PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Learning the Optimal Recommendation from Explorative Users

Fan Yao[1], **Chuanhao Li**[1], **Denis Nekipelov**[2], **Hongning Wang**[1], **Haifeng Xu**[1]

[1]Department of Computer Science, University of Virginia, USA
[2]Department of Economics, University of Virginia, USA
{fy4bc,cl5ev,dn4w,hw5x,hx4ad}@virginia.edu

Abstract

We propose a new problem setting to study the sequential interactions between a recommender system and a user. Instead of assuming the user is omniscient, static, and explicit, as the classical practice does, we sketch a more realistic user behavior model, under which the user: 1) rejects recommendations if they are clearly worse than others; 2) updates her utility estimation based on rewards from her accepted recommendations; 3) withholds realized rewards from the system. We formulate the interactions between the system and such an explorative user in a K-armed bandit framework and study the problem of learning the optimal recommendation on the system side. We show that efficient system learning is still possible but is more difficult. In particular, the system can identify the best arm with probability at least $1-\delta$ within $O(1/\delta)$ interactions, and we prove this is tight. Our finding contrasts the result for the problem of best arm identification with fixed confidence, in which the best arm can be identified with probability $1-\delta$ within $O(\log(1/\delta))$ interactions. This gap illustrates the inevitable cost the system has to pay when it learns from an explorative user's revealed preferences on its recommendations rather than from the realized rewards.

1. Introduction

Recommender systems (RS) are typically built on the interactions among three parties: the system, the users, and the items (Tennenholtz and Kurland 2019). By collecting user-item interactions, the system aims to predict a user's preference over items. This type of problem setting has been extensively studied for decades and has seen many successes in various real-world applications, such as content recommendation, online advertising, and e-commerce platforms (Das et al. 2007; Linden, Smith, and York 2003; Koren, Bell, and Volinsky 2009; Schafer, Konstan, and Riedl 1999; Gopinath and Strickman 2011).

Most previous works have modeled the RS users as an unknown but omniscient "*classifier*" (Das et al. 2007; Li et al. 2010; Linden, Smith, and York 2003) that allows the system to query their preference over the candidate items directly. For instance, any RS algorithm based on supervised learning has to assume that users have the full information beforehand, as its training labels are derived from users' feedback. However, for at least two reasons, such modeling assumptions of static and omniscient users appear less realistic in many modern RS applications. First, given the huge size of candidate choices, a typical user is usually not fully aware of her "true preference" but needs to estimate it via the interactions with the RS. For instance, an ordinary user on video-sharing apps like TikTok or review-sharing apps like Yelp does not have pre-determined rewards on all possible choices or knows the optimal choice in advance. Instead, she has to consume the recommendations in order to discover the desirable content gradually. Second, in many applications, users simply respond to recommendations with accept/reject decisions rather than reveal their consumed items' utility. This situation is reflected in most practical recommendation systems nowadays. Platforms like TikTok and Yelp can easily collect binary feedback like click/non-click while struggling to evaluate the users' actual extent of satisfaction (Schnabel et al. 2018). These two limitations challenge the reliability of existing recommendation solutions in utility estimation from user feedback and thus shake the foundation of modern recommender systems.

To address the challenges mentioned above, we introduce a more realistic user behavior model, which hinges on two assumptions of today's RS users. Firstly, we believe the users are also learning the items' utilities via exploration. Their feedback becomes more relevant to the item's utility only after gaining more experience, e.g., consuming the recommended item. This perspective has been observed and supported in numerous cognitive science (Cohen, McClure, and Yu 2007; Daw et al. 2006), behavior science (Gershman 2018; Wilson et al. 2014), and marketing science studies (Villas-Boas 2004). For instance, Zhang and Angela (2013) showed through a multi-armed bandit experiment that humans maintain confidence levels regarding different choices and eliminate sub-optimal choices to achieve long-term goals when they are aware of the uncertain environment. These works motivate us to study the recommendation problem under a more realistic user model, where the user keeps refining her assessment about an item after consuming it, and she is willing to explore the uncertainty when deciding on the recommendations. Formally, we model her exploration as being driven by her estimated confidence intervals of each item's reward: she will only reject an item when its estimated reward is clearly worse than others.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Secondly, we assume that only the users' *binary* responses of acceptance are revealed to the system, whereas the realized user reward of any consumed items is kept only to the user (to improve her own reward estimation). This thus gives rise to an intriguing challenge of learning user's utility parameters from only the coarse and implicit feedback of "revealed preference" (Richter 1966). Learning from comparative binary feedback is not a novel setting in both empirical studies of recommendation systems and online learning literature (Yue et al. 2012; Yue and Joachims 2011; Komiyama et al. 2015; Zoghi et al. 2014). However, we adopt a new and more general perspective to interpret this binary feedback in the sense that the system does not even know the references in observed comparisons. We will discuss it more in the related work. One may naturally wonder why the user does not simply give all her realized rewards to the system since both the system and user are learning the best recommendation for the user. This is certainly an ideal situation but, unfortunately, is highly unrealistic in practice. As we mentioned before, it is widely observed that very few RS users would bother to provide detailed feedback (even not numerical ratings). This observation is also supported by the 90-9-1 Rule for online community engagement, and the "Lazy User Theory" (Tétard and Collan 2009) in the HCI community, which states that a user will most often choose the solution that will fulfill her information needs with the least effort.

