10-2021

Burst-induced Multi-Armed Bandit for learning recommendation

Rodrigo ALVES

Antoine LEDENT
Singapore Management University, aledent@smu.edu.sg

Marius KLOFT

Citation

ALVES, Rodrigo; LEDENT, Antoine; and KLOFT, Marius. Burst-induced Multi-Armed Bandit for learning recommendation. (2021). *RecSys '21: Proceedings of the 15th ACM Conference on Recommender Systems, September 27 - October 1, Amsterdam*. 292-301.
Available at: https://ink.library.smu.edu.sg/sis_research/7209

Burst-induced Multi-Armed Bandit for Learning Recommendation

Rodrigo Alves
alves@cs.uni-kl.de
TU Kaiserslautern
Kaiserslautern, RP, Germany

Antoine Ledent
ledent@cs.uni-kl.de
TU Kaiserslautern
Kaiserslautern, RP, Germany

Marius Kloft
kloft@cs.uni-kl.de
TU Kaiserslautern
Kaiserslautern, RP, Germany

ABSTRACT

In this paper, we introduce a non-stationary and context-free Multi-Armed Bandit (MAB) problem and a novel algorithm (which we refer to as BMAB) to solve it. The problem is context-free in the sense that no side information about users or items is needed. We work in a continuous-time setting where each timestamp corresponds to a visit by a user and a corresponding decision regarding recommendation. The main novelty is that we model the reward distribution as a consequence of variations in the intensity of the activity, and thereby we assist the exploration/exploitation dilemma by exploring the temporal dynamics of the audience. To achieve this, we assume that the recommendation procedure can be split into two different states: the loyal and the curious state. We identify the current state by modelling the events as a mixture of two Poisson processes, one for each of the possible states. We further assume that the loyal audience is associated with a single stationary reward distribution, but each bursty period comes with its own reward distribution. We test our algorithm and compare it to several baselines in two strands of experiments: synthetic data simulations and real-world datasets. The results demonstrate that BMAB achieves competitive results when compared to state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Reinforcement learning*; • **Mathematics of computing** → Information theory.

KEYWORDS

Multi-Armed bandit, Time series, Bursty methods, Audience dynamics

ACM Reference Format:
Rodrigo Alves, Antoine Ledent, and Marius Kloft. 2021. Burst-induced Multi-Armed Bandit for Learning Recommendation. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3460231.3474250

RecSys '21, September 27-October 1, 2021, Amsterdam, Netherlands
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8458-2/21/09…$15.00
https://doi.org/10.1145/3460231.3474250

1 INTRODUCTION

Cold-start recommendation (CSR) is one of the fundamental problems in recommender systems (RS) [24]. Its key question is: How to profile and recommend items to *new* users? CSR is challenging due to the lack of information about the preferences and behavior of new users. Various techniques have been proposed to solve this problem, including interview-based approaches [41], exogenous attribute exploitation [39] and cross-domain recommendation [11].

Especially challenging is the case of CSR where there is not only a lack of information about user preferences and behavior, but of *any* usable side information. Without the ability to profile users and items densely, a RS can rely only on recent user-item interaction [17, 49]. For example, when a new user visits a news website, a RS needs to choose an article solely based on user-agnostic information (e.g., the average click-through rate) and then observes whether the new user clicks on the recommended article. The website aims to catch the users' attention and maximize the total number of clicks. Providing effective CSR here requires identifying the 'trending' items most popular among the website's audience.

In this challenging scenario, a popular option is to model CSR as a multi-armed bandit (MAB) problem [40]: at each trial, the gambler (RS) selects an arm (e.g., news article) to pull (show to the user) and observes a reward (a click or lack thereof). Throughout the event history, an algorithm improves the policy to maximize the reward (e.g., the number of clicks). The standard MAB setting assumes that the (unknown) item popularity distribution is stationary [6, 16, 43]. This assumption implies there exists a most popular item fixed over time.

However, this standard form of the MAB problem is inadequate in practice: assuming that the items' popularity is *not* changing over time [49] is highly unrealistic. In this paper, we therefore model CSR as a *non-stationary* MAB problem. Interestingly, we therefore continuously face the classic exploration/exploitation dilemma known from reinforcement learning: the RS must maintain a balance between recommending a classic popular item and recommending the object of the current viral fad. To illustrate this dilemma, let us consider the following example. Suppose that a RS must select among videos of two artists: the South Korean singer Psy and the British singer David Bowie. The gray lines in both graphs of Figure 1 show the cumulative[1] level of system activity (only USA audience) associated with both artists. Most of the time, the rate of growth of the system activity is approximately constant. However, this linearity is sometimes broken by sudden bursts of events highlighted by the two vertical lines. The first spike (vertical red line) matches with the "*Gangnam Style*" release. The hit had an

[1] For a time series $T = \{t_1, t_2, \cdots, t_n\}$, we denote the numbers of events that happened before t by $N(t) := \sum_i^n 1_{\{t_i < t\}}$.

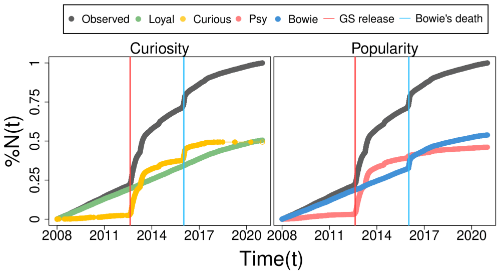

Figure 1: Two examples of time series factorization that motivate our proposed MAB model.Left: loyal and curious systems' audiences (separation of loyal and curious audiences according to [5]). Right: activity related to Psy and David Bowie. The data is taken from Google Trend (Jan/2008 to Dec/2020; country: USA; search engine: Youtube). We give a detailed description of the results in the main text.

unprecedented explosion of popularity[2] and its music video "broke" the YouTube view counter's limit. The second burst of events (vertical blue line) coincides with David Bowie's unexpected death. This unfortunate exogenous event triggered the audience's curiosity.

We explain the variations in the users' activity by the existence of two types of audiences (disentangled in Figure 1, left graph): the *loyal* audience and the *curious* audience. The loyal audience (green curve) is constituted by fans who assiduously follow the topic. In contrast, the curious audience (yellow curve) only turned their attention to the topic due to an extraordinary event. Thus, the environmental context in which the RS must make decisions alternates between calm periods (where the users' behavior is driven by the loyal audience), and disruptive or 'bursty' periods (where the curious audience is driving sudden bursts of interest in certain topics). In our real-life example, during stable periods, the ratio between the singers' popularity is stable and Bowie is consistently more popular than Psy (Figure 1, right graph). On the other hand, during the disruptive period dominated by *curious* behavior, the relative popularity between Psy and Bowie changed drastically.

Motivated by this phenomenon, which we entitle *audience curiosity* [5, 9, 32, 47], we propose to detect the state of the environment (whether the system is in a bursty or calm period) and use this information to guide the exploration strategy of the RS. For instance, in the example above, Psy's sudden burst of popularity after the release of 'Gangnam Style' deeply altered the environment and reward distribution: Psy momentarily became more popular than David Bowie. As we explain in more detail below, whenever our algorithm detects such dramatic changes in the environment, it intensifies its *exploratory behavior* during the turbulent period to keep up with changes in the optimal strategy.

Prior work on non-stationary MAB problems [10, 22] has dealt with the shifts in the items' popularity in both context-free [4, 10, 12, 20, 48] and context-aware [27, 30, 49] situations. While the first group solely uses the observed rewards, the latter group requires user or item features to build its arm-selection strategy. However, previous work did not take into account the effect of audience curiosity in the reward distribution.

