Pessimistic Decision-Making for Recommender Systems*

OLIVIER JEUNEN, Amazon, United Kingdom
BART GOETHALS, University of Antwerp, Belgium and Monash University, Australia

Modern recommender systems are often modelled under the sequential decision-making paradigm, where the system *decides* which recommendations to show in order to maximise some notion of either imminent or long-term reward. Such methods often require an explicit model of the reward a certain context-action pair will yield – for example, the probability of a click on a recommendation. This common machine learning task is highly non-trivial, as the data-generating process for contexts and actions can be skewed by the recommender system itself. Indeed, when the deployed recommendation policy at data collection time does not pick its actions uniformly-at-random, this leads to a selection bias that can impede effective reward modelling. This in turn makes off-policy learning – the typical setup in industry – particularly challenging. Existing approaches for value-based learning break down in such environments.

In this work, we propose and validate a general *pessimistic* reward modelling approach for off-policy learning in recommendation. Bayesian uncertainty estimates allow us to express scepticism about our own reward model, which can in turn be used to generate a conservative decision rule. We show how it alleviates a well-known decision making phenomenon known as the Optimiser's Curse, and draw parallels with existing work on pessimistic policy learning. Leveraging the available closed-form expressions for both the posterior mean and variance when a ridge regressor models the reward, we show how to apply pessimism effectively and efficiently to an off-policy recommendation use-case. Empirical observations in a wide range of simulated environments show that pessimistic decision-making leads to a significant and robust increase in recommendation performance. The merits of our approach are most outspoken in realistic settings with limited logging randomisation, limited training samples, and larger action spaces. We discuss the impact of our contributions in the context of related applications like computational advertising, and present a scope for future research based on hybrid off-/on-policy bandit learning methods for recommendation.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Computing methodologies** → *Reinforcement learning*.

Additional Key Words and Phrases: Contextual Bandits; Offline Reinforcement Learning; Probabilistic Models

ACM Reference Format:
Olivier Jeunen and Bart Goethals. 2022. Pessimistic Decision-Making for Recommender Systems. *ACM Trans. Recomm. Syst.* 1, 1, Article 1 (October 2022), 27 pages. https://doi.org/10.1145/3568029

1 INTRODUCTION AND MOTIVATION

Recommendation systems are tools that are often used to stimulate some form of engagement between users and items on online platforms. In the early days, fuelled by the popularity of the MovieLens datasets and the Netflix Prize [6, 32], the problem was typically framed as that of *rating* prediction. When presented with a dataset consisting of explicit ratings given from users to items, the recommendation system would then generate a model that predicts the rating a certain user would give to a yet unseen item. Items with higher predicted ratings would then be assumed

*A preliminary version of this article appeared as "Pessimistic Reward Models for Off-Policy Learning in Recommendation" in the proceedings of the Fifteenth ACM Conference on Recommender Systems (RecSys), 2021 [42], when the first author was at the University of Antwerp.

Authors' addresses: Olivier Jeunen, jeunen@amazon.com, Amazon, Edinburgh, United Kingdom; Bart Goethals, bart.goethals@uantwerpen.be, University of Antwerp, Antwerp, Belgium and Monash University, Melbourne, Australia.

to make up better recommendations, and these assumptions allowed the field to thrive. As the web evolved, the availability of *implicit* feedback quickly outgrew that of its *explicit* counterpart, and recommendation models evolved with it. The focus on *rating* prediction moved towards *item* prediction, where the quality of a recommendation list is determined via ranking metrics borrowed from the neighbouring field of Information Retrieval [100, 108]. This again allowed the field to thrive for several years.

More recently, several real-world recommendation systems have moved from the *prediction* to the *decision* paradigm, explicitly acknowledging that the recommendations we choose to show have an influence on the state of the world.[1] Moreover, the goals of live systems can diverge, and need not be focused on a single metric [76]. Indeed, many modern web services deploy machine-learnt models on their websites to help steer traffic towards certain items. Retail websites try to predict which of their recommendations might lead to a sale, music streaming platforms suggest songs in your queue to optimise engagement metrics, search engines will often rank items in decreasing estimated probability of receiving a click, et cetera. In these and in many more use-cases, the system consists of a model that estimates the consequences of its actions, and weighs them before making a decision. For example, we might model the probability of receiving a click when showing a recommendation, and decide to show recommendations that maximise the estimated expected number of clicks.

These models are generally part of a (1) collect data, (2) train model, (3) deploy model loop, where models are iteratively retrained and earlier versions influence the training data that is used for future iterations. This correlation between the deployed model and the collected training data can impede effective learning if we are unable to somehow correct for the bias it creates. Recent work has shown how such "algorithmic confounding" leads to feedback loops when left untreated, which can be detrimental to the users, the platforms, and the models themselves [9, 70]. Traditional recommendation research that falls under the *prediction* paradigm bypasses these feedback loops by (often implicitly) assuming *organic* user-item interaction data that was collected independently of any existing recommendation process. Nevertheless, this assumption is often questionable, and can have a significant impact on evaluation results when violated [38]. In deployed systems, feedback loops cannot be dismissed. In this work, we wish to learn directly from the logs of the deployed recommender system, casting the recommendation task in a bandit learning framework [11, 109]. Here, the feedback loop is a *feature*, as it allows us to directly optimise online reward metrics in an offline manner [40]. Notwithstanding this, the biased nature of data collected by deployed recommendation policies should be taken into account appropriately.

Learning from biased data is not a novel problem, and many *unbiased* learning procedures have proven to be effective in counteracting *position*, *presentation*, *trust* and *selection* bias [2, 3, 11, 52, 83]. These methods typically make use of importance sampling or Inverse Propensity Score (IPS) weighting, in order to obtain an unbiased estimate of the counterfactual value-of-interest [86]. They aim to answer questions of the form: "*What click-through-rate would this new policy have obtained, if it were deployed instead of the old policy*"? The policy that maximises the answer to this question is the policy we want to deploy. Answering this question effectively and efficiently, however, is not an easy feat.

IPS is the cornerstone of counterfactual reasoning [8], but by no means a silver bullet. It is plagued by variance issues that are exacerbated at scale, often making it hard to deploy these systems reliably in the real world [29]. Furthermore, the randomisation requirements for IPS to remain unbiased are often unrealistic or simply unattainable. Recent work explores the effectiveness of counterfactual

[1] Note that this line of research is not inherently new [95], but has recently seen increased interest from researchers in industry [11, 12, 74, 78, 109, 116].

models in cases where IPS assumptions in the training data are violated, highlighting an interesting area for future research and a commonly-encountered yet understudied problem [47, 89].

An alternative family of approaches are so-called "value-based" models. These methods rely on an explicit model of the reward conditioned on a context-action pair – for example, the probability of a user clicking on a given recommendation when it is shown [33, 75]. When prompted, the model then simply takes the action that maximises the probability of a positive reward, given the presented context and the learnt model. Aside from the typical problems of model misspecification in supervised learning [71], another issue with value-based methods is that learning an accurate model of the reward is not straightforward when the collected training data is heavily influenced by the model that was deployed in production at the time. Methods that use IPS to re-weight the data as if it were unbiased exist [97], but their performance when deployed as recommendation policies is often disappointing in comparison with policy-based methods or even reward models that do not re-weight the data [47, 80]. Furthermore, the logging policy is not always known before-hand, several logging policies might be at play concurrently [1, 53], and even when we do manage to obtain unbiased value estimates we should expect the true obtained reward from acting on them to be disappointing. Indeed, only considering the action with the highest estimated reward can be a flawed decision procedure in and of itself – a phenomenon known as "the Optimiser's curse" [99].

Contributions. In this paper, we focus on improving the recommendation performance of policies that rely on value-based models of expected reward. We propose and validate a general pessimistic reward modelling framework, with a focus on the task of off-policy learning in recommendation. Bayesian uncertainty estimates allow us to express scepticism about our own reward model, which can then in turn be used to generate conservative decision rules based on the resulting reward predictions – instead of the usual ones based on Maximum Likelihood (MLE) or Maximum A Posteriori (MAP) estimates. We show how closed-form expressions for both the posterior mean and variance can be leveraged to express pessimism when a ridge regressor models the reward, and how to apply them effectively and efficiently to an off-policy recommendation use-case. Our approach is agnostic to the logging policy, and does not require (a model of) propensity scores to quantify selection bias. As a result, we are not bound to the strict assumptions that make IPS work, and abide by statistical conjectures such as the likelihood principle [7]. Pessimistic decision-making not only significantly increases the reward obtained by the learnt policy's recommendations, but we additionally show how our proposed framework lifts the Optimiser's Curse. By essentially accepting an increase in model bias over the *full* action space for reduced variance in the *topmost* actions, we significantly improve the recommender's ability to forecast its own performance. That is, we limit "post-decision disappointment", defined as the difference between the estimated expected reward and the true obtained reward. We discuss the important consequences this has for offline evaluation and downstream applications such as computational advertising.

Bias-variance trade-offs and pessimism are not new to the machine learning field. Especially in general Reinforcement Learning (RL) research, estimators are often replaced by conservative lower-bounds, albeit for slightly different reasons [92, 93, 112].[2] In the case of policy-based methods particularly, IPS' well-known variance issues have caused a plethora of extensions to sprout in recent years. Some of those explicitly recognise their pessimistic nature, others do not. We provide an overview of those relevant to the recommendation use-case, draw parallels with our proposed value-based pessimism, and highlight connections and differences.

One further connection that cannot be overlooked, is how our proposed approach for off-policy learning seemingly goes against the "optimism in the face of uncertainty" adage adopted by decades

[2]These methods are typically proposed as a way to restrict the step size for an *on*-policy algorithm to remedy learning instabilities.

of work on on-policy bandits [4, 10, 59]. We show how the insights presented in this work are complementary to theirs, and propose a research agenda for future work connecting off- and on-policy learning with hybrid approaches.

Evaluation. The empirical performance of counterfactual learning methods is often reported with a supervised-to-bandit conversion on existing multi-class or multi-label classification datasets [51, 69, 102]. As publicly available datasets with propensity information are scarce, this inhibits robust and reproducible evaluation of such methods on off-policy *recommendation* tasks. In line with recent work [41, 45, 47, 91], we adopt the RecoGym simulation environment in our experiments to yield reproducible results that are aligned with the specifics of real-world recommendation scenarios, such as stochastic rewards, limited randomisation and small effect sizes [87]. An added advantage of adopting such a simulation framework is the freedom gained to change environmental parameters and better understand how these changes affect the trade-offs between different methods. This allows us to present insights in our proposed method that offline datasets would not be able to uncover. Empirical observations for a wide range of configurations show that our proposed approach of pessimistic decision-making leads to a significant and robust increase in recommendation performance. The merits of our method are most outspoken in realistic settings where the amount of randomisation in the logging policy is limited, training sample sizes are small, and action spaces are large. Indeed, these are exactly the cases where selection bias will be strong, and overestimation is likely to occur. All source code to reproduce the reported results is available at github.com/olivierjeunen/pessimism-recsys-2021.

To summarise, the main contributions we present in this work are:

(1) We propose the use of explicit pessimism in reward models for off-policy recommendation use-cases.
(2) We introduce the decision-making phenomenon known as the Optimiser's Curse in the context of recommendation, and show how naïve reward models suffer from it. In contrast, principled pessimism lifts the curse.
(3) We show how to leverage closed-form estimates for the posterior mean and variance of a ridge regressor to express pessimism, and how to apply this effectively and efficiently to an off-policy recommendation use-case.
(4) Empirical observations from reproducible simulation experiments highlight that explicit pessimism significantly and robustly improves online recommendation performance, compared to ML or MAP-based decision-making.
(5) We draw parallels with existing work in general on-policy bandits as well as pessimistic policy-learning, and present a scope for future research connecting these problem settings and research areas.

