Incentivizing High-Quality Content in Online Recommender Systems

Xinyan Hu
University of California, Berkeley, CA, USA

Meena Jagadeesan[1]
Stanford University, CA, USA

Michael I. Jordan
University of California, Berkeley, CA, USA
Inria and École Normale Supérieure, Paris, France

Jacob Steinhardt
University of California, Berkeley, CA, USA
Transluce, San Francisco, CA, USA

Abstract

In content recommender systems such as TikTok and YouTube, the platform's recommendation algorithm shapes content producer incentives. Many platforms employ online learning, which generates intertemporal incentives, since content produced today affects recommendations of future content. We study the game between producers and analyze the content created at equilibrium. We prove that standard online learning algorithms, such as Hedge and EXP3, unfortunately incentivize producers to create low-quality content, where producers' effort approaches zero in the long run for typical learning rate schedules. Motivated by this negative result, we design learning algorithms that incentivize producers to invest high effort and achieve high user welfare. At a conceptual level, our work illustrates the unintended impact that a platform's learning algorithm can have on content quality and introduces algorithmic approaches to mitigating these effects.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Information systems → Recommender systems; Theory of computation → Online learning algorithms

Keywords and phrases recommender systems, content quality, producer incentives, online learning, algorithmic game theory, Stackelberg games

Digital Object Identifier 10.4230/LIPIcs.FORC.2026.15

Related Version *Full Version*: `https://arxiv.org/abs/2306.07479` [22]

Acknowledgements Xinyan Hu and Meena Jagadeesan contributed equally to this work.

1 Introduction

Digital content recommendation platforms have given rise to a creator economy [32], where content producers strategically adapt the quality of the content they create in response to the platform's algorithm. Although incentives for content creation have been studied primarily in a static setting [4, 6, 24, 21], *learning* introduces an intertemporal aspect to producer incentives. This is because the content produced today affects the recommendations of future content via learning updates of the recommendation algorithm.

The resulting intertemporal incentives can impact how much effort producers put into creating high-quality content. If learning is too slow, then good content will not be rewarded until after a long delay; as a result, myopic or near-myopic producers may never put in

[1] Work done while at UC Berkeley.

significant effort, and their content will be low-quality. The learning dynamics may even incentivize producers to vary their effort over time. For instance, producers who frequently upload new content (e.g., influencers on YouTube and TikTok, or artists who release many albums) can strategically modulate their effort in real-time.

Motivated by the above phenomena, we study a game-theoretic formulation of the producers' time-varying incentives arising from the recommendation platform's learning algorithm. We model the platform's learning algorithm as an online learning or bandit algorithm [20], which recommends a single producer at each time step. To model producer incentives, we consider a game between producers who compete for recommendations and strategically choose what content to create over time, based on the platform's algorithm. Our focus is on how the platform's algorithm affects the equilibrium content quality (where quality is measured in terms of both producer effort and user welfare), as well as how the equilibrium quality changes over time. Our results, summarized in Table 1, are as follows.

First, we show that standard online learning algorithms (specifically, `LinHedge` and `LinEXP3` [30]) incentivize producers to create low-quality content. We upper bound the producer effort as well as the user welfare at equilibrium for both `LinHedge` (Theorem 1) and `LinEXP3` (Theorem 2). This upper bound shows that under typical learning rate schedules the quality approaches zero as time goes to infinity. The intuition is that if the platform's learning update is too slow, producers will put in low effort due to temporal discounting.

Motivated by these negative results, we design alternative algorithms that incentivize high-quality content. These algorithms draw inspiration from content moderation [16], and demonstrate the benefits of "punishing" creators for producing low-quality content. First, we show that a simple modification to `LinHedge` achieves $\Theta(1)$ producer effort at equilibrium even as time goes to infinity (Theorem 4). Next, we turn to user welfare, which is a more challenging objective that additionally requires content to align with the preferences of a heterogeneous population of users. Our algorithm `PunishUserUtility` (Algorithm 3) takes advantage of producer specialization to achieve a higher user welfare (Theorem 7) and even matches the optimal welfare in limiting cases (Corollary 10).

En route to proving these results, we develop and leverage technical tools for learning in strategic environments which could be of broader interest. First, we develop tools that upper bound the change in the behavior of an online learning algorithm as the reward functions change. This enables us to compare the probability of an arm being pulled across two settings of rewards. Second, we show an algorithm can leverage "punishment" at each time step to shape how agents behave, which enables the algorithm to achieve its own objective (Section 4 and Section 5). The punishment criteria need to be carefully set, especially due to the delayed effect of punishment on non-myopic agents.

1.1 Related Work

Our work connects to research threads on societal effects of recommender systems, learning in Stackelberg games, and online learning algorithms.

Societal effects of recommender systems. In an emerging line of work, [4, 6, 24, 21] have proposed game-theoretic models for producer incentives for content creation in recommender systems. We build on the D-dimensional model of [24] where producers select the quality level (magnitude) of their content in addition to the genre (direction). Several recent works introduce learning into the framework in different ways. For example, [5, 36] incorporate that producers running learning algorithms to converge to an equilibrium. [13] incorporate

Table 1 Bounds on content quality at equilibrium at each time step $t < T$. We measure quality by producer effort $\|p^t\|$ (top row) and user welfare $\mathbb{E}[\langle u^t, p^t_{A^t}\rangle]$ (bottom row). For the standard algorithms `LinEXP3` or `LinHedge`, the equilibrium quality is small for large t. `PunishLinDirectionHedge` (our first incentive-aware algorithm) leads to high producer effort. `PunishUserUtility` (our more sophisticated incentive-aware algorithm) generates high user welfare.

Algorithm	LinHedge ($\eta_t = \tilde{O}(\frac{1}{\sqrt{t}})$)	LinEXP3 ($\eta_t = \tilde{O}(\frac{1}{\sqrt{t}})$)	PunishLinDirectionHedge (Algorithm 2)	PunishUserUtility (Algorithm 3)
Producer Effort	$O\left(\left(\frac{1}{\sqrt{t}}\right)^{\frac{1}{c-1}}\right)$ (Theorem 1)	$O\left(\left(\frac{P}{\sqrt{t}}\right)^{\frac{1}{c}}\right)$ (Theorem 2)	$\Theta\left(\left(\frac{1}{P}\right)^{\frac{1}{c}}\right)$ (Theorem 5)	N/A
User Welfare	$O\left(\left(\frac{1}{\sqrt{t}}\right)^{\frac{1}{c-1}}\right)$ (Theorem 1)	$O\left(\left(\frac{P}{\sqrt{t}}\right)^{\frac{1}{c}}\right)$ (Theorem 2)	$\Theta\left(\left(\frac{1}{P}\right)^{\frac{1}{c}} \|\mathbb{E}[u]\|\right)$ (Corollary 6)	$\Omega\left(\|\mathbb{E}[u]\|\right)$ (1 as $c \to \infty$) (Section 5)

that the platform runs retraining, but assumes that both the platform and producers are myopic; in contrast, our work captures that producers non-myopically react to the platform's learning algorithm.

Our work closely relates to a line of work [15, 14, 28] that aims to incentivize the creation of high-quality content. These works assume each producer creates a single digital good with fixed quality across time; thus, the platform's learning problem is cast within the stochastic bandit framework. In contrast, we allow producers to create new content at each time step and thus can vary their quality over time, which leads our problem to an *adversarial* bandit framework. Another difference is we allow for D-dimensional content vectors which enables us to capture producer specialization, whereas they focus only on 1-dimensional content quality.