Under this more realistic yet challenging environment, it is unclear whether efficient system learning is even possible, i.e., can the system still discover the user's real preference? To answer the question, we formulate the interactions between the system and such an explorative user in a K-armed bandit framework and study best arm identification (BAI) with fixed confidence. We design an efficient online learning algorithm and prove it obtains an $O(\frac{1}{\delta})$ upper bound on the number of recommendations to identify the best arm with probability at least $1-\delta$. We also show that this bound is tight by proving a matching lower bound for any successful algorithm. Our results illustrate the inevitable gap between the performance under the standard best arm identification setting and our setting, which indicates the intrinsic hardness in learning from an explorative user's revealed preferences. Our experiments also demonstrate the encouraging performance of our algorithms compared to the state-of-the-art algorithms for BAI applied to our learning setup.

Related Work The first related direction to this work is the problem of best arm identification (BAI) with fixed confidence (Garivier and Kaufmann 2016). Instead of minimizing regret, the system aims to find the arm with the highest expected reward with probability $1-\delta$, while minimizing the number of pulls T. The tight instance-dependent upper bound for T is known to be $O(H \log \frac{1}{\delta})$ (Carpentier and Locatelli 2016), where H is a constant describing the intrinsic hardness of the problem instance. In our work, the system shares the same goal but under a set of more challenging restrictions posed by learning from an explorative user's revealed preferences. For example, the system cannot directly observe the realized reward of each pulled arm. We prove that under this new learning setup, the budget upper bound increases from $O(\log \frac{1}{\delta})$ to $O(\frac{1}{\delta})$.

There are also previous works that focus on online learning without access to the actual rewards. The dueling bandit problem proposed in (Yue et al. 2012) modeled partial-information feedback where actions are restricted to noisy comparisons between pairs of arms. Our feedback assumption is more challenging than dueling bandit in two aspects. First, we do not assume the system knows the reference arm to which the user compares when making her decisions. Second, the user's feedback is evolving over time as she learns from the realized rewards. Hence, none of existing dueling bandit algorithms (Yue and Joachims 2011; Komiyama et al. 2015; Zoghi et al. 2014) can address our problem. The unobserved reward setting is also studied in inverse reinforcement learning. For instance, Hoiles et al. (Hoiles, Krishnamurthy, and Pattanayak 2020) used Bayesian revealed preferences to study if there is a utility function that rationalizes observed user behaviors. Their work focused on user behavior modeling itself while we studied system learning and analyzed the outcome induced by this type of user behavior assumption.

Another remotely related direction is incentivized exploration in the Internet economy. In such a problem, a system aims to maximize the welfare of a group of users who only care about their short-term utility. Kremer, Mansour, and Perry (2014) first studied this problem and developed a policy that attains optimal welfare by partially disclosing information to different users. Follow-up works extended the setting by allowing users to communicate (Bahar, Smorodinsky, and Tennenholtz 2015) and introducing incentive-compatibility constraints (Mansour et al. 2016; Mansour, Slivkins, and Syrgkanis 2020). Our motivation considerably differs from this line of work in three important aspects. First, incentivized exploration looks at an informationally advantaged principal whereas our system is in an informational disadvantageous position, as it has mere access to the user's revealed preferences. Second, their setting looks at how to influence user decisions through signaling with misaligned incentives, whereas we are trying to help a boundedly rational ordinary user to learn their best action in a cooperative environment. Third, the user in our model is an adaptive learning agent rather than a one-time visitor to the system.

2. Modeling Users' Revealed Preferences

To study the sequential interactions between a recommender system and an explorative user, we adopt a stochastic bandit framework where the time step t is used to index each interaction, and the set of arms $[K] = \{1, \cdots, K\}$ denote the recommendation candidates. At each time step t, the following events happen in order:

1. The system recommends an arm a_t to the user;
2. The user decides whether to accept or reject a_t. If the user accepts a_t, realized reward $r_{a_t,t}$ is disclosed to the user afterwards.
3. The system observes the user's binary decision of acceptance or not, i.e., the revealed preference (Richter 1966).

From the user's perspective, we denote the true reward of each arm $i \in [K]$ as μ_i, and the realized reward after

each acceptance of arm i is drawn independently from a sub-Gaussian distribution with mean μ_i and unit variance. The system has a long-term objective and aims to find the best arm while minimizing the total number of recommendations. This renders our problem a best arm identification (BAI) problem with fixed confidence but based on partial information about the rewards. Throughout the paper, we assume without loss of generality that $\mu_* = \mu_1 > \mu_2 \geq \mu_3 \geq \cdots \geq \mu_K$, and let $\Delta_1 = \mu_1 - \mu_2, \Delta_i = \mu_* - \mu_i > 0, \forall i > 1$. Following the convention in BAI literature (Carpentier and Locatelli 2016; Audibert and Bubeck 2010), we further define the quantity $H = \sum_{i=1}^{K} \frac{1}{\Delta_i^2}$ which characterizes the hardness of the problem instance.

As discussed previously, the user cannot choose from the full arm set but can only decide whether to accept or reject the recommended arms from the system. To make a decision at time t, we assume the user utilizes the information in all previous interactions by maintaining a confidence interval $CI_{i,t} = (lcb_{i,t}, ucb_{i,t})$ for each arm i, which is defined as

$$(lcb_{i,t}, ucb_{i,t}) \triangleq \left(\hat{\mu}_{i,t} - \sqrt{\frac{\Gamma(t;\rho,\alpha)}{n_i^t}}, \hat{\mu}_{i,t} + \sqrt{\frac{\Gamma(t;\rho,\alpha)}{n_i^t}} \right),$$

