Sequential Learning of Product Recommendations with Customer Disengagement

Hamsa Bastani
Wharton School, hamsab@wharton.upenn.edu

Pavithra Harsha
IBM Thomas J. Watson Research, pharsha@us.ibm.com

Georgia Perakis
MIT Sloan School of Management, georgiap@mit.edu

Divya Singhvi
MIT Operations Research Center, dsinghvi@mit.edu

We consider the problem of sequential product recommendation when customer preferences are unknown. We first present empirical evidence of customer disengagement using a sequence of ad campaigns from a major airline carrier. In particular, customers decide to stay on the platform based on the quality of recommendations. We then formulate this problem as a linear bandit, with the notable difference that the customer's horizon length is a function of past actions. We prove that any classical bandit learning algorithm will over-explore in this regime, while a naive greedy policy under-explores; thus, both algorithms incur linear regret with probability one. We propose modifying bandit learning strategies by constraining the action space upfront using an efficient IP procedure. We prove that this simple modification allows our algorithm to achieve sublinear regret with nonzero probability. Furthermore, simulations demonstrate that our algorithm significantly improves both regret and the length of time that a customer is engaged with the platform.

Key words: bandits, online learning, recommendation systems, disengagement
History: This paper is under preparation.

1. Introduction

Personalized customer recommendations are a key ingredient to the success of platforms such as Netflix, Amazon and Expedia. Product variety has exploded, catering to the heterogeneous tastes of customers. However, this has also increased search costs, making it difficult for customers to find products that interest them. Platforms add value by learning a customer's preferences over time, and leveraging this information to match her with relevant products.

The personalized recommendation problem is typically formulated as an instance of collaborative filtering (Sarwar et al. 2001, Linden et al. 2003). In this setting, the platform observes different customers' past ratings or purchase decisions for random subsets of products. Collaborative filtering techniques use the feedback across all observed customer-product pairs to infer a low-dimensional

model of customer preferences over products. This model is then used to make personalized recommendations over unseen products for any specific customer. While collaborative filtering has found industry-wide success (Breese et al. 1998, Herlocker et al. 2004), it is well-known that it suffers from the "cold start" problem (Schein et al. 2002). In particular, when a new customer enters the platform, no data is available on her preferences over *any* products. Collaborative filtering can only make sensible personalized recommendations for the new customer after she has rated at least $\mathcal{O}(k \log n)$ products, where k is the dimension of the low-dimensional model learned via collaborative filtering and n is the total number of products. Consequently, bandit approaches have been proposed in tandem with collaborative filtering (Bresler et al. 2014, Li et al. 2016, Gopalan et al. 2016) to tackle the cold start problem using a combination of exploration and exploitation. The basic idea behind these algorithms is to offer random products to customers during an exploration phase, learn the customer's low-dimensional preference model, and then exploit this model to make good recommendations.

A key assumption underlying this literature is that customers are patient, and will remain on the platform for the entire (possibly unknown) time horizon T regardless of the goodness of the recommendations that have been made thus far. However, this is a tenuous assumption, particularly when customers have strong outside options (e.g., a Netflix user may abandon the platform for Hulu if they receive a series of bad entertainment recommendations). We demonstrate this effect using customer panel data on a series of ad campaigns from a major commercial airline. Specifically, we find that a customer is far more likely to click on a suggested travel product in the current ad campaign if the previous ad campaign's recommendation was relevant to her. In other words, customers may *disengage* from the platform and ignore new recommendations entirely if past recommendations were irrelevant. In light of this issue, we introduce a new formulation of the bandit product recommendation problem where customers may disengage from the platform depending on the rewards of past recommendations, i.e., the customer's time horizon T on the platform is no longer fixed, but is a function of the platform's actions thus far.

Customer disengagement introduces a significant difficulty to the dynamic learning or bandit literature. We prove lower bounds that show that any algorithm achieves regret that scales linearly in T (the customer's time horizon on the platform if they are given good recommendations). This hardness result arises because no algorithm can satisfy *every* customer early on when we have limited knowledge of their preferences; thus, no matter what policy we use, at least some customers will disengage from the platform. The best we can hope to accomplish is to keep a large fraction of customers engaged on the platform for the entire time horizon, and to match these customers with their preferred products.

However, classical bandit algorithms perform particularly badly in this setting – we prove that *every* customer disengages from the platform with probability one as T grows large. This is because bandit algorithms *over-explore*: they rely on an early exploration phase where customers are offered random products that are likely irrelevant to them. Thus, it is highly probable that the customer receives several bad recommendations during exploration, and disengages from the platform entirely. This exploration is continued for the entire time horizon, T, under the principal of optimism. This is not to say that learning through exploration is a bad strategy. We show that a greedy exploitation-only algorithm also under-performs by *under-exploring*. Unlike classical bandit algorithms, the greedy algorithm succeeds in keeping a positive fraction of customers on the platform for the entire time horizon. However, we demonstrate through simple analytical examples that the greedy algorithm's recommendations often get stuck in sub-optimal fixed points, and fail to match customers with the best available product even in settings where exploration is not costly. Consequently, the platform misses out on its key value proposition of learning customer preferences and matching them to their preferred products.

Our results demonstrate that one needs to more carefully balance the exploration-exploitation tradeoff in the presence of customer disengagement. We propose a simple modification of classical bandit algorithms by constraining the space of possible product recommendations upfront. We leverage the rich information available from existing customers on the platform to identify a diverse subset of products that are palatable to a large segment of potential customer types; all recommendations made by the platform are then constrained to be in this set. This approach guarantees that mainstream customers remain on the platform with high probability, and that they are matched to their preferred products over time; we compromise on tail customers, but these customers are unlikely to show up on the platform and catering recommendations to them endangers the engagement of mainstream customers. We formulate the initial optimization of the product offering as an integer program. We then prove that our proposed algorithm achieves sublinear regret in T for a large fraction of customers, i.e., it succeeds in keeping a large fraction of customers on the platform for the entire time horizon, and matches them with their preferred product. Numerical experiments on synthetic and real data demonstrate that our approach significantly improves both regret and the length of time that a customer is engaged with the platform compared to both classical bandit and greedy algorithms.

1.1. Main Contributions

- *Empirical evidence of disengagement:* We hypothesise that customer engagement on recommendation platforms with repeated interactions (for example, periodic email recommendations) is dependent on the relevance of the recommendations made in prior interactions. Using data from

sequential email ad-campaigns of a major airline partner, we show that a customer is less likely to click on the email (*disengage*) if the previously sent email was irrelevant to the customer.

- *Customer disengagement modelling:* Based on the findings of the econometric study, we model the customer disengagement decision as being endogenous to the recommendations made by the platform. Then, we analyze the problem of learning new user preferences with customer disengagement.
- *Hardness result and analytical guarantees on existing algorithms:* We show that user preference learning becomes substantially harder when customers are likely to disengage from the platform. More specifically, we prove a lower bound guarantee on regret that shows that no algorithm can achieve sub-linear regret in the proposed model disengagement model. We also show that classical bandit algorithms fail to keep any customer engaged on the platform when customers are likely to disengage. Similarly, we also show that myopic policies fail to incur sub-linear regret even when customers never disengage from the platform. These results show that widely used policies are bound to perform poorly in the current setting.
- *Constrained Bandit algorithm:* We propose a new algorithm, the Constrained Bandit algorithm that constrains exploration over an optimally selected restricted set of products. We prove that the algorithm keeps a large fraction of customers engaged with the platform and incurs sub-linear regret on this set of customers. This is a considerable improvement over both classical bandit policies and myopic policies since they are guaranteed to loose all customers from the platform.
- *Superior performance on a synthetic and a real world data set:* In order to test the applicability of the proposed algorithm, we perform extensive numerical experiments on synthetic as well as a real world data set (MovieLens data set for movie recommendations. Harper and Konstan (2016)) We show that the proposed algorithm outperforms benchmark algorithms by a wide margin. More specifically, the mean engagement time of customers increases by as much as 80% over other benchmark algorithms if exploration is performed using the benchmark algorithm. This is particularly important since increased engagement time directly translates to increased revenue for platforms.

1.2. Related Literature

As noted earlier, there has been considerable interest in the OR/MS community on the problem of learning customer preferences and optimizing recommendation. Since our work lies in the intersection of these topics we discuss each of them in detail next.

Recommendation Systems and Bandit Problems: Optimally learning customer preferences is central to many decision making problems. Particular applications include personalization (Surprenant and Solomon 1987) and recommendations (Sarwar et al. 2001). For a comprehensive overview of personalization in OM and revenue management applications, we refer the readers

to Murthi and Sarkar (2003) and the references therin. Similarly, Su and Khoshgoftaar (2009) provides a detailed review of the research in recommendation systems. Recently, there has been a surge of literature connecting various decision making problems to optimal recommendations. Particular examples include Besbes et al. (2015), Demirezen and Kumar (2016), Li et al. (2016) and others. All these studies focus on making recommendations for existing customers and hence do not involve learning. Our work differs in that we focus on the bandit learning setting where no previous data on personalized customer response is available.