Other aspects studied in this research thread include *participation decisions by creators* [29, 7], *incentivizing exploration of bandit algorithms* [26], *strategic user behavior in recommender systems* [19], and *preference shaping* [8, 11].

Learning in Stackelberg games. Our model can be viewed as a Stackelberg game where the leader (platform) wants to learn the best responses of competing, non-myopic content followers (producers) from repeated interactions. There has been a rich literature on learning in Stackelberg games, including security games [3, 17], strategic classification [18, 12, 9, 40, 17], and contract theory [39]. However, most of these works focus on a single follower who (myopically) best-responds to the leader at each round or makes a gradient update, while our work allows for non-myopic followers who compete with each other (see Section 2.1 for additional discussion).

One notable exception is [17], who consider non-myopic agents. Interestingly, their approach – delaying the use of feedback from the followers in learning – fails to incentivize high-quality content in our setting. The fundamental difference is that [17] aim to disincentivize gaming whereas we aim to incentivize investment. Moreover, while [17] focus on a single follower, our work studies competition between followers. [10] also studies non-myopic agents, but focuses on the case of a (single) *fully* non-myopic follower.

Online learning algorithms. We build on the rich literature on *online learning algorithms* [20]. Most relevant to our work is the *adversarial linear contextual bandits framework* [30], which extends the stochastic linear contextual bandit setup of [2] and [1] to adversarial losses. In comparison to classical adversarial contextual bandits [31, 33], this framework places a linear

structure on the losses. Our work builds on this framework and employs the linear variants of EXP3 proposed by Neu and Olkhovskaya [30] that achieve $O(\sqrt{T})$ regret. We also consider a full-information version of this framework.

2 Model

To study dynamic interactions between the platform, producers, and users, we extend the model of [24] to an online learning framework. Users and content are embedded in the D-dimensional space $\mathbb{R}^D_{\geq 0}$, and a user u's utility for consuming content p is given by the inner product $\langle u, p\rangle$. There is a user distribution $\mathcal{D}$ over $\mathbb{R}^D_{\geq 0}$, where we assume for simplicity that the vectors in the support are all normalized unit vectors (i.e. $\|u\|_2 = 1$ for all $u \in \text{supp}(\mathcal{D})$). There are $P \geq 1$ producers. In our dynamic setup, there are T time steps, and each producer $j \in [P]$ chooses a sequence of content vectors, $\mathbf{p}_j = (p_j^1, \ldots, p_j^T) \in \left(\mathbb{R}^D_{\geq 0}\right)^T$, where $p_j^t \in \mathbb{R}^D_{\geq 0}$ denotes the content created at time step t. We describe the details of our model below, deferring further discussion of the model to Section 2.1.

Online learning framework. We model the platform's learning task using adversarial linear contextual bandits [30], where arms are producers and contexts are users. In particular, at each time step t, a single user $u^t \sim \mathcal{D}$ arrives on the platform and the platform sees the vector u^t. Before observing any information about the producer content vectors p_j^t for $j \in [P]$, the platform assigns the user u^t to the producer $A^t \in [P] \cup \{\perp\}$ according to some online learning algorithm ALG. The platform receives a reward equal to the user utility $\langle p_{A^t}^t, u^t\rangle$ and gets some feedback from the user about the producers' content that can be used for future recommendations.

The platform's recommendations are based on the history of feedback in the online learning framework. We study both the *full-information* setting, where the platform observes all of the vectors $p_1^{t-1}, \ldots, p_P^{t-1}$ at the end of time step $t-1$, and the *bandit* setting, where the platform only observes feedback of the form $\langle p_{A^t}^{t-1}, u^{t-1}\rangle$. For full-information, the platform's algorithm has access to the history $\mathcal{H}_{\text{full}}^{t-1} = \left\{(j, \tau, u^\tau, A^\tau, p_j^\tau) \mid 1 \leq \tau \leq t-1, j \in [P]\right\}$Footnote 2 at time step t. For bandits, the platform's algorithm has access to the history $\mathcal{H}_{\text{bandit}}^{t-1} = \{(\tau, u^\tau, A^\tau, \langle p_{A^\tau}^\tau, u^\tau\rangle) \mid 1 \leq \tau \leq t-1\}$. [2]

Platform algorithms. Some of the algorithms that we analyze are based on variants of Hedge and EXP3 [20], adapted to the adversarial contextual linear setting [30]. We describe these algorithms (LinHedge and LinEXP3) below for completeness. The parameters of these algorithms are the learning rate schedule $\{\eta_t\}_{t=1}^T$ and the exploration parameter $\gamma \in (0, 1)$.

- LinHedge is the full-information analogue of LinEXP3 algorithm introduced in [30]. Given learning rate parameters $\eta_1 \geq \ldots \geq \eta_T > 0$ and $\gamma \in (0, 1)$, it selects the arm A^t from the following probability distribution:

$$\mathbb{P}[A^t = j \mid \mathcal{H}_{\text{full}}^{t-1}, u^t] = (1-\gamma)\frac{e^{\eta_t \sum_{\tau=0}^{t-1}\langle u^t, p_j^\tau\rangle}}{\sum_{j' \in [P]} e^{\eta_t \sum_{\tau=0}^{t-1}\langle u^t, p_{j'}^\tau\rangle}} + \frac{\gamma}{P}.$$

  The probability $\mathbb{P}[A^t = \perp \mid \mathcal{H}_{\text{full}}^{t-1}, u^t]$ is always equal to zero.

- LinEXP3 [30] operates in the bandit setting. It uses the same selection rule as LinHedge, except that it replaces p_j^τ with a non-negative, unbiased estimate $\hat{p}_j^\tau$.

[2] We initialize the history $\mathcal{H}^0$ so that $u^0 \sim \mathcal{D}$ is an arbitrary draw, $A^\tau = 1$, and $p_j^0 = \vec{0}$ for all j.

Producer incentives. After the platform commits to an online learning algorithm ALG, each producer $j \in [P]$ strategically selects $\mathbf{p}_j$ to maximize its own utility for the whole future time horizon, which means they are *non-myopic* players in the game. In particular, the utility of a producer is a discounted cumulative utility across T time steps. At each time step t, the producer j earns utility 1 if they are recommended, and pays a cost of production regardless. We consider production costs of the form $c(p) = \|p\|_2^c$, where $c \geq 1$ is an exponent, and we will use $\|\cdot\|$ to denote $\|\cdot\|_2$ for ease of exposition. If a producer has the discount factor $\beta \in (0, 1)$, then its expected utility is:

$$u_j(\mathbf{p}_j \mid \mathbf{p}_{-j}) = \mathbb{E}\Big[\sum_{t=1}^{T} \beta^t \left(\mathbb{1}[A^t = j] - \|p_j^t\|^c\right)\Big], \tag{1}$$

where $\mathbf{p}_{-j}$ denotes the sequences of content vectors chosen by other producers $j' \neq j$ and the expectation is over the randomness of the algorithm and the user vectors.

For simplicity of analysis, we assume that producers choose their content vectors p_j^t for all t at the start of the game. We thus study the mixed-strategy Nash equilibria of the resulting game between the P producers, where producer $j \in [P]$ has action space is $(\mathbb{R}^D_{\geq 0})^T$ and the utility function u_j.

Performance metrics. We evaluate the platform's algorithm according to two metrics: *producer effort* $\|p_j^t\|$ (Sections 3 and 4), and *user welfare* $\mathbb{E}\left[\langle u^t, p^t_{A^t}\rangle\right]$ (Section 3 and 5).

2.1 Model Discussion

Now that we have formalized our model, we discuss and justify our design choices in greater detail. Our model captures a stylized online content recommendation marketplace and is also an instance of a generalized Stackelberg game, as we describe below.

