When People Change their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments

Rolf Jagerman
University of Amsterdam
Amsterdam, The Netherlands
rolf.jagerman@uva.nl

Ilya Markov
University of Amsterdam
Amsterdam, The Netherlands
i.markov@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

We consider the novel problem of evaluating a recommendation policy offline in environments where the reward signal is non-stationary. Non-stationarity appears in many Information Retrieval (IR) applications such as recommendation and advertising, but its effect on off-policy evaluation has not been studied at all. We are the first to address this issue. First, we analyze standard off-policy estimators in non-stationary environments and show both theoretically and experimentally that their bias grows with time. Then, we propose new off-policy estimators with moving averages and show that their bias is independent of time and can be bounded. Furthermore, we provide a method to trade-off bias and variance in a principled way to get an off-policy estimator that works well in both non-stationary and stationary environments. We experiment on publicly available recommendation datasets and show that our newly proposed moving average estimators accurately capture changes in non-stationary environments, while standard off-policy estimators fail to do so.

CCS CONCEPTS

• **Information systems → Evaluation of retrieval results**; *Recommender systems.*

KEYWORDS

Off-policy evaluation; Non-stationary rewards

ACM Reference Format:

Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When People Change their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments. In *Proceedings of The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290958

1 INTRODUCTION

Modern Information Retrieval (IR) systems leverage user interactions such as clicks to optimize which items such as articles, music or movies to show to users [11, 14, 22]. A challenge in utilizing interaction feedback is that it is a "partial label" problem: We only observe feedback for items that were shown to a user, but not for other items that could have been shown. The contextual bandit framework [19] provides a natural way to solve problems with this interactive nature. In the contextual bandit setup, an interactive system (e.g., a recommender system), often called a *policy*, observes a context (e.g., a user visiting a website), performs an action (e.g., by showing a recommendation to the user) and finally observes a reward for the performed action (e.g., a click or no click) [19].

To evaluate a policy, it is best to deploy it online, e.g., in the form of an A/B test. However, this is expensive in terms of engineering and logistic overhead [13, 42] and may harm the user experience [30]. *Off-policy* evaluation is an alternative strategy that avoids the problems of deploying and measuring a policy's performance online [20]. In off-policy evaluation, we use historical interaction data, often referred to as *bandit feedback*, collected by an existing logging policy to estimate the performance of a new policy. In existing work, off-policy evaluation has been well studied in the context of a stationary world, one where interactions happen independent of time [2, 9, 20, 27, 33, 35, 39].

However, IR environments are usually *non-stationary* with user preferences changing over time [16, 24, 25, 29, 41]. Existing off-policy evaluation techniques fail to work in such environments. In this paper, we address the problem of off-policy evaluation in non-stationary environments. We propose several off-policy estimators that operate well when the environment is non-stationary. Our estimators are based on applying two types of moving averages to the collected bandit feedback: (1) a sliding window average and, (2) an exponential decay average. These proposed estimators rely more on recent bandit feedback and, thus, accurately capture changes in non-stationary environments.

We provide a rigorous analysis of the proposed estimators' bias in the non-stationary setting and show that the bias does not grow over time. In contrast, we show that the standard Inverse Propensity Scoring (IPS) estimator suffers from a large bias that grows over time when applied to non-stationary environments. Finally, we use the results from our analysis to create adaptive variants of the sliding window and exponential decay estimators that change their parameters in real-time to improve estimation performance.

We perform extensive empirical evaluation of the proposed off-policy estimators on two recommendation datasets to showcase how they behave under varying levels of non-stationarity. Our main finding is that the proposed estimators significantly outperform the regular IPS estimator and provide a much more accurate estimation of a policy's true performance, while the regular IPS fails to capture the changes in non-stationary environments. Moreover, we demonstrate that these results hold for both smooth and abrupt changes in the environment. Our findings open up the way for off-policy evaluation to be applied to real-world settings where non-stationarity is prevalent.

The remainder of this paper is structured as follows: In Section 2 we provide background information about off-policy evaluation and non-stationarity. Next, Section 3 describes our estimators for

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290958

solving the non-stationary off-policy evaluation problem. The experimental setup and results are described in Section 4 and Section 5, respectively. Finally, we conclude in Section 6.

2 BACKGROUND

2.1 Off-policy evaluation

Off-policy evaluation is an important technique for assessing the behavior of a decision making policy, e.g., a new recommendation strategy, ad-placement technique or some other new feature, without deploying the policy in a classical A/B test [27]. In settings where the deployment of a new policy is costly, either in terms of logistic and engineering overhead or in terms of potential harm to the user experience, off-policy evaluation is a safe and efficient alternative to A/B testing [34]. The main idea in off-policy evaluation is to collect data by having an already deployed policy taking actions and logging the corresponding user interactions. The typical approach in off-policy evaluation is to then re-weigh the logged data according to what the new policy *would have done* to obtain an unbiased estimate of the expected return of this new policy. Although a randomized logging policy is typically required for unbiased off-policy evaluation, the amount of randomness can usually be controlled through some parameter, trading off exploration and exploitation [9].

Existing work in off-policy evaluation has focused on creating unbiased estimators [12, 20, 33] and reducing their variance [2, 9, 39]. However, these off-policy estimators usually do not take into account the *temporal* component and assume the world and rewards are stationary. In contrast to existing work, we postulate that the world is *non-stationary* and create off-policy estimators that take this into account. We should note that Dudík et al. [8] have studied policy evaluation in a different non-stationary setting, namely one where a policy's behavior depends on a history of contexts and actions. However, unlike our work, Dudík et al. still assumes a stationary world and rewards.

In the reinforcement learning domain, the application of time series prediction methods to predict future off-policy performance in non-stationary environments has been studied by Thomas et al. [36]. Our work is different in important ways: (1) Our work focuses on the contextual bandit scenario, whereas theirs is in the reinforcement learning domain. (2) We are the first to develop a theory for non-stationary off-policy evaluation. (3) Their work is designed for small-scale problems, with up to a few thousand iterations as the complexity of their method is quadratic in the number of iterations, whereas our method scales linearly with the number of iterations, enabling experimentation that is two orders of magnitude larger. (4) We target the recommendation setting, whereas Thomas et al. [36] focus on proprietary datasets from digital ad marketing, limiting reproducibility. The only publicly available dataset used in their work is a synthetic scenario called mountain car [32]. Our results are produced on more realistic publicly available recommendation datasets from LastFM [17] and Delicious [4, 6].

Finally, Garivier and Moulines [10] studied the use of sliding-window and exponential decay techniques for optimizing contextual bandits in abruptly changing environments. Our work differs in two ways: (1) we study off-policy evaluation and not contextual bandit learning, and (2) we focus on the smooth non-stationary setting instead of the abrupt non-stationary setting, as we will explain in the next section.

