Disposable Linear Bandits for Online Recommendations

Melda Korkut, Andrew Li
Tepper School of Business
Carnegie Mellon University
{melda,aali1}@cmu.edu

Abstract

We study the classic stochastic linear bandit problem under the restriction that each arm may be selected for limited number of times. This simple constraint, which we call disposability, captures a common restriction that occurs in recommendation problems from a diverse array of applications ranging from personalized styling services to dating platforms. We show that the regret for this problem is characterized by a previously-unstudied function of the reward distribution among optimal arms. Algorithmically, our upper bound relies on an optimism-based policy which, while computationally intractable, lends itself to approximation via a fast alternating heuristic initialized with a classic similarity score. Experiments show that our policy dominates a set of benchmarks which includes algorithms known to be optimal for the linear bandit without disposability, along with natural modifications to these algorithms for the disposable setting.

Introduction

Consider the stochastic linear bandit problem: given a set of "arms" $a_1, \ldots, a_K \in \mathbb{R}^d$, the task is to select a single arm in each of a sequence of time periods so as to maximize the total reward gained. The rewards are random, but there exists some (unknown) $\theta^* \in \mathbb{R}^d$ such that the mean reward when an arm a_i is chosen is equal to $\langle a_i, \theta^* \rangle$. Algorithms known to be theoretically optimal (in a sense we will review later on) for this problem successfully balance the trade-off between arms that are known to yield high reward, with arms that may reveal more information about the unknown θ^*.

These algorithms are now a core component of most modern recommender systems, as the linear bandit model provides an extremely close fit to a common problem in recommendations: the *cold-start* problem. Succinctly, this is the problem of making recommendations to newer users whose amount of activity is insufficient to accurately estimate their preferences. To map the cold-start problem to the linear bandit model: the a_i's encode the "features" for each item, which have been previously estimated using past data, and θ^* encodes the unknown features of the new user. The linear form of the mean reward matches the underlying preference model of a variety of recommendation algorithms including those based on matrix factorization.

The key problem we seek to address is for a particular subset of recommender systems in which the number of recommendations of an item made to a person *may not exceed a certain limit*, a property we will refer to as *disposability*. One natural example of disposability in practice is personalized styling services (e.g. Stitch Fix, BeautyFIX, Trunk Club) which by and large operate by sending personalized "boxes" of items to users. Users can choose to purchase any subset of the box, and these companies do not re-send an item to the same user. Another application area is online dating where some online dating platforms do not want to show the same person multiple times to a user.

This Paper: Thus motivated, we study the linear bandit problem under the additional rule that each arm may be selected at most a certain amount for each user. Our primary observation is that this disposability constraint impacts both (a) the performance of algorithms known to be optimal when disposability is not imposed, and (b) the nature of algorithms that achieve meaningful theoretical (regret) guarantees. Our specific contributions are as follows:

1. We prove a minimax lower bound of $\Omega(\phi d\sqrt{T})$ on the regret of the disposable linear bandit problem which suggests that disposability should allow for lower regret than the standard linear bandit. In particular, while the dependence on the dimensionality d and time horizon T remains the same as in the standard linear bandit, a new term $\phi \in [0, 1]$ loosely captures the 'spread' of the optimal set of arms.
2. We propose an optimism-based algorithm, which generalizes the optimal upper confidence bound algorithm (called *LinUCB*) for the standard linear bandit. We prove an upper bound of $O(\sqrt{\gamma} d\sqrt{T})$ on its regret, where γ is a parameter that again encodes a notion of the spread of the optimal arm set. Noting that our algorithm involves a computationally hard optimization formulation, we propose a fast alternating heuristic. While this heuristic is not guaranteed to find the true optimum, we observe that it performs well in practice when combined with a clever initialization based on similarity scores.
3. We evaluate our algorithm's performance on a recommendation task based on synthetically generated data. Compared to a number of benchmarks, including LinUCB and

a natural modification of Thompson sampling, our algorithm (solved via the preceding heuristic) achieves as much as 10% lower regret against all competing algorithms.

Previous Work: There are mainly two types of filtering recommender systems use, content based filtering such as in (Gómez Hidalgo et al. 2006), (Lang 1995) and (Mooney and Roy 2000) and collaborative filtering (Sarwar et al. 2001). Collaborative filtering makes use of the idea that similar people will also prefer similar products.

The problem of recommendation becomes more complicated when a new user is in the network, which is called the *cold-start problem*. (Bobadilla et al. 2012) , (Wei et al. 2017), (Lika, Kolomvatsos, and Hadjiefthymiades 2014) and many more focused especially on algorithms improving this problem. Our paper also contributes to this stream of research, albeit using bandit algorithms.

Another method used to learn about user preferences is Multi-Armed Bandits. The classical multi-armed bandit problem was introduced by (Robbins 1952). The problem is well studied for a couple of decades now, pioneered and put together by (Gittins 1979), who introduced the all-time famous *Gittins Index*. The known best algorithm for the classic simple MAB is the *Upper Confidence Bound (UCB)* algorithm by (Lai and Robbins 1985). How to use certain MAB techniques for recommendations are studied before, such as in (Bouneffouf, Bouzeghoub, and Gançarski 2012), (Zeng et al. 2016). In recommendations setting, however, learning a user's preference is most sought-after, so linear bandits, first introduced in (Auer 2002), then used in recommendations in (Li et al. 2010) are also a recent popular tool.