In contrast, we model CSR as a non-stationary and context-free MAB problem and propose a novel algorithm to solve it. Our formulation is context-free, so it requires no side information about users or items. By carefully modeling the temporal dynamics of the audience, our model exploits variations in user activity (e.g., sudden bursts). We assume that the recommendation environment can be split into two different states: the *loyal state* (stable) and the *curious state* (unstable). We identify the current state by modeling the events as a mixture of two Poisson point processes, one for each of the possible states. The main contributions of this paper can be summarized as follows:

- **New algorithm** We propose the *Burst-induced Multi-Armed Bandit* (BMAB), a non-stationary and context-free MAB algorithm that exploits the temporal audience dynamics to predict changes in the reward distribution.
- **Regret guarantees** We prove regret guarantees for our model BMAB when the states are recoverable and bursts are separable. We also experimentally analyze the proposed regret bounds.
- **Competitive performance in experiments** We evaluate our algorithm and compare it to several baselines in two experimental strands: synthetic data simulations and real-world datasets. We compare our method to six state-of-the-art baselines and achieve competitive results.

2 RELATED WORK

2.1 Related Work on MABs

MABs were introduced in [43] and more formally defined in [36]. Some classic and broadly used algorithms to solve this problem are Thompson sampling (TS) [37, 43], ϵ-greedy policies [42], Exp3 [7] and strategies based on upper confidence bounds (UCB) [40]. The Thompson sampling algorithm enjoys strong empirical performance [14], regret guarantees [3] and has been successfully applied in a wide variety of RSs domains [1, 2, 25, 28, 38].

In *non-stationary* MABs, the reward distribution is allowed to change through time. There are two major classes of non-stationary MABs: *adversarial* MABs and *piece-wise* stationary MABs. In adversarial MABs, an adversary controls the payoff generation, so no statistical assumptions are imposed [7]. However, the problem is still stationary in the sense that the aim is to return a single arm that is the globally optimal action at a fixed time horizon. In contrast, the reward generation is non-stationary on the whole time horizon in *piece-wise* stationary MABs, but it is stationary on several unknown intervals [48]. Our model belongs to the latter category. Previous work on *piece-wise* stationary MABs can be further categorized into *context-aware* and *context-free* approaches, depending on whether side information (user/item features) are exploited.

In related work on (context-aware) piecewise stationary MABs, [30] proposed LogUCB, an extension of UCB that estimates the average reward of a topic through a logistic regression on its features. In line with our work, [49] present a MAB algorithm that also considers temporal influence on the item consumption probability. They construct the policy algorithm as a probabilistic framework

[2] The instant popularity $p(t)$ can be expressed as $p(t) = \partial\mathbb{E}(N(t))/\partial t$

that uses as context a high-dimensional vector containing side information about the users (demographic information) and the items (query keywords). There are two key differences between this work and ours: (1) our method is context-free (does not need feature vectors); and (2) instead of treating the problem as a discrete-time one, our model actively exploits continuous temporal dynamics to detect possible changes in the reward environment.

Related work on *context-free* MABs includes [4], who performed constant exploration inspired by the EXP3 algorithm to detect changes in which arm is the best, while [12] achieved this by simply comparing the rewards in the two last time-intervals of size w. In both cases, whenever such a distributional change is detected, the backbone MAB algorithm is restarted with the aim of finding the best arm under the new distribution. In the same direction, [48] proposed a general framework that can be used together with several algorithms. After also dividing the event horizon into several equal-sized windows (of size w), they compute scores for each arm based on both the rewards and the number of observations in the following two intervals: (1) the last window in which a change in the best arm was observed, and (2) the last observed window. If the absolute difference between the scores in the two windows is greater than a hyperparameter ϵ, the algorithm is re-initialised. Similarly, [21] performed change detection by comparing two previous time windows. Their model also relies on estimates of the probability of false-alarm and the probability of missed-detection to improve robustness. Instead of resetting the algorithm altogether, [10] adopted a fixed sliding training window while [19] used a different window size for each arm. With a slightly different approach, [20] proposed the so called 'Discounted-UCB (DUCB)' algorithm. The main idea is to give a higher selection probability to two classes of arms: the arms which recently returned high rewards and the arms which were not recently selected. Therefore, instead of resetting the whole procedure, DUCB tackles the non stationarity by maintaining a minimum amount of exploration throughout the event horizon. By mixing the two previous approaches (discounted reward and sliding window), [13] assigns more relevance to the recent rewards. The last w rewards are not discounted whilst the remaining ones are. [26] proposed a piece-wise MAB algorithm that detects abrupt changes in the reward distribution through a hypothesis test. As a criterion for this hypothesis test they rely on the Page-Hinkley statistic, which involves a random variable defined as the difference between the reward time t and the average reward, cumulated in the last m steps. Our method's main difference from other context-free methods lies in our shift detection procedure. Instead of detecting changes only in the reward distribution, we analyze the system's temporal dynamics to identify behavioral changes in the audience. Our hypothesis is that such behavioral change is associated with the items' popularity.

2.2 Related Work on the Dynamics of Human Communication

Popularity prediction and online trend detection [33, 35, 47] are fundamentally linked to the recommendation task, especially when no context is available [17]. Previous works show that item popularity increases and decreases over time [23, 45, 47] and it is triggered by bursts [8, 9, 35]. One of the first attempts to associate human communication with the emergence of bursts was [9]. They proposed that human activities tend to alternate between periods of calm and intense activity. Plenty of works substantiate this premise [18, 32, 34, 44, 46, 50]. Such alternating behavior points at the presence of two distinct types of audiences: the *loyal audience* which corresponds to the stable activity which occurs during the calm periods and *curious audience*, which is highly unpredictable, and responsible for the bursts in activity in the system [5].

Stochastic point process form the statistical framework to model random sequences of events [15]. Poisson processes, for example, are broadly used to measure stable audiences [5, 29, 31]. On the other hand, power law distributions and self-exciting point processes have been used to model the unexpected behavior of bursts [9, 32, 44, 46]. In this work, we propose that the loyal and the curious audiences form a mix of two stochastic point processes, formally defined in Section 3. The difference in the intensity of the point processes defines the state of the MAB problem.

3 PROBLEM FORMULATION

Let $\mathcal{K} = \{1, 2, \cdots, K\}$ be a set of K arms and $\mathcal{T} = \{t_1, t_2, \cdots, t_N\}$ denote a sequence of N timestamps in the interval $(0, T]$. At each time t_i, a gambler chooses one of the K arms and observes the reward $r_i \in \{0, 1\}$. The reward distribution at time t_i depends on the state s_i of the system. We set $s(t_i) = 0$, if t_i occurs during the *loyal* state, and $s(t_i) = 1$, if t_i occurs during the *curious* state.

We assume that the time series $\mathcal{T}$ is generated by a mixture of two stochastic point processes: (1) a homogeneous Poisson process (HPP) with intensity[3] $\lambda(t) = \lambda_L$, and (2) a piece-wise homogeneous Poisson process (PW-HPP) with intensity $\lambda_C(t)$. We assume that the intensity $\lambda_C(t)$ of the second process is piecewise constant, with the transitions occurring at the random and *unobserved* timestamps $M = \{m_1, m_2, \cdots, m_n\}$ (by convention we also set $m_0 = 0$), on whose distribution we make no formal assumption [4]. Thus, $\lambda_C(t)$ can assume $(n + 1)$ values in the interval $(0, T]$, denoted by $\{c_0, c_1, c_2, \cdots, c_n\}$.We write c_j for the (unique) value the intensity $\lambda_C(t)$ takes in the interval $[m_j, m_{j+1})$. A key assumption of our work is that the underlying distribution has the property that $c_j \ll \lambda_L$ (w.p. 1), if $j \equiv 0 \mod 2$, and $c_j \gg \lambda_L$, otherwise. This implies that the PW-HPP alternates between silent mode (very low intensity) and bursty mode (very high intensity). Finally, write $B(t) = \sum_{j: m_j < t} (j \mod 2)$ for the number of PW-HPP transitions into the bursty mode which occured before t (thus, if $s(t) = 1$, t belongs to the $B(t)^{\text{th}}$ burst).