2.2 Non-stationary environments

Non-stationarity environments have been studied in the context of learning multi-armed bandits [3, 10, 21, 40] and contextual bandits [41]. Two settings naturally arise when dealing with a non-stationary world:

(1) abrupt non-stationarity [10, 41], sometimes called piecewise-stationary [21], and
(2) smooth non-stationarity [40].

The first setting, abrupt non-stationarity, assumes a stationary world that changes abruptly at certain points in time. This is a natural setting in, for example, news recommendation, where a sudden event causes a shift in users' interests [18].

The second setting, smooth non-stationarity, assumes that the world changes constantly but that it changes only a little bit at a time. This is the natural condition of human attitudes (including likes and dislikes). Social psychologists have found that preferences are neither enduring nor stable [31, 37]. In cognitive psychology, numerous experiments have provided evidence of gradual taste changes, for instance in response to changing constraints and abilities [1] or in relation to perceived risk levels [15].

Specifically, in settings such as e-commerce [23], music recommendation [24, 28] and news recommendation [7], the behavior of users is often non-stationary in a smooth manner. Pereira et al. [26] have studied non-stationarity in user preferences on social media and have found strong correlations between the temporal dynamics of users' preferences and changes in their social network graph. Taking smooth non-stationarity into account may benefit overall search and recommendation performance, e.g., in music recommendation; Quadrana et al. [28] have found that encoding the evolution of users' listening preferences via recurrent neural networks, can lead to substantial improvements in recommendation quality.

In our work, we specifically design off-policy estimators that deal with the *non-abrupt* case, that is, estimators that work well when the environment exhibits *smooth non-stationarity*.

3 NON-STATIONARY OFF-POLICY EVALUATION

In this section we first formulate the problem of off-policy evaluation in non-stationary environments. In this setting, we prove that the upper bound on the bias of the regular IPS estimator grows with time. Then we propose two alternative estimators, a sliding window approach and an exponential decay approach, and show that their bias can be bounded. Finally, we use our theoretical findings to propose a method that can adaptively set the window-size or the decay rate of our proposed estimators, based on the principle of minimizing the Mean Squared Error (MSE).

3.1 Problem definition

We consider the following two policies: (i) π_0 is a stochastic logging policy that collects data, and (ii) π_w is a new policy that we want to evaluate. We observe an infinite stream of log data, generated by the logging policy π_0. At each time $t = 1, \ldots, \infty$, the following occurs:

(1) The environment generates a context vector x_t and rewards r_t for all possible actions at time t:

$$(x_t, r_t) \overset{\text{i.i.d.}}{\sim} \mathcal{D}_t. \tag{1}$$

The context could, for example, be a user who interacts with a recommender system, while actions could be possible recommendations for that user. The true interest of the user, e.g., what recommendation they are actually interested in, is captured by rewards. We build on previous work [20], which assumes that contexts and rewards are sampled i.i.d. from an unknown distribution $\mathcal{D}_t$. However, unlike previous work, we generalize to a non-stationary world that may change over time. More formally, we allow the distribution $\mathcal{D}_t$ to change with t:

$$\mathcal{D}_1 \neq \mathcal{D}_2 \neq \ldots \neq \mathcal{D}_t. \tag{2}$$

(2) After observing the context vector x_t, the logging policy π_0 samples an action (e.g., given a user, π_0 chooses a recommendation for that user):

$$a_t \sim \pi_0(\cdot \mid x_t) \tag{3}$$

and records the corresponding propensity score:

$$p_t \leftarrow \pi_0(a_t \mid x_t). \tag{4}$$

(3) The reward $r_t(a_t)$ for the chosen action is revealed, but not the rewards for other possible actions that could have been chosen. Without loss of generality we assume $r_t(a_t) \in [0, 1]$. In practice, the reward would be a click or no-click on the recommendation that was shown.

Our goal is to estimate the *value* of the new policy π_w at time t, denoted as $\mathrm{V}_t^*(\pi_w)$, based on the data collected by the logging policy π_0. We write this value as the expected reward of the policy π_w:

$$\mathrm{V}_t^*(\pi_w) = \mathbb{E}_{(x_t, r_t) \sim \mathcal{D}_t, a_t \sim \pi_w(\cdot \mid x_t)} \left[r_t(a_t) \right] = \mathbb{E}_{\pi_w} \left[r_t(a_t) \right]. \tag{5}$$

Finding an estimator for the above quantity would be near impossible if no further assumptions are made about the reward function r_t. For example, if a user's preferences completely changed every time they enter a recommendation website, it would be impossible to perform any type of estimation or evaluation. To make this problem approachable, we assume that the change of a policy's value between any two consecutive points in time is bounded. More formally:

Assumption 3.1. The value of a policy is a Lipschitz function of time:

$$|\mathrm{V}_{t_1}^*(\pi_w) - \mathrm{V}_{t_2}^*(\pi_w)| \leq |t_1 - t_2| k, \tag{6}$$

where k is the Lipschitz-constant.

This assumption ensures that the expected reward of a policy cannot abruptly jump between time t_1 and time t_2. This is supported by practical observations that for real-world recommendation systems user behavior changes slowly over time [24, 25].

3.2 Regular IPS

A widely used policy evaluation technique is Inverse Propensity Scoring (IPS) [12], defined as:

$$\mathrm{V}_t^{\mathrm{IPS}}(\pi_w) = \frac{1}{t} \sum_{i=1}^{t} r_i(a_i) \frac{\pi_w(a_i \mid x_i)}{p_i}. \tag{7}$$

Under the assumption of a stationary world, this is an unbiased estimate of $\mathrm{V}_t^*(\pi_w)$ [12]:

Lemma 3.1. $\mathrm{V}^{\mathrm{IPS}}(\pi_w)$ *is an unbiased estimate of* $\mathrm{V}_t^*(\pi_w)$*, under the assumption of a stationary world* ($\mathcal{D}_1 = \mathcal{D}_2 = \ldots = \mathcal{D}_t$)*.*

Proof. First, we show that for any point in time $i \in \{1, \ldots, t\}$, the IPS estimate of a single observation is unbiased. To do this, we take the expectation of the IPS estimate under the logging policy, and show that it is equal to the reward of the policy under evaluation:

$$\begin{aligned}
\mathbb{E}_{\pi_0}\left[r_i(a_i) \frac{\pi_w(a_i \mid x_i)}{p_i} \right] &= \sum_{a'} r_i(a_i') \frac{\pi_w(a_i' \mid x_i)}{\pi_0(a_i' \mid x_i)} \pi_0(a_i' \mid x_i) \\
&= \sum_{a'} r_i(a_i') \pi_w(a_i' \mid x_i) \\
&= \mathbb{E}_{\pi_w}\left[r_i(a_i) \right].
\end{aligned}$$

Using this fact, it is easy to show that $\mathrm{V}_t^{\mathrm{IPS}}(\pi_w)$ is indeed an unbiased estimate of $\mathrm{V}_t^*(\pi_w)$:

$$\begin{aligned}
\mathbb{E}_{\pi_0}\left[\mathrm{V}_t^{\mathrm{IPS}}(\pi_w) \right] &= \mathbb{E}_{\pi_0}\left[\frac{1}{t} \sum_{i=1}^{t} r_i(a_i) \frac{\pi_w(a_i \mid x_i)}{p_i} \right] \\
&= \frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{\pi_w}\left[r_i(a_i) \right] \\
&= \frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{\pi_w}\left[r_t(a_t) \right] \\
&= \mathbb{E}_{\pi_w}\left[r_t(a_t) \right] = \mathrm{V}_t^*(\pi_w). \quad \square
\end{aligned}$$

When generalizing to a non-stationary world (under Assumption 3.1), it can be shown that the standard IPS estimate is biased.