Introduced by (Basu et al. 2019) and then extended to the contextual version by (Basu et al. 2020), blocking bandits represent the case for which the arms become unavailable for a deterministic amount of time after they are played. A similar bandit model to ours is *budgeted bandits*. Budgeted bandits have been studied in both classic MAB and linear models in (Xia et al. 2017), Thompson Sampling on budgeted bandits in (Xia et al. 2015), cost-aware cascading bandits (Gan et al. 2020). In all of these models, however, the budget is a common budget on all of the resources that arms might consume, whereas in our case, we consider independent budgets on arms. In (Slivkins 2013; Combes, Jiang, and Srikant 2015), the model is a classical, not linear, MAB model except that each arm has an individual certain budget. Bandits-with-Knapsacks, as studied in (Badanidiyuru, Kleinberg, and Slivkins 2013; Agrawal and Devanur 2016), are the types of bandits that consume a set of resources each time they are played.

Bandits that *deteriorate* or vanish by time have been studied in (Farias and Madan 2011; Chakrabarti et al. 2009; Levine, Crammer, and Mannor 2017; Seznec et al. 2019). In (Farias and Madan 2011), the agent can either keep playing an arm or decide to discard it. (Chakrabarti et al. 2009) has a stochastic time until when an arm is exhausted. (Levine, Crammer, and Mannor 2017) bandits have a decaying reward function depending on the number of times an arm is pulled. (Kleinberg, Niculescu-Mizil, and Sharma 2010) has varying arm-sets over time, however, the structure differ significantly from ours, since we do not allow an unavailable action to be available again in the future. None of the above models has extensively studied the problem of budgeted linear bandits the way we model in this paper.

Model: Disposable Linear Bandits

We begin by formally defining the *disposable linear bandit* problem that we will study. First, recall the well-studied (stochastic) linear bandit problem, which we will refer to here as the "non-disposable" problem: we have a set $A \subset \mathbb{R}^d$ of available actions, or *arms*. At each of T discrete time periods, we select exactly one arm, and a *reward* is observed and received. The reward from selecting any arm $a \in A$ is drawn independently according to a distribution $\langle a, \theta^* \rangle + \eta$, where $\theta^* \in \mathbb{R}^d$ is an (unknown) vector that we seek to learn over time, and η is some mean-zero *noise*. The goal is, loosely, to collect as much total reward as possible within the T time periods. We quickly summarize some standard assumptions and notation that will be needed:

1. Bounded arms: $\|a\| \leq L$ for all $a \in A$
2. Bounded θ^*: $\|\theta^*\| \leq L$
3. Sub-Gaussian[1] noise: $\|\eta\|_{\psi_2} \leq \sigma$
4. Finite arm set: $A = \{a_1, \ldots, a_K\}$
5. Non-negative mean rewards: $\langle a, \theta^* \rangle \geq 0$ for all $a \in A$

The fact that the same L is used in the first two assumptions is not necessary, and done purely to save on notation. Similarly, the fifth assumption is not necessary, but allows us to avoid problems in which it may be preferable to *not* select any arm. From an applications standpoint, the typical mapping to the recommendation setting is to treat θ^* as the cold user, the arms as items to recommend, and reward as some form of conversion. The embedding of the user and items in some inner-product space is estimated from previous data, e.g. with collaborative filtering via matrix factorization (for example, see (Koren, Bell, and Volinsky 2009)).

The above problem is made *disposable* by imposing just one additional constraint: each arm can only be selected at most once. Implicitly, this also requires assuming that the number of arms K is at least T, though in our target applications it is typically the case that $K \gg T$.

Note that we have *not* required the arms in A to be unique.[2] so this problem includes as special cases: the original non-disposable problem, and versions with arm-specific restrictions on selection such as so-called *budgeted bandits* (Slivkins 2013).

Let π denote any *policy*, i.e. a collection of functions

$$\pi_t : ([K] \times \mathbb{R})^{t-1} \to [K],$$

which at each time t, map the previously selected arms and observed rewards to the next selected arm (the notation $[K]$

[1]The sub-Gaussian norm of a random variable X is defined as $\|X\|_{\psi_2} \equiv \inf\{b > 0 : \mathbb{E}[\exp(|X|^2/b^2)] \leq 2\}$.

[2]The set A should thus technically be referred to as a *multiset*, though we will continue to use the shorter term "set."

refers to the set $\{1, \dots, K\}$). We measure the performance of a policy π via its *regret* with respect to the offline optimal policy which knows θ^*. Specifically, the regret $\text{Reg}(\pi)$ is defined as

$$\text{Reg}(\pi) \equiv \max_{S \subset [K], |S|=T} \sum_{s \in S} \langle a_s, \theta^* \rangle - \mathbb{E}\left[\sum_{t=1}^{T} \langle a_{\pi_t}, \theta^* \rangle\right], \quad (1)$$

where the first term is the maximum achievable expected reward, and the second term is the expected reward of π.

To summarize so far, our goal is to construct a policy π which both (a) satisfies the disposability constraint, and (b) achieves low regret. Before describing our policy, we will state a lower bound on the regret of any policy. Comparing this lower bound with known results for the non-disposable problem will serve to highlight the effect of the disposability constraint. Before proceeding, we conclude this section by defining some additional notation that will be needed.

