Pessimistic Reward Models for Off-Policy Learning in Recommendation

OLIVIER JEUNEN and BART GOETHALS, Adrem Data Lab, University of Antwerp, Belgium

Methods for bandit learning from user interactions often require a 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 is often 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.

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 environments show that being conservative in 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.

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. 2021. Pessimistic Reward Models for Off-Policy Learning in Recommendation. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27–October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3460231.3474247

1 INTRODUCTION

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. 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 [6, 50]. Traditional recommendation research assumes *organic* user-item interaction data to bypass any feedback loops that might occur due to deployed systems. 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 [8, 80].

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, 8, 35, 61]. These methods typically make use of importance sampling or Inverse Propensity Score (IPS) weighting, in order to obtain an unbiased estimate of the

Manuscript submitted to ACM

counterfactual value-of-interest [63]. 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 [5], 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 [22]. Furthermore, the randomisation requirements for IPS to remain unbiased are often unrealistic or simply unattainable. Recent work explores the effectiveness of counterfactual 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 [31, 66].

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 [24, 55]. 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 [51], 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 [71], but their performance when deployed as recommendation policies is often disappointing in comparison with policy-based methods or even vanilla reward models [31, 59]. Furthermore, the logging policy is not always known before-hand, and even when we do obtain unbiased value estimates we should expect the true obtained reward from acting on them to be disappointing with respect to the estimates – a phenomenon known as "the Optimiser's curse" [73].

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 [4]. Additionally, we show how our proposed framework lifts the Optimiser's Curse and effectively limits post-decision disappointment.

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 [34, 49, 75]. 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 [28, 29, 31, 68], 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 [64]. 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.

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. 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 naive 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.

2 BACKGROUND AND RELATED WORK

We are interested in modelling recommendation systems following the "Batch Learning from Bandit Feedback" (BLBF) paradigm [77]: 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 conditioned on contexts: $\mathsf{P}(A|C,\pi)$, often denoted $\pi(A|C)$. 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 $\boldsymbol{a}$. 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 [59], to a specialised neural network architecture that learns a shared embedding for multi-task learning [47, 79, 84]. In the off-policy or counterfactual setting, we have access to a dataset consisting of logged context-action pairs and their associated rewards: $\mathcal{D} \coloneqq \{(c,a,r)\}$, where $c \sim \mathsf{P}(C)$, $a \sim \pi_0(a|c)$ and $r \sim \mathsf{P}(R|C,A)$. Here, r represents the reward that the system obtained from recommending a to c. In the general case this reward can be binary (e.g. clicks), real-valued (e.g. dwell time), or higher-dimensional to support multiple objectives (e.g. fairness and relevance) [56, 57]. 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. 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.

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 [12, 21, 27, 65]. 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 [16, 42, 46, 70], and linear item-based models that achieve state-of-the-art performance whilst being highly efficient to compute [9, 11, 32, 60, 74].

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) [39]. 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 [5, 63]. 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 $\widehat{\pi_0}$. 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 [26], self-normalising the weights [34, 78], imposing variance regularisation [52, 77], imitation learning [49] or distributional robustness [19, 72] 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.

$$\widehat{V}_{\text{IPS}}(\pi, \mathcal{D}) = \sum_{(c,a,r) \in \mathcal{D}} r \cdot \frac{\pi(a|c)}{\widehat{\pi_0}(a|c)} \tag{1}$$

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

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 $\mathcal{D}$. Value-based counterfactual estimators do not rely on a model of the logging policy, but rather learn a model for the context-specific 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:

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

As a consequence, we can directly obtain a decision rule from the reward estimates and train the reward estimator on all available data [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 [8, 80], 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, 17].

A unifying family of doubly robust methods aims to marry these two types of approaches in an attempt to get the best of both worlds [13]. Recent advances in doubly robust learning typically optimise the trade-off between DM and IPS [76], optimise the reward model to minimise the overall variance of the estimator [18], or transform the IPS weights to minimise bounds on the expected error of the estimate [75]. Nevertheless, the performance of the reward model remains paramount for doubly robust approaches to attain competitive performance [28].