Extensions to [42]. The introduction and motivation of our work have been extended to explicitly highlight connections with related research areas, both within and outside of the Recommender Systems field. The background and related work section more clearly depicts what the data-generating process looks like in our use-case – additionally providing more detail on doubly robust methods and their relevance. We have included a stream of related work on using Markov Decision Processes (model-based reinforcement learning) for recommendation, motivating how their use-case differs from ours. The core methodological section of the paper provides more details on reward estimation and decision-making in policy learning, allowing us to highlight some examples of pessimism in these cases, and draw connections with existing well-known reinforcement learning methods such as TRPO and PPO [92, 93]. Further detail on applications in related areas such as computational advertising has been included, where the impact of pessimistic decision-making will

be especially tangible. Deeper connections between optimism in on-policy bandits and our proposed pessimism in off-policy bandits are discussed. We highlight their differences and commonalities, and propose a scope for future work on hybrid approaches. More detail on the experimental setup is included, and we further motivate the simulation framework we have adopted to empirically validate our proposed method. We have further extended the experiments with respect to research questions 1–3 over action spaces with varying sizes, and significantly expanded the discussion of the results and their impact. The conclusion has been extended and rewritten to include a scope for future research, and the abstract reflects the new contents of the paper.

2 BACKGROUND AND RELATED WORK

We are interested in modelling recommendation systems following the "Batch Learning from Bandit Feedback" (BLBF) paradigm [104]: a general machine learning setting that properly characterises the off-policy recommendation use-case as it widely occurs in practice. A recommender system is modelled as a stochastic policy π that samples its recommendations from a probability distribution over actions A conditioned on contexts C: $\mathsf{P}(A|C,\pi)$, often denoted $\pi(A|C)$.[3] Note that π is modelled to be stochastic for generality, but that deterministic systems are implied when $\mathsf{P}(A|C,\pi)$ is a degenerate distribution. Contexts are drawn from some unknown marginal distribution $\mathsf{P}(C)$ and can represent a variety of information about the user visiting the system, such as their consumption history, the time of day and the device they are using. When talking about the feature vector for a specific context, we denote it as $\boldsymbol{c}$. Analogously, feature vectors for specific actions are represented as $\boldsymbol{a}$, which can include discrete identifiers as well as information about interactions with the item or its content. The sets of all possible contexts and actions are $\mathcal{C}$ and $\mathcal{A}$, respectively. The combined feature representation of a context-action pair is $\boldsymbol{x} \coloneqq \Phi(\boldsymbol{c},\boldsymbol{a})$, where Φ is a function that maps context- and action-features to a joint space. Note that this step – including interaction terms between contexts and actions – is necessary to allow for linear models to learn personalised treatments. Φ can be anything from a simple Kronecker product between one-hot-encoded contexts and actions [80], to a specialised neural network architecture that learns a shared embedding for multi-task learning [67, 106, 115]. In the off-policy or counterfactual setting, we have access to a dataset consisting of logged context-action pairs and their associated rewards: $\mathcal{D} \coloneqq \{(c,a,r)_t\}_{t=1}^{t_{\max}}$, where $c \sim \mathsf{P}(C)$, $a \sim \pi_0(a|c)$ and $r \sim \mathsf{P}(R|C,A)$. Here, r represents the immediate reward that the system obtained from recommending a when presented with context c at a given time t. In the general case this reward can be binary (e.g. clicks), real-valued (e.g. dwell time or profit), or higher-dimensional to support multiple objectives (e.g. fairness and relevance) [77, 78]. The policy that was deployed at data collection time is called the logging policy (π_0). This type of setting is called "bandit feedback", as we only observe the reward of the actions chosen by the contextual bandit π_0. It is referred to as being "off-policy", as we have no control over π_0 or its exploration strategy. We place this paradigm at the focal point of our work, as it is the most closely aligned with the recommendation use-case that practitioners typically face in industry. Indeed, truly on-policy methods are often prohibitively costly to implement due to the need for frequent real-time updates [11], and continuous experimentation practices lead to multiple logging policies that interact and give rise to selection bias that needs to be addressed [1, 24, 53]. We discuss deeper connections between typical on- and off-policy approaches, as well as a need for hybrid methods in Section 3.5. Figure 1 visualises our interactive data-generating process on the left-hand side, with the learning objective on the right-hand side.

[3]This work focuses on the case where an *action* corresponds to a *single* recommendation. Tractable decompositions that alleviate the combinatorial explosion of the action space when dealing with top-N recommendations can open up paths forward to more general settings [36].

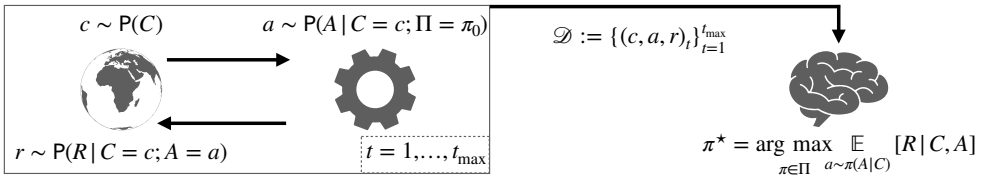


Fig. 1. This schematic shows the typical setup of machine learning systems deployed in the real world. The data-generating process is shown in the box on the left, feeding data into a machine learning system that optimises a policy that maximises the expected reward it will obtain – shown on the right. As we do not observe the outcomes for actions not taken by the logging policy, the collected data $\mathcal{D}$ suffers from selection bias. Because the newly learnt policy is ultimately responsible for new data collection, there is a strong feedback loop that needs to be taken into account.

Learning to recommend from organic user-item interactions. Most traditional approaches to recommendation do not make use of this type of experimental data tying recommendations to observed outcomes. Instead, they typically adopt observational datasets consisting of "organic" interactions between users and items, such as product views on retail websites. By framing the recommendation task as next-item prediction in such a setting, the goal of these systems is no longer that of learning optimal interventions. Maybe unsurprisingly, offline evaluation results in such environments are notoriously uncorrelated with online success metrics based on shown recommendations, making it harder to discern *true* progress with regard to online gains [17, 28, 39, 88]. Nevertheless, it is a very active research area that yields many interesting publications and results every year. Recent trends are geared towards the use of Bayesian techniques that explicitly model uncertainty [23, 61, 66, 96], and linear item-based models that achieve state-of-the-art performance whilst being highly efficient to compute [14, 16, 48, 49, 81, 101].

Off-policy learning from bandit feedback. The bandit feedback setup described above finds its roots in the field of offline reinforcement learning (RL), with the additional simplifying assumption that past actions do not influence future states (more formally, the underlying Markov Decision Process consists of a single time-step) [58]. This type of learning setup is not specific to the recommendation task, and many learning methods are evaluated on simulated bandit feedback scenarios using general purpose multi-class or multi-label datasets. Approaches for off-policy learning optimise a parametric policy for some counterfactual estimate of the reward it would have obtained, if deployed.

The go-to technique that enables this type of counterfactual reasoning is importance sampling [8, 86]. Eq. 1 shows how it obtains an empirical estimate for the value of a policy π, using data $\mathcal{D}$, and a model of the logging policy $\widehat{\pi_0}$ (which can be exact and known, or learnt from data). Many learning algorithms in this family aim to mitigate the increased variance that is a consequence of the IPS weights. Capping the probability ratio to a fixed value [37], self-normalising the weights [51, 105], imposing variance regularisation [72, 104], imitation learning [69] or distributional robustness [26, 98] on the learnt policy are commonly used tools to trade off the unbiasedness of IPS for improved variance properties in finite sample scenarios. Many of these techniques can be interpreted as a form of principled *pessimism*, where we would rather be conservative with the IPS weights than over-estimate the value of an action to a policy.

$$\widehat{V}_{\mathrm{IPS}}(\pi, \mathcal{D}) = \sum_{(c,a,r)\in\mathcal{D}} r \cdot \frac{\pi(a|c)}{\widehat{\pi}_0(a|c)} \tag{1}$$

A conceptually simpler family of approaches are value-based methods, often referred to as Q-learning in the RL community, or the "Direct Method" (DM) in the bandit literature. Eq. 2 shows how DM obtains an empirical estimate of policy π's value w.r.t. a dataset of logged bandit feedback $\mathcal{D}$:

$$\widehat{V}_{\mathrm{DM}}(\pi, \mathcal{D}) = \sum_{(c,a,r)\in\mathcal{D}} \sum_{a'\in\mathcal{A}} \pi(a'|c) \cdot \widehat{r}(a', c). \tag{2}$$

Value-based counterfactual estimators do not rely on a model of the logging policy, but rather learn a model for the context-specific immediate reward of an action: $\widehat{r}(a, c) \approx \mathbb{E}[R|C = c, A = a]$. In practice, the available bandit feedback $\mathcal{D}$ is split into disjoint training sets for the optimisation of the reward model and the resulting policy respectively. Nevertheless, it is easy to see that the optimal policy π^*_{DM} with respect to a given reward model places all its probability mass on the action with the highest estimated reward:

$$\pi^*_{\mathrm{DM}}(a|c) = \begin{cases} 1 & \text{if } a = \arg\max_{a'\in\mathcal{A}} \widehat{r}(a', c), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

As a consequence, we can directly obtain a decision rule from the reward estimates and train the reward estimator on all available data [47]. Note that this simple decision rule leads to a deterministic policy, but stochastic value-based policies can be obtained by explicitly optimising Eq. 2 with an entropy regularisation term [31]. Value-based methods as laid out above are typically biased, but exhibit more favourable variance properties than IPS-based models. While policy-based methods for learning from bandit feedback need (a model of) the logging propensities [11, 109], this is not a constraint for the value-based family. When multiple logging policies are at play (e.g. during an A/B-test), this complicates the use of standard importance sampling techniques even further [1, 24, 53].

A unifying family of Doubly Robust (DR) methods aims to marry these two types of approaches in an attempt to get the best of both worlds [19], as shown in Eq. 4. DR essentially augments DM with an IPS-term that is weighted by the error of the value-based model. When either the propensities $\widehat{\pi}_0$ or the reward model $\widehat{r}$ are correct, this estimator is provably unbiased. However, as it requires a separate model for the policy and the reward, we leave an explicit analysis of pessimism in doubly robust estimators for future work.

$$\widehat{V}_{\mathrm{DR}}(\pi, \mathcal{D}) = \sum_{(c,a,r)\in\mathcal{D}} \left((r - \widehat{r}(a, c)) \cdot \frac{\pi(a|c)}{\widehat{\pi}_0(a|c)} + \sum_{a'\in\mathcal{A}} \pi(a'|c) \cdot \widehat{r}(a', c) \right) \tag{4}$$

Recent advances in doubly robust learning typically optimise the trade-off between DM and IPS [103], optimise the reward model to minimise the overall variance of the estimator [25], or transform the IPS weights to minimise bounds on the expected error of the estimate [102]. Nevertheless, the performance of the reward model remains paramount for doubly robust approaches to attain competitive performance – and it is not uncommon for DR to be outperformed by either DM or IPS [41].

Reinforcement learning for recommendation. The methods introduced above are bandit-based: they aim to learn which actions to take based on the outcomes of previously logged actions. They focus on *immediate* rewards, and include no notion of *planning* to improve future rewards nor notions of long-term value. Indeed, because bandit-based approaches do not explicitly model *state* transitions, they implicitly assume that current actions do not influence the distribution of future contexts or rewards. This assumption significantly simplifies modelling and learning, but can be limiting in more complex scenarios where the optimisation of rewards over a sequence of actions is required. Several works use Markov Decision Processes (MDPs) to incorporate such notions of long-term rewards into recommendation use-cases [36, 82, 95]. We do not explicitly model long-term value in this work, and focus on the bandit setting as opposed to a full RL setting – but note that ideas of pessimism have recently been adopted in general RL use-cases [54, 56, 63, 114]. Exploring whether our presented insights extend to RL recommendation settings is an open and interesting area for future research.