A stylized online content recommendation marketplace. Our model incorporates key aspects of online recommender systems and the resulting creator economy. For example, our model of users and content as D-dimensional embeddings captures the multi-dimensionality of user preferences and content attributes. In fact, high-dimensional user and content embeddings are implicitly learned by many real world recommender systems, which use *two-tower embedding models* based on matrix factorization [23]) or deep learning variants [37, 35]. Moreover, producers choosing new content embeddings every time step captures that real-world creators frequently upload new content (e.g. to their content channel on YouTube[3]). The cost function $c(p) = ||p||^c$ captures that real-world production costs scale with effort and are also independent of consumption because goods are digital [15, 24]. Finally, the assumption on platform information – i.e., that the platform make recommendations before observing content in the current round – captures that the platform faces a cold-start problem for new content due to its reliance on user feedback to learn about content [34].

However, our model does make several simplifying assumptions for mathematical tractability. For example, in our model, producers select all content at the beginning of the game, which from the online learning perspective means that the producers are *oblivious* adversaries, who cannot adapt to the randomness of the algorithm. Additionally, our model assumes that a single user arrives at every round, but this assumption is made for ease of exposition: our results directly generalize to the case where all users arriving at every round

[3] See https://www.youtube.com/creators/how-things-work/getting-started/.

are drawn from the population. Finally, our negative results (Section 3) focus on standard no-regret learning algorithms such as LinHedge and LinEXP3 (see Section 3.3 for additional discussion). Despite these simplifying assumptions, our model is rich enough to provide insight into how a platform's learning process impacts creator incentives over time.

A generalized Stackelberg game. Our model is a (generalized) Stackelberg game, where the platform is the leader and each producer is a follower. Stackelberg games are a general paradigm with many applications beyond content recommender systems, ranging from security games [25] to power grids [38] to marketing [27]. Our generalized Stackelberg game has several unique features: the leader (platform) is *learning* over time, the followers (producers) are *non-myopic* and take into account future time steps when choosing their actions, and *multiple followers* compete with each other leading to a more complicated equilibrium analysis. Since our work is motivated by online recommender systems, our model places several additional structural assumptions on the game: the follower's action space is D-dimensional, the leader's action is to pick one of the followers, the leader's utility function is linear in the followers' actions, and increasing "effort" is costly to the follower but beneficial to the leader. An interesting direction for future work would be to extend our model and findings to more general Stackelberg games, relaxing the structural assumptions on the utility functions and action spaces.

3 LinHedge/LinEXP3 Induces Low-Quality Content

In this section, we show that, in the presence of producer incentives, standard no-regret learning algorithms lead to poor performance along producer effort and user welfare. We prove that `LinHedge` and `LinEXP3` with a typical learning rate schedule incentivize producers to invest *diminishing effort* $\|p\|$ in the long run. Our results imply the same upper bounds on the *user welfare* $\mathbb{E}[\langle u^t, p^t_{A^t}\rangle]$, since the user utility is always upper bounded by the effort via the Cauchy-Schwarz inequality, so we focus on effort in this section for ease of exposition.

Our bounds apply to *any* mixed-strategy Nash equilibrium in the game between producers (see the full version [22] for a proof of equilibrium existence). We first analyze `LinHedge` (Theorem 1), and then extend our analysis to `LinEXP3` (Theorem 2).

3.1 Upper bounds on producer effort

We show the following upper bounds on producer effort in the full-information setting and bandit setting.

Full-information setting. We establish an upper bound on producer effort at Nash equilibrium if the platform runs `LinHedge`. The bound depends on the time step t, learning rate schedule $\{\eta_t\}_{t\in T}$, discount factor β, exploration parameter γ, and cost function exponent c.

▶ **Theorem 1.** *Let $\beta \in (0,1)$ be the discount factor of producers, $c \geq 1$ be the cost function exponent, and $\mathcal{D}$ be any distribution over users. Suppose that the platform runs `LinHedge` with learning rate schedule $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_T > 0$ and $\gamma \in (0,1)$. At any mixed-strategy Nash equilibrium $(\mu_1, \ldots, \mu_P)$, for any producer $j \in [P]$, any strategy $\mathbf{p}_j \in supp(\mu_j)$ and any time t, the quality $\|p^t_j\|$ is upper bounded by*

$$\|p^t_j\| \leq \left(\eta_{t+1}(1-\gamma)\frac{\beta(1-\beta^{T-t})}{1-\beta}\right)^{\frac{1}{c-1}}.$$ [4]

[4] The bound for $c = 1$ (which is undefined as written) is zero if $\eta_t(1-\gamma)\frac{\beta(1-\beta^{T-t})}{1-\beta} < 1$ and ∞ otherwise.

To interpret the bound, we apply typical learning rate schedules for no-regret algorithms. First, for the standard decaying learning rate schedule $\eta_t = O(1/\sqrt{t})$, Theorem 1 implies $\|p_j^t\| = t^{-\Omega(1)}$. This reveals that the equilibrium quality will be low in the long run and in fact approach zero as $t \to \infty$. For the standard fixed learning rate schedule, $\eta_t = \eta = O(1/\sqrt{T})$,Theorem 1 implies $\|p_j^t\| = T^{-\Omega(1)}$ for each $t \le T$. This similarly vanishes as $T \to \infty$, and is actually even more pessimistic than the bound for the decaying learning rate schedule.

The intuition for why producers create low-quality content is that a small learning rate reduces the impact of a producer's action on their future probability of being recommended. Thus, when the producer utility is discounted, the cost of putting in effort at a given time step outweighs the benefit of being recommended a bit more frequently in the future.

Our bound also illustrates how the discount factor β and the cost function exponent c impact content quality. First, the bound decreases in β, approaching zero as $\beta \to 0$. The intuition is that since content created today affects recommendations only in the future, more myopic producers have less incentive to exert effort at the current round. Second, the bound increases in c, approaching 0 as $c \to 1$ and approaching 1 as $c \to \infty$. The intuition is a larger c makes it cheaper to create content with norm $\|p\| < 1$.

Extension to bandit setting. We establish an analogue of Theorem 1 for `LinEXP3`, which similarly implies that typical learning rate schedules cause low-quality content at equilibrium. In particular, the quality decreases at a rate $t^{-\Omega(1)}$ for $\eta_t = O(1/\sqrt{t})$, and at rate $T^{-\Omega(1)}$ for $\eta_t = O(1/\sqrt{T})$.

▶ **Theorem 2.** *Let $\beta \in (0,1)$ be the discount factor of producers, $c \ge 1$ be the cost function exponent, and $\mathcal{D}$ be any distribution over users. Suppose that the platform runs* ***LinEXP3*** *with learning rate schedule $\eta_1 \ge \eta_2 \ge \cdots \ge \eta_T > 0$ and $\gamma \in (0,1)$. At any mixed-strategy Nash equilibrium $(\mu_1, \ldots, \mu_P)$, for any producer $j \in [P]$, any strategy $\mathbf{p}_j \in supp(\mu_j)$ and any time t, the quality $\|p_j^t\|$ is at most*

$$\left(\eta_{t+1} P(1-\gamma)\frac{\beta^{1+\frac{1}{c}}(1-\beta^{T-t}(T-t+1)+\beta^{T-t+1}(T-t))}{(1-\beta)^{2+\frac{1}{c}}}\right)^{\frac{1}{c}}.$$

The extra factor of P in the bound of Theorem 2, compared to Theorem 1, arises from analyzing the additional randomness in bandits versus the full-information setting. The arm pulled at any time step influences the probability distribution of future arms being pulled, complicating producers' choices. However, this extra P may be an artifact of the analysis rather than a fundamental difference between the two settings.