The first graph of Figure 2 presents a realization of a mixture of two stochastic point processes with the properties described above. In this example, $\lambda_L = 3$, $\lambda_C(t)$ alternates between 0.15 and 15, $T = 100$ and the elements of M (vertical red lines) were randomly selected aiming for the expected number of events of both processes to be the same (cf. details in the experiments section below). The HPP models the loyal audience (yellow curve, stable throughout the observed interval) while the PW-HPP models the curious audience (green curve, unstable). Note that the label of the event t_i is not

[3] Recall that the definition of intensity implies that $\lambda(t) = \lambda(t|\mathcal{H}_t) = \lim_{\Delta t \to 0} \mathbb{E}\left(N(t, t + \Delta t)|\mathcal{H}_t\right) / \Delta t$. The intuition behind the intensity function is as follows: for a small time interval Δt, the value of $\lambda(t|\mathcal{H}_t) \times \Delta t$ is approximately the expected number of events in $(t, t + \Delta t)$.

[4] Other than the fact that the total number of transitions n is finite with probability 1.

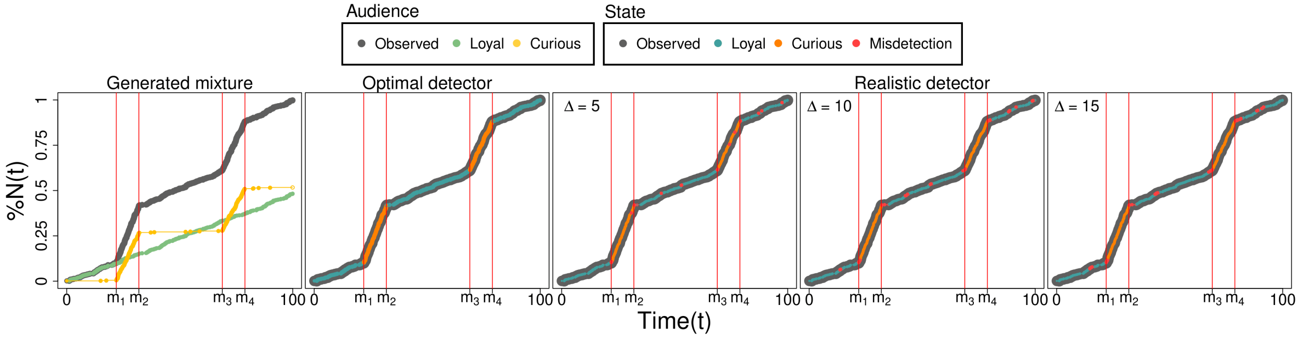

Figure 2: Difference between audiences and states. First graph: mixture of a HPP (green curve, models the *loyal audience*) and a PW-HPP (yellow curve, models the *curious audience*). We only observe the mixture of processes. Parameters: $\lambda_L = 3$, $\lambda_C(t)$ alternates between 0.15 and 15, $T = 100$. Second graph: point-wise state detection (loyal and curious *states*) by an optimal detector. Three last graphs point-wise state detection (loyal and curious states) by the detector proposed in Section 4.4 (we fixed $\delta = 0.95$ and varied $\Delta \in \{5, 10, 15\}$). In all graphs, the vertical lines indicate the actual state transitions (i.e., the set M).

observed, that is, it is unknown whether $t_i \in$ HPP or $t_i \in$ PW-HPP (even in a given stable or bursty period). However, our underlying assumption is that the state $s(t_i)$ of the slot machine at t_i does not depend on the label of t_i, but depends instead on the current dominating audience dynamic: $s(t) = 0$ in the calm periods, and $s(t) = 1$ in the bursty periods. More precisely, $s(t)$ is determined by $s(t) \equiv \max(i|m_i \leq s(t)) \mod 2$.

In our model, the reward distribution is a consequence of the audience variation: we assume that reward distribution is stationary in the absence of bursts. Hence, in all pieces of the interval where $s(t) = 0$ we model the distribution of arm a's reward at time t as $r(a,t) \sim \text{Bernoulli}(\theta_a^0)$. In contrast, in the presence of bursts ($s(t) = 1$) the reward distribution varies: each burst has its own stationary reward distribution. Therefore, we model $r(a,t) \sim \text{Bernoulli}(\theta_a^{B(t)})$ when $s(t) = 1$.

At each time t_i, the agent must choose an arm according to a policy $\pi(t|s(t), B(t))$. Let $\bar{\theta}_a^t = \mathbb{E}[r(a,t)]$ be the expected reward for arm a at time t given $s(t)$ and $B(t)$. As mentioned before, a common goal is to maximize the expected reward $R(T)$ over the entire horizon, which can be written as

$$\mathbb{E}\big(R(T)\big) = \mathbb{E}\sum_{i=1}^{N} \bar{\theta}^{t_i}_{\pi(t_i|s(t_i),B(t_i))},$$

where the expectation runs over both (1) the random choices made by the algorithm policy and (2) the reward distribution. Note that we do not let the expectation run over the distribution of the timestamps. Thus the left hand side (LHS) is technically a random variable. This is a well-principled choice since it corresponds to expressing the regret as a function of the number of decisions to be made by the agent. We also adapt the notion of (expected) regret, which relies on the concept of the optimal *fixed* arm. Let $\bar{\theta}_*^t = \max_j \bar{\theta}_j^t$ be the expected reward of the optimal arm at time t given $s(t)$ and $B(t)$, the expected regret is then defined as:

$$\mathbb{E}\big(\mathcal{R}(T)\big) = \mathbb{E}\sum_{i=1}^{N} \left[\bar{\theta}_*^{t_i} - \bar{\theta}^{t_i}_{\pi(t_i|s(t_i),B(t_i))}\right], \tag{1}$$

where the expectation is also over the random choices made by the algorithm policy and the random rewards.

4 BURST-INDUCED MAB

To solve the problem formulated in Section 3 we propose the BMAB algorithm. In this section we will describe our algorithm according to the following structure: first, we will present the main ideas of the Thompson sampling algorithm, which forms the backbone of our method; second, we will describe the steps of BMAB and show its regret guarantees; finally we will present our state detector, which is a crucial component of our algorithm BMAB.

4.1 Thompson sampling algorithm

The Thompson sampling (TS) algorithm is a classical approach the *stationary* stochastic MAB problem. In this setting, at time t, the slot machine has K arms and, when an arm a is played, the machine produces a reward $r(a)$. The reward distribution of arm a is a Bernoulli distribution with fixed and unknown parameter θ_a. In summary, an arm a has probability θ_a of returning 1 as a reward, and $1 - \theta_a$ of returning 0.

Also known as *posterior sampling*, TS assumes an independent prior belief over each θ_a. In this Bernoulli reward case, it is natural to choose a beta-distribution as a prior (since it is a conjugate prior). Thus for the MAB case, for each arm a, the prior probability density function of θ_a is beta-distributed with parameters α_a and β_a:

$$p(\theta_a) = \frac{\Gamma(\alpha_a + \beta_a)}{\Gamma(\alpha_a)\Gamma(\beta_a)}(\theta_a)^{\alpha_a - 1}(1 - \theta_a)^{\beta_a - 1},$$

where Γ is the gamma function. At each time t, the TS algorithm samples a vector $\Theta = \{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_K\}$, where $\hat{\theta}_i \sim \text{Beta}(\alpha_i, \beta_i)$ (i.i.d). Then the policy selects the arm $\pi(t) = \text{argmax}_i \hat{\theta}_i$. **REMARK 1:** the exploration procedure is probabilistically tackled. At each step, the probability density function $f(\theta)$ of $\text{Beta}(\alpha, \beta)$ is greater than zero in the whole domain $[0, 1]$. Thus any arm has non-zero probability of being selected. **REMARK 2:** In the special case when $\alpha = \beta = 1$, $\theta \sim \text{Uniform}(0, 1)$.

