration of user groups in a population of users (e.g., social norm). Though



(a) Normalized reward on LastFM (b) Normalized reward on Delicious (c) Normalized reward on MovieLens (d) Normalized CTR on Yahoo! Today

Figure 3: Performance comparison on realworld datasets.

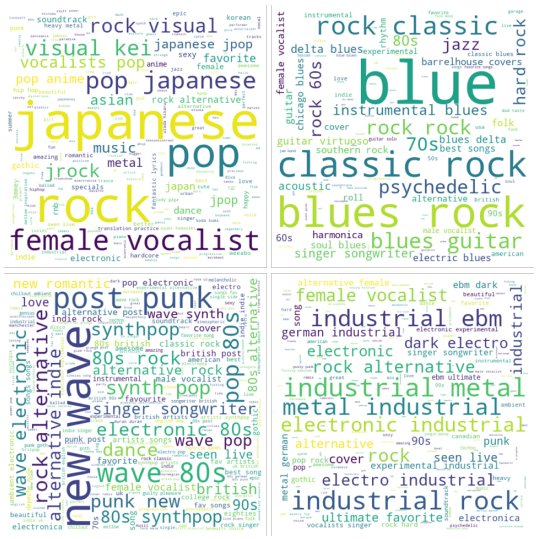


Figure 4: Word clouds for tags associated with the top-100 artists ranked by CoDBand's inferred global bandit models.

the non-stationary bandit algorithms dLinUCB and adTs also show improvement over standard LinUCB, not being able to utilize observations from other users make it hard for them to compete with the collaborative solutions on this dataset.

Yahoo! Today Module: Yahoo! Today Module recommendation dataset is a large-scale click stream dataset from the Yahoo Webscope program, which contains over 45 million user visits to Yahoo Today Module collected in 2009. For each visit, both the user and each of the 10 candidate articles, i.e. $|\mathcal{A}_t| = 10$, are associated with a feature vector of six dimensions ($d = 5$ excluding a bias term) [27]. We adopted the unbiased offline evaluation protocol in [28] to compare the algorithms with data extracted from the first day of the ten-day period from this dataset, which contains 4.6 million user visits. Click through rate (CTR) is used as the performance metric for all bandit algorithms. Similar to [27], we normalized the resulting CTR of different algorithms by the corresponding logged random strategy's CTR. In addition, this dataset does not provide user identities, we followed [42, 43] to cluster users into different groups and view the resulting groups as users.

The results are reported in Figure 3 (d). We can observe that CoDBand and CLUB show a faster and more steady rate in accumulating rewards than the other baselines, suggesting that considering collaboration among users is beneficial for this news recommendation scenario as well. While although AdTS exhibits faster increasing performance at the beginning, as it detects the changes in users' preference, its performance also deteriorates fast as it tends to make more false detections. It is also worth noticing that the simple baseline that attaches LinUCB to each individual user also performs reasonably, beating some of the other more complicated baselines. This suggests incorporating change detection or user clustering come with the risk of errors, e.g., false alarm in change detection causes the algorithm to discard observations when it is unnecessary, and including wrong user in the cluster introduces distortion to the learned model. These directly lead to the added regret comparing with standard baselines like LinUCB and SCLUB. On the other hand, the results in this experiment suggest CoDBand is more accurate in change detection and cluster identification, which ensures its advantage and flexibility against those more restrictive baselines.

6 CONCLUSIONS & FUTURE WORK

In this paper, we propose a collaborative dynamic bandit solution CoDBand for interactive recommendation in a non-stationary environment, where both user preferences and user dependencies change over time. We model the dynamic with Dirichlet process, and propose a Thompson sampling based contextual bandit solution for collaborative online learning. Rigorous regret analysis provides a valid performance guarantee of CoDBand for detecting the changes and correctly selecting the bandit models for recommendation. Extensive experiments on both synthetic and real-world datasets confirmed the effectiveness of CoDBand in recommendation, especially its advantages in addressing the cold start challenge.

In our current formulation, the change points are assumed to happen at arbitrary and unknown time steps, and as a result they are outside of our Bayesian inference framework. A more elegant way is to introduce a prior on the change points [3], and use Thompson sampling to address both change detection and model selection [33]. Also in our current stochastic process model of the changing environment, we only explicitly modeled the popularity of bandit models with a Dirichlet process model. But many other types of important observations can be considered, such as friendship and recency of a model. We would like further extend our Dirichlet process model with other stochastic process models, e.g. Hawkes process [22], to further enhance our solution in handling a complex changing environment.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful and constructive comments. This work is supported by by National Science Foundation under grant IIS-1838615, IIS-1618948 and IIS-1553568.

REFERENCES

[1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *NIPS*. 2312–2320.

[2] Marc Abeille and Alessandro Lazaric. 2017. Linear Thompson Sampling Revisited. In *Proceedings of the 20th AISTATS*. 176–184.