Lemma 3.2. *Under Assumption 3.1,* $\mathrm{V}_t^{\mathrm{IPS}}(\pi_w)$ *is a biased estimate of* $\mathrm{V}_t^*(\pi_w)$ *and the upper bound on the bias grows with* t*.*

Proof. We have:

$$\begin{aligned}
\mathbb{E}_{\pi_0}\left[\mathrm{V}_t^{\mathrm{IPS}}(\pi_w) \right] &= \mathbb{E}_{\pi_0}\left[\frac{1}{t} \sum_{i=1}^{t} r_i(a_i) \frac{\pi_w(a_i \mid x_i)}{p_i} \right] \\
&= \frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{\pi_w}\left[r_i(a_i) \right] \\
&= \frac{1}{t} \sum_{i=1}^{t} \mathbb{E}_{\pi_w}\left[r_i(a_i) - r_t(a_t) + r_t(a_t) \right] \\
&= \frac{1}{t} \sum_{i=1}^{t} \left(\mathbb{E}_{\pi_w}\left[r_t(a_t) \right] + \mathbb{E}_{\pi_w}\left[r_i(a_i) - r_t(a_t) \right] \right) \\
&= \mathbb{E}_{\pi_w}\left[r_t(a_t) \right] + \frac{1}{t} \sum_{i=1}^{t} \left(\mathbb{E}_{\pi_w}\left[r_i(a_i) \right] - \mathbb{E}_{\pi_w}\left[r_t(a_t) \right] \right) \\
&= \mathrm{V}_t^*(\pi_w) + \frac{1}{t} \sum_{i=1}^{t} \left(\mathrm{V}_t^*(\pi_w) - \mathrm{V}_t^*(\pi_w) \right) \\
&\leq \mathrm{V}_t^*(\pi_w) + \frac{1}{t} \sum_{i=1}^{t} |t - i| k = \mathrm{V}_t^*(\pi_w) + \underbrace{\frac{k(t-1)}{2}}_{\text{bias}} \quad \square
\end{aligned}$$

The upper bound on the bias term, i.e., $\frac{k(t-1)}{2}$, grows with t, which is unfortunate because it means that the more data we observe, the larger our bias potentially becomes. We propose two estimators that deal with this problem by avoiding a bias term that grows with t: the sliding window IPS and the exponential decay IPS, which we describe next.

3.3 Sliding window IPS

The first IPS estimator we propose is the *sliding window IPS estimator*, $\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)$. This estimator only takes into account the τ most recent observations and ignores older ones:

$$\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w) = \frac{1}{\tau}\sum_{i=t-\tau}^{t} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}. \tag{8}$$

This estimator has a bias that does not grow with t, but is instead controlled by the window size τ:

Lemma 3.3. *Under Assumption 3.1, $\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)$ is a biased estimate of $\mathrm{V}_t^*(\pi_w)$ and its bias is at most $\frac{k(\tau-1)}{2}$.*

Proof. This proof largely follows the proof of Lemma 3.2, so we will be concise:

$$\begin{aligned}
\mathbb{E}_{\pi_0}\left[\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)\right] &= \mathbb{E}_{\pi_0}\left[\frac{1}{\tau}\sum_{i=t-\tau}^{t} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right] \\
&\le \mathrm{V}_t^*(\pi_w) + \frac{1}{\tau}\sum_{i=t-\tau}^{t}|t-i|k \\
&= \mathrm{V}_t^*(\pi_w) + \frac{1}{\tau}\sum_{i=1}^{\tau}|\tau-i|k \\
&\le \mathrm{V}_t^*(\pi_w) + \underbrace{\frac{k(\tau-1)}{2}}_{\text{bias}}. \quad \square
\end{aligned}$$

The advantage of the sliding window estimator $\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)$ is that its bias term can be controlled by the window size τ. One may consider setting the window size τ to 1, which would effectively produce an unbiased estimate:

$$\frac{k(\tau-1)}{2} = \frac{k(1-1)}{2} = 0. \tag{9}$$

This is a particularly powerful statement because we would obtain an unbiased estimator even in the face of non-stationarity. Unfortunately, a drawback would be that having such a small window size will cause a large variance.

To formally derive the variance of the $\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)$ estimator, we assume that the following variance does not change over time:

$$\mathbb{V}\left[r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right] = \mathbb{V}\left[r_{i+1}(a_{i+1})\frac{\pi_w(a_{i+1} \mid x_{i+1})}{p_{i+1}}\right]. \tag{10}$$

We make this assumption to simplify writing down the variance of our estimator. To further motivate this assumption, we note that the variance of IPS estimators scales quadratically with the inverse propensity scores [9]. As a result, the variance term of the IPS estimator is dominated by the usually large inverse propensity weights and not by the variance in the rewards. Since we do not change our logging policy over time, the distribution of propensity scores will also not change, and hence we expect the variance to remain constant over time. We can write down the variance of $\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)$ as follows:

$$\begin{aligned}
\mathbb{V}\left[\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)\right] &= \mathbb{V}\left[\frac{1}{\tau}\sum_{i=t-\tau}^{t} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right] \\
= \frac{1}{\tau^2}\sum_{i=t-\tau}^{t}\mathbb{V}\left[r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right] &= \frac{1}{\tau}\mathbb{V}\left[r_t(a_t)\frac{\pi_w(a_t \mid x_t)}{p_t}\right].
\end{aligned}$$

As we can see, the variance scales by $\frac{1}{\tau}$, which means larger values of τ reduce variance and conversely smaller values of τ increase variance.

Hence, setting the window size is a trade-off between how much bias and variance we are willing to tolerate. We will see a similar bias-variance trade-off in the next estimator, the exponential decay IPS.

3.4 Exponential decay IPS

The *exponential decay IPS estimator*, $\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)$, uses an exponential moving average to weigh recent observations more heavily than old observations:

$$\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w) = \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}, \tag{11}$$

where $\alpha \in (0,1)$ is a hyper parameter controlling the rate of decay. A large value of α indicates a slow decay, meaning that old observations weigh more heavily. Conversely, a small value of α indicates a rapid decay, which means recent observations weigh more heavily.

