Online Recommendations for Agents with Discounted Adaptive Preferences

Arpit Agarwal* William Brown†

October 26, 2022

Abstract

For domains in which a recommender provides repeated content suggestions, agent preferences may evolve over time as a function of prior recommendations, and algorithms must take this into account for long-run optimization. Recently, Agarwal and Brown (2022) introduced a model for studying recommendations when agents' preferences are adaptive, and gave a series of results for the case when agent preferences depend *uniformly* on their history of past selections. Here, the recommender shows a k-item menu (out of n) to the agent at each round, who selects one of the k items via their history-dependent *preference model*, yielding a per-item adversarial rewards for the recommender.

We expand this setting to *non-uniform* preferences, and give a series of results for γ-*discounted* histories. For this problem, the feasible regret benchmarks can depend drastically on varying conditions. In the "large γ" regime, we show that the regret benchmark from Agarwal and Brown (2022) (the "EIRD set") is attainable for any *smooth* model, relaxing their "local learnability" condition. We introduce "pseudo-increasing" preference models, for which we give an algorithm which can compete against any item distribution with small uniform noise (the "smoothed simplex"). We show NP-hardness results for larger regret benchmarks in each case. We give another algorithm for pseudo-increasing models (under a restriction on the adversarial nature of the reward functions), which works for any γ and is faster when γ is sufficiently small, and we show a super-polynomial regret lower bound with respect to EIRD for general models in the "small γ" regime. We conclude with a pair of algorithms for the memoryless case.

1 Introduction

Today, recommendation systems are an integral part of online platforms for e-commerce, social networks, and content sharing. It has been well-documented that user preferences change over time in response to content recommendations (see e.g. Curmei et al. (2022)). which can lead to a self-reinforcing patterns of content consumption and can have a variety of unintended consequences for the user, platform and society; including but not limited to loss of revenue for the platform, or creation of "filter-bubbles" and "echo-chambers" that drive polarization in society. Hence, it becomes important for recommendation systems to incorporate such self-reinforcing patterns into user modeling.

Motivated by this, Agarwal and Brown (2022) introduce a model for adaptive user preferences that depend on the history of interaction with the recommender. In this problem formulation, the recommender is faced with multi-armed bandit task where items cannot be chosen directly, but rather are selected by an agent with an *adaptive preference model* from a menu shown by the recommender. This interaction happens repeats for T rounds, and the reward for each item at any given round may be adversarial. The goal of the recommender is to minimize its regret with respect to a suitable benchmark; Agarwal and Brown (2022) identify a feasible regret benchmark (which roughly corresponds to the set of high-entropy item distributions), while showing linear regret lower bounds for other benchmarks.

In the model of Agarwal and Brown (2022), the agents' preferences depend uniformly on the sequence of past interactions, i.e. recent selections are given the same weight for preferences as past selections. However,

*arpit.agarwal@columbia.edu

†w.brown@columbia.edu

there is much evidence from psychology that humans are recency-biased, which should be accounted for in models of human behavior (see Curmei et al. (2022) for an overview).

We expand the model of Agarwal and Brown (2022) by considering agent preferences which depend non-uniformly on their history of past interactions with the recommender, in which the influence of past interactions is discounted by a factor of $\gamma < 1$ after each round. Here, we study regret minimization for the online recommendations problem for adaptive agents across the landscape of memory horizons under this new model, ranging from when the memory horizon can be $o(T)$ down to when agents are memoryless. and provide a series of algorithmic and barrier results for a variety of preference model classes and regret benchmarks.

Below, we formally describe this problem setting and outline our main contributions.

1.1 Problem Description

In each round $t \in [T]$, the recommender must choose k out of n items to show in a menu to the agent, who will choose one item probabilistically according to their *preference model* M. Under this model, the choice probability of each item depends on a *memory vector* v which encodes the sequence of prior item selections. Similar to the classical adversarial multi-armed bandit problem, we allow the sequences of rewards for each arm/item to be adversarially chosen. However, the main difference in our problem is that the recommender can only suggest a menu of items to the agents, and the agent will then "pull" the item according to its preference model. As a result, in general it will not be possible to do as well as the best individual item in terms of reward, as an agent may never choose an item more than once every k rounds even if scores are constant and it is shown every round. Hence, we must be careful in selecting benchmarks with which we can feasibly compete (in the form of a sublinear regret bound), and the appropriate benchmark can depend on the intricacies of how preferences are allowed to adapt over time.