3.2 Proof techniques

The key technical ingredient in the proofs of Theorems 1 and 2 is a bound on how much a platform's probability of choosing a producer is affected by the producer's choices of content vectors. In particular, we compare the difference in the expected probability that producer j wins the user at time step t if the producer chooses $\mathbf{p}_{j,1}$ versus $\mathbf{p}_{j,2}$, which we denote as M_t. To formalize this, let $A^t(\mathbf{p}_j; \mathbf{p}_{-j})$ be the arm pulled by the algorithm at time t if producer j chooses the vector $\mathbf{p}_j$ and other producers choose the vectors $\mathbf{p}_{-j}$. We show the following bound (proof deferred to the full version [22]):

▶ **Lemma 3.** *Suppose that the platform runs* `LinHedge` *or* `LinEXP3` *with learning rate schedule* $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_T \geq 0$ *and exploration parameter* $\gamma > 0$*. For any choice of* $\mathbf{p}_{-j}$*,* $\mathbf{p}_{j,1}$*, and* $\mathbf{p}_{j,2}$*, the difference*

$$M_t := \mathbb{E}\left[\mathbb{P}[A^t(\mathbf{p}_{j,1}; \mathbf{p}_{-j}) = j \mid \mathcal{H}^{t-1}, u^t]\right] - \mathbb{E}\left[\mathbb{P}[A^t(\mathbf{p}_{j,2}; \mathbf{p}_{-j}) = j \mid \mathcal{H}^{t-1}, u^t]\right]$$

can be upper bounded as follows for any time t*:*

(1) *For* `LinHedge`*, it holds that*

$$M_t \leq (1-\gamma)\eta_t \sum_{\tau=0}^{t-1} \|p_{j,1}^\tau - p_{j,2}^\tau\|.$$

(2) *For* `LinEXP3`*, if* $1 \leq s \leq T$ *is the minimum value such that* $p_{j,1}^s \neq p_{j,2}^s$*,*[5] *then it holds that*

$$M_t \leq (1-\gamma)\eta_t \sum_{\tau=s}^{t-1} \left(\|p_{j,1}^\tau\| + \sum_{j' \neq j} \|p_{j'}^\tau\| \right).$$

Both parts of Lemma 3 bound M_t in terms of the learning rate η_t and the difference between the two content vectors of producer j. For the full-information setting, our bound depends on the norm of the difference summed across time steps. For the bandit setting, the bound depends on the first time step when the vectors differ and sums the norms of all producers' content vectors from that point. We defer the proof of Lemma 3 along with the proofs of Theorems 1 and 2 from Lemma 3 to the full version [22].

3.3 Discussion: Algorithm Choices and Platform Design Insights

Due to the adversarial linear contextual bandit framework underlying the platform's learning task, we assumed that the platform uses a standard no-regret adversarial linear contextual bandit algorithm (`LinHedge` or `LinEXP3`) in this section. The reason that we considered *adversarial* algorithms is that the platform operates in a non-stochastic environment: producers create new content at each time step, so the "arm" rewards can vary over time. The *contextual* aspect captures that different users arrive at each time step, and the *linear* aspect captures that user utility is linear in the user's context and content vector. We focused on *no-regret algorithms*, because regret minimization is a standard objective in bandit problems.

While real-world recommendation platforms are unlikely to directly deploy these specific algorithms, we expect the conceptual insights of our results apply more broadly to platform design. In particular, `LinHedge` and `LinEXP3` capture a stylized learning process of a *non-incentive-aware platform* that does not take into account how its learning algorithm impacts content creation. The key conceptual insight from our analysis is *if the learning algorithm reacts too slowly to producer actions*, then producers are disincentivized from investing effort in content quality. This suggests that the platform needs to react more quickly and directly to producer actions than is required in typical, non-incentive-aware learning environments. The algorithms that we design in Sections 4 and 5 below build on this incentive-aware design principle.

[5] For $t \leq s$, note that $M_t = 0$ since the platform behavior prior to seeing the producer's content vector at time step s is unaffected by the producer's choice of content vector.

4 Simple Approaches to Incentivizing Producer Effort

Motivated by the negative results in Section 3 for `LinHedge` and `LinEXP3`, we design algorithms that incentivize producers to create higher-quality content.[6]

In this section, we focus on incentivizing high producer effort (we defer incentivizing high user welfare to Section 5). We design a simple modification to `LinHedge` – based on *punishing* producers who invest too little effort – which guarantees non-vanishing effort from producers at equilibrium. Our punishment-based approach is inspired by *content moderation* [16]. However, an interesting difference is that while content moderation typically targets producers who post offensive or dangerous content, our algorithm removes producers who post low-quality content.

First, we instantiate this idea for $D = 1$ (Algorithm 1; Theorem 4). Then, we extend it to $D > 1$ (Algorithm 2; Theorem 5) and show the algorithm achieves a partial recovery of user welfare (Corollary 6). Our results focus on the full-information setting.

4.1 Simple case: Dimension $D = 1$

We start with the case of $D = 1$, where the producer picks a single scalar $p \geq 0$ equal to their effort. We design `PunishHedge` (Algorithm 1) building on `Hedge` (the one-dimensional version of `LinHedge`) with the following punishment principle: stop recommending producers if they ever create content with quality below a certain threshold q. More specifically, at each time t, we maintain an "active" producer set $\mathcal{P}^t$ and only consider producers from this set. If a producer j chooses content with effort p_j^t below q, then the producer is removed from $\mathcal{P}^s$ for all remaining time $s > t$.

Algorithm 1 `PunishHedge` (for dimension $D = 1$).

Require: punishment threshold q, learning rate schedule $\{\eta_t\}_{1 \leq t \leq T}$, exploration rate γ, time horizon T, number of producers P

Initialize $\mathcal{P}^1 = [P]$ (active producers).

Initialize $A^0 = 1$, $u^0 = 1$, and $p_j^0 = \vec{0}$ for all $j \in \mathcal{P}^1$, and let $\mathcal{H}_{\text{full}}^0 = \left\{(j, \tau, u^\tau, A^\tau, p_j^\tau) \mid \tau = 0, j \in [P]\right\}$.

for $1 \leq t \leq T$ **do**

  If $\mathcal{P}^t = \emptyset$, let $A^t = \perp$.

  Otherwise, let A^t be defined so that:

$$\mathbb{P}[A^t = j \mid \mathcal{H}_{\text{full}}^{t-1}, u^t] = \begin{cases} 0 & \text{if } j \notin \mathcal{P}^t \\ (1-\gamma)\dfrac{e^{\eta_t \sum_{\tau=0}^{t-1} p_j^\tau}}{\sum_{j' \in \mathcal{P}^t} e^{\eta_t \sum_{\tau=0}^{t-1} p_{j'}^\tau}} + \dfrac{\gamma}{|\mathcal{P}^t|} & \text{if } j \in \mathcal{P}^t. \end{cases}$$

  Let $\mathcal{H}_{\text{full}}^t = \left\{(j, \tau, u^\tau, A^\tau, p_j^\tau) \mid 1 \leq \tau \leq t, j \in \mathcal{P}^t\right\}$

  Let $\mathcal{P}^{t+1} = \mathcal{P}^t \setminus \left\{j \in \mathcal{P}^t \mid p_j^t < q\right\}$

end for

The following theorem shows `PunishHedge` with the learning rate schedule $\eta = O(\frac{1}{\sqrt{T}})$ induces constant equilibrium effort over time, as long as the punishment threshold q is carefully chosen:

[6] Given that the small learning rate was the driver of low-quality content in Section 3, increasing the learning rate of `LinHedge` might seem like a straightforward fix. However, this naive approach introduces challenges involving even the existence of a (symmetric pure-strategy) equilibrium (see the full version [22]).

