# Contextual-Bandit Based Personalized Recommendation with Time-Varying User Interests

**Xiao Xu,**[1] **Fang Dong,**[2*] **Yanghua Li,**[2] **Shaojian He,**[2] **Xin Li**[2]
[1]Cornell University, Ithaca, NY, USA
[2]Alibaba Group, Hangzhou, Zhejiang, China
xx243@cornell.edu, {dongfang.df, shaojian.he, xin.l}@alibaba-inc.com, yichen.lyh@taobao.com

## Abstract

A contextual bandit problem is studied in a highly non-stationary environment, which is ubiquitous in various recommender systems due to the time-varying interests of users. Two models with disjoint and hybrid payoffs are considered to characterize the phenomenon that users' preferences towards different items vary differently over time. In the disjoint payoff model, the reward of playing an arm is determined by an arm-specific preference vector, which is piecewise-stationary with asynchronous and distinct changes across different arms. An efficient learning algorithm that is adaptive to abrupt reward changes is proposed and theoretical regret analysis is provided to show that a sublinear scaling of regret in the time length $T$ is achieved. The algorithm is further extended to a more general setting with hybrid payoffs where the reward of playing an arm is determined by both an arm-specific preference vector and a joint coefficient vector shared by all arms. Empirical experiments are conducted on real-world datasets to verify the advantages of the proposed learning algorithms against baseline ones in both settings.

## Introduction

Online learning has long been adopted as one of the archetypal formulations in various applications including online advertising (Li et al. 2010b), personalized recommendation (Li et al. 2010a), and information retrieval (Yue and Joachims 2009). A classic framework for online learning is the multi-armed bandit (MAB) model with a set of $K$ arms (representing all possible actions) and a single player. At each time, the player chooses one of the $K$ arms to play and obtains a random reward generated from an unknown distribution specific to the chosen arm (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002). In order to maximize the total expected reward over a time horizon of length $T$, the learner needs to design an arm selection policy that balances an intrinsic tradeoff between exploring the unknown reward model and exploiting the current knowledge to maximize the instantaneous gain. The performance of an arm selection policy is measured by regret, which is the expected cumulative reward loss against an omniscient player who knows the reward model and always plays the best arm.

In recent years, contextual bandits (Langford and Zhang 2007; Li et al. 2010a), a variation of the classical MAB model, has received large attention due to its success in various online services where context information associated with either users or items is available. It has been assumed that the unknown reward model of an arm is determined by the given context. Through leveraging the context information, a number of new learning algorithms have been developed to achieve better performance compared with context-free ones in the classical setting (Li et al. 2010a; Wang, Wu, and Wang 2016).

While most existing studies on contextual bandits assume a stationary environment where the unknown reward model is fixed over time given the context information, real-world applications are usually dynamic due to the time-varying interests of users. For example, it has been observed that the click behaviors of users over different news articles evolve over time in both Google News (Liu, Dolan, and Pedersen 2010) and Yahoo News (Zeng et al. 2016). Without the capability of detecting potential changes in the underlying reward model, existing algorithms may lead to sub-optimal decisions using out-of-date observations.

### Main Results

In this paper, we study a more realistic setting with non-stationary user interests under the contextual bandit framework. Specifically, we assume that the preferences of users towards items are piecewise-stationary, i.e., the reward model may undergo abrupt changes but between two consecutive change points, the model remains fixed. Furthermore, we consider asynchronous and distinct reward changes across different items, which is a common phenomenon in real applications. For example, in news recommendation, changes on the preferences of readers towards different news categories are triggered by the occurrence of related hot events, which are unlikely to happen at the same time. In e-commerce platforms, customers' life-long interests over different products also exhibit distinct changes: a customer is more likely to purchase toys in his childhood while in adulthood, he may become more interested in sport-related

*Corresponding author.

products. However, the preference changes over the two categories can happen asynchronously as there may exist a time period (e.g., adolescence) when the customer likes both toys and sports. Moreover, it is possible that the customer's preferences towards other products (e.g., snakes) remain unchanged over time. To characterize such phenomena, we consider two reward models with disjoint and hybrid payoffs as described below.

In the disjoint payoff model, we assume that the expected reward of playing an arm[1] is the inner product of the given context vector and an arm-specific unknown coefficient vector, which represents the preference of the user towards the arm. The preference vector is assumed to be piecewise-stationary and the change points are different across arms. We propose an upper confidence bound (UCB) based algorithm that selects arms by estimating the unknown preference vectors from past observations. To address the challenge of time-varying interests, the algorithm adopts a change-detection procedure to identify potential changes on the preference vectors. Once a change is detected, an efficient restart is applied to re-estimate the preference vector using up-to-date observations. We provide theoretical regret analysis of the proposed algorithm and show that a sublinear scaling of regret in $T$ is achieved. We further extend the algorithm to a more general setting with hybrid payoffs. In addition to the arm-specific preference vector, the expected reward in the hybrid model also depends on a joint coefficient vector shared by all arms, which corresponds to the time-invariant component of the user interests. We conduct experiments on real-world datasets to evaluate the performance of the proposed algorithms in both settings.

### Related Work

Under the MAB framework, a large number of learning algorithms have been developed to balance the tradeoff between exploration and exploitation. Example algorithms include Thompson sampling (Thompson 1933; Agrawal and Goyal 2012), UCB (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002), and epsilon-greedy (Sutton and Barto 1998) in the classical context-free bandit setting, epoch-greedy (Langford and Zhang 2007) and LinUCB (Li et al. 2010a) in the contextual bandit setting. However, those algorithms assume a stationary environment that hardly holds in real applications.

In addressing the issue of non-stationary environment, various reward models have been studied in the literature. One of the most commonly accepted models is the piecewise-stationary reward model, which allows abrupt reward changes at certain unknown time points but remains fixed between two consecutive change points. Under the piecewise-stationary assumption, the problem has been well studied in the classical context-free setting. A number of learning algorithms have been developed that adapts to the abrupt reward changes by either triggering a reset of the learning algorithm after the detected changes (Hartland et al. 2007; Yu and Mannor 2009; Cao et al. 2019) or applying a discount factor on past observations (Garivier and Moulines 2011). Theoretical regret analysis showed that a sublinear scaling of regret in $T$ is achieved.