Agarwal and Brown (2022) showed that one can compete against the best item distribution in the EIRD set for the case of uniform-memory preferences. Here, the EIRD set is a "core" subset of item distributions which can be instantaneously realized regardless of the current memory vector via an appropriate randomized menu selection process. Negative results are shown for the feasibility of other benchmarks, such as the resulting stationary distribution over items when a fixed menu distribution is used in every round. Our results consider both of these benchmarks, and we introduce a new benchmark which we term the "ϕ-smoothed simplex" $\Delta^{\phi}(n)$, which is the set obtained when each item distribution in the standard simplex $\Delta(n)$ is mixed with mass $\phi(1 + o(1))$ on the uniform distribution. In Section 3.1 we give additional characterization of the structure of the EIRD set which highlights the cases in which it can be restrictive, such as when it is impossible to persuade an agent to highly favor a particular item. We identify a structural condition on preference models which precludes this, which we term the "pseudo-increasing" property, which aligns with many common settings where exposure to an item tends to increase one's fondness for it (see e.g. Curmei et al. (2022)). Item scoring functions with this property can still be non-monotone, non-convex, and dependent on many items in intricate ways, yet are constrained to within some multiplicative factor of a linear increasing function of that item's weight in memory. When this property is satisfied, we that show Δ^{ϕ} becomes feasible as a benchmark; under appropriate settings of other parameters, we can take $\phi = O(\frac{1}{n})$, in which case every item distribution in $\Delta(n)$ is relatively close to some point in the ϕ-smoothed simplex.

1.2 Overview of Results

We give a series of algorithmic and barrier results for this problem under the γ-discounted setting. This coincides exactly with the uniform memory case of Agarwal and Brown (2022) when $\gamma = 1$; on the whole, our results illustrate the regret benchmarks against which one can compete may change drastically depending on the uniformity of memory and structural conditions on the preference model.

Our first key algorithmic contribution (Section 3) considers the case when γ is quite close to 1 (bounded away by $o(1/T)$), and the "effective horizon" for memory is large (but still $o(T)$). Here, we show that the "local learnability" assumption (i.e. that preference models can be estimated accurately everywhere by only querying points in a small region) required by Agarwal and Brown (2022) is no longer needed. It suffices for us to only assume that preference scoring functions are smooth at some constant order (and that their sum is smooth), as we will always be able to maintain a preference model hypothesis which is locally accurate

Memory	Preferences	Benchmark
Uniform ($\gamma = 1$)	Locally Learnable	EIRD
$\gamma = 1 - o(\frac{1}{T})$	Smooth	EIRD
$\gamma = 1 - o(\frac{1}{T})$	Smooth + P-I	Δ^ϕ
$\gamma \in [0, 1)$	P-I	Δ^ϕ $^{(*)}$
Memoryless	Fixed	Menus

Table 1: Sufficient conditions on memory rates and preference models, and their corresponding target benchmarks, for which where $o(T)$ regret is achievable. The first row is due to Agarwal and Brown (2022), and the remainder are contributions of this work. "P-I" denotes that a preference model is pseudo-increasing, and Δ^ϕ denotes the ϕ-smoothed simplex, both of which are introduced in Section 3.4. $^{(*)}$ denotes a restriction on the adversarial nature of rewards, namely that the reward distribution cannot change more than once every $o(T)$ rounds.

in the neighborhood of the memory vector, which cannot change too quickly. We give an algorithm which obtains $o(T)$ regret with respect to EIRD in this case; when the preference model satisfies the additional "pseudo-increasing" assumption we give a similar algorithm which can compete against Δ^ϕ. We also give a characterization of IRD sets which enables efficiently constructing sparse menu distributions. In each of these cases we also give a negative result that substantially larger regret benchmarks are infeasible, by reducing a NP-complete problem to the problem of computing the best stationary item distribution from fixed menu distributions.

On the other extreme, we consider the case where $\gamma \in [0, 1)$ is some fixed constant, and so the effective memory horizon is $O(1)$ with respect to T. This includes cases where the set of feasible memory vectors is essentially discrete, as there may be a finite grid of well-separated points which all feasible vectors, such as when $\gamma = 0$ and the memory vector always has its entire mass on the most recently selected item. Despite this difficulty, we that the pseudo-increasing condition is sufficient (even without smoothness) to compete with Δ^ϕ, and we give an algorithm for this case which obtains sublinear regret with respect to Δ^ϕ for $\gamma \in [0, 1)$, and with a faster rate than the previous algorithm when γ is sufficiently small. The presence of additional structural condition (such as the pseudo-increasing property) on the scoring functions are necessary; we show a super-polynomial query learning lower bound for general smooth models, which corresponds to a regret lower bound holding even with respect to EIRD for any algorithm which aims to learn the preference model.

Finally, we analyze the "memoryless" case when preferences are fixed, and give an algorithm which obtains the optimal $\tilde{O}(\sqrt{T})$ rate yet depends exponentially on k, as well as a $\tilde{O}(T^{2/3})$ algorithm where the dependence on all other parameters is polynomial.

Overall, our contributions range from proposing a model for non-uniform adaptive agents and identifying suitable regret benchmarks, to understanding the landscape of feasibility for a range of values of γ that interpolate between the uniform case and the memoryless case. We summarize all our results in Tables 1 and 2.