Off-policy learning for recommendation. Methods that apply ideas from the bandit and RL literature to recommendation problems have seen increased research interest in recent years – typically in off-policy settings. Chen et al. extend a policy gradient-based method with a top-K IPS estimator and show significant gains from exploiting bandit feedback in online experiments [11]. In the top-1 use-case we consider with the additional bandit assumption, their method yields a policy that is analogous to one optimised for $\widehat{V}_{\text{IPS}}$ (Equation 1). This work has been extended to deal with two-stage recommender systems pipelines that are typically adopted to deal with large action spaces [68]. Xin et al. adopt a Q-learning perspective to deal with sequential recommendation tasks, exploiting both self-supervised (*organic*) and reinforcement (*bandit*) signals [113]. Analogously, Sakhi et al. propose a probabilistic latent model that combines organic and bandit signals in a Bayesian value-based manner [91]. The work of Jeunen et al. studies the performance of both value- and policy-based approaches when the organic data is only used to describe the context, proposing a joint policy-value approach that outperforms stand-alone methods without the need for an external reward model [47]. Their experimental set-up is the closest to the one we tackle in this work.

On-policy learning for recommendation. Off-policy methods learn from data that was collected under a different policy. In contrast, on-policy methods learn from data that they themselves collect. In such cases, the well-known exploration-exploitation trade-off becomes important, as the policy needs to balance the immediate reward with the informational value of an action [60, 74]. Successful methods use variants of Thompson sampling [10, 20, 73] or confidence bounds [59]; recent work benchmarks a number of different exploration approaches to predict clicks on advertisements when the reward model is parameterised as a neural network [30]. Although the use-case we tackle in this work does not include any interactive component, we draw upon existing work in learning from on-policy bandit feedback to obtain improved, uncertainty-aware decision strategies in the off-policy setting.

Uncertainty estimation. Both Thompson sampling and confidence-bound-based methods make use of a posterior distribution for the reward estimates, instead of the usual point estimate that is obtained from uncertainty-agnostic models. Principled Bayesian methods can be used to obtain closed-form expressions for exact or approximate posteriors, but they are often restricted to specific model classes [10, 59]. The Bootstrap principle [21], its extensions [85] (originally proposed in the context of Q-learning), and Monte Carlo Dropout [27] can provide practical uncertainty estimates for general neural network models. The work of Guo et al. proposes a hybrid Bootstrap-Dropout

approach, and validates the effectiveness of the obtained uncertainty estimates in an on-policy recommendation scenario [30]. Finally, other recent work shows promising results in inferring model uncertainty from neuron activation strength [15]. All these uncertainty estimation methods are complementary to the framework we propose in this paper, and can be used to explicitly express either *optimism* in on-policy settings, or our proposed *pessimism* for off-policy learning.

3 METHODOLOGY AND CONTRIBUTIONS

3.1 The Optimiser's Curse in Recommendation

In what follows, we introduce the Optimiser's Curse [99] in the context of off-policy learning in recommendation scenarios. For illustratory purposes, we assume an immediate binary reward (e.g. a click) that follows a Bernoulli distribution with parameter p that is conditioned on the relevance of the given context-action pair. Nevertheless, the Optimiser's Curse is a general phenomenon that is by no means bound to these assumptions.

Suppose we have an action space $\mathcal{A}$ and for simplicity, but without loss of generality, assume that the probability of a positive reward is independent of the context. Now, every action $a_i \in \mathcal{A}$ has a *true* probability of leading to a click: $\mathrm{P}(R = 1|A = a_i) = p_i^*$. The goal of a reward model is to estimate these true Bernoulli-parameters p_i^*, yielding the estimated parameters $\widehat{r}(a_i) = \widehat{p}_i$. Widely used estimation methods include Maximum Likelihood (MLE) and Maximum A Posteriori (MAP) estimation. We aim to learn such a model based on a previously acquired log of training data, and assume that our obtained value estimates are conditionally unbiased in that $\forall i \in \{1, \ldots, |\mathcal{A}|\} : \mathbb{E}[\widehat{p}_i | p_1^*, \ldots, p_{|\mathcal{A}|}^*] = p_i^*$. Note that this assumption is already quite idealistic for many real world applications, and that it cannot be checked when we do not know the true parameters p^*. In practice, we can minimise the bias between the reward model and the empirical reward in the training sample.[4] Nevertheless, even in such an idealised setting, problems arise.

Once we have a reward model, we are ready to start showing recommendations to users. Analogous to Equation 3 we take the action with the highest estimated reward or Bernoulli-parameter, indexed by i^*:

$$a_{i^*} = \arg\max_{a_i \in \mathcal{A}} \widehat{r}(a_i). \tag{5}$$

After showing this recommendation to a user, we get to observe a sample from the true reward distribution: $r_{i^*} \sim \text{Bernoulli}(p_{i^*}^*)$. Now, the difference between the observed and estimated rewards $(r_{i^*} - \widehat{p}_{i^*})$ can be seen as the *post-decision surprise* we get from acting on the model $\widehat{r}$. Repeating this process and averaging the observed post-decision surprise yields the average expected surprise: $\mathbb{E}[p_{i^*}^* - \widehat{p}_{i^*}]$. The Optimiser's Curse states that, even though the reward estimates are conditionally unbiased *over all actions* ($\mathbb{E}_{i \in \mathcal{A}}[p_i^* - \widehat{p}_i] = 0$), this process leads to a *negative* expected surprise: $\mathbb{E}[p_{i^*}^* - \widehat{p}_{i^*}] \leq 0$, meaning that we incur *less* reward than predicted. This *disappointment* on average is not merely a result of the model itself (as it is unbiased), but rather a consequence of the decision making process that only considers the action with the highest estimated value $\widehat{p}_{i^*}$, leaving us especially vulnerable to actions with over-estimated rewards.

Smith and Winkler provide an excellent overview of this phenomenon, showing how it can be mitigated by adopting Bayesian methods with well-chosen priors [99]. They prove that, in the settings they consider, choosing actions based on MAP estimates alleviates any post-decision surprise *when these posteriors are unbiased*. This elegant theoretical finding is of limited practical use in our use-case. Indeed, we have no way to guarantee that the reward estimates we end up with are unbiased with respect to the true reward distribution parameters p^*. We can only check

[4] This type of "calibration" of the reward model with respect to the *empirical* reward distribution is often a requirement in computational advertising [33, 75], as a downstream bidding strategy then depends on the reward model [44].

unbiasedness with respect to the empirically observed reward, which can be highly skewed due to the logging policy. Additionally, underfitting and model misspecification make this assumption of converging to the true parameters sound especially utopian [71]. To make matters worse, the training data $\mathcal{D}$ that is used to obtain the reward model $\widehat{r}$ is also highly dependent on this logging policy π_0, impeding effective reward modelling even before we take part in an ill-suited decision making process. Indeed, standard Empirical Risk Minimisation (ERM) focuses its efforts on context-action regions that are well-explored in the training data. This leaves us vulnerable when naively handling the resulting reward estimator $\widehat{r}$, because a single erroneously optimistic reward estimate can disturb the recommendation policy and decimate performance. The probability of this happening grows with the size of the action space and the level of "determinism" in the logging policy (more formally, decreasing entropy). Using (estimated) propensity scores to redistribute the errors in the model fit does not guarantee performance improvements in such cases [47, 80, 97].

3.2 Heteroscedasticity in Reward Estimates

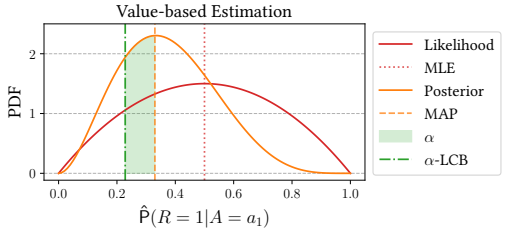
Logging policies typically aren't solely optimised for data collection. The currently deployed system will take actions with a higher estimated reward more often than it will take those with lower estimates. This skews the training data for future model iterations, which in turn leads to heteroscedasticity in the reward estimates. The most common frequentist approaches to reward modelling based on MLE – be it parameterised by simple linear models or deep neural networks – do not provide information about an estimated posterior distribution out-of-the-box. As a result, detecting pathological cases where gross over-estimation occurs is highly non-trivial. Well-chosen priors and the resulting MAP estimates can partially alleviate this, but are hard to validate and yield no guarantees.

In what follows, we will illustrate problems stemming from naïve value-based estimation in a typical recommendation setup, and highlight alternative decision-making strategies. We present a parallel line of thought for existing approaches to policy-based estimation. Although these are not the focal point of this work, the connections deserve mentioning.

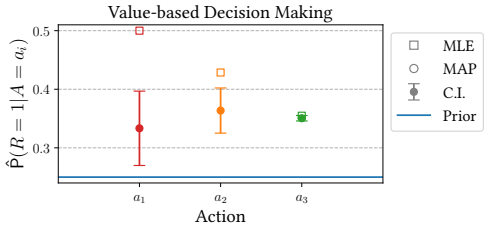
As a simple example, consider a Beta-Bernoulli model with three actions [79, §7.2.1]. Rewards for action a_i are drawn as $r_i \sim \text{Bernoulli}(p_i)$, with $p_i \sim \text{Beta}(\alpha_0 + \alpha_i, \beta_0 + \beta_i)$. In this setup, α_i and β_i can be seen as the number of observed clicks and non-clicks for action a_i. For illustratory purposes, assume $\alpha_1 = \beta_1 = 1, \alpha_2 = 3, \beta_2 = 4, \alpha_3 = 33, \beta_3 = 60$. We assume a prior probability of receiving a click for the posterior predictive of 25%, so we set $\alpha_0 = 1, \beta_0 = 3$. Now, we can compute the ML and MAP estimates for these actions, and deduct the optimal policies.

Reward Estimation. Figure 2a shows the resulting likelihood and posterior distributions for action a_1. The MLE postulates that the true probability of receiving a click for action a_1 is 50%, albeit based on just two samples. Although the likelihood function clearly illustrates large variance and uncertainty – this collapses when the MLE is considered, and over-estimation is likely to occur. The MAP estimate partially alleviates this by relying on well-chosen priors, but we can see that large uncertainty remains. The Lower-Confidence-Bound is introduced in Section 3.3.

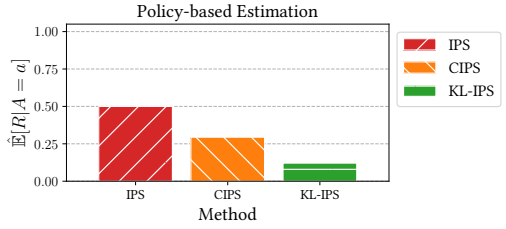
We now consider policy-based estimators instead, which have access to the true logging propensities: $\widehat{\pi_0} \equiv \pi_0$. These propensities are calculated based on the empirical frequencies implied by the α_i, β_i parameters: $\pi_0(a_i) = \frac{\alpha_i + \beta_i}{\sum_{j \in \mathcal{A}} \alpha_j + \beta_j}$. We compute reward estimates using three estimators that are well-known and used in the literature: IPS (Eq. 1), Capped IPS (CIPS, $m = 30$, [29, 37]), and IPS with a penalty on the Kullback-Leibler (KL) divergence between the logging and learnt policies $\text{D}_{\text{KL}}(\pi_0 || \pi)$ (KL-IPS, $\lambda = 10$, [26, 92]). The expected reward for action a_1 under these estimators is shown in Figure 2c. Interestingly – traditional IPS coincides with the MLE. CIPS cuts off the importance weight at a point determined by the hyper-parameter m, which can be interpreted as



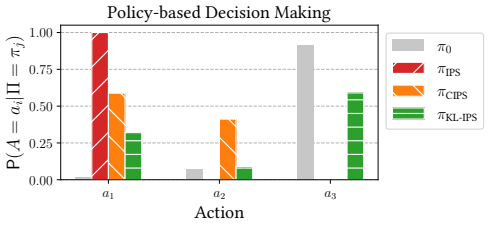
(a) Value-based estimates for action a_1's reward distribution.



(b) Value-based decision-making alternatives.



(c) Policy-based estimates for action a_1's reward distribution.



(d) Policy-based decision-making alternatives.

Fig. 2. Illustration of value- and policy-based estimation and decision-making strategies for the toy example in Section 3.2.

an explicit form of pessimism. KL-IPS penalises policies that diverge from the logging policy, which induces an implicit form of pessimism.