Bresler et al. (2014), Li et al. (2016), Gopalan et al. (2016) propose solving the "cold start" problem of recommending products to new users with no historical data using bandit learning models. Since none of these studies model learning with customer disengagement, the focus of this paper differ substantially from previous work. Similarly, Lika et al. (2014) and Wei et al. (2017) use similarity measures and deep neural networks to alleviate recommendations for new customers or new users. Neither of the two studies provide analytical guarantees on the rate of regret which is part of the focus of the current paper.

OM for new customers has also been widely studied. Keskin and Zeevi (2014), den Boer and Zwart (2013), Javanmard and Nazerzadeh (2016) focus on the problem of dynamically pricing products with unknown demand. Similarly, Agrawal et al. (2016) and Agrawal et al. (2017) analyze the problem of optimal assortment selection with unknown user preferences. These studies rely on optimally balancing the exploration-exploitation tradeoff under bandit feedback. Nevertheless, none of these studies model customer disengagement. Using data from a major airline partner, we first show evidence of customer disengagement being endogeneous to the recommendation decisions of the airline. Our modelling framework explicitly captures the disengagement decision of the customer to as a function of the recommendations made by the system and we optimize learning customer preferences under disengagement.

Customer Disengagement: A second stream of literature that directly connects to this work is that of customer disengagement. Customer disengagement and its relation to service quality has been recognized and well studied; see for example Venetis and Ghauri (2004), Bowden (2009) and Sousa and Voss (2012). OM researchers have also analyzed the problem of optimizing services for better customer engagement. For example, Davis and Vollmann (1990) developed the framework for relating waiting times of customers with service quality perception. Lu et al. (2013) have provided evidence of changing customer purchase behavior due to waiting times. Kanoria et al. (2018) model customer disengagement based on the goodwill model of Nerlove and Arrow (1962). The decision maker chooses one of two decisions: a high reward service level with high likelihood of abandonment and a low reward service level with low likelihood of abandonment. Similarly, Aflaki and Popescu (2013), model the customer disengagement decision as a deterministic known

function of service quality. Similarly, for recommendation systems, Fitzsimons and Lehmann (2004) show that poor recommendations can have considerably negative impact on customer engagement. Furthermore, customers start ignoring recommendations (disengage) if provided with sub-quality recommendations. Our customer disengagement model is motivated from the studies above but differs in that we focus on personalized recommendation and user preference learning which is not the focus of the other papers.

Johari and Schmit (2018) analyze the problem of learning customer preferences under customer disengagement. The decision maker's objective is to learn the optimal engagement level, a scalar quantity. The current work is different from their work in many ways. Since we are concerned with a collaborative filtering model for optimal recommendations, the objective is to learn a complex mapping of unknown latent features to personalized quality or rewards based on product features. This involves learning a vector of unkown features as in the case of linear bandits proposed by Auer (2002). Furthermore, a customer's disengagement decision is endoegeneous to the recommendations provided in contrast to it being an independent exogeneous quantity. This endoginity makes the problem considerably harder. This endoegenous disengagement modelling and the subbsequent use of machine learning techniques to learn customer preferences bridges the gap between the OM modelling literature of service quality optimization with the machine learning literature of user preference learning under bandit feedback.

2. Motivation

We use customer panel data from a major commercial airline to provide evidence for customer disengagement. The airline conducted a sequence of ad campaigns over email to customers that were registered with the airline's loyalty program. Our results suggest that a customer indeed disengages with recommendations if a past recommendation was irrelevant to her. This finding motivates our problem formulation described in the next section.

2.1. Data

The airline conducted 7 large-scale non-targeted ad campaigns over the course of a year. Each campaign involved emailing loyalty customers destination recommendations hand-selected by a marketing team at discounted rates. Importantly, these recommendations were made uniformly across customers regardless of customer-specific preferences.

Our sample consists of 130,510 customers. For each campaign, we observe whether or not the customer clicked on the link provided in the email after viewing the recommendations. We assume that a click signals a positive reaction to the recommendation, while no click could signal either (i) a negative reaction to the recommendation, or (ii) that the customer is already disengaged with the airline campaign and is no longer responding to recommendations.

2.2. Empirical Strategy

Since recommendations were not personalized, we use the heterogeneity in customer preferences to understand customer engagement in the current campaign as a function of the customer-specific quality of recommendations in previous campaigns. To this end, we use the first 5 campaigns in our data to build a score that assesses the relevance of a recommendation to a particular customer. We then evaluate whether the quality of the recommendation in the 6^{th} (previous) campaign affected the customer's response in the 7^{th} (current) campaign after controlling for the quality of the recommendation in the 7^{th} (current) campaign. Our reasoning is as follows: in the absence of customer disengagement, the customer's response to a campaign should depend only on the quality of the current campaign's recommendations; if we instead find that the quality of the previous campaign's recommendations plays an additional negative role in the likelihood of a customer click in the current campaign, then this strongly suggests that customers who previously received bad recommendations have disengaged from the airline campaigns.

We construct a personalized relevance score of recommendations for each customer using click data from the first 5 campaigns. This score is trained using the standard collaborative filtering package available in Python, and achieves an in-sample RMSE of 10%. A version of this score was later implemented in practice by the airline for making personalized recommendations to customers in similar ad campaigns, suggesting that it is an effective metric for evaluating customer-specific recommendation quality.

2.3. Regression Specification

We perform our regression over the 7^{th} (current) campaign's click data. Specifically, we wish to understand if the quality of the recommendation in the 6^{th} (previous) campaign affected the customer's response in the current campaign after controlling for the quality of the current campaign's recommendation. For each customer i, we use the collaborative filtering model to evaluate the relevance score $prev_i$ of the previous campaign's recommendations and the relevance score $curr_i$ of the current campaign's recommendation. We then perform a simple logistic regression as follows:

$$y_i = f(\beta_0 + \beta_1 \cdot prev_i + \beta_2 \cdot curr_i + \epsilon_i)\,,$$

where f is the logistic function and y_i is the click outcome for customer i in the current campaign, and ϵ_i is i.i.d. noise. We fit an intercept term β_0, the effect of the previous campaign's recommendation quality on the customer's click likelihood β_1, and the effect of the current campaign's recommendation quality on the customer's click likelihood β_2. We expect β_2 to be positive since better recommendations in the current campaign should yield higher click likelihood in the current campaign. Our null hypothesis is that $\beta_1 = 0$, and a finding that $\beta_1 < 0$ would suggest that customers disengage from the campaigns if previous recommendations were of poor quality.

2.4. Results

Our regression results are shown in Table 1. As expected, we find that customers are more likely to click if the current campaign's recommendation is relevant to the customer, i.e., $\beta_2 > 0$ (p-value = 0.02). More importantly, we find evidence for customer disengagement since customers are less likely to click in the current campaign if the *previous* campaign's recommendation was not relevant to the customer, i.e., $\beta_2 > 0$ (p-value = 7×10^{-9}). In fact, our point estimates suggest that the disengagement effect dominates the value of the current campaign's recommendation since the coefficient β_1 is roughly three times the coefficient β_2. In other words, it is much more important to have offered a relevant recommendation in the previous campaign (i.e., to keep customers engaged with the campaigns) compared to offering a relevant recommendation in the current campaign to get high click likelihood. These results motivate the problem formulation in the next section explicitly modeling customer disengagement.

Variable	**Point Estimate**	**Standard Error**
(Intercept)	-3.62^{***}	0.02
Relevance Score of Previous Ad Campaign	0.06^{***}	0.01
Relevance Score of Current Ad Campaign	0.02^{*}	0.01

$^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$

Table 1 Regression results from airline ad campaign panel data.

3. Problem Formulation

3.1. Preliminaries

We embed our problem within the popular product recommendation framework of collaborative filtering (Sarwar et al. 2001, Linden et al. 2003). In this setting, the key quantity of interest is a matrix $A \in \mathbb{R}^{m \times n}$, whose entries A_{ij} are numerical values rating the relevance of product j to customer i. Most of the entries in this matrix are missing since a typical customer has only evaluated a small subset of available products. The key idea behind collaborative filtering is to use a low-rank decomposition

$$A = U^T V\,,$$

where $U \in \mathbb{R}^{m \times d}, V \in \mathbb{R}^{d \times n}$ for some small value of d. The decomposition can be interpreted as follows: each customer $i \in \{1, ..., m\}$ has an associated d-dimensional vector U_i (row i of the matrix U) that models their preferences; similarly, each product $j \in \{1, ..., n\}$ has an associated d-dimensional vector V_j (given by column j of the matrix V) that models its attributes. Then, the relevance or utility of product j to customer i is simply $U_i^T V_j$. We refer the reader to Su and Khoshgoftaar (2009) for an extensive review of the collaborative filtering literature. We assume that the platform

has a large base of existing customers from whom we have already learned good estimates of the matrices U and V. In particular, all existing customers are associated with known vectors $\{U_i\}_{i=1}^m$, and similarly all products are associated with known vectors $\{V_j\}_{j=1}^n$.