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. 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 [9]. In the top-1 use-case we consider with the additional independence assumption between current and future iterations, 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 [48]. Xin et al. adopt a Q-learning perspective to deal with sequential recommendation tasks, exploiting both self-supervised (*organic*) and reinforcement (*bandit*) signals [82]. Analogously, Sakhi et al. propose a probabilistic latent model that combines organic and bandit signals in a Bayesian value-based manner [68]. 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 [31]. 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 [41, 54]. Successful methods use variants of Thompson sampling [7, 14, 53] or confidence bounds [40]; recent work benchmarks a number of different exploration approaches to predict clicks on advertisements when the reward model is parameterised as a neural network [23]. 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 [7, 40]. The Bootstrap principle [15], its extensions [62] (originally proposed in the context of Q-learning), and Monte Carlo Dropout [20] 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 [23]. Finally, other recent work shows promising results in inferring model uncertainty from neuron activation strength [10]. All these uncertainty estimation methods are complementary to the framework we propose in this paper.

3 METHODOLOGY AND CONTRIBUTIONS

3.1 The Optimiser's Curse in Recommendation

In what follows, we introduce the Optimiser's Curse [73] 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 $\forall i \in \{1, \dots, |\mathcal{A}|\} : \mathbb{E}[\widehat{p}_i | p^*_1, \dots, p^*_{|\mathcal{A}|}] = p^*_i$. 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.[1] 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^* :

$$a_{i^*} = \arg\max_{a_i \in \mathcal{A}} \widehat{r}(a_i). \tag{4}$$

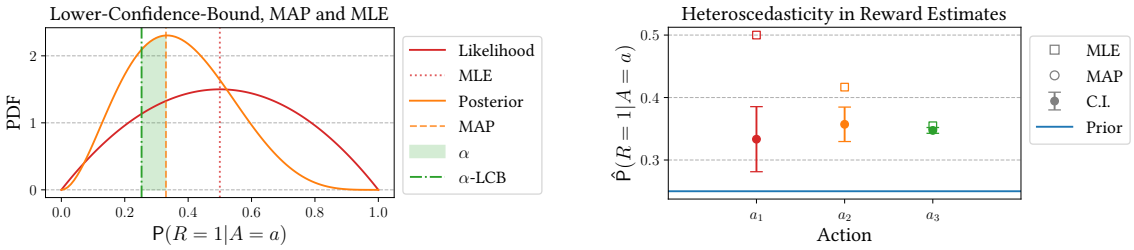
After showing this recommendation to a user, we get to observe a sample from the true reward distribution: $r_{i^*} \sim \text{Bernoulli}(p^*_{i^*})$. Now, the difference between the observed and estimated rewards ($r_{i^*} - \widehat{p}_{i^*}$) can be seen as the *post-decision surprise* we get from 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, 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 [73]. 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 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 [51]. 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 [31, 59, 71].

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

[1] This type of "calibration" of the reward model with respect to the *empirical* reward distribution is often a requirement in computational advertising [24, 55], as a downstream bidding strategy then depends on the reward model.



(a) Different estimates for the reward distribution for action a_1.

(b) Different decision-making strategies for competing actions.

Fig. 1. Illustration of the likelihood, prior and posterior estimates for the toy example in Section 3.2.

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.

As a simple example, consider a Beta-Bernoulli model with three actions [58, §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. Figure 1a shows the resulting likelihood and posterior distributions for action a_1, Figure 1b shows that the MLE prefers action a_1, whereas the MAP estimate prefers a_2 (we will introduce the Lower-Confidence-Bound in Section 3.3). 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. 4), 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: $\mathbb{E}[p_{i^*}^* - \widehat{p}_{i^*}] < 0$ [73].

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.

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 1b. 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 [7], and optimistic extensions to this paradigm are known to further improve performance [53]. 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 [40]. For our example in Figure 1b, this would lead to a ranking of $a_1 > a_2 > a_3$ (which coincides with ranking by the MLE). 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 naive 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" [73]. 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 1b 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 1a. Following our toy example from Figure 1b, 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.