Notation:

- For any subset of arms $S \subset A$, let $S(i)$ denote the arm in S with the i^{th} highest mean reward, with ties broken arbitrarily. In other words,
$$\langle S(1), \theta^* \rangle \geq \langle S(2), \theta^* \rangle \geq \cdots$$
- For any subset of arms S, define the *averaged arm* $\bar{S}$ as
$$\bar{S} = \frac{1}{|S|} \sum_{a \in S} a.$$
- For any policy π and time t, let A_t^π denote the subset of arms remaining (i.e. not selected in the first $t-1$ periods) under policy π.

Lower Regret Bound

Our first result is the following minimax lower bound for our disposable linear bandit:

Proposition 1. *Let $A = \{-1, 1\}^d$ (and thus $T \leq |A| = 2^d$). For every policy π, there exists $\theta^* \in [-1, 1]^d$ such that*

$$\text{Reg}(\pi) \geq C(d-k)\sqrt{T},$$

where C is a universal constant, and k is defined to be the minimum integer in $[0, d]$ satisfying

$$\sum_{j=1}^{k} \binom{d}{j} \geq T.$$

It may be useful to compare this lower bound vis-à-vis existing results for the non-disposable linear bandit: similar results (e.g. Theorem 24.1 in (Lattimore and Szepesvári 2020)) show that the minimax regret for the non-disposable setting is $\Omega(d\sqrt{T})$. Compared to this, Proposition 1 suggests that *lower* regret may be achievable under disposability. The $\sqrt{T}$ scaling remains, but the effect of the ambient dimension d might be reduced. The reduction in regret is dependent on the size of T relative to $|A|$: as T increases, k increases, and thus the lower bound on regret decreases.

Setting	Greedy	LinUCB
Non-disposable	100%	52.9%
Disposable	100%	89.91%

Table 1: Average regret relative to greedy under non-disposable and disposable settings. Experiments were conducted with $T = 200$, with synthetically generated arms and θ^*, and with Bernoulli rewards. Reported values are averaged over 2000 replications.

Another useful interpretation of k is via the relative difference in rewards among the optimal set of arms. Let ϕ be the ratio between the lowest and highest mean rewards among the set of T optimal arms, i.e.

$$\phi = \frac{\langle A(T), \theta^* \rangle}{\langle A(1), \theta^* \rangle}.$$

Intuitively, a larger value of ϕ corresponds to problems where the optimal offline set of arms is "closer" together, and thus disposability has less of an effect and the achievable regret should be higher. This plays out precisely in Proposition 1 where $\phi \sim (d-k)/d$, and so the lower bound can be re-written as $\Omega(\phi d\sqrt{T})$. Similarly, $\phi = 1$ for any non-disposable problem, so the corresponding lower bounds can also be written as $\Omega(\phi d\sqrt{T})$.

Practical Effect of Disposability: Proposition 1 serves as theoretical evidence that disposability may alter the achievable performance in a given problem. Before moving on to describing our policy, it is worth investigating experimentally whether existing algorithms for the non-disposable setting succeed. A more-expanded version of these experiments is described in the Experiments section, but for now we briefly touch on the performance of a policy known to be regret-optimal under non-disposability: *Linear UCB (LinUCB)*.

We performed a set of synthetic experiments under both non-disposable and disposable settings, compared the average regret of LinUCB to that of a greedy policy (both policies will be described in detail later). The results are given in Table 1. The main takeaway is that while LinUCB performed better than the greedy policy in both settings, its relative performance significantly decreased under disposability. This suggests (though it does not guarantee) that better policies may exist. We will propose such a policy now.

Our Algorithm: Generalized UCB

The algorithm we propose generalizes the previously-mentioned LinUCB algorithm (Auer 2002; Dani, Hayes, and Kakade 2008; Li et al. 2010; Abbasi-Yadkori, Pál, and Szepesvári 2011) to the disposable setting. It will be necessary first to recall the details of that algorithm, and in particular the construction of confidence sets on θ^*, as we will use the same in our own algorithm.

Confidence Sets and LinUCB: The LinUCB algorithm relies on using previously-observed rewards to construct carefully tuned confidence sets on θ^*. The following construction is based on that of (Abbasi-Yadkori, Pál, and Szepesvári 2011): Let $\lambda > 0$ be an arbitrary constant (in practice, this can be treated as a tuning parameter). Suppose at the beginning of time period $t+1$, the policy π has selected arms $a_{\pi_1}, \ldots, a_{\pi_t}$ and observed rewards $r_1, \ldots, r_t$. We define the matrix V_t as

$$V_t = \sum_{i=1}^{t} a_{\pi_i} a_{\pi_i}{}^\top + \lambda I,$$

where I is the identity matrix. Then letting $\hat{\theta}_t$ denote the regularized least-squares estimator of θ^*, we have

$$\hat{\theta}_t \equiv V_t^{-1} \sum_{i=1}^{t} a_{\pi_i} r_i,$$

and finally, the confidence set used in the LinUCB algorithm is the following ellipsoid:

$$\Theta_t = \{\theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t\|_{V_t} \leq \beta_t\}, \tag{2}$$

where $\|x\|_V^2 = x^\top V x$ for any $x \in \mathbb{R}^d$, and where

$$\beta_t = \sigma \sqrt{2 \log T + d \log \frac{d\lambda + TL^2}{d\lambda}} + \sqrt{\lambda} L. \tag{3}$$

The LinUCB algorithm, then, selects the arm that has the highest "potential" reward over all $\theta \in \Theta_t$, i.e. it selects

$$\operatorname*{argmax}_{a \in A} \max_{\theta \in \Theta_t} \langle a, \theta \rangle.$$