Due the conjugacy properties of the beta distribution, the Bayesian update of the parameters α and β is particularly simple: at time t, after the algorithm selects arm $\pi(t)$ and observes the reward $r(\pi(t))$, the parameters of the prior distribution of $\theta_{\pi(t)}$ can be updated as follows:

$$[\alpha_{\pi(t)}, \beta_{\pi(t)}] = [\alpha_{\pi(t)} + r(\pi(t)), \beta_{\pi(t)} + (1 - r(\pi(t)))]. \quad (2)$$

4.2 The BMAB algorithm

The BMAB algorithm is described in Algorithm 1. The core idea is to use Thompson sampling on each stationary region with a separate count of α and β for each reward distribution. Thus, for each state, the reward distribution of each arm a is a Bernoulli(θ_a) with prior θ_a ~Beta(α, β). At each time t, the vectors $\alpha_0 \in \mathbb{R}^K$ and $\beta_0 \in \mathbb{R}^K$ (resp. $\alpha_1 \in \mathbb{R}^K$ and $\beta_1 \in \mathbb{R}^K$) denote the parameters of the priors related to the K arms in the loyal (resp. curious) state. Each arm a has reward distribution $r(a)$ ~Bernoulli(θ_a). In the loyal state, the estimated distribution of θ_a is θ_a ~Beta($\alpha_0[a], \beta_0[a]$), whereas in the curious state, we have θ_a ~Beta($\alpha_1[a], \beta_1[a]$) instead.

To support our explanation we will illustrate our algorithm execution for the time series displayed in Figure 2. We assume that $K = 3$ and $[\theta_1, \theta_2, \theta_3] = [0.3, 0.4, 0.5]$ in the loyal state. Regarding the curious state, the parameters assume the values $[\theta_1, \theta_2, \theta_3] = [0.3, 0.9, 0.5]$ (in $(m_1, m_2]$) and $[\theta_1, \theta_2, \theta_3] = [0.9, 0.4, 0.5]$ (in $(m_3, m_4]$). Thus in this case, the best arm during the entire loyal state is arm 3, whilst in the first burst of events it is arm 2, and during the second burst of events it is arm 1. Figure 3 shows the prior distributions of θ_1, θ_2, and θ_3 at the times $\{0, m_1, m_2, m_2^5, m_3, m_4, t_N = 100\}$[5]. The first row of graphs corresponds to the priors of the loyal state while the second row of graphs corresponds to the priors of the curious state.

Our precise algorithm can be split into three main steps: **Initialization [Line 1]:** we initialize all entries of the vectors α and β as 1. Thus, priors are initialised as uniform distributions (at $t = 0$). **Recommendation and learning procedures [Lines 3-7]:** in order to maximize the reward, BMAB aims to learn (by updating its priors of) the reward distributions of *both states* with enough confidence to select the best arm at the event time. Therefore, at each event t_i the algorithm needs to detect the state s_i. We assume that an oracle ω is available to provide an estimate of the state at each timestamp in $\{t_1, t_2, \cdots, t_i\}$ (line 3). When a *perfect* oracle (i.e. with $\omega(t) = s(t)$) is available, we refer to our algorithm as **[BMAB-O]**. In practice, the role of the oracle can be assumed by our realistic state detector from Section 4.4. We refer to the resulting instance of our algorithm as **[BMAB-R]**. In the next step (line 4), we sample $\hat{\theta}_k$ (for each k) according to the current prior distribution Beta($(\alpha_{s_i})_k, (\beta_{s_i})_k$) corresponding to the current state s_i and the arm k. Our policy is to select (recommend) the arm a which has the highest $\hat{\theta}_k$ (line 5). Finally, we observe the reward $r(a)$ of the selected arm and update the priors of the state s_i in accordance with (2). Note that the loyal state is associated with a single stationary reward distribution: our method's estimate of the distribution corresponding to the loyal state keeps improving throughout the whole event horizon (first row of graphs, Figure 3). **Burst separation [Lines 8-12]:** in the problem definition, we further assume that each bursty period comes with its own reward distribution. Accordingly, we aim to treat each bursty period as a separate MAB problem, resetting the Thompson priors at the beginning of each burst, whilst keeping a global count for the periods where the loyal audience dominates. However, due to the uncertainty inherent in the state prediction method ($\omega(t)$), we engineer a *soft* transition procedure: whenever a burst appears to be ending, the priors corresponding to the burst are gradually forgotten rather than discarded immediately. In Figure 3, observe that from $t = 0$ to $t = m_1$ the loyal priors have significantly changed after some reward observations while the curious priors stay as initialized. During the first bursty period $(m_1, m_2]$, the curious priors changed as the model learned the reward distribution of the burst. One can observe at time m_2 that loyal priors remain the same as those as those at m_1, since the interval $(m_1, m_2]$ is governed by the bursty dynamic. This can also be seen at the second bursty period ($(m_3, m_4]$). Similarly, when the bursty period appears to taper off, the learned priors are gradually forgotten as the model gains confidence in its observation of a return to normality. To accomplish it, BMAB employs a forgetting-rate $\gamma \in [0, 1]$: at each step with $s_i = 0$, we compute $\alpha_1 = \gamma\alpha_1$ and $\beta_1 = \gamma\beta_1$. If some entry of α_1 or β_1 is less than 1, we round it to 1 (lines 8-12). The effects of this forgetting procedure can be observed in Figure 3. At time $t = m_2^5$, i.e. five loyal-state events after m_2, an increase of the variance around the expected value of θs is seen for the curious state. This smooth forgetting procedure eventually leads to a return to a uniform prior after sufficiently many loyal-state timestamps, as can be observed at time $t = m_3$.

4.3 BMAB regret guarantees

In this section, we present regret guarantees for the **[BMAB-O]** algorithm (with a perfect oracle). In Section 5.1 we show experimentally that the results also hold for **[BMAB-R]**.

Theorem 4.1. *Write n_b for the total number of bursts and let the set $\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_1, \cdots, \mathcal{N}_{B(t_N)}\}$ contain the number of timestamps in each period (with $\mathcal{N}_0$ corresponding to the entire calm period and $\mathcal{N}_i$ corresponding to the ith burst: $\mathcal{N}_0 = \sum_j 1_{s(t_j)=0}$, and for all i, $\mathcal{N}_i = \sum_j 1_{s(t_j)=1 \wedge B(t_j)=i}$). We assume access to an optimal oracle ω with $\omega(t) = s(t)$ for all t and set $\gamma = 0$. For all configurations $n_b, \mathcal{N}$*

Algorithm 1 BMAB
Input: Number of arms $K \in \{2, 3, 4, \cdots\}$ and forgetting-rate $\gamma \in [0, 1]$

```
 1: $\alpha_0 = \beta_0 = \alpha_1 = \beta_1 = \{1\}^K$
 2: for $i \in \{1, 2, \cdots, N\}$ do
 3:     $s_i = \omega(t_i)$
 4:     $\forall k \in \{1, 2, \cdots, K\}$ sample $\hat{\theta}_k \sim$ Beta($\alpha_{s_i}[k], \beta_{s_i}[k]$)
 5:     $a = \text{argmax}_j \hat{\theta}_j$
 6:     Observe the reward $r(a)$
 7:     $(\alpha_{s_i}[a], \beta_{s_i}[a]) = (\alpha_{s_i}[a] + r(a), \beta_{s_i}[a] + (1 - r(a)))$
 8:     if $s_i == 0$ then
 9:         $\alpha_1 = \gamma\alpha_1$ , $\beta_1 = \gamma\beta_1$
10:         $\forall k \in \{1, 2, \cdots, K\}$ if $\alpha_1[k] < 1$ make $\alpha_1[k] = 1$
11:         $\forall k \in \{1, 2, \cdots, K\}$ if $\beta_1[k] < 1$ make $\beta_1[k] = 1$
12:     end if
13: end for
```

[5] Define m_i^n as the nth element of the ordered set $\{t_j | t_j > m_i\}$, so that m_2^5 is the fifth time stamp in the stationary section (m_2, m_3]