where lcb and ucb stand for the lower/upper confidence bounds respectively, n_i^t is the number of *acceptances* on arm-i up to time t, $\hat{\mu}_{i,t} = \frac{1}{n_i^t} \sum_{s=1}^{t} \mathbb{I}[i \text{ is accepted at } s] r_{i,s}$ is the empirical mean reward of arm i at time t, and $\Gamma(t;\rho,\alpha)$ is a function parameterized by $\{\rho_t, \alpha\}$, which characterize the user's confidence level to her reward estimation at time step t. Following the convention rooted in the UCB1 algorithm (Auer, Cesa-Bianchi, and Fischer 2002), we consider the (flexible) confidence bound form:

$$\Gamma(t;\rho,\alpha) = \max\{0, 2\alpha \log[\rho_t \cdot n(t)]\}, \quad (1)$$

where $n(t) = \sum_{i\in[K]} n_i^t$ is the total number of accepted recommendations up to time t. We note that the choice of Γ is to flexibly cover possibly varied user types captured by parameters α and ρ_t. In particular, α directly controls the span of the CIs and thus represents the user's intrinsic tendency to explore: a larger α indicates a higher tolerance for the past observations, meaning the user is more willing to accept recommendations in a wider range. $\rho_t : \mathbb{N} \to [\rho_0, \rho_1]$ is allowed to be any sequence that can depend on the interaction history and has a bounded range $[\rho_0, \rho_1] \subset (0, +\infty)$, which captures the cases where the user's confidence over the system evolves over time. For example, ρ_t can be a function of the acceptance rate $\frac{n(t)}{t} \in [0,1]$ and increases monotonically with respect to $\frac{n(t)}{t}$. Our results only rely on the lower and upper bound of ρ_t and are oblivious to its specific format. Note that for the special case of $\alpha = 1, \rho_t = 1\,\forall t$, Eq (1) corresponds to the classic confidence interval defined in UCB1. We remark that parameters α and ρ_t are only to characterize different types of users, which provide flexibility in handling different real-world scenarios; but they are *not* introduced for our solution.

The decision rule. When an arm i is recommended, we assume the user will reject it if and only if there exists $j \neq i$ such that $lcb_j \geq ucb_i$. That is, the user only accepts an arm if there is no other arm that is clearly better than the recommended one with a high confidence. The rationale behind our imposed user decision rule is straightforward: first, the user should not miss the arm with the highest lower confidence bound as this is arguably the safest choice for the user at the moment; second, if two arms have chained confidence intervals, the user does not have enough information to distinguish which one is better, and hence should not reject either one, i.e., being explorative.

Algorithm 1: Phase-1 Sweeping

Input: $K > 0$, $\delta \in (0,1)$, $N_1(\delta) > 0$.
Initialization: $F = [K]$, $N = 0$, $n_i = 0, i \in [K]$.
repeat
  Recommend each item in F once, and remove the rejected ones from F.
until F is empty or the time step exceeds $N_1(\delta)$.
repeat
  If F is empty, reset $F = \{1, \cdots, K\}$;
  for $i \in F$ **do**
    Recommend i until rejected, then remove it from F.
until The time step exceeds $N_1(\delta)$.
Output: number of acceptances for each arm $\{n_i\}_{i=1}^K$.

Algorithm 2: Phase-2 Elimination

Input: $K > 0$, $\{n_i\}_{i=1}^K$ from Phase-1.
Initialization: $F = [K]$.
while $|F| > 1$ **do**
  Recommend $a_t = \arg\min_{i\in[K]} n_i$ and update n_{a_t}.
  Remove a_t from F if rejected.
Output: F.

3. Learning from Explorative Users' Revealed Preferences

With stochastic rewards, we know $\mathbb{P}[\mu_i \in CI_{i,t}]$ almost always increases as the number of acceptances $n(t)$ grows. Therefore, the system can confidently rule out a sub-optimal arm once it has collected a reasonable number of acceptances. In light of this, we devise a two-phase explore-then-exploit strategy for system learning: the system first accumulates a sufficient number of acceptances and then examines through the arm set to eliminate sub-optimal ones with a high confidence.

The Phase-1 design is presented in Algorithm 1. Like standard bandit algorithms, the system will execute an initialization procedure by sweeping through the arm set $F = [K]$ and then recommend each arm repeatedly until it collects exactly one rejection on each arm there. This initialization stage is similar to the round-robin style pulls in standard bandit algorithms (e.g., UCB1, ϵ-Greedy). But the key difference is that our algorithm will initialize by collecting one rejection on each arm whereas standard bandit algorithm will initialize by collecting one pull (i.e., acceptance) on each arm. This is because rejections in our setup are more

informative than acceptances to the system. After initialization, Algorithm 1 enters the main loop and do the following: keeps recommending the same arm until it gets rejected and then moves to another arm in F. After each arm gets rejected once, the system starts a new round by resetting $F = [K]$. This procedure continues until the total number of acceptances exceeds $N_1(\delta)$. An inquisitive reader might wonder why repeatedly recommending a specific arm could be practical when deploying a real recommender system. To clarify, we note that an arm in our model represents a type or category of items, rather than just literally an individual. This is also the typical interpretation of arms in the stochastic bandit literature. The sweeping strategy reflects the principle of Phase-1: the system aims to collect a reasonable number of acceptances while minimizing the number of rejections by not recommending any risky arm. For the ease of analysis, we divide Phase-1 into different *rounds* (indexed by r) by the time steps when the system resets F. We will prove later that there is a tailored choice of $N_1(\delta)$ such that when the system enters Phase-2 with $N_1(\delta)$ acceptance, it can identify the best arm with probability $1 - \delta$.