Within the contextual bandit setting, however, only a few recent studies have taken the issue of non-stationary environment into consideration. In (Hariri, Mobasher, and Burke 2015), a contextual Thompson sampling algorithm with a change detection module was proposed but theoretical regret analysis is lacking. In (Wu, Iyer, and Wang 2018), a hierarchical bandit algorithm was developed that detects and adapts to changes by maintaining a suite of contextual bandit models and a regret sublinear in $T$ was proved. However, the existing results assumed a uniform payoff model where all arms share a common coefficient vector representing the user interests, which fails to characterize the fact that users' preferences towards different items vary differently. Recently, a so-called context-dependent property was considered in (Wu et al. 2019) where arms are partitioned into change-invariant and change-sensitive ones based on their context vectors to characterize the distinct reward changes. However, the changes are not completely asynchronous across arms. A more detailed comparison between various models is discussed in the next section.

## Problem Formulation

Consider a contextual bandit problem with $K$ arms and a time horizon of length $T$. At each time $t$, a recommender system observes the current player $u_t$ with a $d$-dimensional feature vector $x_{u_t}$. A subset $\mathcal{A}_t \subseteq [K]$ of arms is available for selection and each arm $a \in \mathcal{A}_t$ is associated with an $m$-dimensional feature vector $y_a$. The system recommends an arm $a_t$ to the user $u_t$ and observes a random reward $r_{u_t,a_t}(t)$ (i.e., clicks, ratings, etc.), which is drawn from an unknown distribution $f(\cdot; x_{u_t}, y_{a_t}, W(t))$ where $W(t) = (w_1(t), ..., w_m(t)) \in \mathbb{R}^{d\times m}$ is a time-varying unknown weight matrix representing the preferences of users towards items in the feature space. The conditional expectation of the reward $r_{u_t,a_t}(t)$ given the feature vectors and the weight matrix is defined as

$$\mathbb{E}[r_{u_t,a_t}(t)|x_{u_t}, y_{a_t}, W(t)] = x_{u_t}^T W(t) y_{a_t}. \quad (1)$$

Without loss of generality, we assume that the probability distribution of the random reward $r_{u_t,a_t}(t)$ is sub-Gaussian with parameter $\sigma$.[2] The objective is an arm selection policy $\pi$ that maximizes the expected cumulative reward over the entire time horizon, i.e., $\mathbb{E}[\sum_{t=1}^{T} r_{u_t,\pi_t}(t)]$ where $\pi_t$ is the arm selected by policy $\pi$ at time $t$. Equivalently, we may find a policy $\pi$ that minimizes the expected cumulative regret defined as the expected reward loss of policy $\pi$ against the best policy in the known model case, i.e.,

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} r_{u_t,a_t^*}(t) - r_{u_t,\pi_t}(t)\right], \quad (2)$$

where $a_t^*$ is the arm with the largest expected reward at $t$.

[1] An arm corresponds to an item in the recommender system. Two terms are used interchangeably thoughout the paper.

[2] A random variable $Y$ with mean $\mu$ is sub-Gaussian with parameter $\sigma$ if $\mathbb{E}[e^{\lambda(Y-\mu)}] \leq e^{\sigma^2\lambda^2/2}, \forall\lambda \in \mathbb{R}$.

In the stationary scenario where $W(t)$ is fixed over time (i.e., $W(t) \equiv W$), the above formulation is equivalent to the standard contextual bandit model with linear payoffs as studied in the literature (Auer 2002; Chu et al. 2011; Agrawal and Goyal 2013). Specifically, let $z_{u_t,a} = \text{vec}(x_{u_t} y_a^T)$ be the context vector[3] associated with arm $a$ at time $t$ and $\beta = \text{vec}(W)$ be an unknown preference vector. It is clear that $\mathbb{E}[r_{u_t,a}(t)|x_{u_t}, y_a, W] = z_{u_t,a}^T \beta$. The unknown preference vector $\beta$ can be efficiently estimated in an online fashion at each time $t$ via ridge regression (see the LinUCB algorithm in (Li et al. 2010a)), and is applied to the reward estimation and the arm selection at time $t+1$.

In the non-stationary scenario, however, estimating $W(t)$ is in general challenging if elements of $W(t)$ vary arbitrarily: without constraints on the variation of the parameters, estimating $W(t)$ is impossible. Moreover, to characterize the fact that the preferences of users towards different items vary asynchronously and distinctly, elements of $W(t)$ should exhibit different varying patterns. However, the effects of different elements of $W(t)$ on the obtained rewards are difficult to be distinguished, which leads to the challenge of detecting unknown changes on each element from reward observations. To address the two challenges, we turn to consider approximated reward models to simplify the problem, and adopt certain assumptions on the varying patterns of the reward parameters. Specifically, we study two reward models, i.e., the *disjoint payoff model* and the *hybrid payoff model*.

## Disjoint Payoff Model

In the disjoint payoff model, we let the combination of $W(t)$ and $y_a$, i.e., $\theta_a(t) = W(t) y_a$ be the unknown preference vector associated with arm $a$ at time $t$. The expected reward of recommending item $a$ to user $u$ at time $t$ is then equivalent to the inner product of $x_u$ and $\theta_a(t)$, i.e.,

$$\mathbb{E}[r_{u,a}(t)|x_u, \theta_a(t)] = x_u^T \theta_a(t). \tag{3}$$

We adopt a piecewise-stationary assumption on $\theta_a(t)$. To be specific, the time horizon is partitioned into $M_a$ stationary segments with $M_a + 1$ change points $\{\nu_a^{(\ell)}\}_{\ell=0}^{M_a}$ where $\nu_a^{(0)} = 0$ and $\nu_a^{(M_a)} = T$. Within each segment, $\theta_a(t)$ is assumed to be fixed, i.e., $\theta_a(t) \equiv \theta_a^{(\ell)}$, $\forall t \in [\nu_a^{(\ell-1)} + 1, \nu_a^{(\ell)}]$, $0 \le \ell \le M_a$. The sequence of changes points may be different across arms, which characterizes the fact that users' preferences towards different items may change asynchronously.