The bias of this estimator does not grow with t and is controlled by the decay rate α:

Lemma 3.4. *Under Assumption 3.1, $\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)$ is a biased estimate of $\mathrm{V}_t^*(\pi_w)$ and its bias is at most $\frac{k\alpha}{(1-\alpha)(1-\alpha^t)}$.*

Proof. For notational simplicity, we define $\mathrm{V}_i^* = \mathrm{V}_i^*(\pi_w)$. Then:

$$\begin{aligned}
\mathbb{E}_{\pi_0}\left[\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)\right] &= \mathbb{E}_{\pi_0}\left[\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right] \\
&= \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}\mathbb{E}_{\pi_w}\left[r_i(a_i)\right] \\
&= \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}(\mathrm{V}_i^* - \mathrm{V}_t^* + \mathrm{V}_t^*) \\
&= \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}\mathrm{V}_t^* + \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}(\mathrm{V}_i^* - \mathrm{V}_t^*) \\
&= \mathrm{V}_t^* + \underbrace{\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}(\mathrm{V}_i^* - \mathrm{V}_t^*)}_{\text{bias}}.
\end{aligned}$$

We can further simplify the bias term as follows:

$$\begin{aligned}
\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}(\mathrm{V}_i^* - \mathrm{V}_t^*) &\le \frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}|t-i|k \\
&= k\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}|t-i|.
\end{aligned}$$

Note that $\sum_{i=1}^{t}\alpha^{t-i}|t-i|$ is a convergent series for $|\alpha| < 1$:

$$\sum_{i=1}^{t}\alpha^{t-i}|t-i| \le \frac{\alpha}{(1-\alpha)^2}.$$

Plugging this expression into the above equation completes the proof:

$$k\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i}|t-i| \le k\frac{1-\alpha}{1-\alpha^t}\frac{\alpha}{(1-\alpha)^2} = \frac{k\alpha}{(1-\alpha)(1-\alpha^t)}. \quad \square$$

The bias term $\frac{k\alpha}{(1-\alpha)(1-\alpha^t)}$ exhibits behavior that we expect: If k is large, and thus the environment is highly non-stationary, the estimate will be more biased. Conversely, when $k = 0$, we recover the stationary case and have an unbiased estimator. Furthermore, when α approaches 1, the bias term grows because we weigh old observations more heavily. Finally, we note that $\lim_{t\to\infty}(1-\alpha^t) = 1$ and thus t vanishes from the bias term as t approaches infinity.

Let us now consider the variance of the exponential decay IPS estimator. Similarly to the sliding window IPS estimator, we assume that the variance does not change over time. This gives us:

$$\begin{aligned}\mathbb{V}\left[\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)\right] &= \mathbb{V}\left[\frac{1-\alpha}{1-\alpha^t}\sum_{i=1}^{t}\alpha^{t-i} r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right]\\ &= \left(\frac{1-\alpha}{1-\alpha^t}\right)^2\sum_{i=1}^{t}\alpha^{2(t-i)}\mathbb{V}\left[r_i(a_i)\frac{\pi_w(a_i \mid x_i)}{p_i}\right]\\ &= \left(\frac{1-\alpha}{1-\alpha^t}\right)^2\left(\frac{1-\alpha^{2t}}{1-\alpha^2}\right)\mathbb{V}\left[r_t(a_t)\frac{\pi_w(a_t \mid x_t)}{p_t}\right].\end{aligned}$$

As expected, the variance scaling factor $\left(\frac{1-\alpha}{1-\alpha^t}\right)^2\left(\frac{1-\alpha^{2t}}{1-\alpha^2}\right)$ decreases as α goes to 1. Conversely, the variance increases as α goes to 0. Similarly to the bias term, we see that t vanishes from the variance as t approaches infinity: $\lim_{t\to\infty}\left(\frac{1-\alpha}{1-\alpha^t}\right)^2\left(\frac{1-\alpha^{2t}}{1-\alpha^2}\right) = \frac{1-\alpha}{1+\alpha}$.

3.5 How to choose τ and α

Compared to regular IPS estimators, $\mathrm{V}_t^{\tau\mathrm{IPS}}$ and $\mathrm{V}_t^{\alpha\mathrm{IPS}}$ have additional parameters τ and α, respectively, that need to be set.

Let us first consider the scenario where an unbiased estimator is the goal. We can set $\tau = 1$ or $\alpha = 0$ to obtain an unbiased estimator. This is equivalent to computing an IPS estimate on only the current observation. It is obvious that such a strategy will suffer from high variance and is not very useful in practice.

Conversely, if we were to consider the scenario where an estimator with minimal variance is the goal, we could set $\tau = \infty$ or α arbitrarily close to 1, resulting in an estimator that would heavily weigh as many old observations as possible. This is also a poor strategy as it would result in potentially unbounded bias.

Setting τ or α comes down to finding a balance between bias and variance. A principled way to trade off these quantities is by minimizing the mean squared error of the estimator [38]:

$$MSE = \mathrm{bias}^2 + \mathrm{variance}.$$

If the Lipschitz constant k is known, we can compute τ or α that minimizes the mean squared error at every time t as follows:

$$\begin{aligned}\tau_t^* &= \operatorname*{argmin}_{\tau\in\mathbb{N}} \lambda\left(\frac{k(\tau-1)}{2}\right)^2 + \mathbb{V}\left[\mathrm{V}_t^{\tau\mathrm{IPS}}(\pi_w)\right],\\ \alpha_t^* &= \operatorname*{argmin}_{\alpha\in[0,1)} \lambda\left(\frac{k\alpha}{(1-\alpha)(1-\alpha^t)}\right)^2 + \mathbb{V}\left[\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)\right],\end{aligned}$$

where λ is a hyperparameter that trades off bias for variance. In practice we would tune λ to achieve a good trade-off.

Finding the optimal values τ_t^* and α_t^* requires knowledge about the Lipschitz constant k which is usually not known in practice. In the next section, we describe a heuristic that estimates k.

3.6 Estimating the Lipschitz constant k

The Lipschitz constant k tells us how fast the true value of a policy is moving (see Eq. (6)). Since the true value $\mathrm{V}_t^*(\pi_w)$ cannot be observed without deploying the policy π_w, we rely on the IPS estimated rewards. To estimate k, we track the difference between two moving averages: one at time t, denoted as V_t and one at time $t-s$, denoted as V_{t-s}, where $s > 0$ is a parameter representing a window size for estimating k.

Now, we can estimate k at every time t as follows:

$$\hat{k}_t = \frac{1}{s}\left(\mathrm{V}_t - \mathrm{V}_{t-s}\right), \tag{12}$$

where V_t is a moving average estimator at time t. For example, V_t could be the exponential decay estimator $\mathrm{V}_t^{\alpha\mathrm{IPS}}(\pi_w)$.

Tracking the difference between two averages at different points in time has previously been used as a change-point detection mechanism for contextual bandits. For example, the windowed mean-shift algorithm uses a very similar method to detect when an abrupt change occurs [43]. Our heuristic is different in the fact that it does not detect an abrupt change, but instead is measuring how fast the true value of the policy is moving up and down.