We now present the design for Phase-2, as shown in Algorithm 2, which follows Phase-1 Sweeping. Here, the system executes arm elimination: always recommend the arm with the minimum number of acceptances; and eliminate an arm when it is rejected by the user, until there is only one arm left. We prove that the stopping time of Algorithm 2 is finite with probability 1, and it outputs the best arm with probability $1 - \delta$ when it terminates. We name our proposed two-phase algorithm **B**est **A**rm **I**dentification under **R**evealed preferences, or BAIR for short.

Next, we analyze BAIR by upper bounding its stopping time given fixed confidence δ. Our main result is formalized in the following theorem. All formal proofs of our theories are omitted due to the space limit and can be found in the full version (Yao et al. 2021).

Theorem 1. *When Γ is defined as in Eq* (1)*, with probability at least $1 - 2\delta$, the system makes at most*

$$O\Big(K^{\frac{1}{\alpha}}\delta^{-\frac{1}{\alpha}} + K^{1+\frac{1}{2\alpha}}\delta^{-\frac{1}{2\alpha}}\sqrt{\log\frac{K}{\delta}} + \frac{\alpha K}{\Delta_1^2}\log\frac{K}{\delta\Delta_1}\Big)$$

recommendations and successfully identifies the best arm by running Algorithm 1 and 2.

Note that the upper bound on the number of rounds above is deterministic while not in expectation. The proof of Theorem 1 requires separate analysis for Phase-1 and Phase-2, which we discuss in the following subsections. At a high level, the first two terms in the bound come from the number of acceptances and rejections in Phase-1, and the last term corresponds to the number of acceptances in Phase-2. We decompose the bound in Theorem 1 in Table 1.

Table 1: Upper bounds on #recommendations in Phase 1,2

Phase	# Acceptance	# Rejection	Prob.
1	$O(K^{\frac{1}{\alpha}}\delta^{-\frac{1}{\alpha}})$	$O(K^{\frac{1+2\alpha}{2\alpha}}\delta^{-\frac{1}{2\alpha}}\log^{\frac{1}{2}}\frac{K}{\delta})$	$1-\delta$
2	$O(\frac{\alpha K}{\Delta_1^2}\log\frac{K}{\delta\Delta_1})$	K	$1-\delta$

Note that there is a clear *tradeoff* between the upper bounds in Phase-1 and Phase-2 in terms of α: a smaller α increases the upper bound in Phase-1 but requires less number of recommendations in Phase-2, while a larger α ensures a lighter Phase-1 but would result in a more cumbersome Phase-2. This is expected because, e.g., when facing a highly explorative user (large α), the system can easily accumulate sufficient acceptances in Phase-1. However, it will need more comparisons in Phase-2 to identify the best arm for such a highly explorative user. Theoretically, there exists an optimal α which minimizes the total number of recommendations; however, this is not particularly interesting to investigate in this paper, as α is not under the system's control, but a characterization of the user.

Upper Bound for Phase-2

We start the analysis for Phase-2 first as it will lead to the correct N_1 for us to run Phase-1. Specifically, we prove that when $N_1 = \frac{1}{\rho_0}\cdot\left(\frac{2K}{\delta}\right)^{\frac{1}{\alpha}}$ acceptances are accumulated in Phase-1, it is safe for the system to move on to Phase-2.

Lemma 1. *If Phase-1 terminates with $N_1 = \frac{1}{\rho_0}\cdot\left(\frac{2K}{\delta}\right)^{\frac{1}{\alpha}}$ acceptances, the Phase-2 Algorithm 2 will output the best arm with probability at least $1 - \delta$.*

The next Lemma shows that no matter when the system enters Phase-2, Algorithm 2 must terminate with probability $1 - \delta$ within $O(\log\frac{1}{\delta})$ steps.

Lemma 2. *With probability $1 - \delta$, Algorithm 2 terminates within $O(K + \sum_{i=1}^{K}\frac{\alpha}{\Delta_i^2}\log\frac{\rho_1 K}{\rho_0\delta\Delta_i})$ steps.*

The first term $O(K)$ in the bound corresponds to the number of rejections in Phase-2, since Phase-2 Elimination incurs at most $K-1$ rejections by definition. The second term characterizes the number of acceptances, which matches the tight lower bound of BAI with fixed budget (Carpentier and Locatelli 2016) in terms of δ with a factor $\sum_{i=1}^{K}\frac{1}{\Delta_i^2}\log\frac{1}{\Delta_i}$ instead of $H = \sum_{i=1}^{K}\frac{1}{\Delta_i^2}$. Thus, the bound provided by Lemma 2 is almost tight. The ρ_1 also plays a role in the upper bound because when ρ_1 is too large, the user could maintain a very wide confidence interval for each arm which requires extra effort for the system to eliminate sub-optimal arms. Combining Lemma 1 and Lemma 2 and take ρ_0, ρ_1 as fixed constants, we conclude that Algorithm 2 will terminate and output the best arm with probability $1 - \delta$ within $O(\sum_{i=1}^{K}\frac{\alpha}{\Delta_i^2}\log\frac{K}{\delta\Delta_i})$ steps after Algorithm 1 is equipped with $N_1(\delta) = O(K^{\frac{1}{\alpha}}\delta^{-\frac{1}{\alpha}})$. Note that compared with the theoretical guarantee for BAI with fixed confidence, our upper bound matches the lower bound in (Garivier and Kaufmann 2016) in terms of δ. This implies that once the system has accumulated sufficient acceptances in Phase-1, the learning from reveal preferences does not bring extra difficulty. However, the bottleneck for the integrated system strategy lies in Phase-1, which we now analyze.