Decision-Making. Figure 2b visualises the actions that will be taken when following competing strategies: the MLE prefers action a_1, whereas the MAP estimate prefers a_2. For well-explored context-action pairs, we see that the variance in the posterior predictive of the reward model is reduced, leading to a tighter credible interval. For under-explored context-action pairs, however, the error and variance grow to be quite substantial. The de facto decision making process of taking the action with the highest reward estimate, is then even more vulnerable to post-decision disappointment due to this type of heteroscedasticity, and thus prone to provide over-estimations of the *true* expected reward. There only needs to be a single action with a badly calibrated reward estimate for this situation to occur (due to the arg max in Eq. 5), and the probability of encountering such lesser explored actions will typically grow with the size of the action space (for realistic logging policies). If our posterior means are unbiased conditional on the value estimates, the results of Smith and Winkler show that the expected post-decision surprise will be zero. This is, however, an unreasonable assumption in complex real-world environments where correct model specification is often impossible, and the range of conjugate priors might not be expressive enough to allow for this to happen. Furthermore, as we often deal with small effect sizes (i.e. $|p_i^* - p_j^*| < \epsilon$), even slight errors in the reward estimates can have significant impact on the actions taken by the resulting policy. If there is *some* probability of our recommendation policy taking a suboptimal action, the inequality bounding expected surprise even becomes strict: $\mathbb{E}[p_{i^*}^* - \widehat{p}_{i^*}] < 0$ [99]. The Optimiser's Curse can lead to a significant disparity between what we expect will happen based on the reward model, and what will actually happen when we act according to its estimates. Nevertheless, we can significantly improve recommendation performance by treating our reward estimates with a

healthy dose of principled scepticism. Figure 2d shows the optimal policies obtained by maximising the three competing policy-based objectives. As is expected, IPS solely focuses on the action with the highest empirical reward. CIPS presents us with a lever than can be used to stipulate how much trust we put in the data – and KL-IPS provides a way to limit the step-size in the policy space. Empirical studies have shown significant performance gains from adopting these pessimistic techniques in the policy-based case. In what follows, we explore their validity and use for value-based recommendation.

3.3 Pessimistic Decision-Making

Small effect sizes, bias and heteroscedasticity in the reward estimates are the main reasons why reward models typically perform more poorly than expected. The conceptually simplest way of mitigating this unevenly distributed variance is to mitigate selection bias altogether by adopting a uniformly random logging policy. However, showing recommendations to users independently of the estimated relevance of the action might not be in the best interest of the platform or the users, at least not from a business perspective. In what follows, we explore our decision-making options, borrowing ideas from the related on-policy bandit literature.

Traditional models generate point estimate predictions, which we can reasonably assume to be contained by the posterior shown in Figure 2b. Possible actions are then ranked by $\widehat{r}(a_i)$, as these approaches cannot quantify differences between recommendations a_1, a_2 and a_3 in any other way. This is problematic due to all the reasons laid out above.

In an on-policy world, typical approaches make use of uncertainty estimates to balance the expected reward with the informational value of an action. Methods based on Thompson Sampling (TS) repeatedly sample reward estimates from an approximate posterior [10], and optimistic extensions to this paradigm are known to further improve performance [73]. Upper Confidence Bound (UCB) methods also follow the "optimism in the face of uncertainty" adage, explicitly taking the action with the highest posterior quantile instead of the MAP estimate [59]. For our example in Figure 2b, this would lead to a ranking of $a_2 > a_1 > a_3$. On-policy approaches are optimistic because it provably pays off; they get to observe the outcome of the chosen action and use this new data point to adjust reward estimates. Intuitively, this makes that reward estimates will never be overly optimistic for long, as the posterior will tend to converge to the lower, true p_i^* as more data comes in. This "self-correcting" property then naturally bounds metrics like regret in an online setting, and makes TS and UCB provably efficient. In an off-policy setting, we do not have the luxury to instantly learn from the outcome of our actions. All we have is a finite log of context-action-reward triplets, collected by a different policy, with which we will have to make do. It is clear that optimism will not help us in such cases, as we cannot reap the fruit of informational value that comes with it.

Optimism is not the way to go – but the naïve decision-making procedure that purely focuses on the maximal reward estimates, is still likely to yield exactly those that were over-estimated. Even without explicitly encoding optimism, this still leads to inflated expectations and subpar performance. We can offset this unwarranted over-estimation by treating our model predictions pessimistically. This is exactly what Smith and Winkler suggest when saying: "model the uncertainty in the value estimates explicitly and use Bayesian methods to interpret these value estimates" [99]. With a suitable prior distribution and unbiased posterior means, their suggested approach effectively encourages principled conservatism which provably limits disappointment. We have argued how their proposed solution breaks down in complex environments, and additionally note that advanced prior distributions tend to complicate the reward modelling procedure and can hurt scalability by surrendering conjugacy. Ranking the actions in Figure 2b according to their posterior means leads to a ranking of: $a_2 > a_3 > a_1$. Because of the vastly reduced variance from a_2 to a_3 and the small difference in their posterior means, we argue that a_3 should be the safe choice. One might argue that

the MAP choosing a_2 is merely the result of an inappropriate prior, but small effect sizes combined with heteroscedasticity make this highly non-trivial to tune and validate properly. Optimising the prior as a hyper-parameter to achieve exactly zero post-decision disappointment is theoretically possible in controlled environments when Bayesian methods are used, but this is highly complex and intractable in real-world environments where we have approximate uncertainty estimates for general model classes like neural networks. Furthermore, as a simple bias term directly influences post-decision surprise without altering the actions that are being taken, it is clear that maximising the online performance of the deployed recommendation policy should still be the overarching objective compared to blindly limiting disappointment.

Instead of pursuing unbiasedness through appropriate priors, we propose to be even more sceptical of our own reward model, and to make decisions based on the maximal lower quantile of the posterior distribution. By adopting a Lower Confidence Bound (LCB)-driven decision-making strategy, we effectively penalise actions with high variance and pick the action with the best worst-case outcome. This is visualised as the α–LCB in Figure 2a. Following our toy example from Figure 2b, this inverts the UCB and flips the MAP ranking to obtain $a_3 > a_2 > a_1$. Reward predictions based on posterior lower bounds are designed to be conservative and thus strictly lower than the MAP estimates: $\widehat{p}_{i^*} > \widehat{p^{\text{LCB}}_{i^*}}$. As a consequence, it naturally follows that the post-decision disappointment from acting on these maximal lower bound predictions (actions j^*) will be strictly lower than if we had picked them according to their posterior mean predictions: $\mathbb{E}[p^*_{j^*} - \widehat{p}_{j^*}] < \mathbb{E}[p^*_{j^*} - \widehat{p^{\text{LCB}}_{j^*}}]$. Note that this result is quite loose and holds for any $\widehat{p^{\text{LCB}}_{j^*}} < \widehat{p}_{j^*}$; the posterior lower bounds still need to be constructed sensibly to improve the online performance of the resulting policy. As backed up by empirical observations from a wide range of experiments, our proposed pessimistic decision-making strategy leads to a significant and robust increase in recommendation performance. Naturally, the potential performance gains will be highest in those settings where traditional reward models fail: limited training sample sizes in large action spaces collected under highly skewed logging policies, as is often the case in real-world systems.

3.3.1 Pessimism in Policy Learning. As we have argued, the idea of scepticism, conservatism or pessimism is not novel in itself and lies at the heart of many advances in policy-based methods for off-policy learning, albeit often implicitly. One of the most widely used extensions to IPS weighting is that of capping the weights to a certain maximum value m [29, 37]. In doing so, we effectively choose to be sceptical about our reweighted rewards when things are too good to be true, replacing the probability ratio in Equation 1 with $\min\left(m, \frac{\pi(a|c)}{\widehat{\pi_0}(a|c)}\right)$. Capped IPS is known to improve the accuracy of the estimator and the performance of the resulting learnt policy, even when the logging propensities are known and exact. The use of such techniques is often justified by claiming an improved bias-variance trade-off, but the connections to over-estimation in reward models deserve mentioning.

Trust-Region Policy Optimisation (TRPO) and Proximal Policy Optimisation (PPO) are two well-known RL methods that adopt KL-regularisation and weight capping respectively, that have fostered significant progress in policy learning in recent years [92, 93]. Subtly different to our use-case, the motivation for these estimator lower-bounds comes from the instability of learning in complex RL environments. Here, the accumulation of many smaller, pessimistic, step-sizes can provably yield a globally optimal policy (under some conditions) [62]. Recommendation environments are typically less complex than those in which RL methods excel, and it is reasonable to assume that exploring poor actions early on would not significantly inhibit the system's ability to attain well-performing policies later on. Our motivation for pessimistic decision-making is much more closely tied to the Optimiser's Curse.

The same parallels between pessimism and existing methods can be drawn for several other policy learning tricks such as variance regularisation [72, 104], imitation learning [69], distributional robustness [26, 98] and other estimator lower bounds [47, 64]. Several concurrent recent works provide a deeper understanding of the value of pessimism in more general offline RL scenarios, be it in policy- or Q-learning-based methods [54, 56, 63, 114]. We point the interested reader towards the work of Jin et al. for more theoretical underpinnings [50].

3.3.2 Applications in Computational Advertising. As we have hinted at earlier on, computational advertising is one application area closely related to recommender systems where the consequences of over-estimation by machine learning models can have direct monetary consequences [33, 44, 75]. In a simplified case, these systems consist of an *ad allocation* step, where a machine learning model estimates the probability of observing a click when showing an ad in a certain context. An ad exchange will then solicit bids for the opportunity to show an ad in a given context from multiple competing advertisers. These advertisers need to decide on a bid given the context c, their estimated click probability on their ads $\widehat{\mathrm{P}}(R|C=c;A=a_i)$, and the value they have for a click on a particular ad a_i: $v(a_i)$. If we assume a truthful Vickrey auction [110], all bidders will be incentivised to bid their expected value: $b(c) = \mathbb{E}[v|c] \approx v(a_{i^*}) \cdot \widehat{\mathrm{P}}(R=1|C=c;A=a_{i^*})$. Here, the ad a_{i^*} is chosen equivalently to Eq. 5, triggering all of the problems associated with it. As a result, the estimated value of an ad impression is an overestimation, and the advertiser will, in expectation, bid an amount that is higher than the advertisement's true worth. These issues are only intensified when $v(a_i)$ is replaced by a machine learning model itself (for example, expected revenue from a click on a_i), as the variance of the final estimate will be the multiplication of its factors. The arg max operation that the advertiser carries out to decide on their highest-value ad is the main culprit for the Optimiser's Curse. However, in this setting, the ad exchange on their part also carries out an arg max operation over all incoming bids, which often leads to a related anomaly known as the Winner's Curse [107]. Because of the interplay of the two and the ubiquity of online advertising – we believe that pessimistic decision-making can have tremendous value in these use-cases, and that an understanding of these phenomena is crucial to success.

3.4 Closed-Form Lower-Confidence-Bounds with Bayesian Ridge Regression

By looking at the problem of learning an optimal recommendation policy through the lens of the "Direct Method", we effectively cast it as a classification or regression problem. As a consequence, the parameterisation of $\widehat{r}$ can take many forms. The pessimistic LCB method we propose in this work is generally applicable and not bound to any specific model class, with the exception that it relies on uncertainty estimates to generate sensible bounds. In what follows, we show how to obtain closed-form expressions for both the posterior mean and variance when a ridge regressor models the reward. The interpretability and efficiency of linear models makes them an attractive and common choice for practitioners that need to decide on a reward model [33, 47, 74, 75, 80]. An ongoing line of research in traditional approaches to recommendation has repeatedly shown the effectiveness of linear models in collaborative filtering tasks as well [14, 48, 81, 94, 101]. Other recent work reports empirical advantages of squared loss over cross-entropy loss [34], which could explain the effectiveness of item-based least-squares models like SLIM [81] and EASE$^{\mathrm{R}}$ [101], even when labels are binary. The model we propose here can be interpreted as a pessimistic, off-policy, bandit variant of the latter.