Now consider a single new customer that arrives to the platform. She forms a new row in A, and all the entries in her row are missing since she is yet to view any products. Like the other customers, she is associated with some vector $U_0 \in \mathbb{R}^d$ that models her preferences, i.e., her expected utility for product $j \in \{1, ..., n\}$ is $U_0^T V_j$. However, U_0 is unknown because we have no data on her product preferences yet. We assume that $U_0 \sim \mathcal{P}$, where $\mathcal{P}$ is a known distribution over new customers' preference vectors; typically, $\mathcal{P}$ is taken to be the empirical distribution of known preference vectors associated with the existing customer base $\{U_1, ..., U_m\}$.

At each time t, the platform makes a product recommendation $a_t \in \{V_1, ..., V_n\}$, and observes a noisy signal of the customer's utility

$$U_0^T a_t + \epsilon_t \,,$$

where ϵ_t is R_1-subgaussian noise. Online service providers such as Netflix, Amazon and others often make single recommendations through email marketing campaigns. In Figure 1 we show screenshots of two such emails where a single product was recommended to the customer.

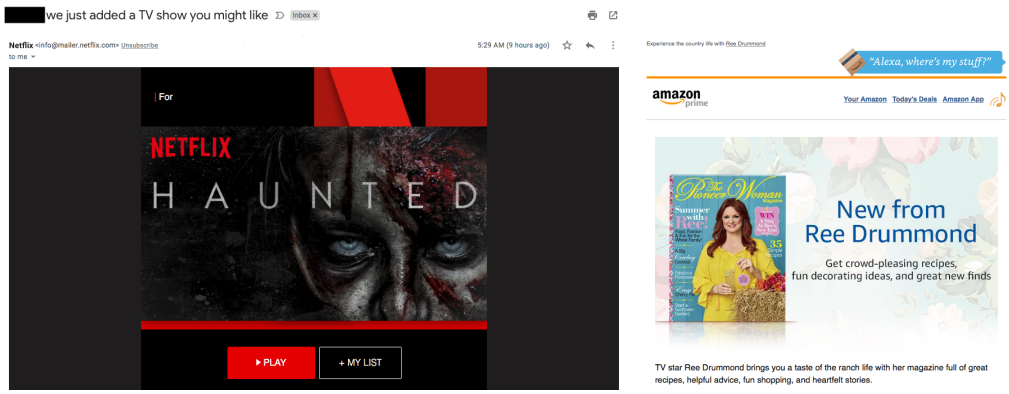

Figure 1 Examples of a single recommendation through digital marketing campaigns. On the left, a personalized email from Netflix which only recommends a single TV show. Similarly, on the right, Amazon Prime promotes the release of a book.

We seek to learn U_0 through the customer's feedback from a series of product recommendations in order to eventually offer her the best available product

$$V_* = \arg \max_{V_j \in \{V_1, ..., V_n\}} U_0^T V_j \,.$$

We impose that $U_0^T V_* > 0$, i.e., the customer receives positive utility from being matched to her most preferred product on the platform; if this is not the case, then the platform is not appropriate for the

customer. The problem of learning U_0 now reduces to a classical linear bandit (Rusmevichientong and Tsitsiklis 2010), where we seek to learn an unknown parameter U_0 given a discrete action space $\{V_j\}_{j=1}^n$ and stochastic linear rewards. However, as we describe next, our formulation as well as our definition of regret departs from the standard setting by modeling customer disengagement.

3.2. Disengagement Model

Let T be the time horizon during which the customer will stay on the platform if she is engaged. Unfortunately, poor recommendations can cause the customer to disengage from the platform. In particular, at each time t, upon viewing the platform's product recommendation a_t, the customer makes a choice $d_t \in \{0,1\}$ on whether to disengage. $d_t = 1$ signifies that the customer has disengaged and receives zero utility for the remainder of the time horizon T; on the other hand, $d_t = 0$ signifies that the customer has chosen to remain engaged on the platform for the next time period.

There are many ways to model disengagement. For simplicity, we consider the following: each customer has a tolerance parameter $\rho > 0$ and a disengagement propensity $p \in [0,1]$. Then, the probability that the customer disengages at time t (assuming she has been engaged until now) upon receiving recommendation a_t is:

$$\Pr[d_t = 1 \mid a_t] = \begin{cases} 0 & \text{if } \; U_0^T a_t \;\geq\; U_0^T V_* - \rho\,, \\ p & \text{otherwise.} \end{cases}$$

In other words, each customer is willing to tolerate a utility reduction of ρ from a recommendation with respect to her utility from her (unknown) optimal product V_*. If the platform makes a recommendation that results in a utility reduction greater than ρ, the customer will disengage with some probability p. Note that when $p = 0$, we recover the classical bandit setting with no disengagement.

We seek to construct a sequential decision-making policy π that learns U_0 over time to maximize the customer's utility on the platform. We measure the performance of π by its *cumulative expected regret*, where we modify the standard metric in the analysis of bandit algorithms (Lai and Robbins 1985) to accommodate customer disengagement. In particular, we compare the performance of our policy π against an oracle policy π^* that knows U_0 in advance and always offers the customer her preferred product. At time t, we define the instantaneous expected regret for a given new customer as

$$r_t(U_0) = \begin{cases} U_0^T V_* & \text{if } \; d_{t'} = 1 \;\; \text{for any} \;\; t' < t\,, \\ U_0^T V_* - U_0^T a_t & \text{otherwise.} \end{cases}$$

This is simply the expected utility difference between the oracle's recommendation and our policy's recommendation, accounting for the fact that the customer receives zero utility for all future recommendations after she disengages. The expectation is taken with respect to ϵ_t, the σ-subgaussian

noise in realized customer utilities that was defined earlier. We seek to minimize expected cumulative regret

$$R(T,\rho) = \mathbb{E}_{U_0 \sim \mathcal{P}} \left[\sum_{t=1}^{T} r_t(U_0) \right], \tag{1}$$

where we additionally take the expectation over the new customer's preference vector.

In what follows, we assume that the second norm of known product features are bounded by a large constant, L. That is,

$$||V_i||_2 \leq L \, \forall i.$$

This assumption is very common in the bandit learning literature. See, for example, Abbasi-Yadkori et al. (2011).

4. Classical Approaches

THEOREM 1 **(Hardness Result)**. *Any policy achieves regret that scales linearly with T.*

Proof: See Appendix B.1 □

The proof relies on constructing a simple example with two possible products (denoted by vectors [1,0] and [0,1]) and two possible customer features ([1,0] and [0,1]) with discrete uniform prior on the two user features. Letting $\rho \in (0,1)$ and $p = 1$, an irrelevant product recommendation leads to customer disengagement in the first round. This implies that in expectation, any recommendation algorithm is bound to incur linearly increasing rate of regret.

Theorem 1 shows that recommendation with customer disengagement is a challenging problem. While the simple example considered in Theorem 1 assumes a uniform prior over the possible latent user features, it is easy to observe that a low value of ρ and a high value of p ensure linear regret for any measurable set of customers with independent preferences. That is, no algorithm can keep all the users engaged without knowing the user's preference a-priori. Another approach could be to ensure that at least a large fraction of customers (*main stream customers*), if not all, can be kept engaged for the entire time horizon and only customers with unique preferences are disengaged from the platform (*tail customers*). In Theorem 2, we show that classical bandit learning algorithms fail to achieve engagement for even the main stream customer for the entire time horizon.

We note that a *consistent* bandit algorithm (Lattimore and Szepesvari (2016)) is any policy π which is consistent with respect to the classical cumulative regret definition used in traditional bandit learning settings. More specifically, if we let $\rho \to \infty$, then (1) reduces to the widely studied bandit setting where customer never disengages from the platform. We define consistency with respect to this classical cumulative regret definition.

DEFINITION 1 (LATTIMORE AND SZEPESVARI (2016)). A bandit algorithm, π, is consistent if, $\forall p > 0$, $R(T,\rho)_{\rho\to\infty} = o(T^p)$. This is equivalent to the following condition:

$$\lim\sup_{T,\rho\to\infty} \frac{R(T,\rho)}{\log(T)} \leq 0$$

We will also use the following Lemma that bounds the matrix norm of suboptimal products in terms of their optimality gap.