## Hybrid Payoff Model

In a more general model with hybrid payoffs, we further assume that $W(t)$ consists of both a time-varying component $W_v(t)$ and a time-invariant component $W_c$, i.e., $W(t) = W_v(t) + W_c$. In particular, $W_v(t)$ represents the dynamically changing preferences of users towards items and $W_c$ represents the stationary internal interests of users that are unaffected by the external environment.

For the time-varying component $W_v(t)$, we adopt the same approximation method as the one used in the disjoint setting and define $\theta_a(t) = W_v(t) y_a$ be the arm-specific preference vector of arm $a$. For the time-invariant component, we define $\beta = \text{vec}(W_c)$ be the joint coefficient vector shared by all arms. It is not difficult to see that the expected reward of recommending arm $a$ to user $u$ at time $t$ satisfies that

$$\mathbb{E}[r_{u,a}(t)|x_u, z_{u,a}, \theta_a(t), \beta] = x_u^T \theta_a(t) + z_{u,a}^T \beta, \tag{4}$$

where $z_{u,a} = \text{vec}(x_u y_a^T)$ is a $k$-dimensional ($k = d \times m$) cross-feature vector of the user-item pair. We adopt the same piecewise-stationary assumption on the arm-specific vectors $\theta_a(t)$ as that assumed in the disjoint setting, which allows asynchronous changes across different arms.

## Comparisons with Existing Models

We first compare the two payoff models with the stationary ones in the classical contextual bandit setting. It is clear that both models are direct extensions of the stationary payoff models studied in (Li et al. 2010a) where the preference vectors $\theta_a(t), \forall a$ are assumed to be fixed over time. As discussed in the introduction section, it is more realistic to consider non-stationary preferences in real applications as users' interests are in general time-varying.

In considering the non-stationary environment within the contextual bandit setting, the majority of existing studies (Wu, Iyer, and Wang 2018; Wu et al. 2019) assumed a uniform (joint) payoff model where all arms share a common coefficient vector $\theta_u(t)$ representing the interests of user $u$. The expected reward is thus defined as

$$\mathbb{E}[r_{u,a}(t)|y_a, \theta_u(t)] = y_a^T \theta_u(t). \tag{5}$$

Notice that the uniform payoff model is another approximation of the bilinear model defined in (1): $\theta_u(t)$ is the combination of $x_u$ and $W(t)$, i.e., $\theta_u(t) = W^T(t) x_u$. In the literature, $\theta_u(t)$ is assumed to be piecewise-stationary to model the time-varying interests of users. The fact that users' preferences change differently towards different items is, however, not characterized.

The issue was partially addressed in (Wu et al. 2019) where the so-called *context-dependent* property was considered. It has been assumed that the expected rewards of certain arms are insensitive to the changes of $\theta_u(t)$ (i.e., for some stationary periods $i$ and $j$, $|y_a^T \theta_u^{(i)} - y_a^T \theta_u^{(j)}| \le \Delta_L$, where $\Delta_L$ is a small constant), while the other arms are change-sensitive. The partition of arms based on their context vectors models the distinct reward changes on different arms. However, the change points across arms are not completely asynchronous: it has been assumed in (Wu et al. 2019) that between any two stationary periods, there should be a sufficient number of change-sensitive arms undergo perceivable changes to distinguish the two periods. As a result, the user preferences towards a large fraction of arms change simultaneously at the change points of $\theta_u(t)$.

Moreover, we further study a general hybrid payoff model consisting of both arm-specific and joint preference vectors that correspond to the time-varying and the time-invariant interests of users respectively. To the best of our knowledge, the hybrid payoff model with dynamically changing user interests has not been studied in the literature.

[3] $\text{vec}(\cdot)$ is the vectorization operator that concatenates columns of a matrix to a single vector.

## Piecewise-Stationary LinUCB Algorithm under the Disjoint Payoff Model

We first consider the disjoint payoff model in this section. The key to achieving the objective of minimizing regret under the assumption of piecewise-stationary payoffs is to i) estimate the preference vectors accurately, and ii) detect the abrupt changes timely and correctly. We propose a Piecewise-Stationary LinUCB (PSLinUCB) algorithm to address the two issues.

To estimate the preference vectors, we adopt a learning structure similar to that of the LinUCB algorithm (proposed in (Li et al. 2010a) in the stationary contextual bandit setting). In particular, the unknown preference vectors $\theta_a(t), \forall a$ are estimated through ridge regression and can be updated incrementally at each time $t$. To detect the preference changes timely and correctly, the key technique adopted in the algorithm is to maintain a sliding window for each arm consisting of the most recent reward observations from the arm. If the preference vector learned from observations before the sliding window cannot accurately predict the rewards observed within the window, it is likely that the preference vector has changed. A new model should then be rebuilt based on the observations after the change point.

To be more specific, the estimation and the change detection of the preference vector $\theta_a(t)$ of every arm $a$ can be executed independently in the disjoint payoff model. For every arm $a$, the algorithm maintains a sliding window $SW_a$ and three different models $M_a^{pre}$, $M_a^{cur}$, and $M_a^{cum}$. The sliding window $SW_a$ of length $\omega$ consists of the $\omega$ latest observations from arm $a$ (including the observed context vectors and the obtained rewards). $M_a^{pre}$ consists of necessary statistics for estimating the preference vector $\theta_a(t)$. It is learned from observations after the last detected change point and before the sliding window $SW_a$. Similarly, $M_a^{cur}$ with the same set of statistics is learned from observations within the sliding window, and $M_a^{cum}$ is learned from all observations from the last detected change point to the current time step. In the following subsections, we describe the details of the three models and their usage in the two key components of the PSLinUCB-Disjoint algorithm: (i) parameter estimation and arm selection, and (ii) change detection and model update.