1.3 Related Work

Stochastic bandits with changing rewards: The stochastic multi-armed bandit problem has been studied in settings where the reward distributions change as a function of arm pulls (Gittins, 1979; Heidari et al., 2016; Levine et al., 2017; Kleinberg and Immorlica, 2018; Leqi et al., 2021; Laforgue et al., 2022; Awasthi et al., 2022; Papadigenopoulos et al., 2022). Most recent work on this problem has focused on specific models for reward evolution that are motivated by user satiation, user boredom, congestion etc. There are several crucial differences between our setting and this setting. Firstly, we consider adversarial rewards, whereas this setting considers stochastic rewards that evolve according to a fixed dynamics. Secondly, we consider a recommendation setting where the learner plays a menu of arms, and the user choice behavior over menus evolves over time. In contrast, these works consider a classic bandit setting where the learner plays a single

Memory	Preferences	Barrier
Uniform	Locally Learnable	Menus [(1)]
Uniform	Non-L.L.	EIRD [(2)]
$\gamma = 1 - o(\frac{1}{T})$	Smooth	Menus [(3)]
$\gamma = 1 - o(\frac{1}{T})$	Smooth + P-I	Menus [(3)]
$\gamma \in (0, \frac{1}{2})$	Smooth	EIRD [(2)]

Table 2: Conditions on memory rates and preference models for which formal barriers exist against competing with particular benchmarks for regret. The uniform barriers are due to Agarwal and Brown (2022), with the rest from this work. Here, [(1)] denotes a linear regret lower bound, [(2)] denotes a superpolynomial runtime lower bound for model learning via queries, and [(3)] denotes NP-hardness.

arm.

Models of preference dynamics: There has also been substantial work in understanding preference dynamics in recommendation systems (Hazla et al., 2019; Gaitonde et al., 2021; Dean and Morgenstern, 2022). Hazla et al. (2019); Gaitonde et al. (2021) studied a model for preference dynamics where preferences are represented by vectors and these vectors drift towards the direction of agreement/disagreement on a randomly drawn issue. Dean and Morgenstern (2022) studied a similar model, but in the context of personalized recommendations of a single item. Even though these papers consider preference dynamics similar to our work, their goal is to study conditions under which polarization occurs. In contrast, our goal is to minimize regret for the platform under these dynamically evolving preferences. Restricted history has also been identified by Schneider and Vodrahalli (2022) as a useful property for modeling agent behavior with online learning algorithms.

Reinforcement Learning: There has also been work on using reinforcement learning for recommendations in order to maximize long-term rewards (Ie et al., 2019; Zhan et al., 2021; Chen et al., 2019). However, either these works do not consider models of preference evolution or lack rigorous theoretical guarantees on the achieved performance. We consider a very general model for preference evolution as compared to these works, and provide a rigorous treatment of achievability/hardness under various benchmarks. Moreover, it does not seem possible to frame our model as a Markov decision process (MDP) and simultaneously achieve good regret guarantees using off-the-shelf reinforcement learning.

Dueling Bandits: The "dueling bandits" framework studies a recommendation problem similar to ours where the learner plays more than one arm in each trial and observes relative feedback between arms (Yue and Joachims, 2009; Yue et al., 2012; Agarwal et al., 2020; Rangi et al., 2021). However, in contrast to our setting, these works consider preference models which are fixed *a priori*, and do not change as a function of item history.

We provide additional related work in the appendix.

1.4 Organization

We introduce the problem setting and key definitions in Section 2. In Section 3, we study the "large γ" regime where memory changes slowly, and in Section 4 we consider the "small γ" regime where memory vectors may change quickly, giving algorithmic and barrier results in each case. In Section 5 we give algorithms for the memoryless case. We defer most proofs to the appendix.

2 Model And Preliminaries

2.1 Setting

We recall the standard setup from Agarwal and Brown (2002) for the online recommedations problem for an agent with adaptive preferences. At any time, there is some *memory vector* $v \in \Delta(n)$, which expresses some

function of the prior selections of the agent. The *preference model* of an agent is a mapping $M : \Delta(n) \to [0,1]^n$ which assigns scores $M(v)_i = f_i(v)$ according to preference functions $f_i : \Delta(n) \to [0,1]$ for each item. In an instance of this problem, there is a universe of n items, a menu size $k < n$, a preference model M, a memory update rule U, and a sequence of linear reward functions $r_1, \ldots, r_T$ for the recommender. In each round $t \in \{1, \ldots, T\}$:

- The recommender chooses a menu K_t, consisting of k distinct items from $[n]$, which is shown to the agent;
- The agent selects one item $i_t \in K_t$, chosen at random according to the distribution given by:

$$p_{K_t, v_t, i} = \frac{f_i(v_t)}{\sum_{j \in K_t} f_j(v_t)};$$

- The memory vector is updated to $v_{t+1} = U(v_t, i_t, t)$ by the update rule;
- The recommender observes receives reward $r_t(e_{i_t})$ for the chosen item, where e_{i_t} is the unit vector for i_t.

The goal of the recommender is to minimize their regret over T rounds with respect to some benchmark set.

For any preference model M and current memory vector v, the set of item choice distributions which can be instantaneously realized by showing the agent a menu sampled from some distribution is denoted $\mathrm{IRD}(v, M)$, or simply the IRD set, and is given by

$$\mathrm{IRD}(v, M) = \operatorname*{convhull}_{K \in \binom{n}{k}} p_{K,v},$$

where $p_{K,v}$ denotes agent's item selection distibution conditioned on being shown a menu K, given by

$$p_{K,v,i} = \frac{f_i(v)}{\sum_{j \in K} f_j(v)}$$

for each item i in K (and 0 otherwise), with f_i being the preference scoring function for item i. The set of item distributions which are instantaneoulsy realizable *everywhere* (i.e. from any memory vector), denoted EIRD, is given by

$$\mathrm{EIRD}(M) = \bigcap_{v \in \Delta(n)} \mathrm{IRD}(v, M).$$

To ensure that EIRD is non-empty, we require that scoring functions f_i are in fact bounded in the range $[\lambda, 1]$ for some constant λ, satisfying $\lambda \geq \frac{k^2}{n}$ in the general case or $\lambda \geq \frac{\sigma k^2}{n}$ for (σ, λ)-pseudo-increasing functions (see Section 3.4 for the full definition of pseudo-increasing functions).