In line with the item-based paradigm, we model users based on their historical organic interactions with other items in the catalogue: $\boldsymbol{c} \in \mathbb{R}^{|\mathcal{A}|}$; additionally normalising samples according to their respective ℓ_1-norms to deal with varying-length user histories. Recommendations are represented as one-hot encoded vectors: $\boldsymbol{a} \in \{0,1\}^{|\mathcal{A}|}$. Action- and context-features are mapped to a joint space

via a Kronecker product: $\boldsymbol{x} = \Phi(\boldsymbol{c}, \boldsymbol{a}) = \boldsymbol{c} \otimes \boldsymbol{a}$. When we denote the model parameters by $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|^2}$, a linear model estimates the reward as shown in Equation 6 (omitting a bias-term for brevity).

$$\hat{\mathrm{P}}(R = 1|C = c, A = a, \boldsymbol{\theta}) = \boldsymbol{x}^\intercal \boldsymbol{\theta} = \boldsymbol{c}^\intercal \boldsymbol{\theta}_{|a} \tag{6}$$

Here, $\boldsymbol{\theta}_{|a}$ holds the parameters that are relevant for action a: the $|\mathcal{A}|$ parameters ending at index $i \cdot |\mathcal{A}|$ for actions $i \in \{1, \ldots, |\mathcal{A}|\}$. The final equation holds because we use a one-hot encoding for actions and a Kronecker product to link context and action features. This implementation trick makes computations significantly less expensive, as we now deal with vectors of size $|\mathcal{A}|$ instead of its square. If we define $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{D}| \times |\mathcal{A}|^2}$ as the design matrix holding joint context-action features for every sample in the training set $\mathcal{D}$, $\boldsymbol{y}$ as the vector of rewards to be predicted, and Θ the parameter space, we can formally define our optimisation problem as follows:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\min} \left(\|\boldsymbol{X}^\intercal \boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \right). \tag{7}$$

The Tikhonov-regularisation in Eq. 7 is key to the Bayesian interpretation of this ridge regression problem. Indeed, it is known that this formulation is equivalent to imposing independent Gaussian priors with constant variance on the parameters, as well as on the errors in the rewards [79]:

$$\boldsymbol{\theta} \sim \mathcal{N}(0, \sigma_x^2), \qquad \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}^\intercal \boldsymbol{\theta}, \sigma_y^2). \tag{8}$$

When $\lambda = \frac{\sigma_y^2}{\sigma_x^2}$, the solution to the ridge regression problem in Eq. 7 is equivalent to the MAP estimate for $\boldsymbol{\theta}$. The posterior mean and covariance can be computed efficiently with the analytical formulas presented in Eq. 9:

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y}, \qquad \hat{\Sigma} = (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}. \tag{9}$$

The main bottleneck here is the inversion of the $|\mathcal{A}|^2 \times |\mathcal{A}|^2$ Gramian matrix, which can quickly grow to be cumbersome for larger action spaces. In similar spirit to the implementation trick in Eq. 6, we can decompose this inversion into one per target action. This leads to $|\mathcal{A}|$ matrix inversions of size $|\mathcal{A}| \times |\mathcal{A}|$, and is possible because of the sparse, block-diagonal structure we acquire from the Kronecker product combined with one-hot encoded action vectors. We now end up with a model that is similar to the disjoint linear models used in the LinUCB procedure for on-policy bandit applications [59], although our prior variance ratio λ can be tuned whereas theirs is fixed (reducing to the MLE when $\lambda = 0$). Also in contrast with their approach, we will use the posterior mean and covariance to obtain a lower confidence bound reward estimate for a given context-action pair:

$$\widehat{\mathrm{P}}_{\mathrm{LCB}}(R = 1|C = c, A = a) = \boldsymbol{x}^\intercal \hat{\boldsymbol{\theta}} - \alpha \sqrt{\boldsymbol{x}^\intercal \hat{\Sigma} \boldsymbol{x}} = \boldsymbol{c}^\intercal \hat{\boldsymbol{\theta}}_{|a} - \alpha \sqrt{\boldsymbol{c}^\intercal \widehat{\Sigma}_{|a} \boldsymbol{c}}. \tag{10}$$

Let $\widehat{\Sigma}_{|a}$ denote the sub-matrix of $\hat{\Sigma}$ that is relevant to action a. That is, the $|\mathcal{A}|$ rows and columns ending at index $i \cdot |\mathcal{A}|$ for actions $i \in \{1, \ldots, |\mathcal{A}|\}$. From this formulation, it is clear that values in $\hat{\Sigma}$ off of this block-diagonal will never be used, and thus never need to be computed. The hyperparameter α is related to the coverage of the approximate posterior induced by $\widehat{\mathrm{P}}_{\mathrm{LCB}}$ [111]. Note that this hyperparameter is not specific to the ridge regression parameterisation, and will also occur when a nonlinear neural network models the reward and uncertainty estimates are obtained from the approximation techniques described in Section 2. Replacing the reward model in the direct method with this pessimistic alternative for the estimator based on the posterior mode ($\widehat{\mathrm{P}}_{\mathrm{LCB}}$ vs $\hat{\mathrm{P}}$), yields the optimal deterministic LCB policy:

$$\pi^*_{\text{LCB}}(a|c) = \begin{cases} 1 & \text{if } a = \arg\max_{a' \in \mathcal{A}} \widehat{r}_{\text{LCB}}(a', c), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

3.5 Connections with On-policy Bandits and the Explore-Exploit Trade-Off

Decades of work on multi-armed and contextual bandits that operate in an *on-policy* fashion has repeatedly shown that the concept of *optimism* is crucial to properly tackle the well-known explore-exploit trade-off [4, 10, 59]. Indeed, the bandit needs to find a way to balance the exploration of actions with uncertain reward estimates, and the exploitation of actions with high and certain reward estimates. One reason why this trade-off is prominently present in the literature, is because the evaluation metric used to validate the bandit's performance is often based on *cumulative regret*. That is, the cumulative difference between the reward obtained by the current policy and the reward obtained by the theoretically optimal policy. As we increase the number of rounds over which we evaluate, it is clear to see that a policy that converges to the optimal policy will eventually always be preferred over one that does not. Indeed, the cumulative regret of the former policy will stagnate, whereas that of the latter will continue to grow unbounded. Nevertheless, evaluation metrics that are adopted in practical scenarios are often focused on a balance between *immediate* and *long-term* rewards. As such, we do not merely focus on cumulative regret, but evaluate policies by their obtained reward over finite samples.

In essence, this discussion very much resembles that of the bias-variance trade-off in importance sampling. In theory, IPS weighting yields an unbiased estimator, and it should be preferred over biased extensions for this reason. In practice, where we have finite samples, non-stationarities and increasing complexity, weight-capping and the likes are known to yield empirical success [11, 29, 47]. As a result, we argue that depending on the environment, a policy that quickly identifies high-reward actions (and mainly *exploits)* could very well be preferable to one that perfectly balances the trade-off in the limit, but might spend too much time *exploring* in finite-samples. This is exactly what pessimism can achieve: by discounting highly uncertain actions, we are effectively favouring exploitation over exploration.

Nevertheless, the distinction between the on- and off-policy set-up is seldom as clear-cut as we have made it out to be. The crucial parameter is the length of the time-window between taking the action and learning from possible rewards (i.e. the delay in feedback Δ_t). Methods that continue to observe and learn instantly are truly on-policy ($\Delta_t = 0$). Those that learn from a fixed dataset and do not update based on new data, are truly off-policy ($\Delta_t = \infty$). In practice, the length of this time-window will not be at the extreme end of either 0 or infinity, but rather somewhere in between. Recent work that explores the use of bandits for top-K music recommendation also validates that in such "delayed feedback" settings, a mixture of Thompson sampling with *pessimistic* priors outperforms purely optimistic approaches [5]. Other work in the neighbouring field of Learning-To-Rank shows how online interventions can be incorporated in counterfactual estimators to greatly improve their sample efficiency in realistic scenarios [84]. Further exploring the connections between off- and on-policy methods is an exciting area for future research. Finally, we wish to emphasise that we are not arguing for exploration to be disregarded in recommender systems research. In contrast, we wish to find ways of exploring more *efficiently*. Indeed, the value of exploration is crucial for the long-term health of multiple involved stake-holders, and balancing it effectively is a challenging but nevertheless important task for user serendipity [13] and fairness of exposure [18, 43], among other key goals of the system beyond accuracy and immediate reward.

4 EXPERIMENTAL RESULTS AND DISCUSSION

A key component of recommender systems is their interactive nature: evaluating recommendation policies on offline datasets is not a straightforward task, and conclusions drawn from offline results often contrast with the online metrics that we care about [28, 39, 88]. Indeed, this is a strong motivation for casting recommendation as a bandit learning problem, allowing for *offline* optimisation of *online* objectives [40, 46].

Methods for BLBF are often evaluated using supervised-to-bandit conversions on multi-class or multi-label classification datasets [26, 104, 105]. This type of empirical validation is warranted in general machine learning use-cases, but it is unclear how these result translate to improved recommendations [47]. Recent work on BLBF *for recommenders* either shows empirical success by adopting the same supervised-to-bandit conversions on *organic* user-item datasets [68], through live experiments [11, 74, 78], or by adopting open-source simulation environments [47, 91] (which have seen a growing interest in the Recommender Systems community as of late [22, 35], among related research areas [44]).

To aid in the reproducibility of our work, we make use of the RecoGym simulation environment [87]. RecoGym provides functionality to simulate organic user-item interactions (e.g. users viewing products on a retail website), as well as bandit interactions under a given logging policy (users clicking on shown recommendations). Publicly available datasets that contain both types of data (observational *and* experimental) are scarce, and still insufficient for reliable counterfactual evaluation. A considerable advantage of RecoGym is the opportunity to simulate online experiments such as A/B-tests, that can then be used to reliably estimate the online performance of an intervention policy in the synthetic environment. We refer the interested reader to the source code of the simulator[5] or the reproducibility appendix of [47] for an overview of the inner workings of the simulation environment, while pointing out that the underlying RecoGym reward model adopts a latent factor model assumption that is often made in recommender systems research [55]. The source code to reproduce our experiments is publicly available at github.com/olivierjeunen/pessimism-recsys-2021. The research questions we wish to answer are the following:

RQ1 Can we find empirical evidence of the Optimiser's Curse in off-policy recommendation environments?

RQ2 Can our proposed LCB decision-making strategy effectively limit post-decision disappointment?

RQ3 Can we increase online performance with a recommendation policy using a reward model with LCB predictions?

RQ4 How are these methods influenced by the amount of randomisation in the logging policy?

RQ5 How are these methods influenced by the number of training samples and the size of the action space?

4.1 Logging Policies

An important factor to take into account when learning from bandit feedback is the logging policy that was deployed at the time of data collection. Deterministic policies make bandit learning nearly impossible, whereas a uniformly random logging policy generates unbiased data, but is an idealised case in practice. Realistic logging policies will aim to show recommendations that they perceive to be relevant, whilst allowing other actions to be taken in an explorative manner. We adopt a simple but effective personalised popularity policy based on the organic user-item interactions that have preceded the impression opportunity. For a context c consisting of historical counts of organic interactions with items (as laid out in the parameterisation in Section 3.4), the logging policy π_{pop}

[5] github.com/criteo-research/reco-gym

samples actions proportionately to their organic occurrences. This policy is deficient, as it does not assign a non-zero probability mass to every possible action in every possible context [89]. Deficient logging policies violate the assumptions made by IPS to yield an unbiased reward estimate [86], which poses a significant hurdle for policy-based methods. Nevertheless, they are realistic to consider in real-world off-policy recommendation scenarios. This extreme form of selection bias impedes effective reward modelling as well, as we will show in the following section. Indeed, when a context-action pair has zero probability of occurring in the training sample, we *need* to resort to appropriate priors or conservative decision making. The deficiency of π_{pop} can be mitigated easily by adopting an ϵ-greedy exploration mechanism, where we resort to the uniform policy with probability $\epsilon \in [0, 1]$. Naturally, this implies both π_{pop} and π_{uni} when ϵ is respectively 0 or 1. For arbitrarily small values of ϵ, π_0 is no longer deficient in theory, but extremely unlikely to explore the full context-action space within finite samples.

$$\pi_0(a|c) = \begin{cases} \pi_{\text{pop}}(a|c) & \text{with probability } 1-\epsilon, \\ \pi_{\text{uni}}(a|c) & \text{otherwise,} \end{cases} \qquad \text{where } \pi_{\text{pop}}(a_i|c) = \frac{\boldsymbol{c}_i}{\sum_{j=1}^{|\mathcal{A}|} \boldsymbol{c}_j}, \text{ and } \pi_{\text{uni}}(a|c) = \frac{1}{|\mathcal{A}|}. \tag{12}$$

We vary $\epsilon \in \{0, 10^{-6}, 10^{-4}, 10^{-2}, 1\}$ in our experimental setup. Note that this type of logging policy is equivalent to the ones used in previous works [41, 45, 47, 80, 91], but that we explore a wider range of logging policy randomisation to highlight the effects on naïve reward modelling procedures.