### Parameter Estimation and Arm Selection

In each of the three models $M_a^{pre}, M_a^{cur}$, and $M_a^{cum}$, the preference vector $\theta_a(t)$ can be estimated by applying ridge regression to the associated set of observations. Without loss of generality, we take $M_a^{cum}$ for an example to illustrate the estimation process. Denote $\{(x_{u_t}, r_{u_t,a})\}_{t\in\mathcal{I}_a^{cum}}$ as the set of observations where $\mathcal{I}_a^{cum}$ is the set of time steps when arm $a$ is played from its last detected change time (initialized to be 0) to the current time step. $\hat{\theta}_a^{cum}$ can be estimated as $\hat{\theta}_a^{cum} = (\mathbf{A}_a^{cum})^{-1}\mathbf{b}_a^{cum}$ where $\mathbf{A}_a^{cum} = \mathbf{I}_d + \sum_{t\in\mathcal{I}_a^{cum}} x_{u_t}x_{u_t}^T$, $\mathbf{I}_d$ is a $d \times d$ identity matrix, and $\mathbf{b}_a^{cum} = \sum_{t\in\mathcal{I}_a^{cum}} r_{u_t,a}(t)x_{u_t}$. The statistics $\mathbf{A}_a^{cum}$ and $\mathbf{b}_a^{cum}$ can be updated incrementally as described in(Li et al. 2010a).

Based on the estimated preference vector $\hat{\theta}_a^{cum}$ of every arm $a \in \mathcal{A}_t$, we select arms according to the UCB principle to balance the tradeoff between exploration and exploitation. Similar to the LinUCB algorithm, we define a UCB index for every arm $a$ at time $t$ as $x_{u_t}^T\hat{\theta}_a^{cum} + \alpha\sqrt{x_{u_t}^T(\mathbf{A}_a^{cum})^{-1}x_{u_t}}$. The arm with the greatest index is selected and the reward observations from the selected arm is used to update the corresponding models.

**Algorithm 1** Piecewise-Stationary LinUCB under the Disjoint Payoff Model (PSLinUCB-Disjoint)

**Input**: $\alpha > 0, \omega \in \mathbb{N}^+, \delta > 0$.
**for** $t = 1, 2, ..., T$ **do**
  Observe the feature vector $x_{u_t}$ of the current user $u_t$ and the set of available arms $\mathcal{A}_t$.
  *//Parameter Estimation and Arm Selection*
  **for** $a \in \mathcal{A}_t$ **do**
    **if** $a$ is new **then**
      $\mathbf{A}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{I}_d$, $\mathbf{b}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times 1}$, $SW_a \leftarrow \emptyset$.
    $\hat{\theta}_a^{cum} \leftarrow (\mathbf{A}_a^{cum})^{-1}\mathbf{b}_a^{cum}$.
    $p_{t,a} \leftarrow x_{u_t}^T\hat{\theta}_a^{cum} + \alpha\sqrt{x_{u_t}^T(\mathbf{A}_a^{cum})^{-1}x_{u_t}}$.
  Play $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$, obtain reward $r_{u_t,a_t}(t)$.
  Append $(x_{u_t}, r_{u_t,a_t}(t))$ to the end of $SW_{a_t}$.
  $\mathbf{A}_{a_t}^{\{cur,cum\}} \leftarrow \mathbf{A}_{a_t}^{\{cur,cum\}} + x_{u_t}x_{u_t}^T$.
  $\mathbf{b}_{a_t}^{\{cur,cum\}} \leftarrow \mathbf{b}_{a_t}^{\{cur,cum\}} + r_{u_t,a_t}(t)x_{u_t}$.
  *//Change Detection and Model Update*
  **if** $|SW_{a_t}| \geq \omega$ **then**
    $\hat{\theta}_{a_t}^{pre} \leftarrow (\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{b}_{a_t}^{pre}$.
    Let $SW_{a_t} = \{(x_s, r_s)\}_{s=1}^{\omega}$.
    **if** $|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T\hat{\theta}_{a_t}^{pre} - r_s)| \geq \delta$ **then**
      $\mathbf{A}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{A}_{a_t}^{cur}$, $\mathbf{b}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{b}_{a_t}^{cur}$,
      $\mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{I}_d$, $\mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times 1}$, $SW_{a_t} \leftarrow \emptyset$.
    **else**
      $(x_1, r_1) \leftarrow$ Popleft$(SW_{a_t})$.
      $\mathbf{A}_{a_t}^{pre} \leftarrow \mathbf{A}_{a_t}^{pre} + x_1x_1^T$, $\mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{A}_{a_t}^{cur} - x_1x_1^T$
      $\mathbf{b}_{a_t}^{pre} \leftarrow \mathbf{b}_{a_t}^{pre} + r_1x_1$, $\mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{b}_{a_t}^{cur} - r_1x_1$.

### Change Detection and Model Update

To detect potential changes on an arm $a$, we use $M_a^{pre}$ to predict the rewards of playing arm $a$ at the time steps within the sliding window. We compare the predicted rewards with the observed ones to test if the model learned from earlier data still fits the current observations. To be specific, let $\{(x_s, r_s)\}_{s=1}^{\omega}$ be the set of observations within the sliding window. We check if $|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T\hat{\theta}_{a_t}^{pre} - r_s)| \geq \delta$, where $\delta$ is an input threshold.

If a change is detected on arm $a$, i.e., the average distance between the predicted rewards and the observed ones in the sliding window exceeds the threshold, we have to restart the learning process of arm $a$ using only observations after the detected change point. Instead of re-constructing a new model without using history data, we exploit the observations within the sliding window again as a warm-start to accelerate learning. In particular, we initialize $M_a^{cum}, M_a^{pre}$, which are used for arm selection and change detection re-

spectively, with $M_a^{cur}$, which is the model learned from the latest observations after the change point (i.e., within the sliding window). The sliding window is then emptied to collect new observations until its length reaches $\omega$ again.