As a side note, implementing this is tractable (unless A is large or infinite) because the ellipsoidal form of Θ_t enables a closed-form expression for the inner maximization above:

$$\mathrm{UCB}_t(a) \equiv \max_{\theta \in \Theta_t} \langle a, \theta \rangle = \langle a, \hat{\theta}_t \rangle + \beta_t \|a\|_{V_t^{-1}}. \tag{4}$$

Generalized UCB: The guiding principle behind LinUCB, and in fact optimism-based algorithms in general, can be viewed as selecting the action which has the highest optimistic (within reason) reward. For the non-disposable linear bandit, "within reason" is precisely encoded by the previously-described uncertainty sets Θ_t, and the "actions" are simply the arms. From this lens, it seems the natural generalization of LinUCB to the disposable setting is to keep the same uncertainty sets, but to treat an "action" as a plan for the set of arms that will be selected in the remaining periods. That is, at any time t, there are $T - t + 1$ selections remaining, and so we should choose optimistically from among the *subsets* of $T - t + 1$ arms. This is precisely the idea behind our algorithm, *Generalized UCB (UCBG)*.

Recall that we use A_t^{UCBG} to denote the remaining (i.e. not yet selected) arms at the beginning of time t. Then the optimistic subset of arms is chosen according to:

$$S_t^{\mathrm{UCBG}} \equiv \operatorname*{argmax}_{\substack{S \subset A_t^{\mathrm{UCBG}} \\ |S| = T-t+1}} \max_{\theta \in \Theta_{t-1}} \sum_{a \in S} \langle a, \theta \rangle. \tag{5}$$

Algorithm 1: Generalized LinUCB (UCBG)

```
Input: λ > 0
Initialization: V_0 = λI, θ̂_0 = [0]^d
Calculate Θ_0 according to (2).
for t = 1, 2, ... T do
    Calculate S_t^UCBG according to (5)
    Select arm a_UCBG_t = argmax_{a ∈ S_t^UCBG} UCB_{t-1}(a)
    Observe reward r_t
    Calculate V_t = V_{t-1} + a_UCBG_t (a_UCBG_t)^⊤
    Calculate Θ_t according to (2).
end
```

The actual arm selected by UCBG, denoted a_{UCBG_t}, is chosen optimistically from this subset:

$$a_{\mathrm{UCBG}_t} \equiv \operatorname*{argmax}_{a \in S_t^{\mathrm{UCBG}}} \mathrm{UCB}_{t-1}(a),$$

with $\mathrm{UCB}_t(a)$ defined as in (4). The UCBG algorithm is described fully in Algorithm 1.

Our Guarantee

Next we prove an upper bound on the regret of our algorithm. To summarize briefly, UCBG achieves $\tilde{O}(\sqrt{\bar{\gamma}} d \sqrt{T})$ regret, which is a reduction of the $\tilde{O}(d\sqrt{T})$ regret for non-disposable linear bandits. This reduction depends on the quantity $\bar{\gamma}$, which averages, over time, a certain *shrinkage factor* that loosely captures the difference in rewards among the best arms.

Shrinkage Factor: Fix any time period t. Note that at time t, there are $T - t + 1$ selections remaining, and irrespective of the previous selections, there are $K - t + 1$ remaining arms from which to select. For any such set of remaining arms, we can compute the ratio between the $(T - t + 1)^{\text{th}}$ highest mean reward and the average mean reward across the best $T - t + 1$ arms. We define γ_t to be an upper bound on this ratio across all such subsets:

$$\frac{\langle S(T-t+1), \theta^* \rangle}{\left\langle \frac{1}{T-t+1} \sum_{i=1}^{T-t+1} S(i), \theta^* \right\rangle} \leq \gamma_t \text{ for all } S \subset A, |S| = K - t + 1 \tag{6}$$

Notice that $\gamma_t \leq 1$ for all t, and is in general smaller when the arms are more "spread out." As a special case, any non-disposable problem, when represented in the disposable framework, has $\gamma_t = 1$.

With γ_t defined, we can now state our regret bound:

Theorem 1. *With probability at least* $1 - O(1/T)$*, the expected regret of* UCBG *satisfies*

$$\mathrm{Reg}(\mathrm{UCBG}) = \tilde{O}\left(\sqrt{\bar{\gamma}} d \sqrt{T}\right),$$

where $\bar{\gamma} = \frac{1}{T} \sum_{t=1}^{T} \gamma_t$ *and* γ_t *is defined according to* (6)*.*

Proof Sketch of Theorem 1. Our proof relies on a framework similar to that used to upper bound the regret for the

LinUCB policy. The key challenge in adapting that framework is that the disposable setting misses an important property – the optimal arm at each time period is a single "best" arm, and so the regret analysis can be almost entirely decomposed over time. On the other hand, in our disposable setting, the optimal sequence of arm is unique only up to permutation, which prevents the same decomposition if the usual method for regret "book-keeping" is applied. To deal with this, we prove the following key lemma, which suggests that there is an alternative book-keeping, with a different form of instantaneous regret, that allows for decomposition over time.