Pessimism in Policy Learning. 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 as well, 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 [22, 26]. 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. The same parallels can be drawn for several other policy learning tricks such as variance regularisation [52, 77], imitation learning [49], distributional robustness [19, 72] and estimator lower bounds [31, 44]. 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 [36, 37, 43, 83]. We point the interested reader towards the work of Jin et al. for more theoretical underpinnings [33].

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 [24, 31, 54, 55, 59]. An ongoing line of research in traditional approaches to recommendation has repeatedly shown the effectiveness of linear models in collaborative filtering tasks as well [9, 32, 60, 69, 74]. Other recent work reports empirical advantages of squared loss over cross-entropy loss [25], which could explain the effectiveness of item-based least-squares models like SLIM [60] and EASE$^{\text{R}}$ [74], 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 5 (omitting a bias-term for brevity).

$$\hat{\mathrm{P}}(R = 1|C = c, A = a, \boldsymbol{\theta}) = \boldsymbol{x}^\intercal \boldsymbol{\theta} = \boldsymbol{c}^\intercal \boldsymbol{\theta}_{|a} \tag{5}$$

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 $i \in \{1, \dots, |\mathcal{A}|\}$. 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:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\min} \left(\|\boldsymbol{X}^\intercal \boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2 \right). \tag{6}$$

The Tikhonov-regularisation in Eq. 6 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 [58]:

$$\boldsymbol{\theta} \sim \mathcal{N}(0, \sigma_x^2), \qquad \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}^\intercal \boldsymbol{\theta}, \sigma_y^2). \tag{7}$$

When $\lambda = \frac{\sigma_y^2}{\sigma_x^2}$, the solution to the ridge regression problem in Eq. 6 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. 8:

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y}, \qquad \hat{\Sigma} = (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}. \tag{8}$$

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. 5, 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 [40], 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:

$$\widehat{\mathrm{P}}_{\mathrm{LCB}}(R = 1|C = c, A = a) = \boldsymbol{x}^\intercal \hat{\boldsymbol{\theta}} - \alpha \sqrt{\boldsymbol{x}^\intercal \hat{\Sigma} \boldsymbol{x}} = \boldsymbol{c}^\intercal \hat{\boldsymbol{\theta}}_{|a} - \alpha \sqrt{\boldsymbol{c}^\intercal \widehat{\Sigma}_{|a} \boldsymbol{c}}. \tag{9}$$

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 $i \in \{1, \dots, |\mathcal{A}|\}$. 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 $\widehat{\mathrm{P}}_{\mathrm{LCB}}$ [81]. 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 mean ($\widehat{\mathrm{P}}_{\mathrm{LCB}}$ vs $\hat{\mathrm{P}}$), yields the optimal deterministic LCB policy:

$$\pi^*_{\mathrm{LCB}}(a|c) = \begin{cases} 1 & \text{if } a = \underset{a' \in \mathcal{A}}{\arg\max}\, \widehat{r}_{\mathrm{LCB}}(a', c), \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

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 [21, 27, 65], which motivates casting recommendation as a bandit learning problem [30]. Recent work either shows empirical success with supervised-to-bandit conversions on organic user-item datasets [48], through live experiments [8, 54, 57], or by adopting open-source simulation environments [31, 68]. To aid in the reproducibility of our work, we make use of the RecoGym simulation environment [64]. 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. We refer the interested reader to the source code of the simulator[2] or the reproducibility appendix of [31] for an overview of the inner workings of the simulation environment. 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?

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 near 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} 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 [66]. Deficient logging policies violate the assumptions made by IPS to yield an unbiased reward estimate [63], 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.