If no change is detected on arm $a$, i.e., the earlier and the current reward observations follow the same model, we should keep both sets of data to enhance the estimation accuracy. Therefore, $M_a^{cum}$ keeps unchanged and the sliding window is right-shifted by one time step. Note that $M_a^{pre}$ and $M_a^{cur}$ should be updated accordingly after the right-shifting of $SW_a$. The detailed implementation is summarized in Algorithm 1. Note that the computation complexity in each time step is $O(Kd^3)$ (a finite number of matrix operations for each arm) and the memory size required for learning is $O(K(d^2+d\omega))$ (three sets of statistics and a sliding window for each arm).

### Regret Analysis

In this subsection, we prove an upper bound on the regret of the proposed PSLinUCB-Disjoint algorithm. We first make several modifications on the algorithm without changing the underlying key strategies to get rid of certain technical difficulties in the theoretical analysis.

The modification includes three steps. First, to avoid heavy dependency between the estimation and the change detection of the underlying parameters and across different time steps, the observations in the sliding-window are not reused for initialization after a detected change. Besides, the change detection procedure only uses observations within the sliding window rather than all observations after the last detected change (note that $M^{pre}$ uses observations before the sliding window). Second, once a change is detected on an arm, the learning procedures of all arms get restarted. Finally, a round-robin exploration step is added to guarantee sufficient exploration of every arm so that the changes in the reward models can be detected timely. Due to the page limit, the details of the modified algorithm are summarized in the appendix. Based on certain mild assumptions, we prove an upper bound on regret in the following theorem.

**Theorem 1.** *The cumulative regret of the modified PSLinUCB algorithm under the disjoint payoff model satisfies:*

$$R(T) \leq C_1\omega\sqrt{MKT} + C_2\sqrt{MTdK\log\frac{T}{d}}, \quad (6)$$

*where $C_1, C_2$ are constants independent of $T$ and $M$ is the number of total piecewise-stationary segments, i.e.,*

$$M = 1 + \sum_{t=1}^{T-1} \mathbb{I}(\theta_a(t) \neq \theta_a(t-1) \text{ for some } a \in \mathcal{A}). \quad (7)$$

*Proof.* See the appendix. □

**Remark 1.** *Assume that $M \ll T$. The cumulative regret achieved by the modified PSLinUCB-Disjoint algorithm has a sublinear scaling in $T$, i.e., $R(T) \sim \tilde{O}(\sqrt{T})$ where the $\tilde{O}$ notation hides the logarithmic factor. In other words, the average regret per time step diminishes to zero as $T \to \infty$.*

## Extension to the Hybrid Payoff Model

In the hybrid payoff model, the preference of a user towards an arm $a$ is determined by both an arm-specific preference vector $\theta_a(t)$ and a joint coefficient vector $\beta$, which should be estimated simultaneously. Therefore, in addition to a sliding window $SW_a$ and three models $M_a^{pre}, M_a^{cur}$, and $M_a^{cum}$ for each arm $a$, the PSLinUCB-Hybrid algorithm maintains two global models $G^{pre}$ and $G^{cum}$ to estimate $\beta$. Specifically, $G^{pre}$ is the model learned from the observations from all arms before their sliding windows and is used for change detection. $G^{cum}$ is the model learned from the observations from all arms up to the current time step and is used for arm selection. The statistics in the two global models are obtained by applying ridge regression to the associated data. Due to the page limit, we omit the detailed theoretical derivations of ridge regression and only describe the process of updating the arm-specific and the global parameters.

### Parameter Estimation and Arm Selection

By applying ridge regression to the observed data, it can be shown that the joint coefficient vector $\hat{\beta}^{cum}$ is estimated as $\hat{\beta}^{cum} = (\mathbf{A}_0^{cum})^{-1}\mathbf{b}_0^{cum}$ where $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are coupled with arm-specific parameters $\mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}$ and $\mathbf{b}_{a_t}^{cum}$. Therefore, the global and the arm-specific parameters should be updated simultaneously. Specifically, $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are initialized to $\mathbf{I}_m, \mathbf{0}_{m\times k}$ respectively and the parameters are updated as follows:

$$\begin{aligned}
\mathbf{A}_0^{cum} \leftarrow& \mathbf{A}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum},\\
\mathbf{b}_0^{cum} \leftarrow& \mathbf{b}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum},\\
\mathbf{A}_{a_t}^{cum} \leftarrow& \mathbf{A}_{a_t}^{cum} + x_{u_t}x_{u_t}^T,\\
\mathbf{B}_{a_t}^{cum} \leftarrow& \mathbf{B}_{a_t}^{cum} + x_{u_t}z_{u_t,a_t}^T,\\
\mathbf{b}_{a_t}^{cum} \leftarrow& \mathbf{b}_{a_t}^{cum} + r_{u_t,a_t}(t)x_{u_t},\\
\mathbf{A}_0^{cum} \leftarrow& \mathbf{A}_0^{cum} + z_{u_t,a_t}z_{u_t,a_t}^T\\
&- (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum},\\
\mathbf{b}_0^{cum} \leftarrow& \mathbf{b}_0^{cum} + r_{u_t,a_t}(t)z_{u_t,a_t}\\
&- (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum}.
\end{aligned} \quad (8)$$

The update procedures of $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}$ and $\mathbf{b}_{a_t}^{cur}$ are similar to the ones of $\mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}$, and $\mathbf{b}_{a_t}^{cum}$ as described above.