Upper Bound for Rejections in Phase-1

Recall that a round in Algorithm 1 is a segment of a sequence of interactions indexed by r, in which the candidate arm set F is reset to $[K]$ at the beginning and each arm gets rejected once in the end. We abuse the notation a bit by using $[t_s^{(r)}, t_e^{(r)}]$ to denote the r-th round that starts from time $t_s^{(r)}$ with $N = n(t_s^{(r)})$ acceptances and ends at time $t_s^{(r)}$ with $N = n(t_e^{(r)})$ acceptances. Next we upper bound the total number of rounds in Phase-1. We prove that Algorithm 1 must terminate in a small number of rounds with probability $1-\delta$, as shown in the following lemmas.

Lemma 3. *For any $K > 0, \delta > 0, N_1 > 0$, with probability $1-\delta$, Algorithm 1 terminates in $O(\sqrt{N_1 \log \frac{K}{\delta}})$ rounds and thus incurs at most $O(K\sqrt{N_1 \log \frac{K}{\delta}})$ rejections.*

In particular, if we choose $N_1 \sim O(K^{\frac{1}{\alpha}}\delta^{-\frac{1}{\alpha}})$ in accordance with Lemma 1, the total number of rejections in Phase-1 can be upper bounded by $O(K^{1+\frac{1}{2\alpha}}\delta^{-\frac{1}{2\alpha}}\sqrt{\log \frac{K}{\delta}}) = o(N_1)$ as $\delta \to 0$. In the next section, we will show that $N_1 \sim O(K^{\frac{1}{\alpha}}\delta^{-\frac{1}{\alpha}})$ is necessary to guarantee the success in Phase-2. The proof of Lemma 3 depends on the following two lemmas.

Lemma 4. *(Lattimore and Szepesvári 2020) Let $\{X_t\}_{t=1}^{\infty}$ be a sequence of $i.i.d.$ sub-Gaussian random variables with zero mean and unit variance, and $\hat{\mu}_n = \frac{1}{n}\sum_{t=1}^{n} X_t$, for any $\delta \in (0,1)$,*

$$\mathbb{P}\Big(\forall n \in \mathbb{N}^+ : |\hat{\mu}_n| \le \sqrt{\frac{2}{n}\log\frac{n(n+1)}{\delta}}\Big) > 1-\delta.$$

Lemma 5. *Let $f(t) = \max_{i=1}^{K} \hat{\mu}_{i,t}$ be the highest empirical mean maintained by the user at time step t. Then for any round r denoted by $[t_s^{(r)}, t_e^{(r)}]$ in Algorithm 1, we have*

$$f(t_e^{(r)}) \le f(t_s^{(r)}) - 2\sqrt{\frac{\underline{\Gamma}^{(r)}}{n(t_e^{(r)})}},$$

where $\underline{\Gamma}^{(r)} = \min_{t\in[t_s^{(r)}, t_e^{(r)}]} \Gamma(t)$.

Lemma 5 shows an interesting property about the user's empirical reward estimation during our Algorithm 1 — the maximum empirical mean will decrease by at least $2\sqrt{\frac{\underline{\Gamma}^{(r)}}{n(t_e^{(r)})}}$ after each round. This implies that Phase 1 cannot run for too many rounds. Finally, assembling Lemma 1, 2, and 3, we can derive the upper bound for the stopping time of BAIR in Theorem 1.

The Lower Bound

It is worthwhile to compare our upper bound on the number of recommendations in Theorem 1 with the $O(\log \frac{1}{\delta})$ upper bound in standard BAI setting. Specifically, our bound is worse due to the leading term $O(\delta^{-\frac{1}{\alpha}})$. In this subsection, we prove that this performance deterioration is inevitable due to our focus on an intrinsically harder setup with only the user's revealed preferences. As our second main result, the following theorem shows that the dependence of δ in the upper bound of Theorem 1 is tight.

Theorem 2. *For any algorithm π and $0 < c < \frac{1}{2}$, there exists a problem instance depending on δ such that if π collects less than $N_0 = \max\{\frac{\delta^{-\frac{1}{\alpha}+c}}{\rho_0}, \frac{2}{\Delta_1^2}\log\frac{1}{4\delta}\}$ accepted recommendations, it must make mistake about the best arm with probability at least δ.*

Proof Sketch. The lower bound $N_0 \ge \frac{2}{\Delta_1^2}\log\frac{1}{4\delta}$ is from the general lower bound result for BAI in a stochastic bandit setting, as the system in our setting can never find the best arm quicker than an BAI algorithm that has access to the realized rewards. To prove $N_0 \ge \delta^{-\frac{1}{\alpha}+c}/\rho_0$, we construct two problem instances ν and ν' such that: 1). they have different best arms; 2). any system interacting with ν or ν' will receive exactly the same sequences of user binary responses with probability at least 2δ, as long as it collects less than N_0 acceptances. Therefore, the system is not able to differentiate between ν and ν' with probability at least $1-\delta$, thus making mistakes about the best arm with probability δ on either ν or ν'. Our final proof is based on the ensemble of the difficult instances in both situations. □

Note that the lower bound $\delta^{-\frac{1}{\alpha}+c}/\rho_0$ illustrates the intrinsic hardness of BAI from revealed preferences: any algorithm has to make at least $\Omega(\delta^{-\frac{1}{\alpha}+c})$ recommendations in order to guarantee the identification of the best arm for any problem instances in our setup. This is in sharp contrast to the well-known $O(\log\frac{1}{\delta})$ lower bound in the standard bandit reward feedback setting.

4. Experiments

In this section, we empirically study BAIR to support our theoretical analysis. We use simulations on synthetic datasets in comparison with several baseline algorithms. Since we propose a new perspective to model user-system interactions in RS, there is no baseline for direct comparison. However, this also gives us an opportunity to demonstrate how problematic it may be when using a wrong user model for the observed system-user interactions.