2.2 Discounted Memory Agents

Throughout, with the exception of Section 5, we consider agents whose memory update rules are γ-*discounted*.

Definition 1 (Discounted Memory Updating)**.** *Under the γ-discounted memory update rule U_γ, for some $\gamma \in [0,1]$, when an item i_t is selected at round t, the memory vector v_t is updated to $v^{t+1} = U_\gamma(v_t, i_t, t)$, with*

$$v_i^{t+1} = \frac{\sum_{s=1}^{t} \gamma^{t-s} \cdot x_i^s}{\sum_{s=1}^{t} \gamma^{t-s}},$$

As in other problem settings involving discount factors (such as MDPs), we can think of larger values of γ as corresponding to having larger "effective horizons" for memory; the case where $\gamma = 1$ is equivalent to the uniform memory setting from Agarwal and Brown (2022).

2.3 Smooth Preference Models

Many of our results consider preference models with the key property that each scoring function is *smooth* at some constant order d, in addition to being bounded above 0.

Definition 2 (Smooth Preference Models)**.** *A preference model M is (λ, d, L)-smooth if each scoring function f_i takes values in $[\lambda, 1]$, is $d+1$-times continuously differentiable and has dth order partial derivatives which are L-Lipschitz over $\Delta(n)$, and if the sum of the scoring functions $\sum_i f_i(v)$ is a constant.*

This property allows for quite a broad class of functions, and is satisfied by each of the example locally learnable classes in Agarwal and Brown (2022). Despite its generality, we show in Section 3 that this assumption alone is sufficient to enable us to always maintain an accurate *local* approximation of the model, provided that memory does not change too quickly, by implementing a query learning rountine for a locally learnable class (namely, for bounded-degree multivariate polynomial models). We view the requirement on the constant sum of functions as being well-justified by psychological phenomena (see e.g. Curmei et al. (2022)), and minimally restrictive, as it merely enforces normalization of scores; agents may rapidly gain interest in certain items based on recent selections, which typically corresponds with a drop in interest for other items.

2.4 Other Preliminaries

We assume the reader is familiar with basic terminology regarding bandit and online convex optimization; see e.g. Hazan (2019) for an overview. We make use of the following result from Agarwal and Brown (2022), which gives a bandit linear optimization algorithm for contracting sets.

Theorem (Agarwal and Brown (2022))**.** *There is an algorithm* RC-FKM *for bandit linear optimization which obtains $O(T^{3/4})$ regret against the best point in the final set in a length-T sequence of contracting convex action sets, with actions taken from the set revealed at each round.*

Throughout, we use $\Delta(n)$ to denote the simplex over n items, $d_{TV}(v, v')$ to denote the total variation distance between distributions; we make use of various norms, which are labeled explicitly (e.g. $\|x\|_1$) if not clear from context.

3 Agents with Long Memory Horizons

We begin by considering cases when γ is close to 1, and the effective horizon for memory is large, i.e. memory can only be truncated to some sequence of prior rounds with length at least $o(T)$ without drastically changing the resulting vector. Here, this implies that memory vectors do not change too quickly, and further that every point in Δ^n is *feasible* up to small error by *some* appropriate sequence of item selections (if we were able to dictate the agent's item choices).

In Sections 3.1 and 3.2 we give results on the structure of IRD sets and the local approximability of smooth models, respectively, which enable key subroutines of our algorithms in Sections 3.3 and 3.4. In Section 3.5

3.1 Characterizing IRD via Menu Times

We introduce a notion of the *menu time* required by each item in order to induce a particular item distribution x from a memory vector v, which enables us to more directly characterize the set of feasible item distributions, as well as avoid the exponential dependence on k in runtime from the linear programming construction of menu distributions in Agarwal and Brown (2022). The menu time quantities μ_i are normalized such that $\sum_i \mu_i = k$, and satisfy:

$$\mu_i \propto \frac{x_i}{f_i(v)},$$

We show that an item distribution x can be realized from a memory vector v if and only if $\max_i \mu_i \leq 1$.

Lemma 1. *An item distribution x belongs to $\mathrm{IRD}(v, M)$ if and only if we have that the menu time μ_i for each item is at most 1. If this condition holds, there is a $O(n/(k\epsilon))$ time algorithm for constructing a menu distribution z, with positive mass on at most $O(\frac{n}{k^2\epsilon})$ menus, which results in an induced item distribution $\hat{x}$ satisfying $\|x - \hat{x}\|_\infty \leq \epsilon$.*

Proof Sketch. For any $x \in \mathrm{IRD}(v, M)$, we consider an arbitrary convex combination of the menu-conditional item distributions given by items' scores $f_i(v)$, and show that converting this menu distribution into menu times by "charging" each menu in proportion with the inverse of the sum of its item scores results in a menu time vector satisfying $\|\mu\|_1 = k$ and $\|\mu\|_\infty \leq 1$.