4.2 Optimiser's Curse (RQ1-3)

To validate whether the theoretical concept of the Optimiser's Curse actually occurs when reward models are learned in off-policy recommendation settings, we adopt the following procedure:

(1) Generate a dataset containing organic and bandit feedback,
(2) train a reward model as described in Section 3.4 – optimising the regularisation strength λ to minimise Mean Squared Error (MSE) on a validation set of 20%,
(3) simulate an A/B-test and log the difference between the reward estimates $\widehat{p}_{i^*}$ and the true reward probability $p^*_{i^*}$ for the actions selected by the competing decision strategies.

We then vary the logging policy in (1), and repeat this process 5 times to ensure statistically robust and significant results. Every generated training set and every simulated A/B-test consists of 10 000 distinct users, leading to approximately 800 000 bandit opportunities in the training set as well as 800 000 online impressions per evaluated policy.

The Optimiser's Curse states that we should expect to be disappointed with respect to our reward estimates. As such, we define the average empirical disappointment as the difference between the true expected reward and the expected reward estimated by the reward model: $\widehat{p_{i^*}} - p^*_{i^*}$. As we have argued earlier, a simple bias term on the estimates $\widehat{p_{i^*}}$ can be tuned to bring the average empirical disappointment to zero. This, however, has no impact on the decision making strategy, and therefore does not solve our problems. Indeed, our goal is two-fold: we wish to *decrease* absolute disappointment, whilst *increasing* the online reward our recommendation policy obtains. Figure 3 plots these two quantities for competing decision strategies, varying the amount of selection bias in the logging policy per column, and increasing the size of the action space over the rows. Plots in the upper right quadrant of the figure correspond to less realistic environments, where the size of the action space and the cost of randomisation are limited. In real-world scenarios, the opposite will often be true. The plots on the lower left side of the figure reflect these constraints. The x-axes show disappointment (closer to zero is better), and the y-axes shows a 95% credible interval for the

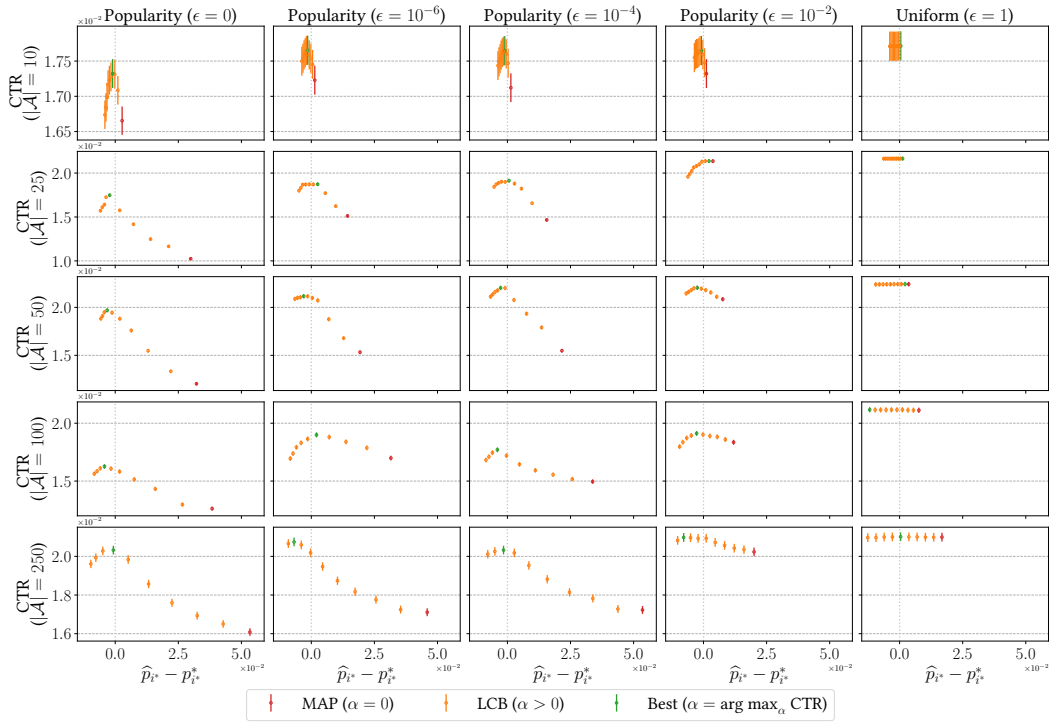


Fig. 3. We evaluate varying degrees of pessimistic decision strategies ($\alpha \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$, corresponding with right- to leftmost measurements in the plots). The x-axis shows the resulting post-decision disappointment with the attained CTR on the y-axis (95% credible intervals). Every column corresponds to a different amount of randomisation in the logging policy (increasing from left to right), every row corresponds to a differently sized action space (increasing from top to bottom). We observe that LCB is effective in minimising post-decision disappointment, and that this is highly correlated with increasing online performance, most notably when the amount of logging randomisation is limited and the action space is large.

obtained click-through-rate (CTR) per recommendation policy in the simulated A/B-test (higher is better). Maximum Likelihood Estimates are consistently so far off that we do not include them in this analysis. The baseline and widely adopted decision strategy of taking the highest MAP action is shown ($\alpha = 0$), along with our pessimistic lower-confidence-bound strategy, varying the lower posterior quantile $\alpha \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$. Increased values of α strictly decrease disappointment, and $\alpha = 0$ always corresponds to the rightmost measurement within a plot, whereas $\alpha = 0.45$ is shown leftmost. Note that this hyperparameter α plays an important role, and that it can always be increased to achieve zero post-decision disappointment (and even lower – indicating that we are being overly pessimistic). While this sets more realistic expectations for the performance of the reward model (and hence is important when the model is used for offline evaluation or computational advertising), this provides no theoretically guaranteed improvement in the online metrics we care about. Also note that this type of experimental procedure would not be feasible without the use of a simulation environment, as we usually don't have access to the true reward probability p^*_i. In such cases, we would need to resort to empirical averages based on the observed reward.

Empirical Observations. First, we see clear empirical evidence of the Optimiser's Curse in action: when acting based on the MAP estimate ($\alpha = 0$), we encounter post-decision disappointment regardless of the logging policy. As our trained reward models are even slightly under-calibrated w.r.t. the empirical training sample (i.e. negative mean error), this result can seem counter-intuitive and is not straightforward to mitigate with a bias term tuned on offline data. Second, we observe that pessimistic decision-making based on predictive uncertainty consistently decreases disappointment, and that it can significantly increase the policy's attained CTR in A/B-tests. The optimal value of α with respect to online performance also brings the average empirical disappointment closer to zero, indicating that these values are closely related. α's interpretation relating to the coverage of the approximate posterior of $\widehat{r}$ helps when tuning it [111]. Naturally, when the variance on the reward estimates is homoscedastic w.r.t. the actions, LCB does not affect the ordering of the reward estimates or the resulting policy. This explains why online performance is not significantly impacted when the logging policy is uniform, while post-decision disappointment can consistently be alleviated. We observe that the expected benefits of pessimism, both in terms of decreased disappointment and in terms of increased online reward, are lower in the upper right quadrant. This is to be expected, as the context-action space is more likely to be well explored in these cases, and the MAP estimate achieves good performance. In the more realistic settings in the lower left, the improvements are significant and consistent. Indeed, we observe that MAP estimates consistently over-estimate the expected reward, by a large margin. In the case of $\epsilon = 0$ and $|\mathcal{A}| = 250$, the MAP strategy obtains a CTR of 1.6%, with a disappointment of 5.2%: over-estimating the reward by a factor of 3.25. Our proposed explicitly pessimistic decision-making strategy removes all empirical disappointment while improving CTR by 28%.

4.3 Performance Comparison (RQ3-5)

To further assess when our proposed pessimistic decision-making procedure can lead to an offline learnt policy with improved online performance, we train models on a range of datasets generated under different environmental conditions and report results from several simulated A/B-tests. The resulting CTR estimates with their 95% credible intervals are shown in Figure 4. Every row corresponds to a differently sized action space ($|\mathcal{A}| \in 10, 25, 50, 100, 250$), every column shows results for a different amount of randomisation in the logging policy. The amount of available training data for the reward model increases over the x-axis for every plot. We report CTR estimates for policies that act according to reward models based on ML or MAP estimates, and those that use lower confidence bounds with a tuned α. Additionally, we show the CTR attained by the logging policy π_0, and an unattainable skyline policy π^* that acts based on the true reward probabilities p^*. This provides an upper bound on the expected CTR that any decision-making strategy can obtain. Every measurement shown in Figure 4 shows a 95% credible interval over 5 runs with 10 000 evaluation users, totalling 1 000 simulated A/B-tests with five competing policies each, or more than three billion impressions summed up. As our reward models are agnostic to the logging propensities, we do not include policy-based approaches that would require them (either purely based on IPS [8], hybrid [47] or doubly robust [19]). We do note that our results are directly comparable to those presented in [47, 80], and both our novel LCB method and MAP baseline show significant improvements over all their policy- and value-based competitors.

Empirical Observations. In line with our observations from Figure 3, we see that LCB decision-making yields a robust and significant improvement over naively acting on ML or MAP estimates. This result is consistent over varying training sample sizes, action spaces and logging policies, but most outspoken in cases where the amount of randomisation and the number of available training samples are limited, and the action space is larger. As explicit randomisation and data collection can

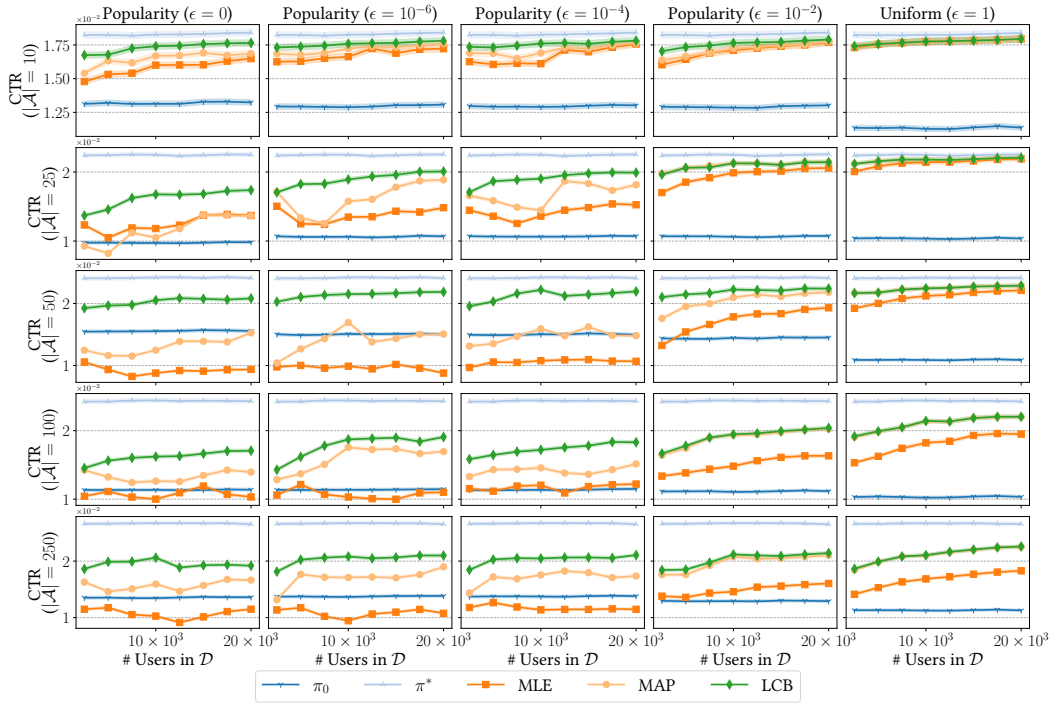


Fig. 4. Experimental results for a range of simulated A/B-tests. The amount of training data is increased over the x-axis, the attained CTR is shown on the y-axis (shaded 95% credible interval). Every column corresponds to a different amount of randomisation in the logging policy (increasing from left to right); every row corresponds to a differently sized action space (increasing from top to bottom). We observe a significant increase in CTR for the pessimistic model, most apparent for smaller training samples, larger action spaces, and limited randomisation. The CTR improvements for LCB over MAP average to 16% over all measurements, and range up to 95%.

be expensive in practice, the environments where LCB excels are the ones that are most commonly encountered in real-world systems. Additionally, we observe more consistent and robust behaviour for policies that use LCB decisions compared to those that do not. This decreased variance in online performance can also be attributed to pessimistic decision making: because we no longer take our chances with high-uncertainty predictions, we fall back to more robust alternatives. We know what the reward model does not know, and this gained knowledge significantly benefits the interpretation of reward predictions, and the resulting decisions.