▶ **Theorem 4.** *Let $\beta \in (0,1)$ be the discount factor of producers, $D = 1$ be the dimension, and $c \geq 1$ be the cost function exponent. Suppose that the platform runs* `PunishHedge` *with learning rate schedule $\eta = O(\frac{1}{\sqrt{T}})$ and quality threshold $q = \left(\frac{\beta}{P}\right)^{1/c}(1-\epsilon)$ for any $\epsilon \in (0,1)$. For T sufficiently large, there is a unique mixed-strategy Nash equilibrium, comprised of symmetric pure strategies that we denote by $\mathbf{p} = \mathbf{p}_1 = \ldots = \mathbf{p}_P$. The content quality is $|p^t| = 0$ for $t = T$ and*

$$|p^t| = \left(\frac{\beta}{P}\right)^{1/c}(1-\epsilon)$$

for $1 \leq t \leq T-1$.

As a direct corollary, the user welfare $\mathbb{E}\left[\langle u^t, p^t_{A^t}\rangle\right]$ is also high since $\langle u^t, p^t_j\rangle = u^t \cdot p^t_j = p^t_j$ when $D = 1$.

Setting the punishment threshold q is a key ingredient of the algorithm design. This is because if q is set too high, then discounted producers might opt for punishment rather than bearing higher production costs for future recommendations. Since removal from the active set occurs one step after producing low-quality content, this delay further incentivizes producers to accept punishment. In light of this, Theorem 4 sets q to be at the break-even point where the benefit of staying in the active set just outweighs the cost of efforts of producing high-quality content. We defer the proof to the full version [22].

4.2 Extension to dimension $D > 1$

We next extend the algorithm and analysis to any $D \geq 1$ case. Here, the norm $\|p\|$ captures the producer's effort for creating content $p \in \mathbb{R}^D_{\geq 0}$. At a high level, we convert this D-dimensional problem into a one-dimensional problem by specifying a *direction criterion* g, and then punishing producers if their effort is less than our threshold *or* if their direction differs from g. We formalize this algorithm in `PunishLinDirectionHedge` (Algorithm 2).

Algorithm 2 `PunishLinDirectionHedge` (for $D \geq 1$).

Require: punishment threshold q, direction criterion g, learning rate schedule $\{\eta_t\}_{1\leq t\leq T}$, exploration rate γ, time horizon T, number of producers P

1: Initialize $\mathcal{P}^1 = [P]$
2: Initialize $u^0 \sim \mathcal{D}$, $A^0 = 1$, and $p^0_j = \vec{0}$ for all $j \in \mathcal{P}^1$, and let $\mathcal{H}^0_{\text{full}} = \left\{(j, \tau, u^\tau, A^\tau, p^\tau_j) \mid \tau = 0, j \in [P]\right\}$.
3: **for** $1 \leq t \leq T$ **do**
4: Let $u^t \sim \mathcal{D}$ be the arriving user at time t.
5: If $\mathcal{P}^t = \emptyset$, let $A^t = \perp$.
6: Otherwise, let A^t be defined so that $\mathbb{P}[A^t = j \mid \mathcal{H}^{t-1}_{\text{full}}, u^t]$ is equal to:

$$\mathbb{P}[A^t = j \mid \mathcal{H}^{t-1}_{\text{full}}, u^t] = \begin{cases} 0 & \text{if } j \notin \mathcal{P}^t \\ (1-\gamma)\dfrac{e^{\eta_t \sum_{\tau=0}^{t-1}\langle u^t, p^\tau_j\rangle}}{\sum_{j'\in\mathcal{P}^t} e^{\eta_t \sum_{\tau=0}^{t-1}\langle u^t, p^\tau_{j'}\rangle}} + \dfrac{\gamma}{|\mathcal{P}^t|} & \text{if } j \in \mathcal{P}^t. \end{cases}$$

7: Let $\mathcal{H}^t_{\text{full}} = \left\{(j, \tau, u^\tau, A^\tau, p^\tau_j) \mid 1 \leq \tau \leq t, j \in \mathcal{P}^t\right\}$
8: Let $\mathcal{P}^{t+1} = \mathcal{P}^t \setminus \left\{j \in \mathcal{P}^t \mid \|p^t_j\| < q \text{ or } \frac{p^t_j}{\|p^t_j\|} \neq g\right\}$
9: **end for**

Regardless of the direction criterion g, `PunishLinDirectionHedge` matches the bound on producer effort from Theorem 4 (proof deferred to the full version [22]).

▶ **Theorem 5.** *Let $\beta \in (0,1)$ be the discount factor of producers, $D \geq 1$ be the dimension, and $c \geq 1$ be the cost function exponent. Suppose that the platform runs `PunishLinDirectionHedge` with fixed learning rate schedule $\eta = O(\frac{1}{\sqrt{T}})$, quality threshold $q = \left(\frac{\beta}{P}\right)^{1/c}(1-\epsilon)$ for any $\epsilon \in (0,1)$ and any direction criterion $g \in \mathbb{R}^D_{\geq 0}$ satisfying $\|g\| = 1$. For T sufficiently large, there is a unique mixed-strategy Nash equilibrium, comprised of symmetric pure strategies that we denote by $\mathbf{p} = \mathbf{p}_1 = \ldots = \mathbf{p}_P$. The content quality is $\|p^t\| = 0$ for $t = T$ and*

$$\|p^t\| = \left(\frac{\beta}{P}\right)^{1/c}(1-\epsilon)$$

for $1 \leq t \leq T-1$. Moreover, the vector p^t points in the direction of g for $1 \leq t \leq T-1$.

4.3 Partial recovery of user welfare

While we focused on *producer effort* $\|p^t\|$, our results also enable partial recovery of the *user welfare* $\mathbb{E}[\langle u^t, p^t_{A^t}\rangle]$. If we take g to the average user direction, `PunishLinDirectionHedge` guarantees nonvanishing user welfare at equilibrium for $1 \leq t \leq T-1$, as the following corollary shows.

▶ **Corollary 6.** *Assume the same setup as Theorem 5, but where direction criterion is taken to be $g = \frac{\mathbb{E}[u]}{\|\mathbb{E}[u]\|}$. For T sufficiently large, the user welfare at equilibrium is equal to $\mathbb{E}[\langle u^t, p^t_{A^t}\rangle] = 0$ for $t = T$ and*

$$\mathbb{E}[\langle u^t, p^t_{A^t}\rangle] = \left(\frac{\beta}{P}\right)^{1/c}(1-\epsilon)\cdot\|\mathbb{E}_{\mathcal{D}}[u]\| \tag{2}$$

for $1 \leq t \leq T-1$.

However, the welfare bound in (2) has suboptimal dependence on both the number of producers P and the user distribution $\mathcal{D}$.

- First, the bound has a $1/P$ factor, leading to poor bounds for a large number of producers. This factor arises because all producers compete along the same direction g: as a result, they each gets only $1/P$ of the reward and puts in correspondingly little effort.
- Second, the bound has a $\|\mathbb{E}_{\mathcal{D}}[u]\|$ factor, which is poor in high dimensions with heterogeneous users. For instance, it is $1/\sqrt{D}$ if users are uniformly distributed in D dimensions.

In the next section, we will introduce a new algorithm that fixes both of these issues and achieves higher welfare.