Given a menu time vector satisfying these conditions, we can construct such a distribution by greedily choosing the k items with highest remaining menu time, and decreasing the remaining menu time of the selected items by ϵ. The mass of each added menu in our final distribution z will be inversely proportional to the sum of scores of items in the menu. This allows cancellation of the terms for sums of menu scores, resulting in a menu distribution where the selection probability of an item is proportional to its score $f_i(v)$ and the number of stages in which it was added to the menu. As the number of stages an item is added to a menu is proportional to its menu time, and its menu time is proportional to $x_i/f_i(v)$, the resulting induced item distribution is simply proportional to x_i as desired. □

3.2 Bounded Local Learnability of Smooth Models

Here we show that any smooth preference model can be well-approximated over a bounded range by a model comprised of multivariate polynomials with degree d. As such, we can directly use the multivariate polynomial local learning method from Agarwal and Brown (2022) to find a hypothesis which is accurate over a bounded radius. When γ is sufficiently large, it is possible to spend $o(T)$ time total in "learning stages", while always maintaining an accurate representation of the preference model in the neighborhood around the current memory vector, enabling the implementation of regret minimization algorithms over the space of item distributions for $T - o(T)$ rounds.

Lemma 2. *For any point $x \in \Delta(n)$ and any (λ, d, L)-smooth preference model M, there is a memory model $\tilde{M}$ where each scoring function $\tilde{f}_i$ is a degree-d polynomial, and which approximates the score vectors of M to within accuracy β for any point $y \in B_\alpha(x) \cap \Delta(n)$, for any $\beta \geq Ln^{d+1}\alpha^{d+1}$, with $\sum_i \tilde{f}_i$ in the range $[C - n\beta, C + n\beta]$ for some constant C.*

Proof. By Taylor's theorem, for each f_i we have that

$$\begin{aligned} f_i(y) &= \sum_{|\chi| \leq d} \frac{\partial^\chi f(x)}{\chi!}(y - x)^\chi + \sum_{|\xi| = d+1} R_\xi(y)(y - x)^\xi \\ &= \tilde{f}_i(y) + \sum_{|\xi| = d+1} R_\xi(y)(y - x)^\xi, \end{aligned}$$

where χ is a multi-index with ∂^χ representing its corresponding partial derivative operator, where $\tilde{f}_i$ is a degree-d polynomial, and where the magnitude of each $R_\xi(y)$ term is bounded by

$$|R_\xi(y)| \leq \frac{1}{\xi!} \max_{|\chi| = |\xi|} \max_{z \in \Delta(n)} \left|\partial^\chi f(z)\right|.$$

By the dth order Lipschitz condition, we can bound the magnitude of each $(d+1)$th order partial derivative $|\partial^\chi f(z)|$ by L, and so for any $y \in B_\alpha(x) \cap \Delta(n)$ we have that

$$\begin{aligned} \left|f_i(y) - \tilde{f}_i(y)\right| &\leq \sum_{|\xi| = d+1} \frac{L}{\xi!}(y - x)^\xi \\ &\leq Ln^{d+1}\alpha^{d+1} \\ &\leq \beta. \end{aligned}$$

This also yields that the sum of each $\tilde{f}_i$ within $n\beta$ from the sum of each f_i, which is some constant C. □

3.3 Targeting EIRD

We are now ready to describe our algorithm for obtaining sublinear regret with respect to EIRD for agents with smooth preference models. Our objective will be to alternate between "learning" and "optimizing" stages, following a burn-in period to ensure memory concentration, such that our memory vector v_t always remains within distance ρ from the center v^* of our previous learning stage, enabling regret minimization during the optimization stages which will constitute $T - o(T)$ steps in total. As a subroutine, we make use of the algorithm from Lemma 2 in Agarwal and Brown (2022), which enables local learning of degree-d multivariate polynomial preference models.

Theorem 1. *For any agent with a (λ, d, L)-smooth preference model and γ-discounted memory for $\gamma \geq 1 - \frac{1}{g(T)}$, where $g(T) = o(T)$, Algorithm 1 obtains regret $o(T)$ with respect to* $\texttt{EIRD}(M)$.

Algorithm 1 (Targeting EIRD for Smooth Models).

Run $\texttt{BurnIn}(\gamma)$ for $t_0 = o(T)$
Initialize RC-FKM to run for $T - o(T)$ rounds
while $t < T$ **do**
  Let $v^* = v_t$
  Get set of points $S^* = \texttt{Queries}(v^*, \alpha)$ to query
  for $s \in S^*$ **do**
    Run $\texttt{QueryPad}(s)$
    Run $\texttt{Sample}(s, \beta)$, observe $\hat{f}_i(s)$ for each i
  end for
  Let $M^* = \texttt{FitModel}(\{\hat{f}_i(s) : i \in [n], s \in S^*\})$
  while $v_t \in B_\rho(v^*)$ **do**
    Get x_t from RC-FKM
    Let $z_t = \texttt{MenuDist}(v_t, x_t, M^*)$
    Sample menu $K_t \sim z_t$, update RC-FKM
  end while
end while

We defer discussion on precise tradeoffs between rates of $g(T)$ versus regret, as well as concrete details of each subroutine, to the appendix. If $g(T) = O(T^{1/c_1})$ and d is a constant, then we can obtain regret $O(T^{1-1/c_2})$ for constants $c_1, c_2 > 1$. We also discuss regimes where slightly modifying the algorithm (e.g. by switching between `QueryPad` and `Sample` more often) can improve rates.