LEMMA 1. *Let π be a consistent policy, $u_0 \in R^d$ such that there is a unique optimal product, V_* amongst the set of feasible products. Then $\forall\ V \in \{V_1, ...V_n\}/\ V_*$*

$$\limsup_{t\to\infty} \log(t)||V||^2_{X_t^{-1}} \leq \frac{\Delta_V}{2}$$

where $\Delta_V = u_0^T V_* - u_0^T V$ and $X_t = \mathbb{E}\left[\sum_{l=1}^{l=t} a_l a_l'\right]$

Proof: See Appendix B.1. □

Finally, we introduce some more notation that defines the set of relevant products for an incoming customer.

DEFINITION 2. Let $\mathcal{S}(u_0)$ be the set of products, amongst all products, that satisfy the tolerance threshold for the customer with feature vector u_0. More specifically,

$$\mathcal{S}(u_0) := \{i : u_0^T V_i\ \geq\ u_0^T V_* - \rho, \forall i = 1, .., n\} \tag{2}$$

THEOREM 2 **(Failure of Bandits)**. *Any classical bandit algorithm as defined above fails to keep any customer engaged on the platform for the entire time horizon T as $T \to \infty$.*

Proof: See Appendix B.1 □

The proof relies on constructing a simple counter example where the number of products in the set, $\mathcal{S}(u_0)$ (2), are less than the dimension of the user feature vector, d. Consistency, ensures that the algorithm recommends products outside of $\mathcal{S}(u_0)$ infinitely often. Intuitively, if products are not recommended outside of $\mathcal{S}(u_0)$ infinitely often, there would be a direction in which customer feature learning would be incomplete. This incomplete learning would lead to the algorithm being inconsistent but at the same time infinite often recommendations outside of set $\mathcal{S}(u_0)$ would in-turn lead to eventual disengagement of the customer. The above result is particularly alarming as it proves the poor performance of classical near optimal bandit algorithms in our setting. Inevidently, not incorporating customer disengagement implies recommending irrelevant products infinitely often which results in customer disengagement over time. Note that this is worst then incurring linear regret. While Theorem 1 shows that all algorithms incur linear regret, Theorem 2 shows that classical bandit algorithms incur linear regret for all types of customers regardless of

the likelihood of the customer arriving on the platform. That is, classical bandit algorithms exhibit poor performance on both main stream and tail customers.

A natural next step is to ask whether a certainty equivalent greedy policy which uses the Maximum likelihood Estimate (MLE) of the unknown user vector, u_0 and recommends the corresponding optimal product in each round. Bastani et al. (2017) have shown that such policies perform well in various contextual bandit settings. Nevertheless, in what follows, we show that unfortunately such a policy would also incur linear regret with high probability in our setting.

EXAMPLE 1. Let $\bar{u}$ be the initial prior mean on unknown customer feature vector, $\Sigma = \sigma I^d$ be the initial covariance matrix of the customer features and let $U_0 \sim \mathcal{N}(\bar{u}, \Sigma)$.

The myopic policy (Algorithm 1) recommends products based on $\bar{u}$ in the first round, observes customer response, updates the prior and offers the optimal product based on the new prior. λ is the regularization parameter for ridge regression that ensures that the estimation of unknown parameters is not degenerate.

Algorithm 1 MyopicBandit(λ)

Initialize and recommend $a_1 = \arg\max_{i=1\ldots,n} u^T V_i$
for $t \in [T]$ **do**
  Observe customer utility, $Y_t = U_0^T a_t + \epsilon_t$
  Update customer feature estimate, $\hat{u}_{t+1} = (V_{1:t}^T V_{1:t} + \lambda I)^{-1} V_{1:t} Y_{1:t}$
  Recommend product $a_{t+1} = \arg\max_{i=1\ldots,n} \hat{u}_{t+1}^T V_i$
end for

Note that the myopic policy uses all customer response ($Y_{1:T}$) in order to estimate the current user feature (Step 1) and then offers the myopic optimal product (Step 3).

In what follows, we will show that the myopic policy is guaranteed to incur linear regret even for the case when $\rho \to \infty$, that is, the customer does not disengage from the platform due to bad recommendations. The proof relies on a self-normalized ellipsoidal uncertainty set constructed for L2 regularized regression coefficients by Abbasi-Yadkori et al. (2011).

THEOREM 3 **(Failure of Greedy)**. *Let* $\Delta_{max} = max_{\ i=1,..,n} \frac{V_{i^*}^T \bar{u} - V_i^T \bar{u}}{V_i^T \Sigma V_i}$ *and* $\bar{B}_{min} = min_{\ i=1,..,n} \bar{B}_i$ *where*

$$\bar{B}_i = \frac{\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+TL^2}{\delta}\right)} + \sqrt{\lambda}\left(\sqrt{\log\left(\frac{2d}{\delta}\right)} 2d\sigma + \sum_{i=1}^{i=d} \mu_i\right)\right) - (V_{i^*} - V_i)^T \bar{u}}{(V_{i^*} - V_i)^T \Sigma (V_{i^*} - V_i)}$$

If

$$\left(\frac{1}{\bar{B}_{min}} - \frac{1}{\bar{B}_{min}^2}\right) \frac{e^{-\bar{B}_{min}^2}}{\sqrt{2\pi}} \geq 1 - \frac{1}{2KT}$$

then, under the conditions of Example 1 as $\rho \to \infty$*, with probability at least* $(1-\delta)^2 \left(\frac{e^{\frac{-\Delta_{max}^2}{2}}}{2\sqrt{2\pi}\Delta_{max}} \right)$ *Algorithm 1 incurs linear regret.*

Proof: See Appendix B.1. □

The proof relies on analyzing the probability of two events: (i) Recommending a sub-optimal product in the first round, (ii) Continuing with the sub-optimal recommendation for the rest of the rounds. Lower bounding the probability of the two events results in a lower bound on the probability of recommending a suboptimal product through out the time horizon. This shows that the myopic policy gets "stuck" on a suboptimal product, due to incomplete learning, even when customers' never disengage.

Customer disengagement decision lies on a spectrum of two extreme choices. On one hand, high tolerance threshold customers (small ρ) would disengage from the platform if poor recommendations are made. On the other hand, low tolerance threshold customers (high ρ) would never disengage regardless of the recommendations made by the platform. In Theorem 2 we have shown that classical bandit algorithms perform poorly when customers are likely to disengage and in Theorem 3 we show that the greedy policy performs poorly even when customers' engagement decision is not dependent on the relevance of the recommendations made.

Next, we consider the following question: can customers remain engaged in a more general setting when they do not lie on one of the two extremes of no disengagement and deterministic disengagement? While classical bandit algorithms failed because of over exploration over products that were of low relevance for the incoming customer, myopic policy failed because of no exploration at all. Motivated by these findings, we will construct a well performing algorithm that interpolates between these two extreme choices: explore but only on carefully selected products that have a high probability of lying within the customers' tolerance threshold.

5. Proposed Algorithm

In the previous section we established that classical bandit policies (Theorem 2) and a greedy policy (Theorem 3) fail to perform well in our setting. Nevertheless, in what follows, we show that a simple two step approach of first constraining the exploration over a restricted set of products and then using a classical UCB approach (Abbasi-Yadkori et al. (2011)) guarantees sub-linear regret over a large fraction of customers. We start by describing an Integer Programming based approach of constraining exploration.

5.1. Intuition

In order to develop intuition for an improved algorithm, we start by discussing the reasons behind the failure of the bandit and the greedy approaches. As we have seen earlier, classical bandit algorithms fail because they fail to discard irrelevant products for the current customer. Intuitively, since there is a constant probability, p, of customer disengagement, unless the recommender system

disregards these sub-optimal products, the customer is eventually going to disengage from the platform. The failure of classical bandit algorithms alludes to the fact that only algorithms that can disregard irrelevant products for the incoming customer have a chance of keeping the customer engaged over time. While finding the set of relevant products without knowing the customer features is impossible, a probabilistic estimate of relevance of each product is still feasible. We use this intuition to generate a constrained exploration set for exploring over the next customer. We develop an Integer Programming formulation that is used to disregard products which do not meet the tolerance threshold of customers with high probability. The procedure ensures that all remaining products in the constrained set satisfy the unknown customer tolerance threshold with a very high probability. Our formulation relies on using the prior distribution information on the unknown latent user feature to calculate relevance probabilities of different products. Intuitively, customers are sampled from the prior distribution and the constrained set is also optimized over the same prior distribution. This implies that products in the constrained set would satisfy the tolerance threshold for a substantial portion of customers (main-stream customers) and only risk being irrelevant for tail customers. While tail customers would eventually disengage, by construction, the likelihood of tail customers will be relatively small which would ensure engagement over a large fraction of customers. Furthermore, we use a constrained threshold (γ) in our formulation, that can be used to tune the amount of constraining based on the prior information on the incoming customers. We present the details of this approach in Section 5.2.