In the arm selection step, we follow (Li et al. 2010a) to define the UCB index of arm $a$ at time $t$ as $x_{u_t}^T\hat{\theta}_a^{cum} + z_{u_t,a}^T\hat{\beta}^{cum} + \alpha\sqrt{s_{t,a}}$ where $\hat{\theta}_a^{cum} = (\mathbf{A}_a^{cum})^{-1}(\mathbf{b}_a^{cum} - \mathrm{B}_a^{cum}\hat{\beta}^{cum})$. The exploration term $s_{t,a} = s_{t,a}^{(1)} + s_{t,a}^{(2)} + s_{t,a}^{(3)}$ is computed as follows:

$$\begin{aligned}
s_{t,a}^{(1)} =& z_{u_t,a}^T(\mathbf{A}_0^{cum})^{-1}z_{u_t,a} + x_{u_t}^T(\mathbf{A}_a^{cum})^{-1}x_{u_t},\\
s_{t,a}^{(2)} =& -2z_{u_t,a}^T(\mathbf{A}_0^{cum})^{-1}(\mathbf{B}_a^{cum})^T(\mathbf{A}_a^{cum})^{-1}x_{u_t},\\
s_{t,a}^{(3)} =& x_{u_t}^T\mathbf{P}(\mathbf{A}_0^{cum})^{-1}\mathbf{P}^Tx_{u_t},
\end{aligned} \quad (9)$$

where $\mathbf{P} = (\mathbf{A}_a^{cum})^{-1}\mathbf{B}_a^{cum}$.

## Change Detection and Model Update

We conduct a change detection process similar to the one adopted in PSLinUCB-Disjoint to test if the preference vector $\theta_{a_t}(t)$ of arm $a_t$ changes or not. The occurrence of a change on $a_t$ is equivalent to $a_t$ being replaced by a new arm with a different set of arm-specific parameters specified by $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}$, and $\mathbf{b}_{a_t}^{cur}$. As a result, the global parameters $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ are coupled with two sets of arm-specific parameters associated with both the old and the new arm. In particular, the original arm-specific parameters (i.e., $\mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}$, and $\mathbf{b}_{a_t}^{cum}$) used in estimating $\mathbf{A}_0^{cum}$ and $\mathbf{b}_0^{cum}$ should be replaced by the aggregation of the parameters corresponding to the old arm (i.e., $\mathbf{A}_{a_t}^{pre}, \mathbf{B}_{a_t}^{pre}$, and $\mathbf{b}_{a_t}^{pre}$) and the new arm (i.e., $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}$, and $\mathbf{b}_{a_t}^{cur}$):

$$\begin{aligned}\mathbf{A}_0^{cum} &\leftarrow \mathbf{A}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{B}_{a_t}^{cum} \\ &\quad - (\mathbf{B}_{a_t}^{pre})^T(\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{B}_{a_t}^{pre} - (\mathbf{B}_{a_t}^{cur})^T(\mathbf{A}_{a_t}^{cur})^{-1}\mathbf{B}_{a_t}^{cur} \\ \mathbf{b}_0^{cum} &\leftarrow \mathbf{b}_0^{cum} + (\mathbf{B}_{a_t}^{cum})^T(\mathbf{A}_{a_t}^{cum})^{-1}\mathbf{b}_{a_t}^{cum}, \\ &\quad - (\mathbf{B}_{a_t}^{pre})^T(\mathbf{A}_{a_t}^{pre})^{-1}\mathbf{b}_{a_t}^{pre} - (\mathbf{B}_{a_t}^{cur})^T(\mathbf{A}_{a_t}^{cur})^{-1}\mathbf{b}_{a_t}^{cur}.\end{aligned} \tag{10}$$

Moreover, $G^{pre}$ is re-initialized to the updated $G^{cum}$ after the detected change and the arm-specific parameters are updated in the same way with that in the disjoint payoff case.

If no change is detected on $a_t$, the updating process is similar to that in PSLinUCB-Disjoint. The detailed implementation is summarized in Algorithm 2.

**Algorithm 2** Piecewise-Stationary LinUCB under Hybrid Payoff Model (PSLinUCB-Hybrid)

**Input**: $\alpha > 0, \omega \in \mathbb{N}^+, \delta > 0, k = d \times m$.
**Initialization**: $\mathbf{A}_0^{pre}, \mathbf{A}_0^{cum} = \mathbf{I}_k$. $\mathbf{b}_0^{pre}, \mathbf{b}_0^{cum} = \mathbf{0}_{k\times 1}$.
**for** $t = 1, 2, ..., T$ **do**
  *// Parameter Estimation and Arm Selection*
  Observe the feature vector $x_{u_t}$ of the current user $u_t$ and the cross-feature $z_{u_t,a}$ for every arm $a \in \mathcal{A}_t$ .
  $\hat{\beta}^{cum} = (\mathbf{A}_0^{cum})^{-1}\mathbf{b}_0^{cum}$.
  **for** $a \in \mathcal{A}_t$ **do**
    **if** $a$ is new **then**
      $\mathbf{A}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{I}_d$,$\mathbf{b}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times 1}$,
      $\mathbf{B}_a^{\{pre,cur,cum\}} \leftarrow \mathbf{0}_{d\times k}$, $SW_a \leftarrow \emptyset$.
    $\hat{\theta}_a^{cum} \leftarrow (\mathbf{A}_a^{cum})^{-1}(\mathbf{b}_a^{cum} - \mathbf{B}_a^{cum}\hat{\beta}^{cum})$.
    $p_{t,a} \leftarrow x_{u_t}^T\hat{\theta}_a^{cum} + z_{u_t,a}^T\hat{\beta}^{cum} + \alpha\sqrt{s_{t,a}}$.
  Play $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$, obtain reward $r_{u_t,a_t}(t)$.
  Append $(x_{u_t}, z_{u_t,a_t}, r_{u_t,a_t}(t))$ to the end of $SW_{a_t}$.
  Update $\mathbf{A}_0^{cum}, \mathbf{b}_0^{cum}, \mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}, \mathbf{b}_{a_t}^{cum}$ using (8).
  Update $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}, \mathbf{b}_{a_t}^{cur}$ in the same way with that in updating $\mathbf{A}_{a_t}^{cum}, \mathbf{B}_{a_t}^{cum}, \mathbf{b}_{a_t}^{cum}$ (replace $cum$ with $cur$).
  *//Change Detection and Model Update*
  **if** $|SW_{a_t}| \geq \omega$ **then**
    $\hat{\beta}^{pre} \leftarrow (\mathbf{A}_0^{pre})^{-1}\mathbf{b}_0^{pre}$.
    $\hat{\theta}_{a_t}^{pre} \leftarrow (\mathbf{A}_{a_t}^{pre})^{-1}(\mathbf{b}_{a_t}^{pre} - \mathbf{B}_{a_t}^{pre}\hat{\beta}^{pre})$.
    Let $SW_{a_t} = \{(x_s, z_s, r_s)\}_{s=1}^{\omega}$.
    **if** $|\frac{1}{\omega}(\sum_{s=1}^{\omega} x_s^T\hat{\theta}_{a_t}^{pre} + z_s^T\hat{\beta}^{pre} - r_s)| \geq \delta$ **then**
      Update $\mathbf{A}_0^{cum}, \mathbf{b}_0^{cum}, \mathbf{A}_0^{pre}, \mathbf{b}_0^{pre}$ using (10).
      $\mathbf{A}_0^{pre} \leftarrow \mathbf{A}_0^{cum}, \mathbf{b}_0^{pre} \leftarrow \mathbf{b}_0^{cum}, SW_{a_t} \leftarrow \emptyset$.
      $\mathbf{A}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{A}_{a_t}^{cur}, \mathbf{A}_{a_t}^{cur} \leftarrow \mathbf{I}_d$.
      $\mathbf{B}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{B}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times k}$.
      $\mathbf{b}_{a_t}^{\{pre,cum\}} \leftarrow \mathbf{b}_{a_t}^{cur}, \mathbf{b}_{a_t}^{cur} \leftarrow \mathbf{0}_{d\times 1}$.
    **else**
      $(x_1, z_1, r_1) \leftarrow$ Popleft$(SW_{a_t})$.
      Update $\mathbf{A}_0^{pre}, \mathbf{b}_0^{pre}, \mathbf{A}_{a_t}^{pre}, \mathbf{B}_{a_t}^{pre}, \mathbf{b}_{a_t}^{pre}$ according to (8) (replace $cum$ with $pre$ and $(x_{u_t}, z_{u_t,a_t}, r_{u_t,a_t}(t))$ with $(x_1, z_1, r_1)$).
      Update $\mathbf{A}_{a_t}^{cur}, \mathbf{B}_{a_t}^{cur}, \mathbf{b}_{a_t}^{cur}$ in the same way with that in updating $\mathbf{A}_{a_t}^{pre}, \mathbf{B}_{a_t}^{pre}, \mathbf{b}_{a_t}^{pre}$ (replace $pre$ with $cur$ and operation $+$ with $-$).