We show in Section 3.5 that it is likely difficult to strictly beat EIRD without additional structural assumptions beyond smoothness; in Section 3.4 we explore such an assumption which enables us to run a similar algorithm, yet compete against a much larger set of item distributions.

3.4 Pseudo-Increasing Functions and $\Delta^\phi(n)$

A major motivation of Agarwal and Brown (2022) for considering EIRD was due to the difficulties of exploration uniform memory, as the current history cannot be repeatedly "washed away" without requiring exponentially many rounds. However considering discount factors of $\gamma < 1$ introduces the possibility that we can efficiently explore the space of feasible vectors and compete against item distributions which lie outside of EIRD, i.e. item distributions which may require the memory vector to lie in a small region in order to be feasible. We identify a structural property which enables this, namely that scoring functions are *pseudo-increasing*.

Definition 3 (Pseudo-Increasing Scoring Functions)**.** *A scoring function $f_i : [0,1]^n \to [\frac{\lambda}{\sigma}, 1]$ is (σ, λ)-pseudo-increasing for $\sigma \geq 1$ and $\lambda > \frac{\sigma k^2}{n}$ if*

$$\frac{(1-\lambda)v_i + \lambda}{\sigma} \leq f(v) \leq \sigma((1-\lambda)v_i + \lambda).$$

We say that a preference model M is pseudo-increasing if, for every item i, the scoring function f_i is pseudo-increasing. When this property is satisfied, it becomes possible to induce any item distribution which is in the neighborhood of the current memory vector v_t, provided that v_t is not too close to the boundary of the simplex. This motivates a target regret benchmark of all such points which are not too close to the boundary in any dimension, which we term the *ϕ-smoothed simplex*.

Definition 4 (ϕ-Smoothed Simplex)**.** *Let b^i_ϕ denote the "smoothed basis vector" for item i, where $b^i_{\phi,i} = 1-\phi$, and $b^i_{\phi,j} = \frac{\phi}{n-1}$ for every other item j. We define the ϕ-smoothed simplex $\Delta^\phi(n)$ as the convex hull of the n smoothed basis vectors.*

This is equivalent to the set which results when any point in the standard simplex is mixed with $\frac{\phi n}{n-1}$ uniform noise. It may be natural to think of σ and k as either large constants, or functions which grow slowly in n, in which case λ can be taken small enough such that $\phi = \tilde{O}(\frac{1}{n})$. For large n, this presents a stark constrast with EIRD, as it may be possible to persuade the agent to pick the best item in nearly every round, rather than in only a $O(1/k)$ fraction of rounds.

3.4.1 A No-Regret Algorithm for $\Delta^\phi(n)$

The key difference between Algorithm 1 and our approach here is that rather than taking a step with our copy of RC-FKM during every round of the optimization stage, we treat each optimization stage as a *single* gradient step, with our domain fixed as the smooth simplex rather than a contracting sequence of sets containing EIRD. We take advantage of the fact that, with appropriately calibrated step sizes, we can complete each step without exiting the ball in which our preference model estimate is ϵ-accurate. While we no longer need the "contracting" feature of RC-FKM, we still make use of it rather than other algorithms for bandit linear optimization due to its ability to tolerate imprecision in action specifications and rewards.

Theorem 2 (Pseudo-Increasing Discounted Regret Bound)**.** *For any agent with a preference model which is (σ, λ)-pseudo-increasing and $(\frac{\lambda}{\sigma}, d, L)$-smooth, and with γ-discounted memory for $\gamma \geq 1 - \frac{1}{g(T)}$ where $g(T) = o(T)$, Algorithm 2 obtains regret $o(T)$ with respect to $\Delta^\phi(n)$, for any $\phi \geq 2k\lambda\sigma^2$.*

Discussions on tradeoffs between precise rates, and their comparison with Algorithm 1, are deferred to the appendix.

Algorithm 2 (Targeting Δ^ϕ for Pseudo-Inc. Models).

Run `BurnIn`(γ) for $t_0 = o(T)$
Initialize RC-FKM to run for $o(T)$ rounds
while $t < T$ **do**
  Let $v^* = v_t$
  Get set of points $S^* =$ `Queries`(v^*, α) to query
  for $s \in S^*$ **do**
    Run `QueryPad`(s)
    Run `Sample`(s, β), observe $\hat{f}_i(s)$ for each i
  end for
  Let $M^* =$ `FitModel`$(\{\hat{f}_i(s) : i \in [n], s \in S^*\})$
  Get x^* from RC-FKM
  for $o(T)$ rounds such that $v_t \in B_\rho(v^*)$ **do**
    Let $z_t =$ `MenuDist`(v_t, x^*, M^*)
    Sample menu $K_t \sim z_t$
  end for
  Update RC-FKM with average reward from x^*
end while

3.5 Hardness for Alternate Benchmarks

We give a set of hardness results for both of the previous cases against the possibility of relaxing to larger benchmark sets. Each of these constructions proceeds by showing that an instance of the "Max Independent Set" problem, which is NP-hard to approximate, can be encoded in a smooth preference model for an agent, such that any strategy attaining a particular average reward corresponds to a solution to the problem.