Next, we discuss the failure of the myopic policy (Algorithm 1). Intuitively, the myopic policy fails because it does not explore amongst the feasible product set. Since the myopic policy has no incentive to explore, this could lead to failure in the identification of the optimal product. In some sense, failure of the myopic policy is more intuitive than the failure of classical bandit algorithms. Clearly a relatively better algorithm would explore the product set before narrowing down on a single product. We use this intuition to add a second step in our learning algorithm that explores over the constrained set of products using a bandit learning policy. Details of the exploration-learning step are presented in Algorithm 2.

5.2. Constrained Exploration

In order to efficiently constrain exploration under customer disengagement, we select the largest set of products that satisfy customer tolerance threshold. Indeed if customer features were known, the optimal product could have been trivially calculated. When customer features are unknown, the problem of optimizing a constrained set of products becomes significantly harder. In order to find this constrained set, we use the features of existing users to infer product relevance for the next

potential customer. probabilistic distribution. In what follows we will assume that the customer tolerance threshold, ρ is known a-priori. We discuss the uncertainty around ρ in Section 7.

Before we describe the problem, we first introduce some more notation to make the exposition clear.

DEFINITION 3. Let C_i define the probability of product i satisfying the tolerance threshold for a customer. That is,

$$C_i = \mathbb{P}(i \in \mathcal{S}(U)),$$

where $\mathcal{S}(u_0)$ is defined as in Definition 2. Note that the probability is measured with respect to the distribution over the random customer feature vector, U. Calculating C_i for each product is computationally inexpensive. For example, one can use Monte Carlo simulations to generate random customer features according to the prior distributional assumptions of the feature vector and the tolerance threshold ρ. C_i captures the probability that product i would satisfy the customer tolerance. Naturally, products that are likely to satisfy customer tolerance should be the ones over which exploration should be performed. Larger ρ implies loose tolerance thresholds which would result in higher C_i values and vice versa. Nevertheless, it is easy to notice that the relevance score for a set of products is monotonic in the size of the set. That is, the more the products, the more the chances that one of them is relevant for the next incoming customer. Because the next customer is unknown a-priori, this constraint free exploration would lead to exploration over irrelevant products and potentially cause the disengagement of the next customer from the platform.

Hence, in order to constrain exploration, we consider the Euclidean distance between the known features of different products. Let

$$D_{ij} = ||V_i - V_j||_2,$$

where D_{ij} defines the Euclidean distance between products i and j. D_{ij} captures the similarity between two products. If a product is more likely to be relevant, then it is more likely that products similar to it will be relevant. For example, users who like science fiction movies would be more likely to "tolerate" adventure movie recommendations than horror movie recommendations. Furthermore, science fiction movies and adventure movies would be close to each other in the Euclidean space. We use this intuition to constrain exploration. Let γ denote the threshold parameter that governs the size of exploration set. More precisely, let $T_{ij}(\gamma)$ be an indicator function that determines whether $D_{ij} \leq \gamma$. Hence,

$$T_{ij}(\gamma) = \begin{cases} 1, & \text{if } D_{ij} \leq \gamma \\ 0, & \text{otherwise} \end{cases}$$

Consider the following Integer Programming formulation ($IP_{constrained}(\gamma)$) to select the optimal set of products to explore with.

$$\max_{x,z} \sum_{i=1}^{k} C_i x_i \tag{3a}$$

$$\text{s.t. } z_{ij} \leq x_i, \quad i = 1, \ldots, k, \tag{3b}$$

$$z_{ij} \leq x_j, \quad j = 1, \ldots, k, \tag{3c}$$

$$z_{ij} \geq x_i + x_j - 1, \quad i = 1, \ldots, k, \quad j = 1, \ldots, k \tag{3d}$$

$$z_{ij} \leq T_{ij}, \quad i = 1, \ldots, k, \quad j = 1, \ldots, k \tag{3e}$$

$$x_i \in \{0, 1\} \quad i = 1, \ldots, k \tag{3f}$$

The decision variables in the above problem are $x_i, i = 1, ..., n$ and $z_{i,j}, i = 1, ..., n, j = 1, ..., n$. In particular, x_i in $IP_{constrained}(\gamma)$ defines whether product i is included in the exploration set. Furthermore, $z_{i,j}$ takes value 1 if both products i and j are included in the exploration set. Constraints (3b) to (3e) ensure that only products that are "close" to each other are selected for exploration.

Solving ($IP_{constrained}(\gamma)$) for a threshold parameter γ results in a set of products (products for which the corresponding x_i is 1) which are within γ distance from each other and have the highest likelihood of satisfying the customer tolerance level. In Section 7, we perform computational experiments to show that $IP_{constrained}(\gamma)$ can be solved in reasonable computational time for relatively large scale problems using off-the-shelf IP solvers.

Next,we state the Constrained Bandit algorithm that uses exploration in a constrained manner to learn unknown customer features.

Algorithm 2 Constrained Bandit(λ,γ)

Step 1: Constrained Exploration:

Solve $IP_{constrained}(\gamma)$ to get the constrained set of products to explore over, $S_{constrained}$. Let $a_1 = \arg\max_{\{i \in S_{constrained}\}} \bar{u}^T V_i$

Step 2: Bandit Learning:
for $t \in [T]$ **do**

Observe customer utility, $Y_t = U_0^T a_t + \epsilon_t$

Let $\hat{u}_t = (V_{a_1:a_t}^T V_{a_1:a_t} + \lambda I)^{-1} V_{a_1:a_t} Y_{1:t}$ and

$$\mathcal{T}_t = \left\{ u \in \mathbb{R}^d : ||\hat{u}_t - u||_{\bar{X}_t} \leq \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \sqrt{\lambda} S \right) \right\}$$

Let $(u_{opt}, a_t) = \arg\max_{\{i \in S_{constrained}, u \in \mathcal{T}_t\}} u^T V_i$.

Recommend product a_t at time t if the customer is still engaged. Stop if the customer disengages from the platform.
end for

The constrained Bandit algorithm (Algorithm 2) takes as an input two parameters: λ, the regularization parameter for parameter estimation and γ, the constrained set tuning parameter for constraining the exploration set. We note that λ is introduced to make the estimation non degenerate and the analysis tractable. Moreover, Abbasi-Yadkori et al. (2011) and others in the bandit learning literature also use regularized regression in the parameter estimation step for similar reasons. The only parameter that we introduce is the constraint threshold γ. We discuss the selection of γ and the corresponding tradeoffs in Section 6.1.

The Constrained Bandit algorithm works in two steps. In the first step, we use $IP_{constrained}(\gamma)$ to generate a subset of products over which to explore. This constrained exploration ensures that we satisfy tolerance thresholds for mainstream customers and explore over a set of products that are relevant for a large set of customers. In the second step we use confidence ellipsoids on the unknown user vector (Abbasi-Yadkori et al. (2011)) to perform bandit learning on the constrained set. The main idea remains simple: since customers leave the system, the platform should be cautious while exploring with customers and hence only explore in a constrained set of products. Given that we do not know customer features, constrained exploration is optimized for mainstream customers which will be more likely to visit the platform.

Next, we discuss the regret guarantee of ContrainedBandit(λ,γ,T) and show that with positive probability, ContrainedBandit(λ,γ,T) gets sub-linear regret for a large portion of customers.

6. Theoretical Guarantees

In this Section we establish that the Contrained Bandit performs well and incurs sub linear regret over a large fraction of customers. Intuitively, since exploration is constrained to products that are relevant for a large fraction of customers, sublinearity of regret is also ensured for this set of customers.

Definition 4. Let,

$$L_{t,\rho} = \begin{cases} 1 & \text{Customer with quality threshold } \rho \text{ engaged until time t}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly,

$$\mathbb{1}\{L_{T,q} = 1\} = \Pi_{t=1}^{T} \mathbb{1}\{d_t = 0\}$$

where recall, by definition, d_t is the disengagement decision of the customer at time t and $L_{t,\rho}$ denotes the overall engagement decision of the user until time t. Also note that $\mathbb{P}(L_{T,q})$ is monotonically decreasing in T. We are now in a position to discuss the rate of regret for Constrained Bandit.

In order to analyze the regret of the Constrained Bandit algorithm, we consider a simple setting where $U_0 \sim \mathcal{N}(\bar{u}, \frac{\sigma}{d} I^d)$, $||\bar{u}||_2 = 1$. Also let the product features, $V \in R^d$ be uniformly distributed in $[-1, 1]^d$. This simple setting of product features in the d dimensional space ensures that the analysis is clean and succinctly shows the relevance of constraining exploration before attempting to learn user features. Recall that from Section 4, we have already shown that no other policy performs well in the setting when customers can possibly disengage from the platform. Theorem 4 shows that the Constrained Bandit algorithm indeed performs well for a large fraction of customers under the stylized setting. In Section 7, we will analyze the numerical performance of the algorithm in a real world setting and further provide evidence of the usefulness of the Constrained Bandit algorithm.