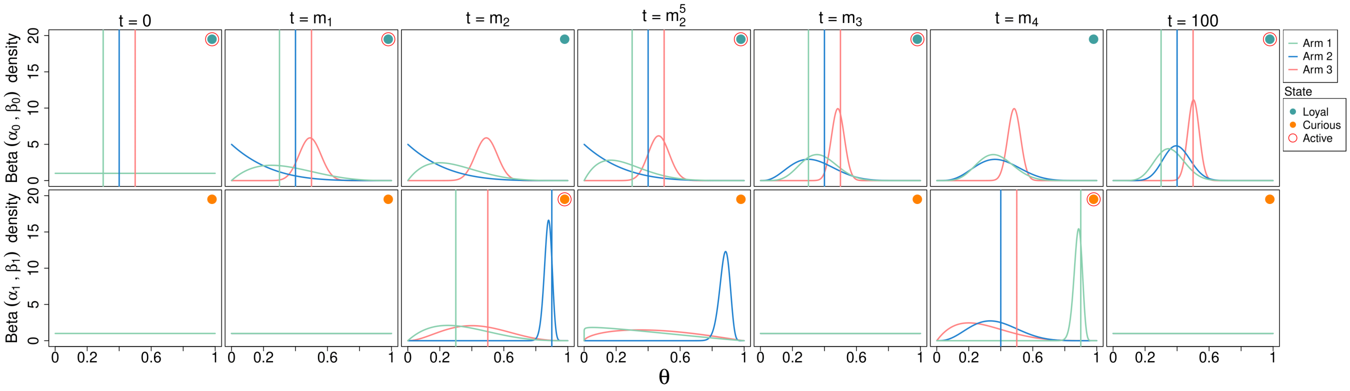

Figure 3: Realization of the BMAB algorithm in the mixture of point processes presented in Figure 2 ($K = 3$; Loyal state: $[\theta_1, \theta_2, \theta_3] = [0.3, 0.4, 0.5]$; and Curious state: $[\theta_1, \theta_2, \theta_3] = [0.3, 0.9, 0.5]$ (in $(m_1, m_2]$) and $[\theta_1, \theta_2, \theta_3] = [0.9, 0.4, 0.5]$ (in $(m_3, m_4]$)). First row of graphs: prior distribution of θ_1, θ_2, and θ_3 at the times $\{0, m_1, m_2, m_2^5, m_3, m_4, t_N = 100\}$ concerning the loyal state. Second row of graphs: prior distribution of θ_1, θ_2, and θ_3 at the times $\{0, m_1, m_2, m_2^5, m_3, m_4, t_N = 100\}$ concerning the curious state.

satisfying the (mild) condition $\mathcal{N}_i > 0$ for all i, we have

$$\mathbb{E}\big(\mathcal{R}(T)\big) \leq O\Big(\sqrt{n_b K N \log K}\Big),$$

where $N = \sum_{i=0}^{n_b} \mathcal{N}_i$ is the total number of events.

Proof. From (1) we have:

$$\begin{aligned}\mathcal{R}(T) &= \sum_{i=1}^{N} \left[\bar{\theta}_*^{t_i} - \bar{\theta}^{t_i}_{\pi(t_i|s(t_i), B(t_i))}\right] \\ &= \sum_{i \in \Omega_0} \left[\bar{\theta}_*^{t_i} - \bar{\theta}^{t_i}_{\pi(t_i|0,-)}\right] + \sum_{l=1}^{n_b} \sum_{j \in \Omega_1(l)} \left[\bar{\theta}_*^{t_j} - \bar{\theta}^{t_j}_{\pi(t_j|1,l)}\right] \\ &= \mathcal{R}_0(T) + \sum_{l=1}^{n_b} \mathcal{R}_l(T), \end{aligned} \tag{3}$$

where $\Omega_0 = \{a|s(t_a) = 0\}$, $\Omega_1(l) = \{b|s(t_b) = 1 \text{ and } B(t_b) = l\}$, $\mathcal{R}_0(T)$ is the loyal-state regret and for $l > 0$, $\mathcal{R}_l(T)$ is the component of the regret corresponding to the ith burst. The condition $\mathcal{N}_i > 0$ guarantees that the bursts are separable in the sense that $B(t) = \#\left\{j : \omega(t_j) = 0 \wedge \omega(t_{j+1}) = 1 \wedge t_{j+1} \leq t\right\}$ can be computed by the oracle. Therefore, we essentially have $(n_b + 1)$ stationary Thompson sampling algorithms. Accordingly, applying Theorem 2 from [3] to equation (3) we obtain:

$$\begin{aligned}\mathbb{E}[\mathcal{R}(T)] &= \sum_{i=0}^{n_b} \mathbb{E}[\mathcal{R}_i(T)] = O\left(\sum_{i=0}^{n_b} \sqrt{K\mathcal{N}_i \log K}\right) \\ &\leq O\left(\sqrt{(n_b + 1) \sum_{l=0}^{n_b} K\mathcal{N}_l \log K}\right) = O\left(\sqrt{n_b K N \log K}\right),\end{aligned} \tag{4}$$

where at the second line, we have used Jensen's inquality (more precisely, $\|x\|_1 \leq \sqrt{d}\|x\|_2$ for all $x \in \mathbb{R}^d$). The theorem follows. □

As expected, we observe that stable systems, where bursts are rare, expect to have lower regret, as the number of bursts influences the regret bound by a factor of $\sqrt{n_b}$.

4.4 A realistic state detector

In this section, we will propose a realistic state detector, a crucial step of **[BMAB-R]**. We assume that the loyal audience rate λ_L is known (or can be easily learned [5, 29, 31]). Note that such a rate does not require prior knowledge of the reward distributions and can be easily measured through the system's traffic logs. Our method requires a positive integer sensitivity hyperparameter Δ as well as a confidence parameter $\delta \in (0, 1)$. For all i, we then write $\Delta_i = t_i - t_{i-\Delta+1}$ (if $i < \Delta$, $\Delta_i = t_i$). We write $q_\delta(\mu, \rho) = q_{\text{Gamma}}(\mu, \rho, \delta)$ for the (left) quantile function of the Gamma distribution with shape μ and scale ρ, i.e., $\mathbb{P}(X \leq q_\delta(\mu, \rho)) = 1 - \delta$ where X follows a Gamma distribution with shape μ and scale ρ.

In order to detect the the state of the event t_i we aim to test the hypothesis that the elements of the set $\mathcal{T}_{\Delta,i} = \{t_{i-\Delta+1}, t_{i-\Delta+2}, \cdots, t_i\}$ (if $i < \Delta$, $\mathcal{T}_{\Delta,i} = \{0, t_1, t_2, \cdots, t_i\}$) are timestamps generated by a uniform Poisson process with intensity λ_L. Our state detector is described in Algorithm 2. Firstly, we compute the size of the interval Δ_i that is covered by the set $\mathcal{T}_{\Delta,i}$ (lines 1-5). If the timestamps in $\mathcal{T}_{\Delta,i}$ were indeed indeed generated by a Poisson process with intensity λ_L, then the distribution of $\Delta_i = t_i - t_{i-\Delta+1}$ will be a Gamma distribution with shape $\Delta - 1$ and scale λ_L.

Accordingly, we calculate the quantile function $q_\delta(\Delta - 1, \lambda_L)$ and test the hypothesis stated by comparing $q_\delta(\Delta - 1, \lambda_L)$ and Δ_i (lines 6-11), returning the state 0 (loyal), if the hypothesis is accepted, and 1 (curious) otherwise. Experiments analysed in Section 5.1 show that our state detector has comparable performance to the optimal oracle. The three last graphs of Figure 2 illustrate our state detector for a fixed $\delta = 0.95$ and different values of $\Delta \in \{5, 10, 15\}$.