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

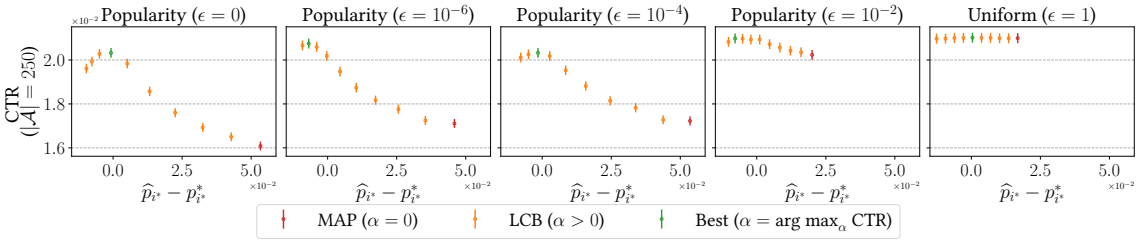


Fig. 2. We evaluate varying degrees of pessimistic decision strategies ($\alpha \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$, corresponding with right- to leftmost measurements in the plots). The x-axis shows the resulting post-decision disappointment with the attained CTR on the y-axis (95% credible intervals). Every plot corresponds to a different amount of randomisation in the logging policy (increasing from left to right). 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.

$$\pi_0(a|c) = \begin{cases} \pi_{\text{pop}}(a|c) & \text{with probability } 1-\epsilon, \\ \pi_{\text{uni}}(a|c) & \text{otherwise,} \end{cases} \qquad \text{where } \pi_{\text{pop}}(a_i|c) = \frac{\boldsymbol{c}_i}{\sum_{j=1}^{|\mathcal{A}|} \boldsymbol{c}_j}, \text{ and } \pi_{\text{uni}}(a|c) = \frac{1}{|\mathcal{A}|}. \tag{11}$$

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 [28, 29, 31, 59, 68], but that we explore a wider range of logging policy randomisation to highlight the effects on naïve reward modelling procedures.

4.1 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 users, leading to approximately 800 000 bandit opportunities in the training set as well as 800 000 online impressions per evaluated policy. We report the average empirical disappointment ($\widehat{p_{i^*}} - p^*_{i^*}$) for both the standard decision-making strategy of taking the MAP action ($\alpha = 0$), and our pessimistic lower-confidence-bound strategy, varying the lower posterior quantile α. Note that the hyperparameter α plays an important role here, 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, this does not guarantee an improvement in the online metrics we care about. For this, we additionally report an estimate of the policy's attained click-through-rate (CTR) in the A/B-test. 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. Figure 2 visualises the results from these experiments.