## Numerical Results

We use both synthetic and real-world data to evaluate the performance of the proposed learning algorithms under the disjoint and the hybrid payoff models. Due to the page limit, we only present the simulation results on two real-world datasets in this section. The results on synthetic data can be found in the appendix.

The first real-world dataset is a collection of user-visit log information from Yahoo! front page, which is widely used for algorithm evaluation in the contextual bandit setting (Li et al. 2010a; 2011). The Yahoo! dataset contains 45,811,883 user-visits to Yahoo Today Module in a ten-day period in May 2009. The log information of each user-visit includes a feature vector of the current user, a pool of candidate articles (arms) for recommendation associated with feature vectors, the recommended article, and the feedback from the user (click or not). It has been observed in (Wu et al. 2019) that the preferences of users towards different items are dynamically changing in this dataset.

The second dataset is extracted from the Last.fm online music system, which is made available on the HetRec 2011 workshop. This dataset contains 1892 users, 17,632 artists (arms), and 92,834 user-artist listening records. Each user may assign multiple tags to the listened artists, which can be preprocessed as the context information to fit into the contextual bandit setting. Following (Hartland et al. 2006), a non-stationary environment can be simulated.

We compare the proposed learning algorithms with the following baselines:

1. *Random*: a policy that selects arms uniformly at random.
2. *UCB* (Auer, Cesa-Bianchi, and Fischer 2002): one of the most well-known algorithms developed in the stationary context-free bandit setting.
3. *MUCB* (Cao et al. 2019): an extension of UCB to the context-free setting with piecewise-stationary rewards.
4. *LinUCB* (Li et al. 2010a; Chu et al. 2011): a representative algorithm for stationary contextual bandits. There are three versions of LinUCB corresponding to three different models with uniform, disjoint, and hybrid payoffs.
5. *DenBand* (Wu et al. 2019): a new algorithm developed under the uniform payoff model with piecewise-stationary rewards. Under the assumption of continuous rewards with little noise, the original algorithm only compares the predicted reward at a single time step with the

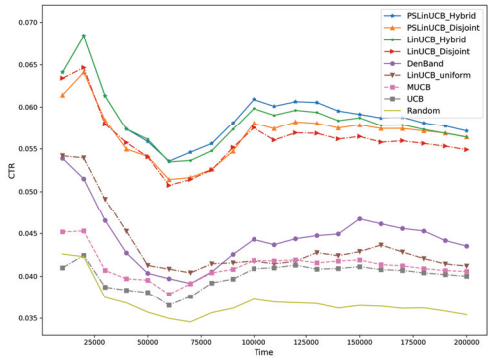


Figure 1: Average CTR v.s. time in the Yahoo! dataset.

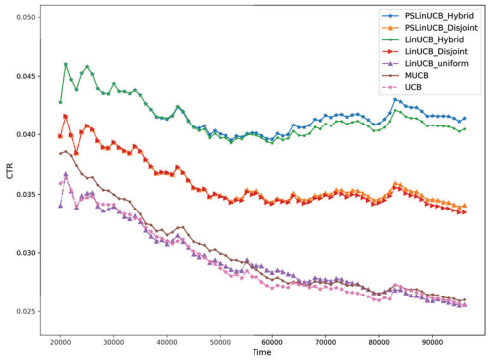


Figure 2: Average CTR v.s. time in the LastFM dataset.

observed one to detect potential changes. In cases with larger noise (e.g., binary rewards), we modify the algorithm by using observations at multiple time steps for change detection.

## Yahoo! Dataset

We randomly sample a subset of data from the original dataset for testing (i.e., each user-visit is selected independently with probability $0.1$). We adopt an unbiased offline evaluation method proposed in (Li et al. 2010a; 2011) to evaluate the online performance of the proposed learning algorithms and the baseline ones.