5 Incentivizing High User Welfare

We now address the suboptimality of the welfare bound for `PunishLinDirectionHedge` and develop algorithms that incentivize high user welfare $\mathbb{E}[\langle u^t, p^t_{A^t}\rangle]$. `PunishUserUtility` (Algorithm 3), which encourages different producers to specialize their content in different directions (Section 5.1). `PunishUserUtility` can significantly beat the welfare bound from the previous section (Sections 5.2- 5.3), and even achieves optimal welfare in limiting cases (Corollary 10).

5.1 Algorithm and equilibrium

Our algorithm, `PunishUserUtility`, sets individualized criteria $\bar{\mathbf{p}} = (\bar{p}_j)_{j=1}^P$ for each producer and uses these criteria both to punish producers and to determine recommendations. To instantiate punishment, the algorithm stops recommending a producer j if their content p_j^t does not meet the user-specific utility requirement $\langle u, \bar{p}_j \rangle$ for any user u. (This utility-based punishment differs from the effort-based punishment for `PunishHedge`.) To determine recommendations, the algorithm chooses the producer j from the set of active producers whose individualized criterion $\bar{p}_j$ maximizes the arriving user's utility. The recommendation step is thus based entirely on the criteria $\bar{\mathbf{p}}$ and not on the content created by producers at each round.

Algorithm 3 `PunishUserUtility`.

Require: number of producers P, individualized criteria $\bar{\mathbf{p}} = (\bar{p}_j)_{j=1}^P$, time horizon T, user distribution $\mathcal{D}$

Initialize $\mathcal{P}^1 = [P]$

Initialize $u^0 \sim \mathcal{D}$, $A^0 = 1$, and $p_j^0 = \vec{0}$ for all $j \in \mathcal{P}^1$, and let $\mathcal{H}_{\text{full}}^0 = \{(j, \tau, u^\tau, A^\tau, p_j^\tau) \mid \tau = 0, j \in [P]\}$.

for $1 \le t \le T$ **do**

  Let $u^t \sim \mathcal{D}$ be the arriving user at time t.

  If $\mathcal{P}^t = \emptyset$, let $A^t = \perp$.

  Otherwise, let A^t be defined so that

$$\mathbb{P}[A^t = j \mid \mathcal{H}_{\text{full}}^{t-1}, u^t] = \frac{\mathbb{1}\left[j = \arg\max_{j' \in [P^t]} \langle u^t, \bar{p}_{j'} \rangle\right]}{\left|\arg\max_{j' \in [P^t]} \langle u^t, \bar{p}_{j'} \rangle\right|} \tag{3}$$

  Let $\mathcal{H}_{\text{full}}^t = \left\{(j, \tau, u^\tau, A^\tau, p_j^\tau) \mid 1 \le \tau \le t, j \in \mathcal{P}^t\right\}$

  Let $\mathcal{P}^{t+1} = \mathcal{P}^t \setminus \left\{j \in \mathcal{P}^t \mid \exists u \in \text{supp}(\mathcal{D}), \text{s.t.} \langle u, p_j^t \rangle < \langle u, \bar{p}_j \rangle\right\}$

end for

We set the individualized criteria $\bar{\mathbf{p}}$ to maximize user welfare while guaranteeing that producers prefer to avoid punishment:

$$\begin{aligned} &W(c, \beta, \mathcal{D}, P) := \max_{p_1, \ldots, p_P} \mathbb{E}\Big[\max_{j \in [P]} \langle u, p_j \rangle\Big] \\ &\text{s.t. } \|p_j\| \le \beta^{\frac{1}{c}} \mathbb{E}\left[\frac{\mathbb{1}\left[j = \arg\max_{j' \in [P]} \langle u, p_{j'} \rangle\right]}{\left|\arg\max_{j' \in [P]} \langle u, p_{j'} \rangle\right|}\right]^{\frac{1}{c}}, \forall j \in [P]. \end{aligned} \tag{4}$$

In this optimization problem, the objective is to maximize expected user welfare when each user is matched to the content that maximizes its utility. The constraint limits the production cost, guaranteeing that producers prefer creating high-quality content over facing punishment.

When we set the criteria based on the optimal solution to (4), we can lower bound the equilibrium user welfare of `PunishUserUtility`.

▶ **Theorem 7.** *Let $\beta \in (0, 1)$ be the producer discount factor and $c \ge 1$ be the cost function exponent. Suppose that the platform runs `PunishUserUtility` with individualized criteria $\bar{\mathbf{p}}$, where $\bar{\mathbf{p}}$ is $(1 - \epsilon)$ times an optimal solution of (4). Then, there exists a pure strategy Nash equilibrium. Furthermore, for all $t \in [T - 1]$, at any (mixed-strategy) Nash equilibrium, the user welfare $\mathbb{E}\left[\langle u^t, p_{A^t}^t \rangle\right]$ is at least $(1 - \epsilon) \cdot W(c, \beta, P, \mathcal{D})$.*

Proof sketch. The constraint in (4) guarantees that producers weakly prefer to stay in the active set over being punished at equilibrium. The slack $\epsilon > 0$ guarantees that this preference is *strict*. Altogether, we achieve user welfare $(1-\epsilon) \cdot W(c, \beta, \mathcal{D}, P)$, since the objective function in (4) is exactly the user welfare guaranteed by the individualized criteria. A formal proof is given in the full version [22]. ◀

Theorem 7 shows that the welfare of `PunishUserUtility` as $\epsilon \to 0$ is at least $W(c, \beta, P, \mathcal{D})$. To analyze the welfare of `PunishUserUtility`, it thus suffices to analyze $W(c, \beta, P, \mathcal{D})$.

In the following subsections, by analyzing the welfare $W(c, \beta, P, \mathcal{D})$, we show that `PunishUserUtility` addresses the two shortcomings of `PunishLinDirectionHedge` described in Section 4.3. More specifically, we show that the welfare $W(c, \beta, P, \mathcal{D})$ achieves an improved dependence on the number of producers P (Section 5.2) and the user distribution $\mathcal{D}$ (Section 5.3).

5.2 Welfare analysis: Improved dependence on P

We show that the welfare of `PunishUserUtility` achieves a better dependence on the number of producers P compared to the welfare of `PunishLinDirectionHedge`. In particular, in contrast to Equation (2), `PunishUserUtility` achieves welfare that is independent of P:

▶ **Proposition 8.** *For $W(c, \beta, \mathcal{D}, P)$ as defined in (4), the following lower bound holds: $W(c, \beta, \mathcal{D}, P) \geq \beta^{\frac{1}{c}} \|\mathbb{E}[u]\|$.*

The bound in Proposition 8 improves over the welfare bound of `PunishLinDirectionHedge` ($\left(\frac{\beta}{P}\right)^{\frac{1}{c}} \|\mathbb{E}[u]\|$, Theorem 5) by a factor of $P^{1/c}$. Setting the criteria $(\bar{p}_j)_{j \in [P]}$ equal to zero for all but one producer already achieves the bound in Proposition 8: a single producer wins all of the users, which weakens the constraint in (4). This naive setting of the criteria achieves the lower bound for the welfare of `PunishUserUtility` in Proposition 8; we defer the full proof to the full version [22].

The economic intuition is that reducing homogeneous competition can enable greater effort investment. In particular, while `PunishLinDirectionHedge` incurred a $1/P$ factor from all producers competing for the same outcome, `PunishUserUtility` implicitly restricts competition between producers.

5.3 Welfare analysis: Improved dependence on $\mathcal{D}$

Compared with the naive setting of criteria in the previous subsection, we can do even better by having different producers specialize to different subsets of users. The potential for specialization improves the dependence on $\mathcal{D}$ and even results in the optimal welfare in some cases.