Theorem 4 **(Guarantee for Proposed Algorithm)**. *Let $U_0 \sim \mathcal{N}(\bar{u}, \frac{\sigma}{d} I^d)$, $||\bar{u}||_2 = 1$. Also let the product features, $V \in R^d$ be uniformly distributed in $[-1, 1]^d$. Let,*

$$w = \left(1 - 2de^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}\right)\left(1 - 2de^{-\left(\frac{\frac{\rho}{\gamma} - \sum_{i=1}^{i=d} \bar{u}_i}{2d\sigma}\right)^2}\right).$$

Then for at least w fraction of customers, with probability at least $(1-\delta)$, the cumulative regret of Constrained Bandit is,

$$\begin{aligned} R(T) &= 4\sqrt{Td\log\left(\lambda + \frac{TL}{d}\right)}\left(\sqrt{\lambda}\frac{\rho}{\gamma} + R_1\sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{TL}{\lambda d}\right)}\right) \\ &= \tilde{\mathcal{O}}\left(\sqrt{T}\right) \end{aligned}$$

We briefly discuss the implications of Theorem 4 before detailing the proof. Constrained Bandit algorithm (Algorithm 2) operates in two steps. In the first step, the exploration is constrained on a restricted set, $S_{constrained}$. Bandit learning follows in the second step where the decison maker learns the optimal recommendation over the restricted set of products to experiment with. Naturally, the regret depends on two factors: (i) If the optimal product is contained in the constrained exploration set and (ii) If the relevance of recommendations can ensure user engagement over the time horizon. If the constrained set does not contain the optimal product, the decision maker is bound to incur linear regret due to the fact that she can never recommend the optimal product. Similarly, if the customer leaves the platform due to sub quality recommendations, the decision maker incurs linear regret on account of lost customer opportunity. In Theorem 4, we bound the probability of the above two events. Finally, sub linearity of regret over this set of customers follows from self normalized tail bounds of Abbasi-Yadkori et al. (2011).

Proof of Theorem 4: We will prove the above result in three steps. In the first step we will lower bound the probability that the constrained exploration set, $S_{constrained}$, contains the optimal product for an incoming vector. In the second step we will lower bound the probability of customer engagement over the constrained set. Finally, in the last step, we use the above lower bounds on probabilities to upper bound regret from the ConstrainBandit algorithm.

Step 1 (Lower bounding the probability of not choosing the optimal product for an incoming customer in the constrained set): Let, $\mathcal{E}_{no-optimal}$, be the event that the optimal product, V_* for the incoming user is not contained in $S_{constrained}$. Also let $\tilde{i} = \arg\max_{V\in[-1,1]^d} \bar{u}^T V$, denote the features of the prior optimal product. Notice that $V_{\tilde{i}} = \bar{u}$ since $||\bar{u}||_2 = 1$. Also recall that

$$V_* = \arg\max_{V\in[-1,1]^d} U_0^T V$$

denotes the current optimal product which is unknown because of unknown customer features. We are interested in

$$\mathbb{P}\left(\mathcal{E}_{no-optimal}\right) = \mathbb{P}\left(V_* \notin S_{contrained}\right)$$

In order to characterize the above probability, we focus on the structure of the constrained set, $S_{contrained}$. Recall that $S_{constrained}$ is the outcome of Step 1 of Constrained Bandit (Algorithm 2) and uses $IP_{constrained}(\gamma)$ to restrict the exploration space. It is easy to observe that $S_{constrained}$ in the continuous feature space case would be centred around the prior optimal product vector ($\bar{u}$) and will contain all products that are at most γ *away* from each other. Figure 2 plots the constrained set under consideration. Note that any realized $U_0 \notin [\bar{u}_l, \bar{u}_r]$ is contained in $\mathcal{E}_{no-optimal}$. Hence, we characterize this probability next.

$\bar{u}_l$ and $\bar{u}_r$ denote the farthest products inside a γ constrained sphere. Furthermore, $\bar{u}_l$ and $\bar{u}_r$ are divided in two equal halves of length $\frac{\gamma}{2}$ by a perpendicular line segment between $\bar{u}$ and the

origin due to the symmetry of the constrained set. Simple geometric analysis yields that $\bar{u}$ and $\bar{u}_l$ are $\sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)}$ apart. The distance between $\bar{u}$ and $\bar{u}_r$ follows symmetrically.

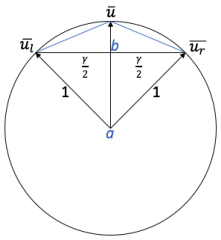

Figure 2 Analyzing the probability of $\mathcal{E}_{no-optimal}$. Note that we are interested in characterizing the length of line segments $(\bar{u}_l, \bar{u})$ and $(\bar{u}, \bar{u}_r)$

Having calculated the distance between $\bar{u}$ and $\bar{u}_l$, we are now in a position to characterize the probability of $\mathcal{E}_{no-optimal}$.

$$\mathbb{P}\left(\mathcal{E}_{no-optimal}\right) = \mathbb{P}\left(V_* \notin S_{contrained}\right) = \mathbb{P}\left(||U_0 - \bar{u}||_2 \geq \sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)}\right)$$

Note by Holder's inequality that,

$$\sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)} \leq |||U_0 - \bar{u}||_2 \leq |||U_0 - \bar{u}||_1$$

which implies that,

$$\mathbb{P}\left(\mathcal{E}_{no-optimal}\right) = \mathbb{P}\left(||U_0 - \bar{u}||_2 \geq \sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)}\right) \leq \mathbb{P}\left(||U_0 - \bar{u}||_1 \geq \sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)}\right)$$

Note that $U_0 \sim \mathcal{N}(\bar{u}, \frac{\sigma}{d} I^d)$. Hence, using Lemma 2, we have that,

$$\mathbb{P}\left(||U_0 - \bar{u}||_1 \leq \sqrt{2\left(1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}\right)}\right) \geq 1 - 2de^{-\left(\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}\right)}$$

Hence,

$$\mathbb{P}\left(\mathcal{E}_{no-optimal}\right) \leq 2de^{-\left(\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}\right)}$$

Step 2 (Lower bounding the probability of customer disengagement due to relevance of the recommendation): Recall that customer disengagement decision is driven by the relevance of the recommendation and the tolerance threshold of the customer. Hence,

$$\begin{aligned}\mathbb{P}(U_0^T V_* - U_0^T V_i < \rho) &= \mathbb{P}(U_0^T U_0 - U_0^T u_i < \rho | U_0, u_i \in S_{constrained}) \\ &= \mathbb{P}(U_0^T (U_0 - u_i) < \rho | U_0, u_i \in S_{constrained}) \\ &\geq \mathbb{P}\left(||U_0||_2 < \frac{\rho}{\gamma} | U_0, u_i \in S_{constrained} \right) \\ &\geq \left(1 - 2de^{-\left(\frac{\frac{\rho}{\gamma} - \sum_{i=1}^{i=d} \bar{u}_i}{2d\sigma}\right)^2} \right)\end{aligned}$$

where the last inequality follows by Lemma 2. This in-turn shows that with probability at least $\left(1 - 2de^{-\left(\frac{\frac{\rho}{\gamma} - \sum_{i=1}^{i=d} \bar{u}_i}{2d\sigma}\right)^2} \right)$, customers will not leave the platform because of irrelevant product recommendations.

Step 3 (Sub-linearity of Regret): Recall that

$$r_t(U_0) = \begin{cases} U_0^T V_* & \text{if } \ d_{t'} = 1 \ \text{ for any } \ t' < t\,, \\ U_0^T V_* - U_0^T a_t & \text{otherwise.} \end{cases}$$

In other words,

$$r_t = (U_0^T V_* - U_0^T a_t) \mathbb{1}\{L_{t,\rho} = 1\} + U_0^T V_* \mathbb{1}\{L_{t,\rho} = 0\}.$$

Nevertheless,

$$\begin{aligned} r_t(U_0) &= \left(U_0^T V_* - U_0^T a_t\right) \mathbb{1}\{L_{t,\rho} = 1\} + (U_0^T V_*) \mathbb{1}\{L_{t,\rho} = 0\} \\ &= \left(U_0^T V_* - U_0^T a_t\right) \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\} + (U_0^T V_*)(1 - \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\}) \\ &= \left(U_0^T V_* - U_0^T a_t\right) \Pi_{t=1}^T \mathbb{1}\{dt, \rho = 0\} + (U_0^T V_* + u_0^T a_t - u_0^T a_t)(1 - \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\}) \\ &= (U_0^T V_* - U_0^T a_t) + U_0^T a_t (1 - \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\}). \end{aligned}$$

Note that the first part in the above expression is related to the regret of the classical bandit setting where the customer does not disengage while the second part is associated with the regret when the customer disengages from the platform and the platform incurs maximum regret.