4 EXPERIMENTAL SETUP

In this section we describe our experimental setup. The goal of our experiments is to answer the following research questions: (1) How well do the proposed estimators perform in a non-stationary environment? (2) How well do the estimators function when Assumption 3.1 is violated? E.g., when the environment changes abruptly? (3) Can the proposed estimators be applied to stationary environments? (4) How do the estimators behave under different parameters?

To answer these questions we consider a simulated non-stationary contextual bandit setup as described in [41]. Note that although our setup is the same as in [41], we are solving a different problem: particularly, we perform off-policy evaluation whereas Wu et al. [41] perform online learning.

4.1 Experimental methodology

We evaluate our proposed off-policy estimators in the context of recommendation, where a policy recommends an item to a user. We use the non-stationary contextual bandit setup of Wu et al. [41], which, in turn, builds on the experimental setup of Cesa-Bianchi et al. [5]. In this experimental setup, we use two datasets made available as part of the HetRec2011 workshop [4, 6, 17] and convert them into a contextual bandit problem: LastFM and Delicious. For the LastFM dataset [17], we consider a random artist that the user has listened to as positive feedback and an artist that the user has not listened to as negative feedback. For the Delicious dataset [6], we consider a website that the user has bookmarked as positive feedback and websites that the user has not bookmarked as negative feedback. For each user we consider a random positive item and 24 random negative items as the set of candidate actions. Correspondingly, a reward of 1 is given if a policy chooses the positive item and 0 otherwise. Each item is described by a TF-IDF feature vector comprised of the item's tags, e.g., "metal", "electronic", "rock", etc. in the case of music recommendation (LastFM), and "social", "games", "tech", etc. in the case of bookmark recommendation (Delicious). This feature vector is reduced to 25 dimensions via PCA, as described in [5].

To introduce non-stationarity we follow the setup of Wu et al. [41]: We cluster users into 10 user groups (or super-users) via spectral clustering based on the social network graph structure. Users who are close in the social network graph are hypothesized to have similar preferences.

Then, a single hybrid user is created from the 10 super-users by stacking the preferences of the 10 user groups chronologically. This hybrid user is non-stationary because its preferences change when it moves from one super-user to the next. In [41], the hybrid user switches abruptly between the 10 super-users at certain points in time. We experiment with the existing abrupt case of [41] and introduce a setup where a mixture of the 10 super-users slowly changes over time as observed in real-world recommender systems [24, 25]. Below we describe both setups in detail.

The hybrid user can be represented by a mixture with 10 components which add up to 1. For example:

$$[0, \quad 1, \quad 0, \quad 0, \quad \ldots, \quad 0].$$

To simulate a *smooth* non-stationary setup, we introduce a transition period from time t_1 to t_2. In this transition period we change two components, linearly reducing one component while linearly increasing the other. For example, changing from the second to the third super-user would happen as follows:

$$\begin{array}{llllllll}
t_1: & [0, & 1, & 0, & 0, & \ldots, & 0] \\
t_1+1: & [0, & 0.9, & 0.1, & 0, & \ldots, & 0] \\
t_1+2: & [0, & 0.8, & 0.2, & 0, & \ldots, & 0] \\
\vdots & & & & & & \\
t_2: & [0, & 0, & 1, & 0, & \ldots, & 0].
\end{array}$$

This setup is in line with Assumption 3.1, which states that the environment changes only a little bit at a time and not abruptly.

In the *abrupt* setup, one component is set to 1 and the other components are set to 0. An abrupt change happens by changing which component is set to 1. This is equivalent to the setup in [41] where the hybrid user switches abruptly between the 10 super-users.

Using the non-stationary setups described in this section, we can now deploy a logging policy that collects bandit feedback and evaluate a set of candidate policies using the proposed off-policy estimators. The choice of a logging policy and candidate policies are described next.

4.2 Logging policy

As described in Section 3.1, the deployed logging policy π_0 logs the data on which we evaluate our candidate policies. The logging policy was trained via LinUCB [19], which is a state-of-the-art contextual bandit method, across all super-users and is expected to function well on average.

To ensure a logging policy that explores, we make the logging policy stochastic and give it full support (that is, every action has a non-zero probability). This is accomplished by using ϵ-greedy exploration (with $\epsilon > 0$) [32]; ϵ-greedy exploration selects an action uniformly at random with probability ϵ and the best action (according to the logging policy) with probability $(1-\epsilon)$. The ϵ parameter allows us to trade off the exploration aggressiveness and the performance of the logging policy.

On the one hand, we want a logging policy that explores aggressively, so as to obtain as much information as possible in the logged feedback. On the other hand, we want a logging policy that performs well, as it is the only component that is exposed to users of the system and we would not want to hurt the user experience. We use $\epsilon = 0.2$ in our experiments, resulting in a policy that exploits 80% of the time and explores 20% of the time. Exploration is necessary for off-policy evaluation and $\epsilon = 0.2$ strikes a decent balance where the logging policy is expected to still function well. We definitely want to avoid $\epsilon = 0$ because it would result in a deterministic policy which is problematic for off-policy evaluation and we want to avoid $\epsilon = 1$ because it is unrealistic to expect a purely random policy to be deployed. In practice one would want to deploy a policy that mostly performs the best actions but performs a little bit of exploration, thus ϵ tends to be closer to 0 than 1.

4.3 Candidate policies

To perform off-policy evaluation we need a set of candidate policies. These are policies whose performance we wish to estimate. In our experiments, candidate policies are trained via LinUCB on each of the 10 super-users, thus, resulting in 10 candidate policies. Each of the 10 candidate policies is expected to work well when the hybrid user switches to the super-user the candidate policy was trained on, but is expected to underperform at any other point in time.

4.4 Ground-truth and metrics

To evaluate how well an estimator predicts a policy's performance we require a ground-truth, i.e., the true performance of a policy. The ground-truth can obtained by deploying the policy and measuring how well it actually performs [8]. According to our task definition, this cannot be done in practice as we have only one deployed logging policy, which does not change. However, in our experimental setup we have full control over the environment and so can simulate the deployment of any candidate policy and measure its true performance. This is done by, at any point in time t, running the candidate policy for 20,000 contextual bandit interactions (observing a context, playing an action and obtaining a reward) and then averaging the rewards.

To evaluate estimators, we measure the Mean Squared Error (MSE) between the estimated policy performance, given by the estimators, and the ground-truth, obtained as described above. The reported MSE values are averaged across the 10 candidate policies described in the previous section. Lower values of MSE correspond to better performance. To measure statistical significance, we run each experiment 20 times and compare the outcomes of the considered off-policy estimators using a paired two-tailed t-test.