5 EXPERIMENTS

To compare BMAB with the baselines we conducted experiments with two data strands: synthetic data (Section 5.1) and real-world data (Section 5.2). In the first case, several simulations were performed to verify the performance of BMAB in different ground truth regimes. Using synthetic data we also experimentally analyze the reward guarantees (stated in Section 4.3) and show a good match

Algorithm 2 State Detector ω-R

Input: Event set $\{t_1, t_2, \cdots, t_i\}$, integer window Δ, confidence index $\delta \in (0,1)$ and loyal audience intensity λ_L

```
if $i < \Delta$ then
    $\Delta_i = t_i$
else
    $\Delta_i = t_i - t_{i-\Delta+1}$
end if
$q_\delta(\Delta - 1, \lambda_L) = q_{\text{Gamma}}(\Delta - 1, \lambda_L, \delta)$
if $q_\delta(\Delta - 1, \lambda_L) \geq \Delta_i$ then
    return 1
else
    return 0
end if
```

between the bounds and the observed reality, even when used in conjunction with our realistic state detector. In the second strand, we validated our model on four real recommender systems datasets. We show that our methods exhibit state-of-the-art performance in all cases.

Baselines and parameter selection: all the following baselines were evaluated.

- **[TS] Thompson Sampling:** traditional stationary MAB algorithm [43]. For more details, see Section 4.1.
- **[EXP3] EXP3:** broadly used MAB algorithm that considers a non-stationary environment. EXP3 uses a parameter γ to control exploration and exploitation during all the period (γ was selected following Corollary 3.2 of [7]).
- **[EXP3DD] EXP3 with Drift Detection:** EXP3 with a reward distribution shift detection procedure. When the best arm changes, EXP3DD re-initializes the algorithm. Hyperparameter selection was performed acording to Section V of [4].
- **[DUCB] Discounted UCB:** A UCB-type method which tackles non stationarity by maintaining exploratory behavior throughout the event horizon. Hyperarameter tuning following Section 3.1 of [20].
- **[MUCB] Monitored UCB:** MUCB detects the change on the arms' reward distribution by comparing the rewards in the two last time-intervals of same size. The parameters w and b were setting according to Section 5 (Remark 1)[12].
- **[WMD] Windowed mean-shift detection:** WMD is a framework that uses time-windows to detect shifts in the arms' reward distribution. As in [48] we set ϵ and τ according to Section 5 and Theorem 4.1.

In all cases, the cited parameters and sections follow the notation of the respective papers. We can split the baselines into three groups depending on which rewards environment they were designed to work in: stationary and non-stationary **[TS]**; non-stationary **[EXP3]**; and piece-wise stationary, **[EXP3DD]**, **[DUCB]**, **[MUCB]** and **[WMD]**. For **[BMAB-O]** and **[BMAB-R]**, we empirically selected the forgetting-rate $\gamma = 0.70$, the detector confidence index $\delta = 0.95$, and the detector window $\Delta = 10$. We assume we know λ_L in the synthetic strand. To find λ_L in the real-world data, we approximate the PW-HPP by a self-feeding point process (for details, see [5]; for an illustration, see Figure 1).

5.1 Synthetic data experiments

Generation of the audience dynamics : We proceed in five steps. **STEP 1:** Choose a set of parameters $\{\lambda_L, \tilde{N}, P_H, n_b, b\}$. λ_L will be the loyal intensity. The (curious) intensity in the bursty periods will be set to $b\lambda_L$. P_H and $\tilde{N}$ will have the following properties: $\tilde{N}$ will be the expected number of timestamps so that $\tilde{N} = \mathbb{E}(N)$, and the expected number of timestamps attributed to the curious audience (during bursts) will be $P_H\tilde{N}$. n_b will be the (fixed) number of bursts. **STEP 2:** Generate a Poisson process with event rate λ_L along the time interval $(0, T]$, where $T = P_H \times \mathbb{E}[N]/\lambda_L$. **STEP 3:** Split the interval $(0, T]$ into the set of n_b contiguous sub-intervals of the same size $\mathcal{U} = \{(0, T_1], (T_1, T_2], \cdots, (T_{n_b-1}, T]\}$. **STEP 4:** Let $\lambda_C(t) = b\lambda_L$, when $s(t) = 1$, and $\lambda_C(t) = 0.05\lambda_L$, otherwise. We consider burst intervals of size $T_b = ((1 - P_H) \times T)/(n_b \times b)$. This guarantees that the expected number of (curious) events during bursts is $(b\lambda_L)n_n((1-P_H)\times T)/(n_b \times b) = (1-P_H)\lambda_L T = (1-P_H)\tilde{N}$. For each $\mathcal{U}_i = (x, y]$ generate $m_{2(i-1)+1} \sim \text{Uniform}(x, y - T_b)$ and $m_{2(i-1)+2} = m_{2(i-1)+1} + T_b$. **STEP 5:** Generate the PW-HPP given the set M and $\lambda_C(t)$ of step 4.

Comparison with baselines: To compare our method with baselines, we designed a reward setting where the rewards depend strongly on the state of the system. We set the parameters as follows. We set $\lambda_L = 1$ and varied $\mathbb{E}[N] \in \{1000, 2000, 5000\}$, $P_H \in \{0.25, 0.5, 0.75\}$, $n_b \in \{1, 2, 3\}$ and $b \in \{5, 10, 20\}$. We set $K = 3$ and set the loyal-state parameters as follows $[\theta_1^0, \theta_2^0, \theta_3^0] = [0.3, 0.4, 0.5]$. During the each burst, the reward parameters are the same as in the loyal state except for one arm, whose parameter is set to 0.9. The sequence of arms whose reward changes is selected from $\{1, 2, 3\}$ by uniform sampling without replacement. Figure 2 shows an instance of the generation procedure for $\lambda_L = 3$, $\mathbb{E}[N] = 500$, $P_H = 0.5$, $n_b = 2$ and $b = 5$. The effect of step 5 can be visualized in Figure 3.

For each set of $\{\lambda_L, \mathbb{E}[N], P_H, n_b, b\}$ we generated 20 samples. For each sample, we performed **[BMAB-O]**, **[BMAB-R]** and the baseline algorithms. Figure 4 and Table 1 show the performed simulation results. The green lines ("Optimal") in Figure 4 show the performance of a hypothetical algorithm with the ability to always select the best arm (with the largest θ). Thus, in theory, no algorithm can achieve better performance. Each point in Figure 4 is the average *normalized* reward $R(T)/R_{\text{Optimal}}(T)$, averaged over 20 samples. Table 1 shows the summary of the results for all simulations. Note that both **[BMAB-O]** and **[BMAB-R]** consistently outperform the baselines in all the considered scenarios. In addition, the proximity of the two BMAB curves reveals **[BMAB-R]**'s ability to recover the correct states with high accuracy, with this fact being particularly marked in the case where there is an even mix of both point processes ($P_H = 0.5$). Smaller values of n_b led to better performance. This result matches with our theoretical results presented in Section 4.3.

Experimental verification of regret guarantees and state detector: we performed 1000 simulations to verify empirically the theoretical results of Section 4.3 and the accuracy of the state detector proposed in Section 4.4. We explore the following parameter