Lemma 1.

$$\mathrm{Reg}(\pi) = \sum_{t=1}^{T} \left(\langle A_t^{\pi}(T-t+1), \theta^* \rangle - \langle a_{\pi_t}, \theta^* \rangle \right)^+$$

A Fast Heuristic

So far, we have proposed and analyzed the UCBG policy, and while it enjoys the theoretical regret guarantee given in Theorem 1, there is the computational issue of solving (5) at each time step. That is, we need to maximize a function of the form

$$f(\theta, S) \equiv \sum_{a \in S} \langle a, \theta \rangle,$$

where θ ranges over ellipsoidal confidence sets, and S ranges over subsets of A. So even though the "inner" maximization over θ can be done in closed form, the "outer" maximization is still over exponentially many subsets S.

In this section, we propose a heuristic for dealing with this (this heuristic will be the algorithm we test in the coming experiments). To simplify the notation, assume that all of the arms in A and θ^* lie on the unit-ball (these have already been assumed to be bounded, so the extra assumption is that they are of the same size; taking $L = 1$ is done without loss of generality). Our heuristic is based on the idea that if the arms are fairly dense on the unit-ball (a reasonable assumption in our target applications), then whatever θ^* is, there is likely to be an arm "close" to it, where closeness is measured in Euclidean distance.

To be more precise, consider functions $g(\theta)$ of the form

$$g(\theta) \equiv \max_{\substack{S \subset A \\ |S|=k}} \sum_{a \in S} \langle a, \theta \rangle,$$

and for any given vector $v \in \mathbb{R}^d$, let $a_i(v)$ be the i^{th} closest arm (from the set A) to v in Euclidean distance. Then, $g(\theta)$ is equivalently

$$g(\theta) = \sum_{i=1}^{k} \langle a_i(\theta), \theta \rangle.$$

By holding out the first term, and making the (approximate) substitution $a_1(\theta) \approx \theta$ twice, we have

$$\begin{aligned} g(\theta) &= \langle a_1(\theta), \theta \rangle + \sum_{i=2}^{k} \langle a_i(\theta), \theta \rangle \\ &\approx \langle a_1(\theta), \theta \rangle + \sum_{i=2}^{k} \langle a_i(a_1(\theta)), \theta \rangle \\ &\approx \langle a_1(\theta), \theta \rangle + \sum_{i=2}^{k} \langle a_i(a_1(\theta)), a_1(\theta) \rangle \end{aligned} \tag{7}$$

The key property we use from the final expression above is that the "nearest arm" operations $(a_1(\cdot), a_2(\cdot), \ldots)$ are only applied to $a_1(\theta)$. Indeed, the second term in (7) depends only on the set of arms (i.e. not on θ), and can be precomputed for each $a \in A$. We call this term the *similarity score* of an arm (this is indeed the usual cosine similarity).

This enables the following heuristic: instead of maximizing $g(\theta)$ over an ellipsoid, we iterate over all of the arms, computing the maximum of (7) over this same ellipsoid (which can be done in closed-form). Another lens toward this heuristic is that it balances the usual objectives of exploration and exploitation with a third objective, which is to select arms for which there are other similar arms. This makes sense intuitive sense in the disposable setting: there is little reason to learn the mean reward of an arm if there are not similar arms to leverage later on.

To formally state the method, we must define two functions: $\mathrm{Similarity}(v, B, s)$ and $\mathrm{Closest}(v, B, s)$. $\mathrm{Similarity}(v, B, s)$ is the cumulative similarity score of vector v with the set of arms that are closest to it in cosine-similarity measure. $\mathrm{Closest}(v, B, s)$ gives a subset of arms of size s from B with the highest inner products with vector v.

$$\mathrm{Similarity}(v, B, s) = \max_{S \subset B, |S|=s} \sum_{a \in S} \langle v, a \rangle$$

$$\mathrm{Closest}(v, B, s) = \operatorname*{argmax}_{S \subset B, |S|=s} \sum_{a \in S} \langle v, a \rangle$$

Our heuristic, which we call the Alternating Heuristic, works as follows. We first choose the arm $\tilde{a}$ with the highest sum UCB + Similarity. Intuitively this corresponds to an arm which has a high reward estimate *and* a high number of arms close to it. Then, we find the set of closest arms to this arm, which we denote as S_t^{AH}. Note that the θ (from the uncertainty set) that will maximize the optimistic reward for S_t^{AH} is different from that of $\tilde{a}$. Let $\theta^{\mathrm{UCB}}(a) = \operatorname{argmax}_{\theta \in \Theta_{t-1}} \langle \theta, a \rangle$ and $\theta^{\mathrm{UCBG}}(S) = \operatorname{argmax}_{\theta \in \Theta_{t-1}} \sum_{a \in S} \langle \theta, a \rangle$. We find the $\tilde{\theta} \in \Theta_{t-1}$ that maximizes $\langle \overline{S_t^{\mathrm{AH}}}, \tilde{\theta} \rangle$. Here we use the fact that maximizing $\mathrm{UCB}(\overline{S_t^{\mathrm{AH}}})$ is equivalent to maximizing $\mathrm{UCBG}(S_t^{\mathrm{AH}})$:

$$\theta^{\mathrm{UCBG}}(S) = \operatorname*{argmax}_{\theta \in \Theta_{t-1}} \sum_{a \in S} \langle a, \theta \rangle = \operatorname*{argmax}_{\theta \in \Theta_{t-1}} \langle \bar{S}, \theta \rangle = \theta^{\mathrm{UCB}}(\bar{S}).$$

For a fixed θ, we can compute $g(\theta)$ by finding a set of arms closest to it, where we denote this set as S_t^{θ}. We iterate