Limitations of the Study Design. Off-policy approaches for learning from bandit feedback are typically evaluated in set-ups where the size of the action space is a few dozen at most [51, 102, 104]. As a result, methods for counterfactual learning in recommendation are often evaluated in modestly sized action spaces too [47, 80, 91]. Therefore, the reported results are most relevant to personalisation use-cases where the number of alternatives is limited, such as personalising tiles or rows on a homepage, recommending news articles from a set of recently published ones, or predicting clicks within a slate. The size of the item catalogue in general purpose recommendation scenarios can be in the hundreds of thousands, warranting further research into off-policy recommendation for very large action spaces [65]. In such environments, learning continuous item embeddings as opposed to the discrete representation we have adopted can provide a way forward. Moreover,

the lack of publicly available datasets for the off-policy recommendation task can be prohibitive for reproducible empirical validation of newly proposed methods. The few alternatives that do exist [57, 90], still deal with comparatively small action spaces and need to resort to counterfactual evaluation procedures with high variance and limited statistical power (compared to simulated online experiments). Furthermore, a single dataset would be comparable to a single measurement in Figure 4, limiting the range of environmental parameters we can change to observe effects on the online performance for competing methods. Because of these reasons, we believe the RecoGym environment to be an appropriate choice for the experimental validation of our methods [87].

5 CONCLUSIONS AND FUTURE WORK

Recommender systems are evolving, turning from *prediction*-based systems into *decision*-based systems. Under this new paradigm, effective and efficient learning from bandit feedback is crucial in order to flourish. One problematic aspect is that bandit feedback is typically collected under some logging policy, which leads to selection biases that can be difficult to deal with. Policy-based methods based on importance sampling are often adopted in these cases – and pessimistic variants have been known to improve empirical performance. Nevertheless, they often rely on strict randomisation assumptions and their high variance remains especially troublesome. Moreover, several application areas rely on calibrated predictions for the probability of the outcome conditional on the action that the system takes, which is exactly what policy-based methods avoid to model.

In this work, we aim to increase the reward obtained through value-based recommendation methods that rely on explicit reward models. We have argued that in the off-policy setting, selection bias is the most prominent and problematic, and have introduced the decision-making phenomenon of the "Optimiser's Curse". In order to lift the curse, we have proposed a general framework for the use of principled pessimism. For the specific case where a ridge regressor models the reward, we have shown how to translate closed-form uncertainty estimates into a conservative decision rule. Extensive experiments with synthetic data show that our proposed method lifts the Optimiser's Curse whilst achieving a significant and robust boost in recommendation performance for a variety of settings. When randomisation in the logging policy is limited, the action space is large, and the size of the training sample is limited, our Lower-Confidence-Bound approach yields the highest improvements over decision-making alternatives. This is a promising and encouraging result, as these settings are exactly those that widely occur in practice.

Pessimism has widely been implicitly accepted as a tool to improve policy learning performance for recommendation problems. We draw parallels with existing work and highlight key differences and overlap. Furthermore, we explore connections with on-policy use-cases where *optimistic* decision-making reigns supreme, emphasising that our novel insights are not in conflict with those presented in earlier work. Indeed, the goals of deployed recommender systems might not be best measured in terms of cumulative regret, but rather in terms of obtained reward. How to best balance efficient exploration in these settings is a largely open problem, although its value is clear [13].

Further directions for future work include to investigate whether pessimistic reward predictions can lead to improved doubly robust learning [41], whether our results can be generalised to larger action spaces, and to investigate the effects of scepticism on the informational value of data collected under such a policy. Moving to realistic settings with multiple iterations of logging and learning, we wish to make our proposed decision-making method more widely applicable in real-world deployments. In order for this to be successful, we need a notion of long-term consequences of actions, and may need to balance optimism with pessimism when appropriate.

ACKNOWLEDGMENTS

This work received funding from the Flemish Government (AI Research Program).

REFERENCES

[1] A. Agarwal, S. Basu, T. Schnabel, and T. Joachims. 2017. Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers. In *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '17)*. ACM, 687–696.

[2] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, 5–14.

[3] A. Agarwal, X. Wang, C. Li, M. Bendersky, and M. Najork. 2019. Addressing Trust Bias for Unbiased Learning-to-Rank. In *Proc. of the 2019 World Wide Web Conference (WWW '19)*. ACM, 4–14.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2 (2002), 235–256.

[5] W. Bendada, G. Salha, and T. Bontempelli. 2020. Carousel Personalization in Music Streaming Apps with Contextual Bandits. In *Proc. of the 14th ACM Conference on Recommender Systems (RecSys '20)*. ACM, 420–425.

[6] J. Bennett, S. Lanning, et al. 2007. The Netflix prize. In *Proc. of the KDD cup and workshop*, Vol. 2007. 35.

[7] J. O. Berger and R. L. Wolpert. 1988. The Likelihood Principle. IMS.

[8] L. Bottou, J. Peters, J. Quiñonero-Candela, D. Charles, D. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14, 1 (2013), 3207–3260.

[9] A. Chaney, B. Stewart, and B. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 224–232.

[10] O. Chapelle and L. Li. 2011. An Empirical Evaluation of Thompson Sampling. In *Proc. of the 24th International Conference on Neural Information Processing Systems (NIPS'11)*. 2249–2257.

[11] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proc. of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 456–464.

[12] M. Chen, B. Chang, C. Xu, and E. H. Chi. 2021. User Response Models to Improve a REINFORCE Recommender System. In *Proc. of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. ACM, 121–129.

[13] M. Chen, Y. Wang, C. Xu, Y. Le, M. Sharma, L. Richardson, S. Wu, and E. Chi. 2021. Values of User Exploration in Recommender Systems. In *Proc. of the Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. ACM, 85–95.

[14] Y. Chen, Y. Wang, X. Zhao, J. Zou, and M. de Rijke. 2020. Block-Aware Item Similarity Models for Top-N Recommendation. *ACM Trans. Inf. Syst.* 38, 4, Article 42 (Sept. 2020), 26 pages.

[15] Z. Chen, Y. Wang, D. Lin, D. Z. Cheng, L. Hong, E. H. Chi, and C. Cui. 2021. Beyond Point Estimate: Inferring Ensemble Prediction Variation from Neuron Activation Strength in Recommender Systems. In *Proc. of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. ACM, 76–84.

[16] M. Choi, J. Kim, J. Lee, H. Shim, and J. Lee. 2021. Session-aware Linear Item-Item Models for Session-based Recommendation. In *Proc. of the 2021 World Wide Web Conference (WWW '21)*.

[17] M. F. Dacrema, P. Cremonesi, and D. Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 101–109.

[18] F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. 2020. Evaluating Stochastic Rankings with Expected Exposure. In *Proc. of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. ACM, 275–284.

[19] M. Dudík, J. Langford, and L. Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proc. of the 28th International Conference on International Conference on Machine Learning (ICML'11)*. 1097–1104.

[20] B. Dumitrascu, K. Feng, and B. E. Engelhardt. 2018. PG-TS: Improved Thompson Sampling for Logistic Contextual Bandits. In *Proc. of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. 4629–4638.

[21] B. Efron and R. J. Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

[22] M. D. Ekstrand, A. Chaney, P. Castells, R. Burke, D. Rohde, and M. Slokom. 2021. SimuRec: Workshop on Synthetic Data and Simulation Methods for Recommender Systems Research. In *Proc. of the Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. ACM, 803–805.

[23] E. Elahi, W. Wang, D. Ray, A. Fenton, and T. Jebara. 2019. Variational Low Rank Multinomials for Collaborative Filtering with Side-information. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 340–347.

[24] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo. 2019. Generalized Multiple Importance Sampling. *Statist. Sci.* 34, 1 (02 2019), 129–155.

[25] M. Farajtabar, Y. Chow, and M. Ghavamzadeh. 2018. More Robust Doubly Robust Off-policy Evaluation. In *Proc. of the 35th International Conference on Machine Learning (ICML'18, Vol. 80)*. PMLR, 1447–1456.
[26] L. Faury, U. Tanielian, F. Vasile, E. Smirnova, and E. Dohmatob. 2020. Distributionally Robust Counterfactual Risk Minimization. In *Proc. of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*. AAAI Press.
[27] Y. Gal and Z. Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. of The 33rd International Conference on Machine Learning (ICML '16)*. PMLR, 1050–1059.
[28] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. 2014. Offline and Online Evaluation of News Recommender Systems at Swissinfo.Ch. In *Proc. of the 8th ACM Conference on Recommender Systems (RecSys '14)*. 169–176.
[29] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proc. of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, 198–206.
[30] D. Guo, S. I. Ktena, P. K. Myana, F. Huszar, W. Shi, A. Tejani, M. Kneier, and S. Das. 2020. Deep Bayesian Bandits: Exploring in Online Personalized Recommendations. In *Proc. of the 14th ACM Conference on Recommender Systems (RecSys '20)*. ACM, 456–461.
[31] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proc. of the 35th International Conference on Machine Learning (ICML '18, Vol. 80)*. PMLR, 1861–1870.
[32] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (2015), 19 pages.
[33] X. He, O. Pan, J.and Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. Candela. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proc. of the 8th International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, 1–9.
[34] L. Hui and M. Belkin. 2021. Evaluation of Neural Architectures Trained with Square Loss vs Cross-Entropy in Classification Tasks. In *Proc. of the 9th International Conference on Learning Representations (ICLR '21)*. arXiv:2006.07322 [cs.LG]
[35] E. Ie, C. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, and C. Boutilier. 2019. RecSim: A Configurable Simulation Platform for Recommender Systems. arXiv:1909.04847 [cs.LG]
[36] E. Ie, V. Jain, J. Wang, S. Narvekar, R. Agarwal, R. Wu, H. Cheng, T. Chandra, and C. Boutilier. 2019. SlateQ: A Tractable Decomposition for Reinforcement Learning with Recommendation Sets. In *Proc. of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. 2592–2599.
[37] E. L. Ionides. 2008. Truncated Importance Sampling. *Journal of Computational and Graphical Statistics* 17, 2 (2008), 295–311.
[38] A. H. Jadidinejad, C. Macdonald, and I. Ounis. 2020. Using Exploration to Alleviate Closed Loop Effects in Recommender Systems. In *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, 2025–2028.
[39] O. Jeunen. 2019. Revisiting Offline Evaluation for Implicit-feedback Recommender Systems. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 596–600.
[40] O. Jeunen. 2021. *Offline Approaches to Recommendation with Online Success*. Ph. D. Dissertation. University of Antwerp.
[41] O. Jeunen and B. Goethals. 2020. An Empirical Evaluation of Doubly Robust Learning for Recommendation. In *Proc. of the ACM RecSys Workshop on Bandit Learning from User Interactions (REVEAL '20)*.
[42] O. Jeunen and B. Goethals. 2021. Pessimistic Reward Models for Off-Policy Learning in Recommendation. In *Proc. of the Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. ACM, 63–74.
[43] O. Jeunen and B. Goethals. 2021. Top-K Contextual Bandits with Equity of Exposure. In *Proc. of the Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. ACM, 310–320.
[44] O. Jeunen, S. Murphy, and B. Allison. 2022. Learning to Bid with AuctionGym. In *Proc. of the AdKDD Workshop at the 28th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (AdKDD '22)*.
[45] O. Jeunen, D. Mykhaylov, D. Rohde, F. Vasile, A. Gilotte, and M. Bompaire. 2019. Learning from Bandit Feedback: An Overview of the State-of-the-art. In *Proc. of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation (REVEAL '19)*.
[46] O. Jeunen, D. Rohde, and F. Vasile. 2019. On the Value of Bandit Feedback for Offline Recommender System Evaluation. In *Proc. of the ACM RecSys Workshop on Reinforcement Learning and Robust Estimators for Recommendation (REVEAL '19)*.
[47] O. Jeunen, D. Rohde, F. Vasile, and M. Bompaire. 2020. Joint Policy-Value Learning for Recommendation. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 1223–1233.
[48] O. Jeunen, J. Van Balen, and B. Goethals. 2020. Closed-Form Models for Collaborative Filtering with Side-Information. In *Proc. of the 14th ACM Conference on Recommender Systems (RecSys '20)*. ACM, 651–656.