Welfare bound in the limit as $c \to \infty$. We first analyze the welfare $W(c, \beta, \mathcal{D}, P)$ as a function of c, focusing on the limiting case of $c \to \infty$ for analytic simplicity.

▶ **Proposition 9.** *Suppose that $\mathcal{D}$ has finite support. The limiting welfare $\lim_{c \to \infty} W(c, \beta, \mathcal{D}, P)$ is equal to the following optimization problem:*

$$G(\beta, \mathcal{D}, P) := \max_{p_1, \ldots, p_P} \mathbb{E}\Big[\max_{j \in [P]} \langle u, p_j \rangle\Big] \quad s.t.\ \|p_j\| = 1, \forall j \in [P]. \tag{5}$$

Moreover, if $P \geq |supp(\mathcal{D})|$, then $\lim_{c \to \infty} W(c, \beta, \mathcal{D}, P) = 1$. Finally, for any $P \geq 1$, it holds that $\lim_{c \to \infty} W(c, \beta, \mathcal{D}, P) \geq \|\mathbb{E}_{\mathcal{D}}[u]\|$.

Proposition 9 relates the limiting welfare of `PunishUserUtility` to the optimum of (5), which is easy to analyze. Moreover, the optimum of (5) is 1 once the number of producers is as high as the number of users in the support (see the full version [22]). The intuition is that there are enough producers to specialize their content at the direction of each individual user so that all users are catered to. Finally, the optimum of (5) is always at least as large as $\|\mathbb{E}_{\mathcal{D}}[u]\|$ since we can set $p_j = \mathbb{E}_{\mathcal{D}}[u]/\|\mathbb{E}_{\mathcal{D}}[u]\|$ for all j. We defer the full proof to the full version [22].

This analysis illustrates that while `PunishLinDirectionHedge` incurs a $\|\mathbb{E}_{\mathcal{D}}[u]\|$ factor from the lack of specialization, `PunishUserUtility` can achieve better dependence on the user distribution $\mathcal{D}$ when P is sufficiently large. The economic intuition is that different producers can point at different directions of user subgroups rather than a single direction of the average user, allowing content to align with user preferences in a more fine-grained way.

Moving beyond `PunishLinDirectionHedge`, this analysis also enables us to compare the welfare of `PunishUserUtility` to *arbitrary* algorithms. The following corollary of Proposition 9 shows that the welfare bound $W(c, \beta, \mathcal{D}, P)$ is *optimal* relative to any algorithm.

▶ **Corollary 10.** *Suppose that $\mathcal{D}$ has finite support and $P \geq |supp(\mathcal{D})|$. The limiting welfare $\lim_{c\to\infty} W(c, \beta, \mathcal{D}, P)$ is equal to the optimal welfare that can be achieved by any algorithm at any (mixed-strategy) Nash equilibrium.*

We defer the proof of Corollary 10 to the full version [22]

Since $W(c, \beta, \mathcal{D}, P)$ lower bounds the welfare of `PunishUserUtility` in the limit as $\epsilon \to 0$, Corollary 10 shows that `PunishUserUtility` in the limit as $c \to \infty$ achieves the *optimal* welfare in this limiting regime. (We note that for any $\epsilon > 0$, `PunishUserUtility` is suboptimal due to the multiplicative factor of $(1 - \epsilon)$.) This result highlights that `PunishUserUtility` successfully leverages specialization to maximize user welfare.

Finite c: the case of 2 users. Since the above analysis focused on the case of $c \to \infty$, we now turn to finite c and examine the gap between $W(c, \beta, \mathcal{D}, P)$ and the upper bound in Proposition 12. For analytic tractability, we focus on the case of 2 users and explicitly compute $W(c, \beta, \mathcal{D}, P)$ in closed-form.

▶ **Proposition 11.** *Suppose that $\mathcal{D}$ is a uniform distribution over $u_1, u_2 \in \mathbb{R}^2_{\geq 0}$. Let $\theta = \cos^{-1}\left(\frac{\langle u_1, u_2\rangle}{||u_1||_2 \cdot ||u_2||_2}\right)$ be the angle between the users. Let $\theta^*(c) = 2\cos^{-1}(2^{-1/c})$. For any $P \geq 2$ and any $\beta \in (0, 1]$, the welfare is equal to:*

$$W(c, \beta, \mathcal{D}, P) = \begin{cases} \beta^{\frac{1}{c}} \|\mathbb{E}[u]\|_2 & \text{if } \theta < \theta^*(c) \\ (\beta/2)^{\frac{1}{c}} & \text{if } \theta \geq \theta^*(c). \end{cases}$$

We defer the proof to the full version [22].

Proposition 11 reveals that when $\theta < \theta^*(c)$ (the two users' preferences are similar), producers do not specialize their content to individual users even when P is arbitrarily large. This contrasts with Proposition 9 where specialization occurred where the number of producers was as large as the number of users. This finding bears resemblance to [24] where specialization did not always occur at equilibrium when preferences were similar across users; however, the results in [24] focus on the equilibrium behavior in a static game, whereas we focus on the solutions to a welfare-maximizing optimization program with cost constraints.

To analyze the gap between $W(c, \beta, \mathcal{D}, P)$ and the optimal welfare, we compare $W(c, \beta, \mathcal{D}, P)$ with the following upper bound on the welfare of any algorithm (proof deferred to the full version [22]).

▶ **Proposition 12.** *For any platform algorithm, at any (mixed-strategy) Nash equilibrium of producer strategies* $(\mu_1, \ldots, \mu_P)$*, the user welfare at any time* t *satisfies*

$$\mathbb{E}\left[\langle u^t, p^t_{A^t}\rangle\right] \leq \left(\frac{\beta}{1-\beta}\right)^{\frac{1}{c}}.$$

When comparing the bound in Proposition 11 with the bound in Proposition 12, there are two sources of suboptimality. First, even when $\theta \geq \theta^*(c)$, the lower bound is a factor of $((1-\beta)/2)^{1/c}$ smaller than the upper bound. Second, when $\theta < \theta^*(c)$ there is an additional factor of $||\mathbb{E}[u]||_2$ gap. This arises from the lack of specialization. Closing the gap between the upper and lower bound for finite c is an interesting direction for future work.

6 Discussion

We studied how a platform's learning algorithm shapes the content created by producers in a recommender system, focusing on the equilibrium quality of the content from the point of view of both the producer effort and the user welfare. We showed that the equilibrium quality decays across time when the platform runs `LinHedge` or `LinEXP3`. In light of these negative results, we designed punishment-based algorithmic approaches to incentivize producers to invest effort in content creation and achieve near-optimal user welfare in limiting cases.

Our model motivates several future directions that seem promising. First, generalizing our punishment-based approach in Section 4 and 5 to the bandit setting is an interesting open question. Furthermore, while we study the Nash equilibria of non-adaptive producers, it would be interesting to consider other solution concepts, such as the subgame perfect equilibria of adaptive producers. Finally, there are several open questions about where natural producer dynamics would converge to.

More broadly, we envision that our model can be extended to incorporate additional aspects of content recommender systems. On the platform side, before making recommendation decisions, the platform may have access to side information about the current content beyond just the producer's identity. The platform algorithm could thus leverage this additional information about the content to make more informed recommendations. Second, the platform may not directly observe the user utility and may have to infer this from noisy feedback (e.g., clicks, likes, or comments).

Our assumptions about producer behavior could also be relaxed in several ways. First, producers may face a cost of changing their content between time steps, which could be captured by making the cost function history-dependent. Second, producers may not participate on the platform for the same time horizon, which could be captured by allowing different producers to have different entry and exit timings. Finally, a producer may have specific skills enabling them to easily improve their content along certain dimensions; this could be captured by heterogeneity in the cost functions across producers.