Algorithm 2: Alternating Heuristic

Input: $\lambda, \alpha, c > 0$
Initialization: $V = \lambda I; \hat{\theta}_0 = [0]^d; X, Y = []$
for $t := 1, 2, \ldots T$ **do**
  $\tilde{a} = \underset{a \in A_t^{\text{AH}}}{\text{argmax}} \langle \hat{\theta}_{t-1}, a \rangle + c\|a\|_{V_{t-1}^{-1}} +$
  α Similarity$(a, A_t^{\text{AH}}, T-t)$
  $S_t^{\text{AH}} = \text{Closest}(\tilde{a}, A_t^{\text{AH}}, T-t+1)$
  $\tilde{\theta} = \theta^{\text{UCB}}(\overline{S_t^{\text{AH}}})$
  $S_t^{\theta} = \text{Closest}(\tilde{\theta}, A_t^{\text{AH}}, T-t+1)$
  while $S_t^{AH} \neq S_t^{\theta}$ **do**
    $S_t^{\text{AH}} = \text{Closest}(\overline{S_t^{\theta}}, A_t^{\text{AH}}, T-t+1)$
    $\tilde{\theta} = \theta^{\text{UCB}}(\overline{S_t^{\text{AH}}})$
    $S_t^{\theta} = \text{Closest}(\tilde{\theta}, A_t^{\text{AH}}, T-t+1)$
  end
  Play the arm $a_{\text{AH}_t} = \underset{a \in S_t^{\text{AH}}}{\text{argmax}} \langle \hat{\theta}_{t-1}, a \rangle + c\|a\|_{V_{t-1}^{-1}}$
  Observe reward r_t
  $V_t = V_{t-1} + a_{\text{AH}_t}(a_{\text{AH}_t})^\top$
  $X = [X \; a_{\text{AH}_t}]$
  $Y = [Y^\top \; r_t]^\top$
  Calculate $\hat{\theta}_t = V_t^{-1} X^\top Y$
end

until the two arm sets, S_t^{θ} and S_t^{AH} are equal. Finally, the arm in the final subset with the maximum UCB index is selected.

Note that in each alternating step, the objective value f increases. Thus, since the set of arms is finite, the heuristic is guaranteed to converge. However, since f is non-convex, the point of convergence is not guaranteed to be S_t^{UCBG}. We find in experiments that the choice of initialization is crucial to the performance of the method. We investigated three different methods to initialize this set, and compared both the final objective values and the convergence rates. The data and experiments are explained in detail in the Experiments section, but in Figure 1, we provide the comparisons of the objective function between those different initial sets given to the Alternating Heuristic. These trends were observed during $t = 5$.

We tested three different initialization methods over the arm set.

- *Random*: Choose $T-t+1$ arms randomly and iterate until convergence. As seen in the Figure 1, this method takes more iterations to converge and converges to a lower value of the objective function.
- *UCB*: Choose $T-t+1$ arms by

$$\underset{\substack{S \in A_t^{\text{AH}} \\ |S| = T-t+1}}{\text{argmax}} \sum_{a \in S} \text{UCB}(a),$$

  i.e. arms with the highest combined UCB values. This method is significantly better than random initialization, but still has a lower final value than that of Similarity initialization.

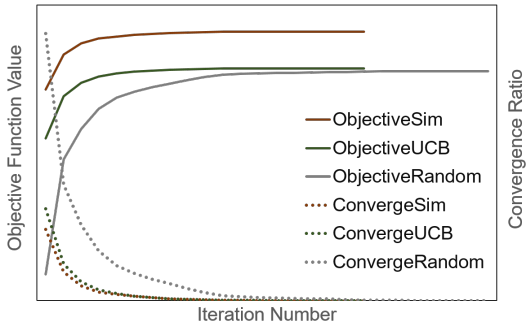

Figure 1: Performance and convergence rate of alternating heuristic with three different initialization methods. Results are averaged over 200 instances, all at period $t = 5$. *Performance (Solid):* Objective function vs. iteration number. *Convergence Rate (Dashed):* Gap fraction to converged value vs. iteration number.

- *Similarity*: Choose $T-t+1$ arms by maximizing the similarity score of one arm, and choosing $T-t$ arms closest to it. This initialization not only has the highest function value it converges, but also has a faster convergence than random, similar to that of LinUCB.

Experiments

We tested our final algorithm (the alternating heuristic with similarity score initialization) against a number of baseline algorithms. These experiments reveal the following:

1. Bernoulli reward setting (see Table 2 top and Figure 2): for all three time horizons tested, the alternating heuristic outperformed all benchmarks in terms of average regret. These benchmarks included LinUCB (with "perfect" choice of tuning parameter) and the greedy algorithm.
2. Gaussian reward setting (see Table 2 bottom): the alternating heuristic outperformed both methods again, despite perfectly tune the parameters for LinUCB, but not for the heuristic.

We begin by describing the experimental setup. We generated K arms, $i = 1, \ldots, K$ in $d-$dimensional space where $K = 5000$ and $d = 15$. Similarly, we generated a set of θs that lie in the same space, where the total number of θs is 5000. [3]

As described in the Model section, the mean reward of pulling arm a while having θ as the unknown vector is $\langle a, \theta \rangle$, with the observed reward following the distribution:

$$r(a) = \langle a, \theta \rangle + \eta$$

where η is a mean-0 sub-Gaussian noise. We now give the disposable adaptations of the benchmarks we use.