Theorem 3. *Unless* RP $\supseteq$ NP*, there is no algorithm which runs in time polynomial in n and T and which obtains regret $o(T)$ for any $T = O(\text{poly}(n))$ against either the ϕ-smoothed simplex for any $\phi < 1/2$, the set of all menu distributions, or against the set of all item distributions contained in their own* IRD *set, for agents with arbitrary (λ, d, L)-smooth memory models.*

Additionally, for agents with arbitrary σ-pseudo-increasing and (λ, d, L)-smooth memory models there is no such algorithm which obtains regret $o(T)$ for any $T = O(\text{poly}(n))$ against the set of all menu distributions, or against the set of all item distributions contained in their own IRD *set.*

This gives evidence that competing with any benchmark is strictly better than the best point in EIRD will require additional structural assumptions on the preference model, as otherwise it is possible for the unique memory states which can beat the best point in EIRD, even for fixed rewards, to be computationally difficult to find.

4 Agents with Short Memory Horizons

When the discount factor of the agent is small enough that memory vectors may move rapidly, we lose the precision required by the algorithms in Section 3 in order to implement local learning, and in fact the feasible state space may more closely resemble a discrete grid, with memory vectors encoding the sequence of items chosen over an effective horizon which is constant with respect to T. Nonetheless, for pseudo-increasing models we give an algorithm which we call EXP-ϕ, which obtains $o(T)$ regret with respect to $\Delta^{\phi}(n)$ for *any* value of $\gamma \in [0, 1)$, with a few conditions. Here we assume that rewards are stochastic rather than adversarial for windows of length $o(T)$, but distributions may change adversarially between each window; we require a slightly larger lower bound on ϕ (yet still potentially $O(\frac{1}{n})$, and our regret rate will typically be worse than that obtained by Algorithm 2 when γ is large.

4.1 A No-Regret Algorithm for Pseudo-Increasing Models

The idea behind EXP-ϕ is to view each vertex of the smoothed simplex as an action for a multi-armed bandit problem, where each "pull" corresponds to several rounds. When we "commit" to playing an item in the menu for a sufficiently long time, while otherwise playing items with the smallest weight in memory, the pseudo-increasing property will ensure that the selection frequency of that item gravitates towards its vertex in the smoothed simplex. Further, as we are longer attempting to learn the preference model explicitly, we can relax the smoothness requirement, and so scoring functions may in fact be discontinuous.

Theorem 4. *For any agent with a preference model which is (σ, λ)-pseudo-increasing and $(\frac{\lambda}{\sigma}, d, L)$-smooth, and with γ-discounted memory for $\gamma \in [0, 1]$, when losses are drawn from a distribution which changes at most once every $O(1/(1-\gamma))$ rounds, Algorithm 2 obtains regret $\tilde{O}(\sqrt{T(1-\gamma)})$ with respect to $\Delta^{\phi}(n)$, for any $\phi \geq 4k^2\lambda\sigma^4$.*

We discuss comparisons between the rates for EXP-ϕ and Algorithm 2 in the appendix.

4.2 Barriers for General Models

If we cannot assume that preferences are pseudo-increasing, then it appears difficult to compete even against EIRD for arbitrary smooth models without additional structural assumptions. We give a barrier result in the form of a runtime lower bound for any algorithm which attempts to learn preference models via queries, and where locally accurate estimates are necessary to compete with EIRD, which holds even for fixed losses. For the instances we construct, there is a single optimal item, yet in order to play it with maximal frequency, one must identify the $k-1$ items with the lowest scores to also include with high frequency, yet this set of other

Algorithm 3 (EXP-ϕ).

Initialize EXP3 to run for $O(T(1-\gamma))$ steps
while $t < T$ **do**
  Sample arm i^* from EXP3
  for $O(1/(1-\gamma))$ rounds **do**
    Let $K_t = \{i^*\} + \text{argmin}_{j \neq i}^{k+1} v_j$
  end for
  Update EXP3 with average reward of i^*
end while

items may change completely and unpredictably across each round, as memory vector updates correspond to state transitions on a quasipolynomially large graph.

Theorem 5. *For any $\gamma \in (0, 1/2)$, there is a class of smooth preference models $\mathcal{M}$ for which any exact query algorithm must make $\Omega\left(n^{\log(n)}\right)$ to obtain a worst-case error bound of a constant c over the set of feasible memory vectors, and for which $o(T)$ regret with respect to* EIRD *cannot be obtained by any algorithm which does not have an estimate with error at most c of the preference scores at the current memory vector in at least $T - o(T)$ rounds.*

5 Memoryless Preferences

Finally, to complete the landscape of potential memory horizons, we consider the "memoryless" case where agent preferences are fixed, and independent of past selections. This variant of the problem is notably simpler, as per-round item distributions are fixed as a linear function of the menu distribution used, which will allow us to compete against the benchmark set of all fixed menus; however, it remains an interesting question as to whether the structure of the model can be exploited to improve dependence on parameters other than T, as enumerating all menus will be undesirable when k is large.