4.5 Hyperparameters

Some of the estimators require setting a parameter. For example, $V^{\tau\text{IPS}}$ and $V^{\alpha\text{IPS}}$ require a window size τ and a decay rate α, respectively. The adaptive variants require us to set λ, which trades off variance and bias, and s, which is the Lipschitz estimation window. We found parameters that minimize MSE via a grid search. The final parameters are displayed in Table 1.

5 RESULTS

In this section, we present the results of our empirical evaluation. We separate our results in four sections, each answering one of our research questions. The overall results of the proposed methods are presented in Tables 2 and 3. The figures for the sliding window estimator $V^{\tau\text{IPS}}$ and exponential decay estimator $V^{\alpha\text{IPS}}$ are very similar to each other, so due to the lack of space we present the

Table 1: The best parameters (in terms of minimizing MSE) for each estimator after a grid search.

Estimator	LastFM	Delicious
$V^{\tau\text{IPS}}$	$\tau = 10{,}000$	$\tau = 50{,}000$
$V^{\tau\text{IPS}}$ (adaptive)	$\tau = 10{,}000$	$\tau = 50{,}000$
	$\lambda = 0.00005$	$\lambda = 0.00001$
	$s = 50{,}000$	$s = 30{,}000$
$V^{\alpha\text{IPS}}$	$\alpha = 0.9999$	$\alpha = 0.99995$
$V^{\alpha\text{IPS}}$ (adaptive)	$\alpha = 0.99995$	$\alpha = 0.99995$
	$\lambda = 0.00005$	$\lambda = 0.00005$
	$s = 50{,}000$	$s = 100{,}000$

figures for $V^{\alpha\text{IPS}}$ in the paper, while the figures for $V^{\tau\text{IPS}}$ can be found given in the supplementary material.

Table 2: Mean Squared Error ($\times 10^{-3}$) on the LastFM dataset. Lower is better. We use $\triangledown$ and $\vartriangle$ to denote statistically significantly ($p < 0.01$) lower and higher MSE respectively compared to V^{IPS}. For the adaptive estimators we use $\blacktriangledown$ and $\blacktriangle$ to denote statistically significantly ($p < 0.01$) lower and higher MSE compared to their non-adaptive counterparts.

Estimator	Smooth	Abrupt	Stationary
V^{IPS}	0.029	7.787	1.183
$V^{\tau\text{IPS}}$	1.709 $^{\triangledown}$	3.041 $^{\triangledown}$	1.657 $^{\vartriangle}$
$V^{\tau\text{IPS}}$ (adaptive)	1.565 $^{\triangledown\blacktriangledown}$	2.881 $^{\triangledown\blacktriangledown}$	1.407 $^{\vartriangle\blacktriangledown}$
$V^{\alpha\text{IPS}}$	1.541 $^{\triangledown}$	2.981 $^{\triangledown}$	1.408 $^{\vartriangle}$
$V^{\alpha\text{IPS}}$ (adaptive)	1.546 $^{\triangledown}$	3.067 $^{\triangledown\blacktriangle}$	1.278 $^{\vartriangle\blacktriangledown}$

Table 3: Mean Squared Error ($\times 10^{-3}$) on the Delicious dataset. Lower is better. Statistical significance is denoted in the same way as in Table 2.

Estimator	Smooth	Abrupt	Stationary
V^{IPS}	0.312	0.469	0.022
$V^{\tau\text{IPS}}$	0.111 $^{\triangledown}$	0.268 $^{\triangledown}$	0.058 $^{\vartriangle}$
$V^{\tau\text{IPS}}$ (adaptive)	0.116 $^{\triangledown\blacktriangle}$	0.260 $^{\triangledown\blacktriangledown}$	0.045 $^{\vartriangle\blacktriangledown}$
$V^{\alpha\text{IPS}}$	0.099 $^{\triangledown}$	0.218 $^{\triangledown}$	0.070 $^{\vartriangle}$
$V^{\alpha\text{IPS}}$ (adaptive)	0.104 $^{\triangledown\blacktriangle}$	0.230 $^{\triangledown\blacktriangle}$	0.047 $^{\vartriangle\blacktriangledown}$

5.1 Smooth non-stationarity

Our first and main research question is:

> *How well do the proposed estimators perform in a non-stationary environment?*

The first column of Tables 2 and 3 shows that the proposed $V^{\tau\text{IPS}}$ and $V^{\alpha\text{IPS}}$ off-policy estimators have significantly lower MSE than the standard V^{IPS} estimator, being three times more effective in estimating the actual performance of a recommendation policy on both the LastFM and Delicious datasets. Note that the rewards on the LastFM dataset are higher than those on the Delicious dataset, which is in line with the results of [41] and can be attributed to the fact that it is easier to recommend correct artists (and, thus, accumulate higher reward) than to recommend correct websites, because the number of artists is smaller than the number of websites.

To better understand the behavior of the proposed off-policy estimators over time, we plot the actual and estimated rewards of one of the 10 candidate policies in Figs. 1 and 2 (the choice of a policy is not important, we use policy 6 in all figures). We only present figures for the exponential decay estimator $V^{\alpha\text{IPS}}$ here; the figures for the sliding window estimator $V^{\tau\text{IPS}}$ are similar.

The top plots in Figs. 1 and 2 show that the $V^{\alpha\text{IPS}}$ estimator closely follows the actual performance of a recommendation policy on both the LastFM and Delicious datasets. The standard V^{IPS} estimator, instead, fails to approximate the policy's actual performance and accumulates a large amount of bias.

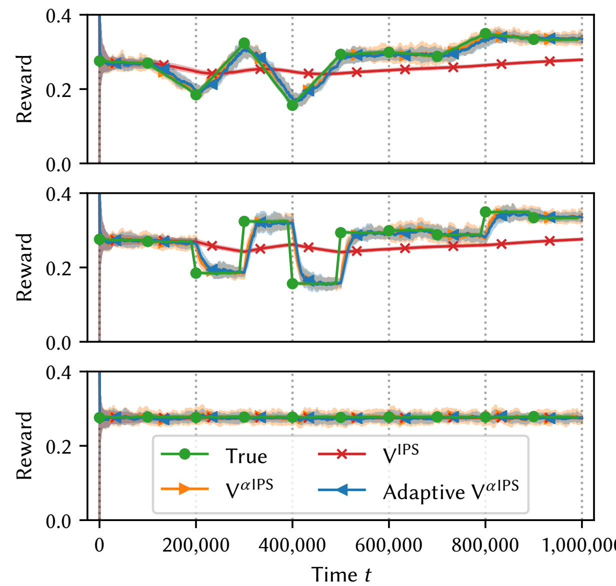

Figure 1: Exponential decay estimators in a smooth (top row), abrupt (middle row) and stationary (bottom row) setting on the LastFM dataset. The shaded areas indicate the standard deviation across 20 runs.

The adaptive variants of our proposed off-policy estimators perform similarly to their non-adaptive counterparts, outperforming or underperforming the latter in a few cases (see Tables 2 and 3). This means that in the smooth non-stationary setup we can use either type of estimator. Below we will show that in other setups adaptive estimators should be preferred over non-adaptive ones.