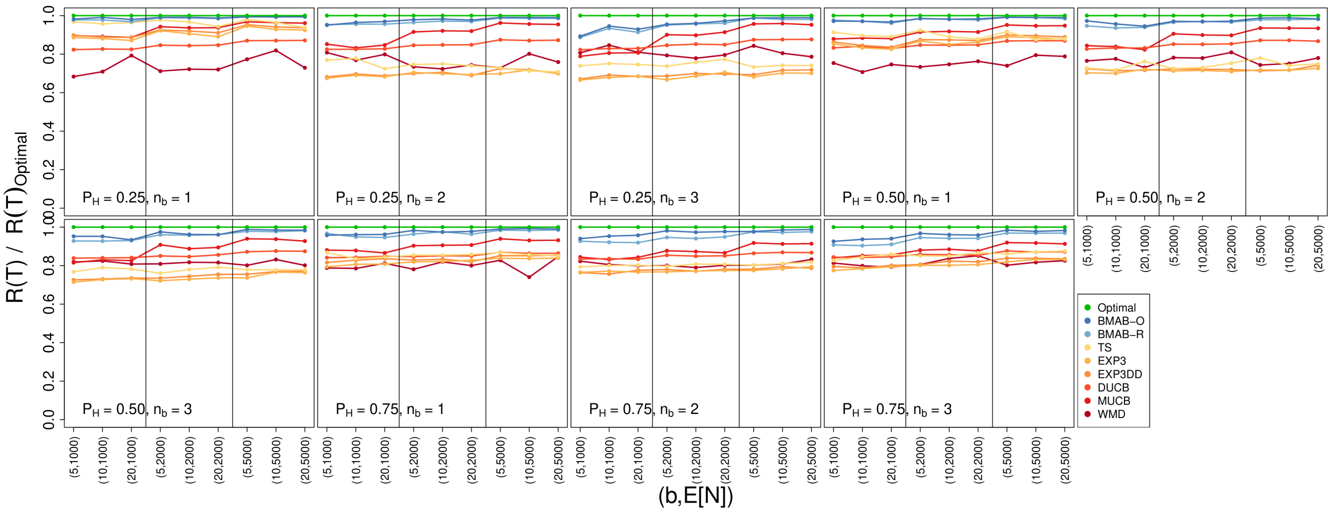

Figure 4: Summary of the results of the synthetic data experiments. The graphs are organized according to the tuple $\{\mathbb{E}[N], P_H, n_b, b\}$. The green line ("Optimal") is the theoretical reward of the best possible algorithm. Each data point corresponds to the average of 20 simulations.

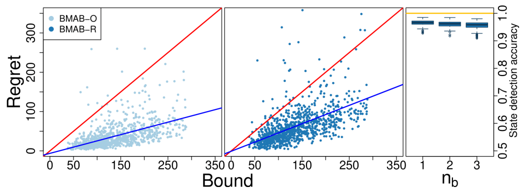

Figure 5: First two graphs: comparison between theoretical guarantees and experimental results of the regret ($\mathcal{R}(T)$). The red line is the function $f(x) = x$. The second line shows the function $f(x) = ax + b$, where a and b are the coefficients of the linear regression fitting of $\mathcal{R}(T)$ versus $O(\mathcal{R}(T))$. Last graph: boxplot of the dispersion of the state detector accuracy.

values: $\lambda_L = 1$, $\mathbb{E}[N] \in \{1000, 1001, \cdots, 5000\}$, $P_H \in [0.4, 0.6]$, $n_b \in \{1, 2, 3\}$, $b \in \{5, 6, \cdots, 20\}$, $K \in \{2, 3, 4\}$. The reward distribution parameters θ_i^0, θ_i^p ($i \leq K, p \leq n_b$) were generated as iid $U(0, 1)$. For each simulation, we chose a random combination of the above parameters and compared the regret the bound to the regret observed when running BMAB.

The two first graphs of Figure 5 are plots of the theoretical regret bound versus the empirical regret related to the 1000 sampled simulations. As expected, all graphs exhibit a linear relation which matches with our theoretical results. At last, the boxplots on the right side of Figure 5 show the distribution of the state detector accuracy as a function of the number of bursts. As can be seen, in all cases, we achieved high accuracy (in average, more than 90% of the states where recovered correctly).

5.2 Real-world data experiments

In this section, we present our results on real-data. We selected the four following recommender systems datasets:

- **[Behance]:** Behance is a social media platform devoted to the dissemination and discussion of creative work. In this RS, each user has the option to appreciate ("like") art.
- **[Google trends]:** We collected the time series related to the singers Psy and David Bowie ($K = 2$) from 2008 to 2020 (YouTube search engine, only USA). The Google trends API only returns a normalized audience (an integer value, maximum 100) for each month. As a result, we modeled each month as 1 unit of time: Jan/2008 $\in (0, 1]$, Feb/2008 $\in (1, 2]$, $\cdots$, Dec/2020 $\in (155, 156]$. To convert normalized audiences into timestamps we generated events uniformly along the corresponding month. For instance, if the API returned x events for Feb/2008, we generated $10x$ events between 1 and 2.
- **[Outbrain]:** Outbrain is a web advertising platform that displays content within websites. The data set contains users' clicks at recommended content.[6]
- **[MovieLens]:** MovieLens is a non-commercial website for movie recommendations. We used MovieLens (25M) which is a broadly used recommendation dataset.[7]

For the Behance, Outbrain and MovieLens datasets we selected the five most popular items as the arms of our bandit problem. The number of arms K, the observed time T and number of events N of all datasets are available in Table 1. In all cases, we split the time period T into two subsets: the first one (T_{λ_L}, cf. Table 1) is used to estimate λ_L with the procedure described in the beginning of

[6] The data is available at https://www.kaggle.com/c/outbrain-click-prediction/data
[7] The data is avaiable at https://grouplens.org/datasets/movielens/

Table 1: Description of the real-world databases and summary of the results of the two experiments strands: synthetic data and real-world data. Metric: average of the observed reward ($R(T)/N$) and its standard deviation (higher values are better). For synthetic data the rewards are normalized by the reward of the Optimal algorithm ($R(T)/R_{\text{Optimal}}(T)$).

	Synthetic data	Behance	Google Trends	Outbrain	MovieLens
(K,N)	–	(5, 7122)	(2, 19850)	(5, 86689)	(5, 270403)
T	–	[Jun to Nov/2011]	[2008,2020]	[14 to 28/Jun/2016]	[Sep/2001,Oct/2019]
T_{λ_L}	–	[Jun/2011]	[2008]	[14/Jun/2016]	[Sep/2001,Dez/2003]
BMAB-O	**0.9721 ± 0.027**	–	–	–	–
BMAB-R	*0.9622 ± 0.034*	**0.5937 ± 0.004**	**0.7756 ± 0.002**	**0.5449 ± 0.008**	**0.22656 ± 0.001**
TS	0.8201 ± 0.091	0.3975 ± 0.042	0.6972 ± 0.033	0.4123 ± 0.018	**0.22652 ± 0.002**
EXP3	0.7777 ± 0.078	0.2202 ± 0.011	0.5249 ± 0.020	0.3053 ± 0.021	0.2223 ± 0.001
EXP3DD	0.7869 ± 0.080	0.2320 ± 0.013	0.5337 ± 0.030	0.3062 ± 0.024	0.2244 ± 0.001
DUCB	0.8516 ± 0.021	0.5014 ± 0.003	0.7616 ± 0.001	0.4852 ± 0.001	0.1701 ± 0.001
MUCB	0.8976 ± 0.047	0.5055 ± 0.006	0.7640 ± 0.002	0.4637 ± 0.001	0.2182 ± 0.001
WMD	0.7854 ± 0.139	0.4197 ± 0.037	0.6951 ± 0.021	0.4237 ± 0.014	0.2039 ± 0.005

Section 5 ; whilst the rest of the data is used to evaluate **[BAMB-R]** and the baselines. Note that both BMAB and the baselines are stochastic. As a result, all the algorithms for each dataset were performed 50 times. For the real datasets the reward is deterministic. At each time t_i, we consider that $r(t_i) = 1$, if the algorithm (correctly) recommends the item that the user liked (Behance, MovieLens), searched (Google trends) or clicked on (Outbrain), and $r(t_i) = 0$ otherwise. We note that our general strategy can be adapted and incorporated in different RS contexts. For example, if rewards or user-reward pairs can be embedded in a dictionary space, we could use our method with a modified version of Thompson Sampling where at each observation, each positive component of a virtual feature vector is Thompson Sampling-updated.

The results are presented in Table 1. We evaluated the performance of **[BMAB-R]** against the baselines by considering the average reward ($R(T)/N$). In real datasets, Optimal and **[BMAB-O]** cannot be defined due to the lack of well-defined bursts. We observe that our algorithm significantly outperformed all the baselines in all the real datasets except MovieLens, where the performance was comparable to that of the Thompson Sampling baseline. This is likely due to the fact that the behavior of the five most popular items is in fact stationary: thus, our algorithm detects no bursts and behaves exactly like Thompson Sampling in that case. Note that in particular, in the Outbrain and Behance datasets, our method outperformed the second best algorithm (**[MUCB]** and **[DUCB]**, respectively) by 12% and 17% (respectively).