In the appendix, we give two $o(T)$ algorithms for this case. The first is a direct application of EXP3 to the set of all menus, which obtains optimal dependence on T but is exponential in k. We also give an algorithm which removes the dependence on k entirely in exchange for a weaker rate with respect to T, in which we estimate each preference score directly and conduct bandit linear optimization using the "menu time" distribution construction from Section 3.1.

Theorem 6. *For an agent with fixed preferences, there is an algorithm which obtains regret $\tilde{O}(\sqrt{Tn^k})$, as well as an algorithm which obtains regret $\tilde{O}(T^{2/3}\,\text{poly}(n))$ with polynomial time per-round computation.*

We leave open the question as to whether the optimal $\tilde{O}(\sqrt{T})$ rate and improved dependence on k can be obtained simultaneously.

References

Arpit Agarwal and William Brown. Diversified recommendations for agents with adaptive preferences. In *To appear in NeurIPS*, 2022.

Arpit Agarwal, Nicholas Johnson, and Shivani Agarwal. Choice bandits. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18399–18410. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d5fcc35c94879a4afad61cacca56192c-Paper.pdf.

Pranjal Awasthi, Kush Bhatia, Sreenivas Gollapudi, and Kostas Kollias. Congested bandits: Optimal routing via short-term resets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1078–1100. PMLR, 2022.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. Top-k off-policy correction for a REINFORCE recommender system. In J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman, editors, *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 456–464. ACM, 2019. doi: 10.1145/3289600.3290999. URL https://doi.org/10.1145/3289600.3290999.

Mihaela Curmei, Andreas A. Haupt, Benjamin Recht, and Dylan Hadfield-Menell. Towards psychologically-grounded dynamic preference models. In Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge, editors, *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 35–48. ACM, 2022.

Sarah Dean and Jamie Morgenstern. Preference dynamics under personalized recommendations. In David M. Pennock, Ilya Segal, and Sven Seuken, editors, *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, pages 795–816. ACM, 2022.

Jason Gaitonde, Jon M. Kleinberg, and Éva Tardos. Polarization in geometric opinion dynamics. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 499–519. ACM, 2021.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019. URL http://arxiv.org/abs/1909.05207.

Jan Hazla, Yan Jin, Elchanan Mossel, and Govind Ramnarayan. A geometric model of opinion polarization. *CoRR*, abs/1910.05274, 2019.

Hoda Heidari, Michael J. Kearns, and Aaron Roth. Tight policy regret bounds for improving and decaying bandits. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1562–1570. IJCAI/AAAI Press, 2016. URL http://www.ijcai.org/Abstract/16/224.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2592–2599. ijcai.org, 2019. doi: 10.24963/ijcai.2019/360. URL https://doi.org/10.24963/ijcai.2019/360.

Robert Kleinberg and Nicole Immorlica. Recharging bandits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 309–319. IEEE Computer Society, 2018. doi: 10.1109/FOCS.2018.00037. URL https://doi.org/10.1109/FOCS.2018.00037.

Pierre Laforgue, Giulia Clerici, Nicolò Cesa-Bianchi, and Ran Gilad-Bachrach. A last switch dependent analysis of satiation and seasonality in bandits. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 971–990. PMLR, 28–30 Mar 2022.

Liu Leqi, Fatma Kilinç-Karzan, Zachary C. Lipton, and Alan L. Montgomery. Rebounding bandits for modeling satiation effects. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4003–4014, 2021.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3074–3083, 2017.

Orestis Papadigenopoulos, Constantine Caramanis, and Sanjay Shakkottai. Non-stationary bandits under recharging payoffs: Improved planning with sublinear regret. *CoRR*, abs/2205.14790, 2022. doi: 10.48550/arXiv.2205.14790. URL `https://doi.org/10.48550/arXiv.2205.14790`.

Anshuka Rangi, Massimo Franceschetti, and Long Tran-Thanh. Sequential choice bandits with feedback for personalizing users' experience, 2021. URL `https://arxiv.org/abs/2101.01572`.

Jon Schneider and Kiran Vodrahalli. History-restricted online learning, 2022. URL `https://arxiv.org/abs/2205.14519`.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1201–1208, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553527.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2011.12.028. URL `https://www.sciencedirect.com/science/article/pii/S0022000012000281`. JCSS Special Issue: Cloud Computing 2011.

Ruohan Zhan, Konstantina Christakopoulou, Ya Le, Jayden Ooi, Martin Mladenov, Alex Beutel, Craig Boutilier, Ed H. Chi, and Minmin Chen. Towards content provider aware recommender systems: A simulation study on the interplay between user and provider utilities. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3872–3883. ACM / IW3C2, 2021. doi: 10.1145/3442381.3449889. URL `https://doi.org/10.1145/3442381.3449889`.