5.2 Abrupt non-stationarity

Our work builds on the assumption of a smooth non-stationary environment, one where the world changes slowly over time. We wish to investigate how well our estimators work when this assumption is violated, i.e., when the world behaves in an abrupt non-stationary way. This leads to our second research question:

> *How well do the estimators function when Assumption 3.1 is violated? E.g., when the environment changes abruptly?*

The second column of Tables 2 and 3 indicates that in the abrupt non-stationary setup the MSE of $V^{\tau\text{IPS}}$ and $V^{\alpha\text{IPS}}$ is about two times lower than the MSE of V^{IPS} (all differences are statistically significant). This shows that our proposed estimators approximate the actual performance of a policy well even when the theoretical upper bounds on the estimators' bias are no longer valid.

The second row of Figs. 1 and 2 further confirms this by showing that the $V^{\alpha\text{IPS}}$ estimator closely follows the true reward of a policy

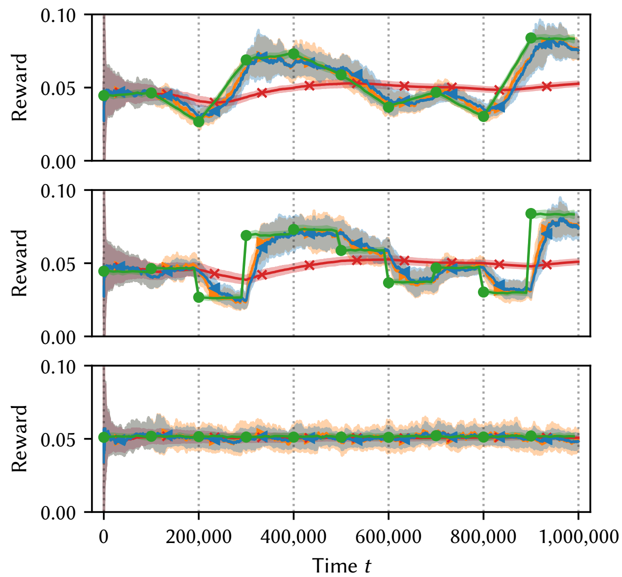

Figure 2: Exponential decay estimators in a smooth (top row), abrupt (middle row) and stationary (bottom row) setting on the Delicious dataset. The shaded areas indicate the standard deviation across 20 runs. See Fig. 1 for the legend.

even if the changes in the environment are abrupt. The standard V^{IPS} estimator still cannot follow the true reward in this setup.

5.3 Stationary environment

Our third research question is:

> *Can the proposed estimators be applied to stationary environments?*

In the stationary environment, the regular V^{IPS} estimator is guaranteed to perform the best: in this setup it is unbiased and has variance that goes to zero when t grows [9]. Our proposed estimators are also unbiased in the stationary environment, but their variance does not decrease over time. Thus, we expect the V^{IPS} estimator to outperform $V^{\tau IPS}$ and $V^{\alpha IPS}$ in the stationary setup.

The above intuitions are confirmed by the results in the last column of Tables 2 and 3. The V^{IPS} estimator indeed has the lowest MSE compared to all other estimators. Interestingly, $V^{\tau IPS}$ and $V^{\alpha IPS}$ are also able to approximate the true reward of a policy relatively well. Particularly, the MSE of $V^{\tau IPS}$ and $V^{\alpha IPS}$ on the LastFM dataset is at most 0.4 times higher than the MSE of V^{IPS} (recall, that V^{IPS} has 2–3 times higher MSE in the non-stationary setups). On the Delicious dataset the differences in MSE are larger, but the absolute MSE values are an order of magnitude smaller than in the non-stationary setups. The adaptive variants of our estimators are significantly better than the non-adaptive ones in the stationary environment, having much lower MSE: the adaptive variants are able to detect the stationary situation, adapt their parameters appropriately and reduce their overall variance.

Thus, we can conclude that although designed for non-stationary environments, the $V^{\tau IPS}$ and $V^{\alpha IPS}$ estimators, and especially their adaptive variants, can be applied in stationary environments. This is further confirmed by the bottom plots in Figs. 1 and 2, where all estimators closely follow the true (stationary) reward.

5.4 Impact of parameters

The final research question that we answer is:

> *How do the estimators behave under different parameters?*

To answer this question, we have investigated different parameter settings for τ, α and λ. In Fig. 3, we plot the true and estimated rewards for different values of α for the $V^{\alpha IPS}$ estimator on the LastFM dataset. The observations for τ and λ are very similar, so we omit these results to save space. From Fig. 3, we see that setting α is a trade-off in bias and variance. This is in line with our theoretical results (Section 3), which state that as α approaches 1, we expect lower variance but higher bias, and vice versa for $\alpha \longrightarrow 0$. The same results hold for the window size τ (a higher value causes lower variance and higher bias) and the λ parameter, which, by design, trades off variance and bias.

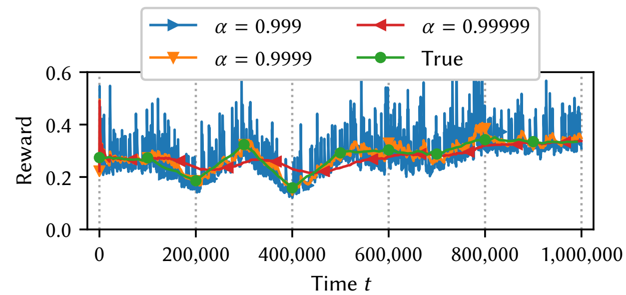

Figure 3: Impact of the α parameter on the exponential decay estimator $V^{\alpha IPS}$ in a smooth non-stationary setting on the LastFM dataset.

6 CONCLUSION

In this paper we studied *non-stationary* off-policy evaluation. We showed that in non-stationary environments the traditional IPS off-policy estimator fails to approximate the true performance of a recommendation policy and suffers from a large bias that grows over time. To address the problem of non-stationary off-policy evaluation, we proposed two estimators that closely follow the changes in the true performance of a policy: one using a sliding window average and one using an exponential decay average. Our analysis of the proposed estimators shows that their bias that does not grow over time and can be bounded. The bias of our estimators can be controlled by the window size τ and the decay rate α. Using the results of our analysis, we proposed a principled way to adapt τ and α automatically according to the changing environment.

We evaluated the proposed estimators in non-stationary recommendation environments using the LastFM and Delicious data sets. The experimental results show that our estimators approximate the policy's actual performance well, having MSE that is 2–3 times lower than that of the standard IPS estimator. We showed that these results hold not only in smooth non-stationary environments, where we can derive upper bounds on the bias of our estimators, but also in the abrupt non-stationary setup, where the theory does not hold. Finally, our results suggest that the proposed off-policy estimators, although designed for non-stationary environments, can be applied in the stationary setup with adaptive variants of the proposed estimators being particularly effective. These findings open up the way for off-policy evaluation to be applied to practical non-stationary real-world scenarios.