Finally, going beyond recommender systems, it would be interesting to extend our insights to more general Stackelberg games with a leader is *learning* over time and *multiple non-myopic* followers. Since our work is motivated by online recommender systems, our model places several additional structural assumptions on the game (see Section 2.1). An interesting future direction would be to extend our model and findings to more general Stackelberg games, relaxing these structural assumptions.

References

1 Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2312–2320, 2011. URL: https://proceedings.neurips.cc/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

2 Naoki Abe and Philip M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 3–11. Morgan Kaufmann, 1999.

3 Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in Stackelberg security games. In *Proceedings of the 16th ACM Conference on Economics and Computation*, pages 61–78, 2015. doi:10.1145/2764468.2764478.

4 Ran Ben Basat, Moshe Tennenholtz, and Oren Kurland. A game theoretic analysis of the adversarial retrieval setting. *J. Artif. Int. Res.*, 60(1):1127–1164, 2017. doi:10.1613/JAIR.5547.

5 Omer Ben-Porat, Itay Rosenberg, and Moshe Tennenholtz. Content provider dynamics and coordination in recommendation ecosystems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

6 Omer Ben-Porat and Moshe Tennenholtz. A game-theoretic approach to recommendation systems with strategic content providers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1118–1128, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/a9a1d5317a33ae8cef33961c34144f84-Abstract.html.

7 Omer Ben-Porat and Rotem Torkan. Learning with exposure constraints in recommendation systems. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 3456–3466. ACM, 2023. doi:10.1145/3543507.3583320.

8 Micah D. Carroll, Anca D. Dragan, Stuart Russell, and Dylan Hadfield-Menell. Estimating and penalizing induced preference shifts in recommender systems. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2686–2708. PMLR, 2022. URL: https://proceedings.mlr.press/v162/carroll22a.html.

9 Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. *Advances in Neural Information Processing Systems*, 33:15265–15276, 2020.

10 Natalie Collina, Eshwar Ram Arunachaleswaran, and Michael Kearns. Efficient stackelberg strategies for finitely repeated games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pages 643–651. ACM, 2023. doi:10.5555/3545946.3598695.

11 Sarah Dean and Jamie Morgenstern. Preference dynamics under personalized recommendations. In *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, pages 795–816. ACM, 2022. doi:10.1145/3490486.3538346.

12 Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018. doi:10.1145/3219166.3219193.

13 Itay Eilat and Nir Rosenfeld. Performative recommendation: Diversifying content via strategic incentives. *CoRR*, abs/2302.04336, 2023. doi:10.48550/arXiv.2302.04336.

14 Arpita Ghosh and Patrick Hummel. Learning and incentives in user-generated content: multi-armed bandits with endogenous arms. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 233–246. ACM, 2013. doi:10.1145/2422436.2422465.

15 Arpita Ghosh and R. Preston McAfee. Incentivizing high-quality user-generated content. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 137–146. ACM, 2011. doi:10.1145/1963405.1963428.

16 Tarleton Gillespie. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media*. Yale University Press, New Haven, 2018.

17 Nika Haghtalab, Thodoris Lykouris, Sloan Nietert, and Alex Wei. Learning in stackelberg games with non-myopic agents. *CoRR*, abs/2208.09407, 2022. doi:10.48550/arXiv.2208.09407.

18 Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 111–122, 2016. doi:10.1145/2840728.2840730.

19 Andreas A. Haupt, Dylan Hadfield-Menell, and Chara Podimata. Recommending to strategic users. *CoRR*, abs/2302.06559, 2023. doi:10.48550/arXiv.2302.06559.

20 Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019. arXiv:1909.05207.

21 Jiri Hron, Karl Krauth, Michael I. Jordan, Niki Kilbertus, and Sarah Dean. Modeling content creator incentives on algorithm-curated platforms. *CoRR*, abs/2206.13102, 2022. doi:10.48550/arXiv.2206.13102.

22 Xinyan Hu, Meena Jagadeesan, Michael I. Jordan, and Jacob Steinhardt. Incentivizing high-quality content in online recommender systems. *CoRR*, abs/2306.07479, 2023. Full version with all proofs. doi:10.48550/arXiv.2306.07479.

23 Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 263–272. IEEE Computer Society, 2008. doi:10.1109/ICDM.2008.22.

24 Meena Jagadeesan, Nikhil Garg, and Jacob Steinhardt. Supply-side equilibria in recommender systems. *CoRR*, abs/2206.13489, 2022. doi:10.48550/arXiv.2206.13489.

25 Debarun Kar, Thanh H. Nguyen, Fei Fang, Matthew Brown, Arunesh Sinha, Milind Tambe, and Albert Xin Jiang. *Trends and Applications in Stackelberg Security Games*, pages 1223–1269. Springer International Publishing, Cham, 2018.

26 Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the "wisdom of the crowd". In *Proceedings of the fourteenth ACM Conference on Electronic Commerce, EC 2013*, pages 605–606, 2013. doi:10.1145/2492002.2482542.

27 Tao Li and Suresh P. Sethi. A review of dynamic Stackelberg game models. *Discrete and Continuous Dynamical Systems - B*, 22(1):125–159, 2017.

28 Yang Liu and Chien-Ju Ho. Incentivizing high quality user contributions: New arm generation in bandit learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1146–1153. AAAI Press, 2018. doi:10.1609/AAAI.V32I1.11464.

29 Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. Optimizing long-term social welfare in recommender systems: A constrained matching approach, 2020. arXiv:2008.00104.

30 Gergely Neu and Julia Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3049–3068. PMLR, 2020. URL: http://proceedings.mlr.press/v125/neu20b.html.

31 Alexander Rakhlin and Karthik Sridharan. BISTRO: an efficient relaxation-based method for contextual bandits. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48, pages 1977–1985, 2016. URL: http://proceedings.mlr.press/v48/rakhlin16.html.

32 Goldman Sachs. The creator economy could approach half-a-trillion dollars by 2027, 2023.

33 Vasilis Syrgkanis, Haipeng Luo, Akshay Krishnamurthy, and Robert E. Schapire. Improved regret bounds for oracle-based adversarial contextual bandits. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3135–3143, 2016. URL: https://proceedings.neurips.cc/paper/2016/hash/dfa92d8f817e5b08fcaafb50d03763cf-Abstract.html.

34 Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.*, 69:29–39, 2017. doi:10.1016/J.ESWA.2016.09.040.

35 Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H. Chi. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 441–447. ACM / IW3C2, 2020. doi:10.1145/3366424.3386195.

36 Fan Yao, Chuanhao Li, Denis Nekipelov, Hongning Wang, and Haifeng Xu. How bad is top-k recommendation under competing content creators? *CoRR*, abs/2302.01971, 2023. doi:10.48550/arXiv.2302.01971.

37 Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed H. Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pages 269–277. ACM, 2019. doi:10.1145/3298689.3346996.

38 Mengmeng Yu and Seung Ho Hong. Supply–demand balancing for power management in smart grid: A stackelberg game approach. *Applied Energy*, 164:702–710, 2016.

39 Banghua Zhu, Stephen Bates, Zhuoran Yang, Yixin Wang, Jiantao Jiao, and Michael I. Jordan. The sample complexity of online contract design. *CoRR*, abs/2211.05732, 2022. doi:10.48550/arXiv.2211.05732.

40 Tijana Zrnic, Eric Mazumdar, S. Shankar Sastry, and Michael I. Jordan. Who leads and who follows in strategic classification? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15257–15269, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/812214fb8e7066bfa6e32c626c2c688b-Abstract.html.