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,

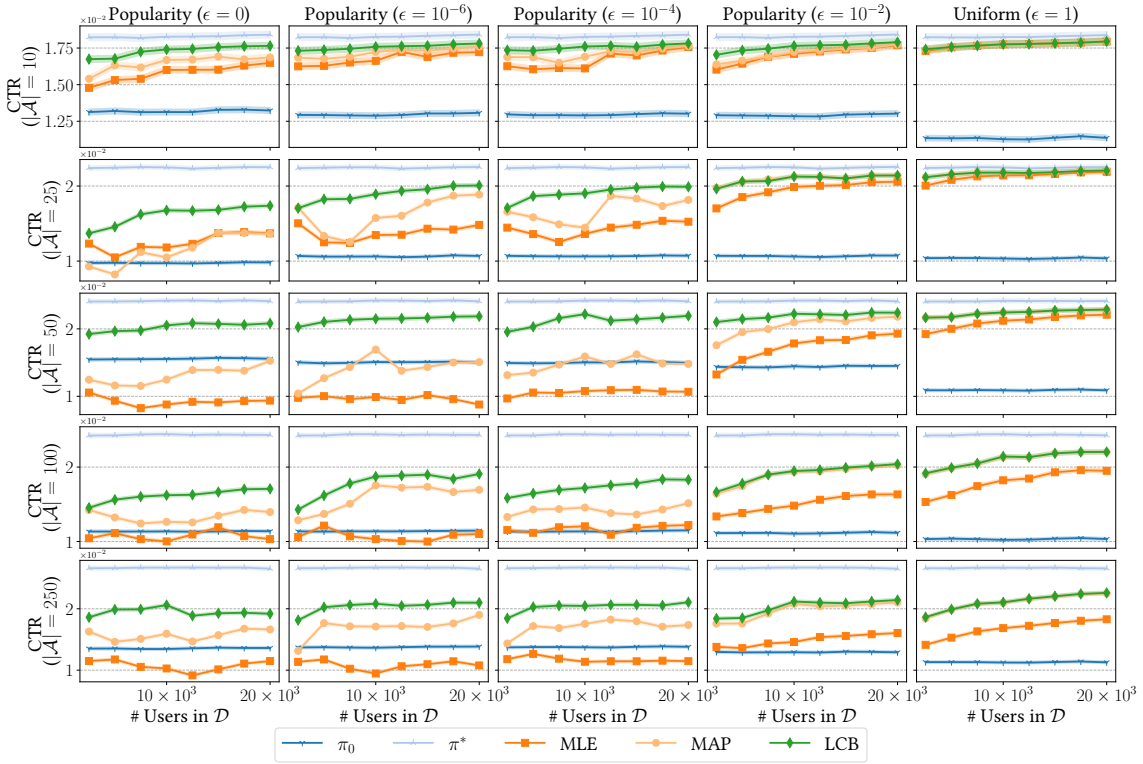


Fig. 3. Experimental results for a range of simulated A/B-tests. The amount of training data is increased over the x-axis, the attained CTR is shown on the y-axis (shaded 95% credible interval). Every column corresponds to a different amount of randomisation in the logging policy (increasing from left 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%.

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 *absolute* surprise 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 [81]. 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.

4.2 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 3. 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^*. Every measurement shown in Figure 3 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 [5], hybrid [31] or doubly robust [13]). We do note that our results are directly comparable to those presented in [31, 59], 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 2, 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 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 our 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 [34, 75, 77]. As a result, methods for counterfactual learning in recommendation are often evaluated in modestly sized action spaces too [31, 59, 68]. 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 [45]. 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 [38, 67], 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 3, 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 [64].

5 CONCLUSION AND FUTURE WORK

In this work, we have advocated in favour of *pessimistic* reward modelling in bandit feedback settings. We have proposed a general framework for sceptic decision-making in off-policy recommendation use-cases, and have shown how to translate uncertainty estimates for ridge regressors into a conservative decision rule. Our proposed method lifts the Optimiser's Curse whilst achieving a significant and robust boost in recommendation performance for a variety of settings. In future work, we wish to investigate whether pessimistic reward predictions can lead to improved doubly robust learning [28], to study the generalisability of our results to larger action spaces, and to investigate the effects of scepticism on the informational value of data collected under such a policy.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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)*.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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*. ACM, 456–461.

[24] 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.

[25] 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]

[26] E. L. Ionides. 2008. Truncated Importance Sampling. *Journal of Computational and Graphical Statistics* 17, 2 (2008), 295–311.

[27] 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.

[28] 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)*.

[29] 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)*.

[30] 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)*.

[31] 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.

[32] 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.

[33] Y. Jin, Z. Yang, and Z. Wang. 2020. Is Pessimism Provably Efficient for Offline RL? arXiv:2012.15085 [cs.LG]

[34] 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)*.

[35] 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.

[36] 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)*.

[37] 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)*.

[38] 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).

[39] 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]

[40] 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.

[41] 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.

[42] 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.

[43] 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)*.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] A. Maurer and M. Pontil. 2009. Empirical Bernstein Bounds and Sample Variance Penalization. *Stat.* 1050 (2009), 21.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] K. P. Murphy. 2021. *Probabilistic Machine Learning: An introduction.* MIT Press.

[59] 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)*.

[60] 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.

[61] 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.

[62] 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.

[63] A. B. Owen. 2013. *Monte Carlo theory, methods and examples.*

[64] 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)*.

[65] 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.

[66] 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*. ACM, 965–975.

[67] 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]

[68] 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.

[69] 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).

[70] 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.

[71] 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.

[72] 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)*.

[73] 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.

[74] H. Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference (WWW '19)*. ACM, 3251–3257.

[75] 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.

[76] 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.

[77] 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.

[78] A. Swaminathan and T. Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *Advances in Neural Information Processing Systems*. 3231–3239.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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)*.

[84] 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.