Disposable Setting Adaptations: The modification for all benchmark algorithms to the disposable setting is straight-

[3]The corresponding data can be found in https://github.com/MeldaKor/DisposableLinearBandits.

Type	Horizon	Greedy	LinUCB1	LinUCB2	LinUCB3	UCBG1	UCBG2	UCBG3	TS-Adapted
Bernoulli	50	100%	94.81%	94.36%	99.46%	94.82%	94.15%	**88.15%**	-
	100	100%	91.67%	87.66%	95.65%	86.94%	84.75%	**78.97%**	-
	200	100%	91.42%	89.91%	99.59%	93.4%	88.60%	**83.07%**	-
Gaussian	50	100%	94.29%	88.97%	94.92%	86.13%	83.11%	**79.67%**	106.65%

Table 2: Final Regret Comparisons. We report the final regret with respect to the Greedy method. For all experiments, we set the tuning parameter α for the heuristic the same as LinUCB's c. In experiments, $\alpha, c = (1/2)^i$ where $i = 3, 4, 5$.

forward: after playing an arm, we remove it from the future available arm sets. The decision making procedures for Greedy and LinUCB remain the same: LinUCB selects the arm with the highest potential whereas Greedy plays the arm with the highest *expected* potential, i.e. it uses the MLE estimator for θ. Both of these algorithms and Alternating Heuristic could be run in either Bernoulli or Gaussian reward functions.

In the Gaussian setting, we also test a version of Thompson Sampling that is given the (true) multivariate Gaussian prior, similar to that studied in (Agrawal and Goyal 2013). Formally, we assume that rewards are drawn according to $r(a) = \langle \theta, a \rangle + \eta$ where $\eta \sim \mathcal{N}(0, \sigma^2)$, and that θ^* is drawn from $\mathcal{N}(\theta_0, \Sigma_0)$. We propose the following adaptation for Thompson Sampling in the disposable setting:choose an arm, uniformly at random, from among the closest $T - t + 1$ arms to θ_t^{TS}, which is sampled from the running posterior distribution.

Results

We ran two sets of experiments with a modification on the reward function: when the reward function has an underlying Gaussian distribution and Bernoulli distribution. Both means are, naturally, $\langle a, \theta \rangle$ when an arm a is pulled with the unknown parameter θ is present. We ran the experiments with 5000 arms with horizons of $T = 50, 100, 200$. The number of instances, i.e. different θ values we worked with, ranges from 200 to 2500. The immediate pseudo regret was calculated as in Lemma 1 using the new book-keeping method for disposable bandits. The regret value of each instance was averaged over 10 runs, and the final results were averaged over all instances.

1. Gaussian Reward: The reason we work with Gaussian reward distribution, even though it is more realistic that the matching rewards come from Bernoulli distribution in online platforms, is that we know the posterior of a Gaussian prior in linear Thompson Sampling. The posterior is calculated as given in the previous subsection. The methods we implement are Thompson Sampling Adaptation, Greedy, LinUCB and Alternating Heuristic. To give Thompson Sampling the benefit of the doubt, we sample $\theta \in \mathbb{R}^d$ from the standard multivariate normal distribution $\mathcal{N}([0]^d, I_d)$. However, the arms $a_i \in \mathbb{R}^d, i = 1, \ldots, 5000$ are as generated. So, for an arm a_i the reward we observe becomes $r(a_i) = \langle a_i, \theta \rangle + \eta$ where $\eta \sim \mathcal{N}(0, 1)$.
2. Bernoulli Reward: In Bernoulli case, the reward we observe from pulling an arm a_i has an underlying distribution of Bernoulli($\langle a_i, \theta \rangle$). The methods we test are Greedy, LinUCB and Alternating Heuristic.

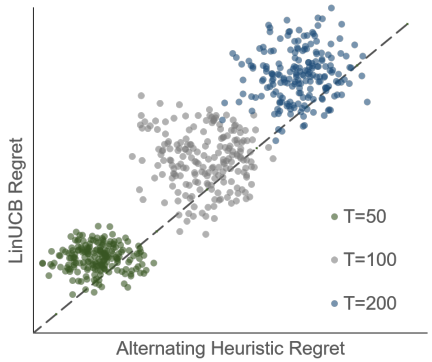

Figure 2: The tuned parameter for LinUCB and the corresponding Heuristic parameter were chosen.

In Figure 2, we work with Bernoulli-type feedback, and we provide a point-wise comparison of each instance based on the final regret value. For each horizon value, we plotted 200 different randomly chosen instances. Alternating Heuristic outperforms LinUCB in majority of the instances for all those 3 horizons implemented.

In Table 2, we give the final regret fraction of these methods with varying parameters and horizons comparing to that of the Greedy Algorithm. In both cases of Gaussian and Bernoulli, we see that even though we did not perfectly tune the Similarity parameter α and took it equal to that of LinUCB, c, Alternating Heuristic outperforms all other benchmarks. In the Gaussian case, the main purpose of which was to test the computationally tractable adaptation of Thompson Sampling, surprisingly, we see that the TS adaptation had a higher regret averaged over the instances compared to other methods.

Conclusion

We tackled the problem of making budgeted recommendations using linear bandits. We first introduced an ideal generalization of the LinUCB algorithm, and due to its computational difficulty, we gave an alternating heuristic approximated from UCBG-index. We tested the heuristic along with other methods on synthetically generated data, and saw that it outperforms all others when arms are disposable.