[49] O. Jeunen, J. Van Balen, and B. Goethals. 2022. Embarrassingly shallow auto-encoders for dynamic collaborative filtering. *User Modeling and User-Adapted Interaction* (2022).
[50] Y. Jin, Z. Yang, and Z. Wang. 2020. Is Pessimism Provably Efficient for Offline RL? arXiv:2012.15085 [cs.LG]
[51] T. Joachims, A. Swaminathan, and M. de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In *Proc. of the 6th International Conference on Learning Representations (ICLR '18)*.
[52] T. Joachims, A. Swaminathan, and T. Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proc. of the 10th ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, 781–789.
[53] N. Kallus, Y. Saito, and M. Uehara. 2021. Optimal Off-Policy Evaluation from Multiple Logging Policies. In *Proc. of the 38th International Conference on Machine Learning (ICML '21, Vol. 139)*. PMLR, 5247–5256.
[54] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. 2020. MOReL: Model-Based Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*.
[55] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37.
[56] A. Kumar, A. Zhou, G. Tucker, and S. Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*.
[57] D. Lefortier, A. Swaminathan, X. Gu, T. Joachims, and M. de Rijke. 2016. Large-scale validation of counterfactual learning methods: A test-bed. *arXiv preprint arXiv:1612.00367* (2016).
[58] S. Levine, A. Kumar, G. Tucker, and J. Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643 [cs.LG]
[59] L. Li, W. Chu, J. Langford, and R. E. Schapire. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proc. of the 19th International Conference on World Wide Web (WWW '10)*. ACM, 661–670.
[60] S. Li, A. Karatzoglou, and C. Gentile. 2016. Collaborative Filtering Bandits. In *Proc. of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, 539–548.
[61] D. Liang, R. G. Krishnan, M. D Hoffman, and T. Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proc. of the 2018 World Wide Web Conference (WWW '18)*. ACM, 689–698.
[62] B. Liu, Q. Cai, Z. Yang, and Z. Wang. 2019. Neural Proximal/Trust Region Policy Optimization Attains Globally Optimal Policy. In *Proc. of the 33rd International Conference on Neural Information Processing Systems (NeurIPS '19)*. Article 948, 12 pages.
[63] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill. 2020. Provably Good Batch Off-Policy Reinforcement Learning Without Great Exploration. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*.
[64] B. London and T. Sandler. 2019. Bayesian Counterfactual Risk Minimization. In *Proc. of the 36th International Conference on Machine Learning (ICML '19, Vol. 97)*. PMLR, 4125–4133.
[65] R. Lopez, I. Dhillion, and M. I. Jordan. 2021. Learning from eXtreme Bandit Feedback. In *Proc. of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*. AAAI Press.
[66] C. Ma, L. Ma, Y. Zhang, R. Tang, X. Liu, and M. Coates. 2020. Probabilistic Metric Learning with Adaptive Margin for Top-K Recommendation. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 1036–1044.
[67] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi. 2018. Modeling Task Relationships in Multi-Task Learning with Multi-Gate Mixture-of-Experts. In *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18)*. ACM, 1930–1939.
[68] J. Ma, Z. Zhao, X. Yi, J. Yang, M. Chen, J. Tang, L. Hong, and E. H. Chi. 2020. Off-Policy Learning in Two-Stage Recommender Systems. In *Proc. of the 2020 World Wide Web Conference (WWW '20)*. ACM.
[69] Y. Ma, Y. Wang, and B. Narayanaswamy. 2019. Imitation-Regularized Offline Learning. In *Proc. of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) (AIStats '19, Vol. 89)*. PMLR, 2956–2965.
[70] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke. 2020. Feedback Loop and Bias Amplification in Recommender Systems. In *Proc. of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*. ACM, 2145–2148.
[71] A. Masegosa. 2020. Learning under Model Misspecification: Applications to Variational and Ensemble methods. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*. 5479–5491.
[72] A. Maurer and M. Pontil. 2009. Empirical Bernstein Bounds and Sample Variance Penalization. *Stat.* 1050 (2009), 21.
[73] B. C. May, N. Korda, A. Lee, and D. S. Leslie. 2012. Optimistic Bayesian Sampling in Contextual-Bandit Problems. *J. Mach. Learn. Res.* 13, 1 (June 2012), 2069–2106.
[74] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, and R. Mehrotra. 2018. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 31–39.
[75] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proc. of the 19th ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining. ACM, 1222–1230.
[76] R. Mehrotra. 2021. Algorithmic Balancing of Familiarity, Similarity, & Discovery in Music Recommendations. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21)*. ACM, 3996–4005.
[77] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems. In *Proc. of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, 2243–2251.
[78] R. Mehrotra, N. Xue, and M. Lalmas. 2020. Bandit Based Optimization of Multiple Objectives on a Music Streaming Platform. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 3224–3233.
[79] K. P. Murphy. 2021. *Probabilistic Machine Learning: An introduction*. MIT Press.
[80] D. Mykhaylov, D. Rohde, F. Vasile, M. Bompaire, and O. Jeunen. 2019. Three Methods for Training on Bandit Feedback. In *Proc. of the NeurIPS Workshop on Causality and Machine Learning (CausalML '19)*.
[81] X. Ning and G. Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proc. of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE Computer Society, 497–506.
[82] H. Oosterhuis and M. de Rijke. 2018. Ranking for Relevance and Display Preferences in Complex Presentation Layouts. In *Proc. of the 41st International ACM SIGIR Conference on Research & in Information Retrieval (SIGIR '18)*. ACM, 845–854.
[83] H. Oosterhuis and M. de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. In *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, 489–498.
[84] H. Oosterhuis and M. de Rijke. 2021. Unifying Online and Counterfactual Learning to Rank: A Novel Counterfactual Estimator That Effectively Utilizes Online Interventions. In *Proc. of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. ACM, 463–471.
[85] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. 2016. Deep Exploration via Bootstrapped DQN. In *Advances in Neural Information Processing Systems*, Vol. 29. 4026–4034.
[86] A. B. Owen. 2013. *Monte Carlo theory, methods and examples*.
[87] D. Rohde, S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou. 2018. RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. In *Proc. of the ACM RecSys Workshop on Offline Evaluation for Recommender Systems (REVEAL '18)*.
[88] M. Rossetti, F. Stella, and M. Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proc. of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 31–34.
[89] N. Sachdeva, Y. Su, and T. Joachims. 2020. Off-Policy Bandits with Deficient Support. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 965–975.
[90] Y. Saito, S. Aihara, M. Matsutani, and Y. Narita. 2020. Large-scale Open Dataset, Pipeline, and Benchmark for Bandit Algorithms. arXiv:2008.07146 [cs.LG]
[91] O. Sakhi, S. Bonner, D. Rohde, and F. Vasile. 2020. BLOB : A Probabilistic Model for Recommendation that Combines Organic and Bandit Signals. In *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 783–793.
[92] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. 2015. Trust Region Policy Optimization. In *Proc. of the 32nd International Conference on Machine Learning*, Vol. 37. PMLR, 1889–1897.
[93] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347
[94] S. Sedhain, A. Menon, S. Sanner, and D. Braziunas. 2016. On the Effectiveness of Linear Models for One-Class Collaborative Filtering. *Proc. of the AAAI Conference on Artificial Intelligence* 30, 1 (2016).
[95] G. Shani, D. Heckerman, and R. I. Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005).
[96] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proc. of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. ACM, 528–536.
[97] H. Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90, 2 (2000), 227 – 244.
[98] N. Si, F. Zhang, Z. Zhou, and J. Blanchet. 2020. Distributionally Robust Policy Evaluation and Learning in Offline Contextual Bandits. In *International Conference on Machine Learning (ICML'20)*.
[99] J. E. Smith and R. L. Winkler. 2006. The Optimizer's Curse: Skepticism and Postdecision Surprise in Decision Analysis. *Management Science* 52, 3 (2006), 311–322.
[100] H. Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Washington, DC, USA) *(KDD '10)*.

ACM, 713–722.
[101] H. Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference (WWW '19)*. ACM, 3251–3257.
[102] Y. Su, M. Dimakopoulou, A. Krishnamurthy, and M. Dudik. 2020. Doubly robust off-policy evaluation with shrinkage. In *Proc. of the 37th International Conference on Machine Learning (ICML '20)*. PMLR, 9167–9176.
[103] Y. Su, L. Wang, M. Santacatterina, and T. Joachims. 2019. CAB: Continuous Adaptive Blending for Policy Evaluation and Learning. In *International Conference on Machine Learning (ICML'19)*. 6005–6014.
[104] A. Swaminathan and T. Joachims. 2015. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *Proc. of the 32nd International Conference on International Conference on Machine Learning (ICML'15)*. JMLR.org, 814–823.
[105] A. Swaminathan and T. Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*. 3231–3239.
[106] H. Tang, J. Liu, M. Zhao, and X. Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Proc. of the 14th ACM Conference on Recommender Systems (RecSys '20)*. ACM, 269–278.
[107] R. H. Thaler. 1988. Anomalies: The Winner's Curse. *Journal of Economic Perspectives* 2, 1 (March 1988), 191–202.
[108] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells. 2018. On the Robustness and Discriminative Power of Information Retrieval Metrics for Top-N Recommendation. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 260–268.
[109] F. Vasile, D. Rohde, O. Jeunen, and A. Benhalloum. 2020. A Gentle Introduction to Recommendation as Counterfactual Policy Learning. In *Proc. of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20)*. ACM, 392–393.
[110] W. Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance* 16, 1 (1961), 8–37.
[111] T. J. Walsh, I. Szita, C. Diuk, and M. L. Littman. 2009. Exploring Compact Reinforcement-Learning Representations with Linear Regression. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, 591–598.
[112] Y. Wang, H. He, and X. Tan. 2020. Truly Proximal Policy Optimization. In *Proc. of The 35th Uncertainty in Artificial Intelligence Conference (UAI '21, Vol. 115)*. PMLR, 113–122.
[113] X. Xin, A. Karatzoglou, I. Arapakis, and J. M. Jose. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. In *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, 931–940.
[114] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. 2020. MOPO: Model-Based Offline Policy Optimization. In *Advances in Neural Information Processing Systems (NeurIPS '20, Vol. 33)*.
[115] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. H. Chi. 2019. Recommending What Video to Watch next: A Multitask Ranking System. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 43–51.
[116] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin. 2019. Reinforcement Learning to Optimize Long-Term User Engagement in Recommender Systems. In *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, 2810–2818.