Experiment Setup and Baselines

As we discussed in the introduction, prior works treat users as an unknown but omniscient classifier, and therefore stochastic bandits are the typical choices to learn from user feedback. Moreover, since the users' responses in our problem setup are not necessarily stochastic, adversarial bandits could be another choice. Therefore, we employ the corresponding state-of-the-art algorithms, Track-and-stop (Garivier and Kaufmann 2016) (for BAI) and EXP3 (Auer et al. 2002) (for adversarial bandits), to compare with BAIR. Besides, we also propose a heuristic baseline, uniform exploration, to directly compete with BAIR. The details of these baselines are as follows.

Uniform exploration (UNI): The system recommends candidate arms uniformly until the number of recommen-

Table 2: Comparison between BAIR and three baselines on proposed metrics. ($\alpha = 1$)

		Stopping Time		Rejection Rate (%)				Prob. of Success			
δ	K	BAIR	T&S	BAIR	UNI	EXP3	T&S	BAIR	UNI	EXP3	T&S
0.1	2	405	539	0.8	8.5	2.5	1.0	0.999	0.786	0.621	0.999
	5	737	1069	1.4	27.6	8.8	1.7	1.000	0.568	0.549	0.971
	20	2113	3107	1.9	33.7	11.6	4.6	1.000	0.229	0.408	0.966
	100	10449	16523	2.0	34.4	13.4	11.9	1.000	0.092	0.400	0.965
0.05	2	413	557	0.7	9.4	2.8	1.1	1.000	0.799	0.654	1.000
	5	787	1123	1.2	29.2	7.6	1.9	1.000	0.599	0.581	0.978
	20	2421	3161	1.5	40.4	12.3	4.4	1.000	0.429	0.577	0.965
	100	10826	16577	2.2	43.2	14.1	11.1	1.000	0.150	0.534	0.945
0.02	2	422	544	0.7	12.2	3.9	1.2	1.000	0.806	0.638	0.999
	5	879	1148	1.2	31.8	7.7	2.1	1.000	0.670	0.607	0.962
	20	3593	3210	1.6	50.2	11.9	4.4	1.000	0.436	0.680	0.957
	100	11528	16955	2.5	46.7	14.2	12.3	1.000	0.153	0.634	0.963
0.01	2	437	554	0.8	17.6	2.7	1.3	1.000	0.821	0.632	0.998
	5	940	1153	1.3	34.9	7.7	2.0	1.000	0.701	0.604	0.959
	20	4017	3223	1.4	51.6	13.6	3.0	1.000	0.476	0.737	0.947
	100	20344	27548	1.5	52.8	15.9	12.4	1.000	0.130	0.725	0.930
0.005	2	512	570	0.8	27.6	4.0	1.4	1.000	0.992	0.901	1.000
	5	1692	1580	0.8	50.0	8.3	1.6	1.000	0.942	0.844	0.993
	20	7827	5983	0.7	71.4	12.4	5.4	1.000	0.938	0.921	0.979
	100	40246	37931	0.7	73.1	15.0	16.0	1.000	0.794	0.930	0.970

dations reaches the given threshold T. When the algorithm terminates, it outputs the arm with the maximum number of acceptances; ties are broken arbitrarily. **Track-and-stop (T&S)**: This is the state-of-the-art solution for BAI with fixed confidence (Garivier and Kaufmann 2016). The expected stopping time of T&S provably matches its information-theoretic lower bound $O(\log \frac{1}{\delta})$ in the stochastic bandit setting. The effectiveness of T&S relies on the independent and stationary reward assumptions on each arm, which fail to hold in our setup as user responses are not a simple function of their received rewards. We will investigate how the theoretical optimality of the T&S breaks down under our problem setting. **EXP3**: To the best of our knowledge, there is no BAI algorithm under an adversarial setting. As a result, we adopt EXP3 (Auer et al. 2002) for comparison. Given the number of arms K and a time horizon $T > K \log K$, EXP3 is provably a no-regret learning algorithm if taking $\gamma \sim O(\sqrt{\frac{\log K}{KT}})$ and $\epsilon \sim O(\sqrt{\frac{K \log K}{T}})$. We run EXP3 with this configuration and output the arm with the maximum number of acceptances in the end.

Simulation Environment and Metrics

For different configurations of (δ, K, Δ_1) for BAI, we generate 1000 independent problem instances $(\mu_i)_{i=1}^K$ by sampling each $\mu_i \in N(0, 1)$ and then reset μ_* to meet the given value of Δ_1. Observing that our conclusion does not vary much under different Δ_1, we present the result for $\Delta_1 = 0.5$ in this section and leave more results in the full version (Yao et al. 2021) due to space limit. The parameters in the user model are set to $\alpha = 1, \rho_t = 1 + \frac{n(t)}{t} \in [1, 2]$, i.e., $\rho_0 = 1, \rho_1 = 2$, and results for different choice of α can be found in (Yao et al. 2021). We run BAIR with $N_1 = \frac{2K}{\delta}$ and compare its performance with UNI, EXP3 and T&S on the entire set of problem instances and calculate the following three metrics. **Probability of success:** When each algorithm terminates, we examine whether the output arm is the best arm (i.e., success). The probability of success (p) is then given by the empirical frequency of success over all problem instances. We also calculate the value $\frac{1-p}{\delta}$ to measure if and how much the probability of success falls below the given confidence level δ, which is presented right after the probability of success. **Rejection rate:** When each algorithm terminates at step T, we count the total number of rejections $\#Rej$ the system receives. The rejection ratio is given by $\frac{\#Rej}{T}$, and then averaged over all problem instances. **Stopping time:** It is the total number of interactions needed to terminate an algorithm. BAIR and T&S stop by their own termination rules; UNI and EXP3 stop by the input time T, since these two algorithms terminate by a preset time horizon. To make a fair comparison, we set T for UNI/EXP3 as the average stopping time of BAIR under the corresponding problem instance. Hence, this metric is only set to compare BAIR and T&S.