In Figure 1, we report the average Click-Through-Rate (CTR) of different algorithms versus time. We first observe that algorithms exploiting the context information (i.e., PSLinUCB, LinUCB, and DenBand) outperform context-free ones (i.e., UCB and MUCB). This observation is rather intuitive since context vectors provide significant side information on the preferences of users towards items. In addition, under each reward model (i.e., classical context-free bandits and contextual bandits with uniform, disjoint, and hybrid payoffs), the algorithm that adapts to reward changes outperforms the one that does not (i.e., MUCB v.s. UCB, DenBand v.s. LinUCB-uniform, PSLinUCB-Disjoint v.s. LinUCB-Disjoint, and PSLinUCB-Hybrid v.s. LinUCB-Hybrid). In particular, PSLinUCB-Disjoint achieves a performance gain of $2.7\%$ ($2.9\%$ at the peak) against LinUCB-Disjoint and PSLinUCB-Hybrid achieves an improvement of $1.3\%$ ($2\%$ at the peak) against LinUCB-Hybrid (see the appendix for details). The comparison results verify the assumption that users' interests are dynamically changing and should be taken into consideration in learning.

Moreover, within the contextual bandit setting, algorithms developed under the hybrid payoff model (i.e., PSLinUCB-Hybrid and LinUCB-Hybrid) or the disjoint payoff model (i.e., PSLinUCB-Disjoint and LinUCB-Disjoint) achieve better performance compared with the ones developed under the uniform payoff model (i.e., DenBand and LinUCB-Uniform). This is because the uniform payoff model fails to exploit the personalized interests of different users. An alternative approach is to learn the preferences of every user individually. However, the amount of data associated with a single user is rather limited. Furthermore, the performance gain of PSLinUCB over DenBand ($31.2\%$ under the hybrid model and $29.5\%$ under the disjoint model) verifies the fact that users' preferences towards different items vary differently. We also conduct experiments to anaylze the sensitivity of the proposed algorithms to the hyper-parameters. Due to the page limit, we leave the results in the appendix.

## LastFM Dataset

Given that the original LastFM dataset dose not provide context vectors of neither users nor items, we first preprocess the dataset to fit into the contextual bandit setting. Specifically, following the settings in (Cesa-Bianchi, Gentile, and Zappella 2013; Wu et al. 2019), we treat the 'listened artists' of each user as positive feedback. For each artist, we use its associated tags to create a TF-IDF feature vector and then apply PCA to reduce the dimension to 10. For each user, we adopt a method similar to the one used in (Li et al. 2010a) to generate a feature vector: we use matrix factorization to obtain a raw feature vector and then apply the K-means method to group users into 10 clusters. The final user feature is a 10-dimensional vector corresponding to the soft-membership of the user in the 10 clusters (computed with a Gaussian kernel and then normalized). In the experiment, we only consider artists that have been listened by at least 100 users and we follow (Wu, Iyer, and Wang 2018) to generate the log data. The results are presented in Figure 2 and similar conclusions with those in the experiment on the Yahoo! dataset can be drawn. In particular, PSLinUCB-Disjoint achieves a performance gain of $2\%$ against LinUCB-Disjoint and PSLinUCB-Hybrid achieves a performance gain of $2.4\%$ against LinUCB-Hybrid, which again verify the advantages of the proposed algorithms.

## Conclusions and Future Work

We studied a contextual bandit problem for personalized recommendation in a non-stationary environment. To characterize the fact that users' interests towards different items vary

asynchronously and distinctly, two models with disjoint and hybrid piecewise-stationary payoffs were considered. Efficient learning algorithms were developed under both models and theoretical analysis validating a vanishing per-time regret was provided under the disjoint payoff model. Numerical results on real-world datasets verified the advantages of the proposed learning algorithms against baseline ones.

Several issues in this work ask for future studies. First, theoretical regret analysis under the hybrid payoff model is still lacking. Moreover, one limitation of the proposed algorithm is that estimating a preference vector for every arm is costly in computation and memory, especially when the number of arms is extremely large. A potential solution is to cluster similar arms into groups and collectively learn the preferences of users towards arms within the same group.

## References


Agrawal, S., and Goyal, N. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, 39–1.

Agrawal, S., and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

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

Cao, Y.; Wen, Z.; Kveton, B.; and Xie, Y. 2019. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 418–427.

Cesa-Bianchi, N.; Gentile, C.; and Zappella, G. 2013. A gang of bandits. In *Advances in Neural Information Processing Systems*, 737–745.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Garivier, A., and Moulines, E. 2011. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, 174–188. Springer.

Hariri, N.; Mobasher, B.; and Burke, R. 2015. Adapting to user preference changes in interactive recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Hartland, C.; Gelly, S.; Baskiotis, N.; Teytaud, O.; and Sebag, M. 2006. Multi-armed bandit, dynamic environments and meta-bandits.

Hartland, C.; Baskiotis, N.; Gelly, S.; Sebag, M.; and Teytaud, O. 2007. Change point detection and meta-bandits for online learning in dynamic environments.

Lai, T. L., and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1):4–22.

Langford, J., and Zhang, T. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 817–824. Citeseer.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010a. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Li, W.; Wang, X.; Zhang, R.; Cui, Y.; Mao, J.; and Jin, R. 2010b. Exploitation and exploration in a performance based contextual advertising system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 27–36. ACM.

Li, L.; Chu, W.; Langford, J.; and Wang, X. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 297–306. ACM.

Liu, J.; Dolan, P.; and Pedersen, E. R. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, 31–40. ACM.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press.

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

Wang, H.; Wu, Q.; and Wang, H. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1633–1642. ACM.

Wu, Q.; Wang, H.; Li, Y.; and Wang, H. 2019. Dynamic ensemble of contextual bandits to satisfy users' changing interests. In *The World Wide Web Conference*, 2080–2090. ACM.

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

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

Yue, Y., and Joachims, T. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 1201–1208. ACM.

Zeng, C.; Wang, Q.; Mokhtari, S.; and Li, T. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2025–2034. ACM.