6 CONCLUSION

In this paper, we introduced BMAB, a novel algorithm for the piecewise stationary multi-armed bandit problem in a continuous-time context. The main novelty is that we took into account variations in audience activity when modeling the reward distribution. We proved regret guarantees and evaluated them experimentally. Synthetic and real-data experiments demonstrate that our algorithm outperforms many state-of-the-art baselines. By using timestamp information to guide exploration, our work adds a new perspective to the discourse on the exploration/exploitation dilemma kicked off by reinforcement learning. In future work, we plan to investigate how to exploit audience dynamics in more complex reinforcement learning scenarios and more deeply analyze the interaction between the state detector and the reward process. Another relevant question is how susceptible our approach might be to false popularity attacks. This discussion is related to how connected the popularity of the items is to the users' ratings. Thus a reasonable extension is to assume that, instead of each burst having a stationary rewards distribution, an adversary influences the observation of the rewards.

REFERENCES

[1] Deepak Agarwal. 2013. Computational advertising: the linkedin way. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 1585–1586.

[2] Deepak Agarwal, Bo Long, Jonathan Traupman, Doris Xin, and Liang Zhang. 2014. Laser: A scalable response prediction platform for online advertising. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 173–182.

[3] Shipra Agrawal and Navin Goyal. 2013. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*. PMLR, 99–107.

[4] Robin Allesiardo and Raphaël Féraud. 2015. Exp3 with drift detection for the switching bandit problem. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 1–7.

[5] Rodrigo Augusto da Silva Alves, Renato Martins Assuncao, and Pedro Olmo Stancioli Vaz de Melo. 2016. Burstiness scale: A parsimonious model for characterizing random series of events. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1405–1414.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2 (2002), 235–256.

[7] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[8] Peng Bao. 2016. Modeling and predicting popularity dynamics via an influence-based self-excited Hawkes process. In *International Conference on Information and Knowledge Management, Proceedings*.

[9] Albert-László Barabási. 2005. The origin of bursts and heavy tails in human dynamics. *Nature* 435, 7039 (may 2005), 207–211.

[10] Omar Besbes, Yonatan Gur, and Assaf Zeevi. 2014. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems* 27 (2014), 199–207.

[11] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender systems handbook*. Springer, 919–959.

[12] Yang Cao, Zheng Wen, Branislav Kveton, and Yao Xie. 2019. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 418–427.

[13] Emanuele Cavenaghi, Gabriele Sottocornola, Fabio Stella, and Markus Zanker. 2021. Non Stationary Multi-Armed Bandit: Empirical Evaluation of a New Concept Drift-Aware Algorithm. *Entropy* 23, 3 (2021), 380.

[14] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. *Advances in neural information processing systems* 24 (2011), 2249–2257.

[15] Daryl J Daley and David Vere-Jones. 2007. *An introduction to the theory of point processes: volume II: general theory and structure.* Springer Science & Business Media.

[16] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2002. PAC bounds for multi-armed bandit and Markov decision processes. In *International Conference on Computational Learning Theory.* Springer, 255–270.

[17] Crícia Z Felício, Klérisson VR Paixão, Celia AZ Barcelos, and Philippe Preux. 2017. A multi-armed bandit model selection for cold-start user recommendation. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization.* 32–40.

[18] Alceu Ferraz Costa, Yuto Yamaguchi, Agma Juci Machado Traina, Caetano Traina, and Christos Faloutsos. 2015. RSC: Mining and Modeling Temporal Activity in Social Media. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15.* ACM Press, New York, New York, USA, 269–278.

[19] Edouard Fouché, Junpei Komiyama, and Klemens Böhm. 2019. Scaling multi-armed bandit algorithms. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1449–1459.

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

[21] Gourab Ghatak. 2020. A Change-Detection Based ThompsonSampling Framework for Non-Stationary Bandits. *IEEE Trans. Comput.* (2020).

[22] John Gittins. 1974. A dynamic allocation index for the sequential design of experiments. *Progress in statistics* (1974), 241–266.

[23] J. P. Gleeson, D. Cellai, J.-P. Onnela, M. A. Porter, and F. Reed-Tsochas. 2014. A simple generative model of collective online behavior. *Proceedings of the National Academy of Sciences* 111, 29 (jul 2014), 10411–10415.

[24] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA).* IEEE, 133–138.

[25] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *ICML.*

[26] Cédric Hartland, Sylvain Gelly, Nicolas Baskiotis, Olivier Teytaud, and Michele Sebag. 2006. Multi-armed bandit, dynamic environments and meta-bandits. (2006).

[27] Xu He, Bo An, Yanghua Li, Haikai Chen, Qingyu Guo, Xin Li, and Zhirong Wang. 2020. Contextual User Browsing Bandits for Large-Scale Online Mobile Recommendation. In *Fourteenth ACM Conference on Recommender Systems.* 63–72.

[28] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. 2015. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in neural information processing systems.* 1297–1305.

[29] Jon Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD (KDD '02).* ACM, New York, NY, USA, 91–101.

[30] Dhruv Kumar Mahajan, Rajeev Rastogi, Charu Tiwari, and Adway Mitra. 2012. Logucb: an explore-exploit algorithm for comments recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management.* 6–15.

[31] R. Dean Malmgren, Jake M. Hofman, Luis A.N. Amaral, and Duncan J. Watts. 2009. Characterizing individual communication patterns. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09.* ACM Press, New York, New York, USA, 607.

[32] Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. 2012. Rise and fall patterns of information diffusion. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12.* ACM Press, New York, USA.

[33] Swapnil Mishra. 2019. Bridging models for popularity prediction on social media. In *WSDM 2019 - Proceedings of the 12th ACM International Conference on Web Search and Data Mining.*

[34] Julio Cesar Louzada Pinto, Tijani Chahed, and Eitan Altman. 2015. Trend detection in social networks using Hawkes processes. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15.* ACM Press, New York, NY, USA.

[35] Marian-Andrei Rizoiu, Lexing Xie, Scott Sanner, Manuel Cebrian, Honglin Yu, and Pascal Van Hentenryck. 2017. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web.* 735–744.

[36] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.

[37] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2017. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038* (2017).

[38] Javier Sanz-Cruzado, Pablo Castells, and Esther López. 2019. A simple multi-armed nearest-neighbor bandit for interactive recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 358–362.

[39] Suvash Sedhain, Aditya Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. 2017. Low-rank linear cold-start recommendation from social data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[40] Aleksandrs Slivkins. 2019. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272* (2019).

[41] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining.* 445–454.

[42] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

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

[44] Pedro Olmo S Vaz de Melo, Christos Faloutsos, Renato Assunção, and Antonio Loureiro. 2013. The self-feeding process: a unifying model for communication dynamics in the web. In *Proceedings of the 22nd international conference on World Wide Web.* 1319–1330.

[45] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, and Tao Mei. 2016. Time Matters. In *Proceedings of the 24th ACM international conference on Multimedia.* ACM, New York, NY, USA, 1336–1344.

[46] Shuang-hong Yang and Hongyuan Zha. 2013. Mixture of Mutually Exciting Processes for Viral Diffusion. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Sanjoy Dasgupta and David Mcallester (Eds.), Vol. 28. JMLR Workshop and Conference Proceedings, 1–9. http://jmlr.csail.mit.edu/proceedings/papers/v28/yang13a.pdf

[47] Honglin Yu, Lexing Xie, and Scott Sanner. 2015. The Lifecyle of a Youtube Video: Phases, Content and Popularity. http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10537

[48] Jia Yuan Yu and Shie Mannor. 2009. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th annual international conference on machine learning.* 1177–1184.

[49] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. 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.

[50] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15.* ACM Press, New York, New York, USA, 1513–1522.