Ethics Statement

The primary motivation for this work is recommender systems. Such systems are in broad use, and so it is impossible to predict all future ethical and societal issues that may arise from their use. One foreseeable issue is that these systems may introduce biases against certain groups.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Agrawal, S.; and Devanur, N. 2016. Linear contextual bandits with knapsacks. In *Advances in Neural Information Processing Systems*, 3450–3458.

Agrawal, S.; and Goyal, N. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 127–135. Atlanta, Georgia, USA: PMLR. URL http://proceedings.mlr.press/v28/agrawal13.html.

Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3(Nov): 397–422.

Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2013. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 207–216. IEEE.

Basu, S.; Papadigenopoulos, O.; Caramanis, C.; and Shakkottai, S. 2020. Contextual Blocking Bandits. *arXiv preprint arXiv:2003.03426* .

Basu, S.; Sen, R.; Sanghavi, S.; and Shakkottai, S. 2019. Blocking Bandits. *CoRR* abs/1907.11975. URL http://arxiv.org/abs/1907.11975.

Bobadilla, J.; Ortega, F.; Hernando, A.; and Bernal, J. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems* 26: 225–238.

Bouneffouf, D.; Bouzeghoub, A.; and Gançarski, A. L. 2012. A Contextual-Bandit Algorithm for Mobile Context-Aware Recommender System. In Huang, T.; Zeng, Z.; Li, C.; and Leung, C. S., eds., *Neural Information Processing*, 324–331. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-34487-9.

Chakrabarti, D.; Kumar, R.; Radlinski, F.; and Upfal, E. 2009. Mortal Multi-Armed Bandits 273–280. URL http://papers.nips.cc/paper/3580-mortal-multi-armed-bandits.pdf.

Combes, R.; Jiang, C.; and Srikant, R. 2015. Bandits with budgets: Regret lower bounds and optimal algorithms. *ACM SIGMETRICS Performance Evaluation Review* 43(1): 245–257.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic Linear Optimization under Bandit Feedback. In *COLT*.

Farias, V. F.; and Madan, R. 2011. The Irrevocable Multi-armed Bandit Problem. *Oper. Res.* 59(2): 383–399. ISSN 0030-364X. doi:10.1287/opre.1100.0891. URL http://dx.doi.org/10.1287/opre.1100.0891.

Gan, C.; Zhou, R.; Yang, J.; and Shen, C. 2020. Cost-Aware Cascading Bandits. *IEEE Transactions on Signal Processing* 68: 3692–3706.

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

Gómez Hidalgo, J. M.; Bringas, G. C.; Sánz, E. P.; and García, F. C. 2006. Content based SMS spam filtering. In *Proceedings of the 2006 ACM symposium on Document engineering*, 107–114. ACM.

Kleinberg, R.; Niculescu-Mizil, A.; and Sharma, Y. 2010. Regret bounds for sleeping experts and bandits. *Machine learning* 80(2-3): 245–272.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42(8): 30–37.

Lai, T.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6(1): 4 – 22. ISSN 0196-8858. doi:https://doi.org/10.1016/0196-8858(85)90002-8. URL http://www.sciencedirect.com/science/article/pii/0196885885900028.

Lang, K. 1995. NewsWeeder: Learning to Filter Netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*.

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

Levine, N.; Crammer, K.; and Mannor, S. 2017. Rotting Bandits. In *NIPS*.

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*, WWW '10, 661–670. New York, NY, USA: ACM. ISBN 978-1-60558-799-8. doi:10.1145/1772690.1772758. URL http://doi.acm.org/10.1145/1772690.1772758.

Lika, B.; Kolomvatsos, K.; and Hadjiefthymiades, S. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41(4): 2065–2073.

Mooney, R. J.; and Roy, L. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, 195–204. ACM.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58(5): 527–535. URL https://projecteuclid.org:443/euclid.bams/1183517370.

Sarwar, B. M.; Karypis, G.; Konstan, J. A.; Riedl, J.; et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www* 1: 285–295.

Seznec, J.; Locatelli, A.; Carpentier, A.; Lazaric, A.; and Valko, M. 2019. Rotting bandits are no harder than stochastic ones. In Chaudhuri, K.; and Sugiyama, M., eds., *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, 2564–2572. PMLR. URL http://proceedings.mlr.press/v89/seznec19a.html.

Slivkins, A. 2013. Dynamic Ad Allocation: Bandits with Budgets. *ArXiv* abs/1306.0155.

Wei, J.; He, J.; Chen, K.; Zhou, Y.; and Tang, Z. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications* 69: 29–39.

Xia, Y.; Li, H.; Qin, T.; Yu, N.; and Liu, T.-Y. 2015. Thompson sampling for budgeted multi-armed bandits. *arXiv preprint arXiv:1505.00146* .

Xia, Y.; Qin, T.; Ding, W.; Li, H.; Zhang, X.; Yu, N.; and Liu, T.-Y. 2017. Finite budget analysis of multi-armed bandit problems. *Neurocomputing* 258: 13 – 29. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2016.12.079. URL http://www.sciencedirect.com/science/article/pii/S0925231217304216. Special Issue on Machine Learning.

Zeng, C.; Wang, Q.; Mokhtari, S.; and Li, T. 2016. Online Context-Aware Recommendation with Time Varying Multi-Armed Bandit. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 2025–2034. New York, NY, USA: ACM. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939878. URL http://doi.acm.org/10.1145/2939672.2939878.