Next, focusing on cumulative regret and taking expectation over the random customer response on quality feedback (ratings), we have that

$$\begin{aligned} R(T) = \sum_{t=1}^{T} r_t(U_0) &\leq \mathbb{E}\left[\sum_{t=1}^{t=T} (U_0^T V_* - U_0^T a_t) + U_0^T a_t (1 - \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\}) \right] \\ &= \sum_{t=1}^{t=T} \mathbb{E}\left[(U_0^T V_* - U_0^T a_t) \right] + \mathbb{E}\left[U_0^T a_t \left(1 - \Pi_{t=1}^T \mathbb{1}\{d_{t,\rho} = 0\}\right) \right]. \end{aligned}$$

Note that conditional on fraction w of customers, we have that these customers would never disengage from the platform due to irrelevant personalized recommendations.

$$1 - \Pi_{t=1}^{T} \mathbb{1}\{d_{t,\rho} = 0\} = 0$$

since there is no probability of leaving when a product within the constraint set is recommended and meets the tolerance threshold (w, analyzed in the previous step). Hence,

$$R(T) = \sum_{t=1}^{t=T} \mathbb{E}\left[(U_0^T V_* - U_0^T a_t)\right].$$

Now notice that for any realization of U_0, Theorem 4 of Abbasi-Yadkori et al. (2011) shows that

$$R(T) \leq 4\sqrt{Td\,\log\left(\lambda + \frac{TL}{d}\right)}\left(\sqrt{\lambda}S + R_1\sqrt{2\log\frac{1}{\delta} + d\log\,(1 + \frac{TL}{\lambda d})}\right)$$

with probability at least 1-δ if $||U_0||_2 \leq S$. From Step 2, we have that all w fraction of customers have $||U_0||_2 \leq \frac{\rho}{\gamma}$. Hence replacing S with $\frac{\rho}{\gamma}$ gives the final result. Recall that R_1 is the error sub-Gaussian parameter and λ is the L2 regularization parameter.

□

6.1. Selecting constrained threshold parameter, γ

In the previous section, we proved that the Constrained Bandit algorithm achieves sublinear regret for a large fraction of customers. This fraction depends on the constrained threshold tuning parameter γ and other problem parameters (see Theorem 4). In this section, we explore this dependence in more detail and provide intuition on the selection of γ that maximizes this .

Recall, from Theorem 4, that the fraction of customers who remain engaged with the platform is lower bounded by,

$$w \;=\; \left(1 - 2de^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}\right)\left(1 - 2de^{-\left(\frac{\frac{\rho}{\gamma} - \sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}\right).$$

This fraction comprises of two parts. The first part, $\left(1 - 2de^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}\right)$, denotes the fraction of customers for which the corresponding optimal product is contained in the constrained exploration set, $S_{constrained}$. Notice that the fraction of customers for which the optimal product is contained in the constrained set increases as the constraint threshold, γ, increases. This follows since a larger γ implies a larger exploration set and more customer that can be served with their most relevant recommendation. Similarly, the second part, $\left(1 - 2de^{-\left(\frac{\frac{\rho}{\gamma} - \sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}\right)$, denotes the fraction

of customers who will not disengage from the platform due to irrelevant recommendations in the learning phase. Contrary to the previous case, as the constraint threshold γ increases, the fraction of customers guaranteed to engage decreases. Intuitively, as the exploration set becomes larger, their is wider range of offerings with more variability in the relevance of the recommendations for a particular customer. This wider relevance in turn leads to a decrease in the probability of engagement of a customer. Hence, γ can either increase or decrease the fraction of engaged customers based on the other problem parameters.

In Figure 3, we plot the fraction of customers who will remain engaged with the platform as a function of the constrained threshold parameter, γ, for different values of tolerance threshold, ρ. As noted earlier, the fraction of engaged customers is not monotonically increasing in γ. When γ is small, the constrained set for exploration (from Step 1 of Algorithm 2) is over constrained. Hence, increasing γ leads to an increase in the fraction of engaged customers. Nevertheless, increasing it above a threshold implies that customers are more likely to disengage from the platform due to irrelevant recommendations. Hence, increasing γ further leads to a decrease in the fraction of engaged customers. We also note that as customers become less quality concious (small ρ), the fraction of engaged customers increases for any chosen value of γ. This again follows from the fact that a higher value of ρ implies a higher probability of customer engagement in the learning phase. This increase in engagement probability during the learning phase encourages less conservative exploration (larger γ).

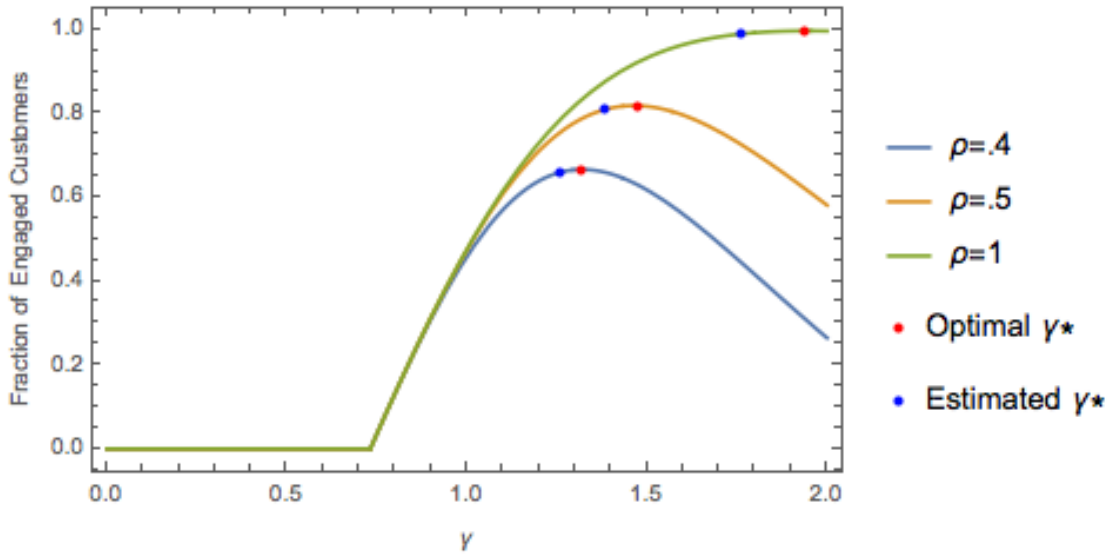

Figure 3 Fraction of engaged customer as a function of the constrained threshold parameter γ for different values of tolerance threshold, ρ. A higher ρ implies that the customer is less quality concious. Hence, for any γ, this ensures higher chance of engagement. We also plot the optimal γ that ensures maximum engagement and an approximated γ that can be easily approximated. The approximated γ is considerably close to the optimal γ and ensures high level of engagement.

The above discussion alludes to the fact that the optimal γ that maximizes the fraction of engaged customers is a function of different problem parameters and is hard to optimize in general. Nevertheless, consider the following:

$$
\begin{aligned}
w &= \left(1-2de^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}\right)\left(1-2de^{-\left(\frac{\frac{\rho}{\gamma}-\sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}\right)\\
w &= 1-2de^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}-2de^{-\left(\frac{\frac{\rho}{\gamma}-\sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}+4d^2e^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}e^{-\left(\frac{\frac{\rho}{\gamma}-\sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}\\
w &\approx \frac{1}{2d^2}-\frac{1}{2d}e^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}-\frac{1}{2d}e^{-\left(\frac{\frac{\rho}{\gamma}-\sum_{i=1}^{i=d}\bar{u}_i}{2d\sigma}\right)^2}\\
w &\approx \frac{1}{4d^2}-\frac{1}{2d}e^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}-\frac{1}{2d}e^{-\left(\frac{\frac{\rho}{\gamma}}{2d\sigma}\right)^2}.
\end{aligned}
$$

Hence, in order to maximize w, we have to solve the following minimization problem:

$$\min_{\gamma}\ e^{-\frac{1-\sqrt{\left(1-\frac{\gamma^2}{4}\right)}}{\sigma}}+e^{-\left(\frac{\rho}{2\gamma d\sigma}\right)^2} \tag{4}$$

While Problem (4) has no closed form solution, we consider the following problem:

$$\min_{\gamma}\ \frac{1}{\sigma}\sqrt{\left(1-\frac{\gamma^2}{4}\right)}-\left(\frac{\rho}{2\gamma d\sigma}\right)^2. \tag{5}$$

Note that (5) is an approximation of (4) based on the Taylor series expansion of the exponent function. Solving (4) using FOC conditions, a suitable choice of γ yields the following:

$$\gamma^* \in \{\gamma : \rho = \frac{\sqrt{\sigma} d \gamma^2}{(4-\gamma^2)^{1/4}} \text{ and } \gamma > 0\}$$

While γ^* is not optimal, it provides directional insights to managers on suitable choices of γ. For example, as ρ increases the estimated optimal γ also increases. Furthermore, it decreases with the prior variance σ and the dimension of the latent feature vector d. A lower variance yields better understanding of the unknown customer and leads to lower size of the optimal exploration set. Similarly, as the latent vector dimension, d, increases, there are higher chances of not satisfying customer relevance thresholds in the learning phase. This leads to a more constrained exploration.