Experiment Results

The results are reported in Table 2. Based on the comparison results for BAIR and the baselines, we have the following observations.

BAIR vs. T&S. As shown in Table 2, T&S enjoys the best performance among three baselines on rejection rate and probability of success, but still does not work well in our problem setting. Given the confidence threshold δ, T&S fails to identify the best arm with probability $1 - \delta$ for $K > 2$ and $\delta < 0.05$. We also find the stopping time of T&S is worse than BAIR in most cases and fails to meet its theoretical lower bound $O(\log \frac{1}{\delta})$. This is expected: our binary user feedback cannot be simply modeled as independent and

stationary rewards, which are the fundamental assumptions behind the design of T&S. Since T&S wrongly models user responses, it is easily misinformed by the user's potentially inaccurate feedback in the early stage. As a result, it is very likely to miss the best arm and spend most of the rest time on a wrong subset. In contrast to T&S, BAIR is aware that the revealed preferences from the early stage are very likely to have a large variance. Therefore, it chooses to make safe recommendations at first to help the user gain more experiences (Phase-1 preparation) such that that the user will reveal more accurate feedback later on (Phase-2 elimination). This explains how BAIR achieves the goal more efficiently, even with the additional cost in Phase-1.

BAIR vs. UNI/EXP3. The other two baselines, UNI and EXP3, exhibit worse performance in both the rejection ratio and the probability of success than BAIR. Given the same time budget, UNI always suffers from the largest proportion of rejections because it does not take any measures to eliminate bad arms. As rejections do not update the user's empirical reward estimation, the given time budget is insufficient for UNI to differentiate the arms with similar expected rewards, thus causing a low probability of success. EXP3 enjoys a lower rejection rate than UNI, because it pulls those empirically bad arms less. The mandatory exploration in EXP3 helps correct the inaccurate early observations and gives a more competitive probability of success when K gets larger. However, due to the larger variance of EXP3, if the user's estimated reward for the best arm is low at the beginning, EXP3 tends to overly focus on differentiating a group of suboptimal arms, which decreases its chance of discovering the best arm.

To summarize, the fundamental reason for the failure of these baselines lies in the insufficient system exploration when facing an explorative user. These baselines either treat the user as a black-box or assume independent and stationary user feedback, which leads to a worse empirical result in terms of both accuracy and efficiency in finding the best arm.

The result in Table 2 supports our theoretical analysis in Theorem 1. When $\Delta_1 = 0.5$ and $\delta < 0.1$, $\frac{1}{\delta}$ dominates $\frac{1}{\Delta_1^2}$ and Theorem 1 suggests the algorithm's stopping time grows approximately linear in $\frac{1}{\delta}$. As expected, the first column in Table 2 confirmed our theory. The first column in Table 2 also suggests an approximately linear dependency between BAIR's stopping time and K. Although it is not fully supported by our theory (the leading term in the upper bound result is $O(K^{1.5})$ when $\alpha = 1$), we believe this observation is informative and could be an interesting target for future work.

Additional Experiments on the Robustness

In practice, it might be too restrictive to assume that the user strictly follows our proposed confidence interval (CI) based behavior model. Thus, it is interesting and also crucial to test the robustness of BAIR under the situation where the user's behavior might deviate from the CI-based model. To this end, we extended the user model to a stochastic setting by incorporating "decision randomness". Specifically, we assume at each time step, with some constant probability p, the user makes a random decision (accept/reject the recommendation with an equal probability); otherwise, she would follow the CI-based behavior model. We demonstrate that a minor modification of Phase-2 still guarantees a competitive empirical performance of BAIR, against the three baselines. The idea is, instead of eliminating an arm after the first rejection in Phase-2, the system only discards an arm after its m-th rejection. It turns out that if we choose $m = O(\frac{K}{\delta})$, BAIR still successfully finds the best arm with probability $1 - 2\delta$ in this stochastic setting, with merely an additional cost up to a multiplicative constant. Due to space limit, a detailed discussion can be found in the full version (Yao et al. 2021).

5. Discussions and Future Work

To bring user modeling to a more realistic setting in modern recommender systems, we proposed a new learning problem of best arm identification from explorative users' revealed preferences. We relax the strong assumptions that users are omniscient by modeling users' learning behavior, and study the learning problem on the system side to infer user's true preferences given only the revealed user feedback. We proved efficient system learning is still possible under this challenging setting by developing a best arm identification algorithm with complete analysis, and also disclosed the intrinsic hardness introduced by the new problem setup. Our result illustrates the inevitable cost a recommender system has to pay when it cannot directly learn from a user's realized utilities. As concluding remarks, we point out some interesting open problems in this direction:

The optimal choice of N_1. Although our lower bound result in Theorem 2 is tight in δ, it does not match the upper bound in Theorem 1 in terms of K. The mismatch comes from the choice of $N_1 = (2K/\delta)^{1/\alpha}/\rho_0$, which might be overly pessimistic as Theorem 2 only indicates a necessary condition of $N_1 > \delta^{-1/\alpha}/\rho_0$. To bridge this gap, a tighter upper bound is needed to improve the choice of N_1. We believe this is promising because the experiment results in Table 2 demonstrate that the choice of $N_1 = (2K/\delta)^{1/\alpha}/\rho_0$ almost guarantees a success probability 1.0 even when δ takes a large value, e.g., 0.1. This implies the stopping time of BAIR could be improved by setting a smaller N_1. In practice, we can fine-tune N_1 to pin down the optimal choice. For example, one can simply apply binary search within $(0, (2K/\delta)^{1/\alpha}/\rho_0)$ with $N_1 = O(\delta^{-1/\alpha})$ as a starting point.