[3] Ryan Prescott Adams and David JC MacKay. 2007. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742* (2007).

[4] Shipra Agrawal and Navin Goyal. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *Proceedings of the 30th ICML*. 1220–1228.

[5] Charles E. Antoniak. 1974. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics* 2, 6 (11 1974), 1152–1174.

[6] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Maching Learning* 47, 2-3 (May 2002), 235–256.

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

[8] John S. Breese, David Heckerman, and Carl Kadie. 1998. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Technical Report MSR-TR-98-12. Microsoft Research. 18 pages. http://research.microsoft.com/apps/pubs/default.aspx?id=69656

[9] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. 2013. A gang of bandits. (2013), 737–745.

[10] Olivier Chapelle and Lihong Li. 2011. An Empirical Evaluation of Thompson Sampling. In *NIPS 2011*. 2249–2257.

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

[12] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. 2019. Learning to optimize under non-stationarity. In *The 22nd AISTATS*. 1079–1087.

[13] Michael D Escobar and Mike West. 1995. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association* 90, 430 (1995), 577–588.

[14] Thomas S. Ferguson. 1973. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics* 1, 2 (03 1973), 209–230.

[15] Leon Festinger. 1954. A Theory of Social Comparison Processes. *Human Relations* 7, 2 (1954), 117–140.

[16] Aurélien Garivier and Eric Moulines. [n.d.]. On Upper-Confidence Bound Policies for Non-stationary Bandit Problems. In *arXiv preprint arXiv:0805.3415 (2008)*.

[17] Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. 2017. On context-dependent clustering of bandits. In *ICML*. 1253–1262.

[18] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online Clustering of Bandits. In *ICML'14*. 757–765.

[19] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2015. Adapting to user preference changes in interactive recommendation. In *24th IJCAI*.

[20] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[21] Cedric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michele Sebag. 2006. Multi-armed Bandit, Dynamic Environments and Meta-Bandits. (2006).

[22] Alan G Hawkes and David Oakes. 1974. A cluster process representation of a self-exciting process. *Journal of Applied Probability* 11, 3 (1974), 493–503.

[23] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When People Change their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments. In *Proceedings of 12th WSDM*. ACM, 297–306.

[24] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In *NIPS*. 1297–1305.

[25] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[26] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[27] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of 19th WWW*. ACM, 661–670.

[28] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of 4th WSDM*. ACM, 297–306.

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

[30] Shuai Li, Claudio Gentile, and Alexandros Karatzoglou. 2016. Graph clustering bandits for recommendation. *arXiv preprint arXiv:1605.00596* (2016).

[31] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative Filtering Bandits. In *Proceedings of the 39th ACM SIGIR*. 539–548.

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

[33] Joseph Mellor and Jonathan Shapiro. 2013. Thompson sampling in switching environments with bayesian online change point detection. *arXiv preprint arXiv:1302.3721* (2013).

[34] Joshua L Moore, Shuo Chen, Douglas Turnbull, and Thorsten Joachims. 2013. Taste Over Time: The Temporal Dynamics of User Preferences.. In *ISMIR*. 401–406.

[35] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. 2012. Modeling and predicting behavioral dynamics on the web. In *Proceedings of the 21st international conference on World Wide Web*. 599–608.

[36] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[37] Yoan Russac, Claire Vernade, and Olivier Cappé. 2019. Weighted Linear Bandits for Non-Stationary Environments. In *NIPS*. 12017–12026.

[38] Daniel Russo and Benjamin Van Roy. 2014. Learning to optimize via posterior sampling. *Mathematics of Operations Research* 39, 4 (2014), 1221–1243.

[39] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of 10th WWW*. ACM, 285–295.

[40] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. 2007. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM KDD*. ACM, 717–726.

[41] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization Bandits for Interactive Recommendation. In *AAAI*. 2695–2702.

[42] Qingyun Wu, Naveen Iyer, and Hongning Wang. 2018. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR*. ACM, 495–504.

[43] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. 2016. Contextual Bandits in a Collaborative Environment. In *Proceedings of the 39th International ACM SIGIR*. ACM, 529–538.

[44] Qingyun Wu, Hongning Wang, Liangjie Hong, and Yue Shi. 2017. Returning is believing: Optimizing long-term user engagement in recommender systems. In *Proceedings of the 26th ACM CIKM*. ACM, 1927–1936.

[45] Kaige Yang, Laura Toni, and Xiaowen Dong. 2020. Laplacian-regularized graph bandits: Algorithms and theoretical analysis. In *AISTATS*. 3133–3143.

[46] Jia Yuan Yu and Shie Mannor. 2009. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th ICML*. ACM, 1177–1184.

[47] Peng Zhao, Lijun Zhang, Yuan Jiang, and Zhi-Hua Zhou. 2020. A simple approach for non-stationary linear bandits. In *Proceedings of the 23rd AISTATS*, Vol. 2020.