An interesting direction for future work is to investigate the use of more advanced off-policy estimators such as Doubly Robust [9] or Switch [39] in non-stationary environments. We hypothesize that such estimators will also suffer from a large bias, while the moving average estimators will be able to solve this issue.

Code

The code for re-running all of the experiments in the paper is available at https://github.com/rjagerman/wsdm2019-nonstationary.

Acknowledgments

This research was supported by Ahold Delhaize, the Innovation Center for Artificial Intelligence (ICAI), the Netherlands Institute for Sound and Vision, and the Netherlands Organization for Scientific Research (NWO) under project nrs CI-14-25, and 612.001.551. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Elliot Aronson. 2008. *The Social Animal* (10th ed.). Worth/Freeman.
[2] Heejung Bang and James M Robins. 2005. Doubly Robust Estimation in Missing Data and Causal Inference Models. *Biometrics* 61, 4 (2005), 962–973.
[3] Omar Besbes, Yonatan Gur, and Assaf Zeevi. 2014. Stochastic Multi-Armed-Bandit Problem with Non-stationary Rewards. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 1. 199–207.
[4] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*. ACM, New York, NY, USA.
[5] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. 2013. A Gang of Bandits. In *Advances in Neural Information Processing Systems 26*. 737–745.
[6] Delicious. 2018. Delicious website. http://www.delicious.com. (2018). Accessed: 2018-08-09.
[7] Fernando Diaz. 2009. Integration of News Content into Web Results. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. ACM, 182–191.
[8] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. 2012. Sample-efficient Nonstationary Policy Evaluation for Contextual Bandits. *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence* (2012), 247–254.
[9] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. *Proceedings of the 28th International Conference on International Conference on Machine Learning* (2011), 1097–1104.
[10] Aurélien Garivier and Eric Moulines. 2011. On Upper-Confidence Bound Policies for Switching Bandit Problems. *Proceedings of the 22nd International Conference on Algorithmic Learning Theory* (2011), 174–188.
[11] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing Historical Interaction Data for Faster Online Learning to Rank for IR. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 183–192.
[12] Daniel G Horvitz and Donovan J Thompson. 1952. A Generalization of Sampling Without Replacement From a Finite Universe. *J. Amer. Statist. Assoc.* 47, 260 (1952), 663–685.
[13] Rolf Jagerman, Krisztian Balog, and Maarten de Rijke. 2018. OpenSearch: Lessons Learned from an Online Evaluation Campaign. *J. Data and Information Quality* 10, 3 (2018), Article 13.
[14] Thorsten Joachims. 2002. Optimizing Search Engines using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 133–142.
[15] Daniel Kahneman, Paul Slovic, and Amos Tversky (Eds.). 1982. *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge University Press.
[16] Anagha Kulkarni, Jaime Teevan, Krysta M. Svore, and Susan T. Dumais. 2011. Understanding Temporal Query Dynamics. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. ACM, 167–176.
[17] Last.fm. 2018. Last.fm website. http://www.lastfm.com. (2018). Accessed: 2018-08-09.
[18] Damien Lefortier, Pavel Serdyukov, and Maarten de Rijke. 2014. Online Exploration for Detecting Shifts in Fresh Intent. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 589–598.
[19] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 661–670.
[20] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased Offline Evaluation of Contextual-Bandit-Based News Article Recommendation Algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 297–306.
[21] Fang Liu, Joohyun Lee, and Ness Shroff. 2018. A Change-Detection based Framework for Piecewise-stationary Multi-Armed Bandit Problem. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (2018).
[22] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized News Recommendation Based on Click Behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*. ACM, 31–40.
[23] Tucker S McElroy, Brian C Monsell, and Rebecca J Hutchinson. 2018. *Modeling of Holiday Effects and Seasonality in Daily Time Series*. Technical Report. Center for Statistical Research and Methodology.
[24] Joshua L Moore, Shuo Chen, Douglas Turnbull, and Thorsten Joachims. 2013. Taste Over Time: The Temporal Dynamics of User Preferences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*. 401–406.
[25] Olfa Nasraoui, Jeff Cerwinske, Carlos Rojas, and Fabio Gonzalez. 2007. Performance of Recommendation Systems in Dynamic Streaming Environments. In *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 569–574.
[26] Fabíola S. F. Pereira, João Gama, Sandra de Amo, and Gina M. B. Oliveira. 2018. On Analyzing User Preference Dynamics with Temporal Social Networks. *Machine Learning* 107, 11 (2018), 1745–1773.
[27] Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the 17th International Conference on Machine Learning*. 759–766.
[28] Massimo Quadrana, Marta Reznakova, Tao Ye, Erik Schmidt, and Hossein Vahabi. 2018. Modeling Musical Taste Evolution with Recurrent Neural Networks. *arXiv preprint arXiv:1806.06535* (2018).
[29] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. 2012. Modeling and Predicting Behavioral Dynamics on the Web. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 599–608.
[30] Tobias Schnabel, Paul N Bennett, Susan T Dumais, and Thorsten Joachims. 2018. Short-Term Satisfaction and Long-Term Coverage: Understanding How Users Tolerate Algorithmic Exploration. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. ACM, 513–521.
[31] Norbert Schwarz and Fritz Strack. 1991. Context Effects in Attitude Surveys: Applying Cognitive Theory to Social Research. *European Review of Social Psychology* 2 (1991), 31–50.
[32] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT press.
[33] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy Evaluation for Slate Recommendation. In *Proceedings of the 31st Conference on Neural Information Processing Systems*. 3632–3642.
[34] Philip S. Thomas and Emma Brunskill. 2016. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, Vol. 48. JMLR.org, 2139–2148.
[35] Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. 2015. High-Confidence Off-Policy Evaluation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 3000–3006.
[36] Philip S Thomas, Georgios Theocharous, Mohammad Ghavamzadeh, Ishan Durugkar, and Emma Brunskill. 2017. Predictive Off-Policy Policy Evaluation for Nonstationary Decision Problems, with Applications to Digital Marketing.. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 4740–4745.
[37] Roger Tourangeau. 1992. Context Effects on Attitude Responses: The Role of Retrieval and Necessary Structures. In *Context Effects in Social and Psychological Research*, Norbert Schwarz and Seymour Sudman (Eds.). Springer, 35–47.
[38] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. 2014. *Mathematical Statistics with Applications*. Cengage Learning.
[39] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudik. 2017. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning*. 3589–3597.
[40] Peter Whittle. 1988. Restless Bandits: Activity Allocation in a Changing World. *J. Applied Probability* 25, A (1988), 287–298.
[41] Qingyun Wu, Naveen Iyer, and Hongning Wang. 2018. Learning Contextual Bandits in a Non-stationary Environment. *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018).
[42] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2227–2236.
[43] 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*. ACM, 1177–1184.