Beyond a single user. We note that our problem formulation and solution for the system and a single user also shed light on learning from a *population* of users. For example, users sometimes learn or calibrate their utility from third-party services that evaluate the quality of items by aggregating users' feedback across different platforms. As a result, users equipped with these services are inclined to exhibit an exploratory pattern and make decisions based on the comparison of confidence intervals. We believe that our problem setting also provides a prototype to study the optimal strategy for the system under this new emerging situation.

6. Ethics Statement

This work studies the principle of recommendation algorithm design under a more realistic user behavior model. We tackle the problem mainly from a theoretical perspective and thus do not introduce any immediate ethical concern. In a nutshell, our algorithmic solution BAIR aims at helping a bounded rational user to realize her best options from a large collection of candidates. As our result shows, a typical user would easily get stuck in sub-optimal choices without the system's help or simply relying on off-the-shelf best arm identification algorithms. Moreover, we should emphasize that BAIR does not harm the users' satisfaction much as it does not allow many rejections. In fact, BAIR implicitly controls the fraction of rejections to achieve the minimal stopping time, as when the user rejects, the system gets penalized for collecting little information. According to our experiment, BAIR enjoys the least fraction of rejections compared to other baseline algorithms, i.e., higher user satisfaction about the recommendations.

References

Audibert, J.-Y.; and Bubeck, S. 2010. Best arm identification in multi-armed bandits.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2): 235–256.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1): 48–77.

Bahar, G.; Smorodinsky, R.; and Tennenholtz, M. 2015. Economic recommendation systems. *arXiv preprint arXiv:1507.07191*.

Carpentier, A.; and Locatelli, A. 2016. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Conference on Learning Theory*, 590–604. PMLR.

Cohen, J. D.; McClure, S. M.; and Yu, A. J. 2007. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481): 933–942.

Das, A. S.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, 271–280.

Daw, N. D.; O'doherty, J. P.; Dayan, P.; Seymour, B.; and Dolan, R. J. 2006. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095): 876–879.

Garivier, A.; and Kaufmann, E. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, 998–1027. PMLR.

Gershman, S. J. 2018. Deconstructing the human algorithms for exploration. *Cognition*, 173: 34–42.

Gopinath, D.; and Strickman, M. 2011. Personalized advertising and recommendation. US Patent App. 12/871,416.

Hoiles, W.; Krishnamurthy, V.; and Pattanayak, K. 2020. Rationally inattentive inverse reinforcement learning explains youtube commenting behavior. *Journal of Machine Learning Research*, 21(170): 1–39.

Komiyama, J.; Honda, J.; Kashima, H.; and Nakagawa, H. 2015. Regret lower bound and optimal algorithm in dueling bandit problem. In *Conference on learning theory*, 1141–1154. PMLR.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Kremer, I.; Mansour, Y.; and Perry, M. 2014. Implementing the "wisdom of the crowd". *Journal of Political Economy*, 122(5): 988–1012.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Linden, G.; Smith, B.; and York, J. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1): 76–80.

Mansour, Y.; Slivkins, A.; and Syrgkanis, V. 2020. Bayesian incentive-compatible bandit exploration. *Operations Research*, 68(4): 1132–1161.

Mansour, Y.; Slivkins, A.; Syrgkanis, V.; and Wu, Z. S. 2016. Bayesian exploration: Incentivizing exploration in bayesian games. *arXiv preprint arXiv:1602.07570*.

Richter, M. K. 1966. Revealed preference theory. *Econometrica: Journal of the Econometric Society*, 635–645.

Schafer, J. B.; Konstan, J.; and Riedl, J. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, 158–166.

Schnabel, T.; Bennett, P. N.; Dumais, S. T.; and Joachims, T. 2018. Short-term satisfaction and long-term coverage: Understanding how users tolerate algorithmic exploration. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 513–521.

Tennenholtz, M.; and Kurland, O. 2019. Rethinking search engines and recommendation systems: a game theoretic perspective. *Communications of the ACM*, 62(12): 66–75.

Tétard, F.; and Collan, M. 2009. Lazy user theory: A dynamic model to understand user selection of products and services. In *2009 42nd Hawaii International Conference on System Sciences*, 1–9. IEEE.

Villas-Boas, J. M. 2004. Consumer learning, brand loyalty, and competition. *Marketing Science*, 23(1): 134–145.

Wilson, R. C.; Geana, A.; White, J. M.; Ludvig, E. A.; and Cohen, J. D. 2014. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6): 2074.

Yao, F.; Li, C.; Nekipelov, D.; Wang, H.; and Xu, H. 2021. Learning the Optimal Recommendation from Explorative Users. *arXiv preprint arXiv:2110.03068*.

Yue, Y.; Broder, J.; Kleinberg, R.; and Joachims, T. 2012. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5): 1538–1556.

Yue, Y.; and Joachims, T. 2011. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 241–248. Citeseer.

Zhang, S.; and Angela, J. Y. 2013. Forgetful Bayes and myopic planning: Human learning and decision-making in a bandit setting. In *NIPS*, 2607–2615.

Zoghi, M.; Whiteson, S.; Munos, R.; and Rijke, M. 2014. Relative upper confidence bound for the k-armed dueling bandit problem. In *International conference on machine learning*, 10–18. PMLR.