In order to analyze the estimated optimal γ, we compare the estimated optimal γ with the numerically calculated optimal γ for different values of ρ, the customer tolerance threshold. In Table 2, we show the gap in the lower bound of engaged customers from choosing the optimal γ vs the estimated γ. Note that the approximated optimal γ performs well in terms of the fraction of engaged customers. More specifically, the estimated γ loses at most 1% customers because of the approximation.

Tolerance Threshold (ρ)	***Optimal γ^****	***Estimated γ^****	***% Gap in Engagement***
0.4	1.31	1.25	1.1%
0.5	1.47	1.37	1.1%
1.0	1.93	1.76	0.07%

Table 2 Optimal vs. estimated γ threshold for different values of customer tolerance threshold, ρ. Note that the % gap between the lower bound on engaged customers is below 1.1% showing that the estimated γ is near optimal.

We note that this optimal selection of γ is based on the model setting of Theorem 4. In Section 7, we discuss the selection of γ for more general settings.

7. Numerical Experiments

In this section, we perform numerical experiments to show the applicability of the proposed algorithm and compare its performance with other benchmark algorithms. We present results from a synthetic data study and a recommendation setting with real data of customer recommendations.

7.1. Synthetic Data

We start this section by analyzing the performance of the Constrained Bandit algorithm related to other benchmarks in the synthetic data regime. The objective is to analyze the performance of the algorithm as we increase the disengagement probability. The disengagement probability

drives the chance of customer disengagement due to irrelevant recommendations. A relatively low disengagement probability implies that customers would not disengage from the platform. On the contrary, if the disengagement probability is high, recommendation relevance plays an important role in keeping customers engaged with the platform.

We will compare the Constrained Bandit algorithm with two benchmark algorithms: (i) Linear TS by Russo and Van Roy (2014) and (ii) the Myopic Bandit (Algorithm 1). Finally, we consider a Thompson Sampling version of the Constrained Bandit algorithm (Algorithm 3) based on Russo and Van Roy (2014) which show a one-to-one equivalence between UCB based algorithms and Thompson Sampling based algorithms. Note that the CTS algorithm again uses two steps as the Constrained Bandit algorithm. In the first step, exploration is constrained to a smaller set of products based on $IP_{constrained}(\gamma)$. In the second step, posterior sampling is used to select products to recommend. We consider a TS implimentation for ease of implimentation and superior empirical performance of TS algorithms over UCB algorithms in various settings (see Chapelle and Li (2011), Russo and Van Roy (2014) and references therein).

Algorithm 3 CTS(λ,γ)

Step 1: Constrained Exploration:

Solve $IP_{constrained}(\gamma)$ to get the constrained set of products to explore over, $S_{constrained}$. Let $\hat{u}_1 = \bar{u}$.

Step 2: Bandit Learning:

for $t \in [T]$ **do**

Sample $u(t)$ from distribution $\mathcal{N}(\hat{u}_t, \sigma I^d)$.

Recommend $a_t = \arg\max_{\{i \in S_{constrained}\}} u(t)^T V_i$ if the customer is still engaged.

Observe customer utility, $Y_t = U_0^T a_t + \epsilon_t$, and update $\hat{u}_t = (V_{a_1:a_t}^T V_{a_1:a_t} + \lambda I)^{-1} V_{a_1:a_t} Y_{1:t}$

Stop if the customer disengages from the platform.

end for

Data generation: Our synthetic data study considers a simple recommendation problem of optimally recommending a single product out of a total of 10 (K) products to incoming users. This problem can be formulated as an instance of the collaborative filtering problem. Using data from past ratings (possibly only over a subset of products), CF techniques infer a low rank model of customer preferences over the set of feasible products. CF algorithms take as an input, the rank of the model to optimize. The rank of the latent user and product features in our synthetic

data study is assumed to be 2. The choice of the rank is driven by empirical experiments of Chen and Chi (2018) who have shown that low rank models perform better in comparison to models with higher rank. We note that the directional results continue to hold for models with higher rank. We assume that product features are generated from a Multivariate Normal distribution with mean $\mu^{product} \in \mathbb{R}^2$ and $\Sigma^{product} \in \mathbb{R}^{2x2}$. Similarly, the latent user features are generated from a multivariate normal with with mean $\mu^{user} \in \mathbb{R}^2$ and $\Sigma^{user} \in \mathbb{R}^{2x2}$. These can be thought of as the outcome of the low rank matrix completion problem in CF. As in the "cold start" problem, we assume that product features are known but user features of the next incoming customer are unknown. Customer response to the recommendation is assumed induced with the error following a mean 0 normal distribution and variance σ ($\mathcal{N}(0, \sigma)$). Finally, maximum time of engagement is $T = 1000$. We let tolerance threshold, ρ, to follow the truncated normal distribution with mean 0 and standard deviation 1.

Each of the three benchmarks are provided with distributional parameters on the unknown users (Normal with mean, μ^{user} and covariance Σ^{user}) and the overall time. The constrained bandit algorithm is also provided with the distributional assumption on the tolerance threshold, ρ. None of the algorithms are provided with reward variance, σ, which needs to be updated over time. Finally the disengagement probability, p, is ranged from 1% to 100% (1%, 10%, 50% and 100%) where 1% refers to the low disengagement probability regime and 100% refers to very high disengagement probability.

Before we discuss the regret and engagement results of all the benchmarks, we first discuss the first step of the constrained bandit algorithm and the insights from the first step.

Constrained Thompson Sampling (CTS): Recall that CTS (Algorithm 3) works in two steps. In Step 1, we constrain the exploration set to reduce the exploration set size. This optimization takes as an input, γ, the threshold tuning parameter. In Section 6.1, a near optimal choice of γ in a more stylized setting. In this section, we discuss optimizing γ using cross validation. More specifically, we run $IP_{constrained}(\gamma)$ for 10 different values of γ equally spaced between 0 and the maximum Euclidean distance between the product vectors. Each γ selection results in a constrained exploration set. One could simulate different customer arrival trajectories and choose the exploration set that optimizes the benchmark metric over the simulated arrivals. In order to show why Step 1 is crucial in our model, in Figure 4, we plot the size of the constrained set as a function of the chosen γ threshold. Inherently $IP_{constrained}(\gamma)$ solves a non trivial subset selection problem which is hard due to its combinatorial nature. As a result, the constrained set size does not increase linearly with γ. An increasing γ results in an increasing exploration set because the constrained set can contain products that are farther away from each other but this increase is exponential and depends on the structure of the product features.

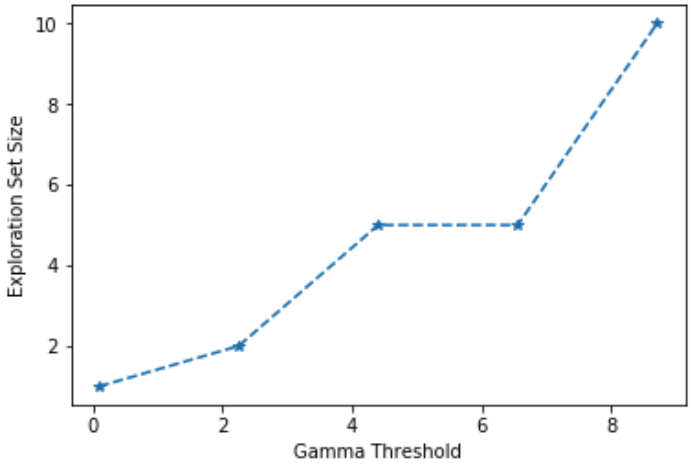

Figure 4 Cardinality of the constrained set by solving IP(γ) for various values of γ between 0 and the maximum Euclidean distance. As γ increases, the distance constraint becomes less stringent and exploration is allowed over more products. Nevertheless, the increase in exploration set size follows a non linear trend.

***Engagement Time**:* In this section we discuss the results of the different benchmark algorithms in comparison with the CTS algorithm. We start by comparing the algorithms on the total time of engagement. In this section we focus our attention on another important performance metric that we call *Engagement Time*. Engagement time or time of engagement measures the total time that a customer remained engaged to the platform when recommendations were made from different algorithms. Indeed, the higher the engagement time, the more are the chances that customer latent features can be learned with high accuracy which can indeed lead to better recommendations. For managers, a higher time of engagement is an important metric to track. A higher engagement time is directly related with higher revenue potential because of improved chances of better recommendations. While the time of engagement metric is related to cumulative regret, it captures regret as a 0-1 loss (higher engagement regardless of the reward earned is better than lower engagement). Indeed, in many recommendation settings, the objective is to provide recommendations that are relevant regardless of the revenue made from these recommendations (proxy of engagement).

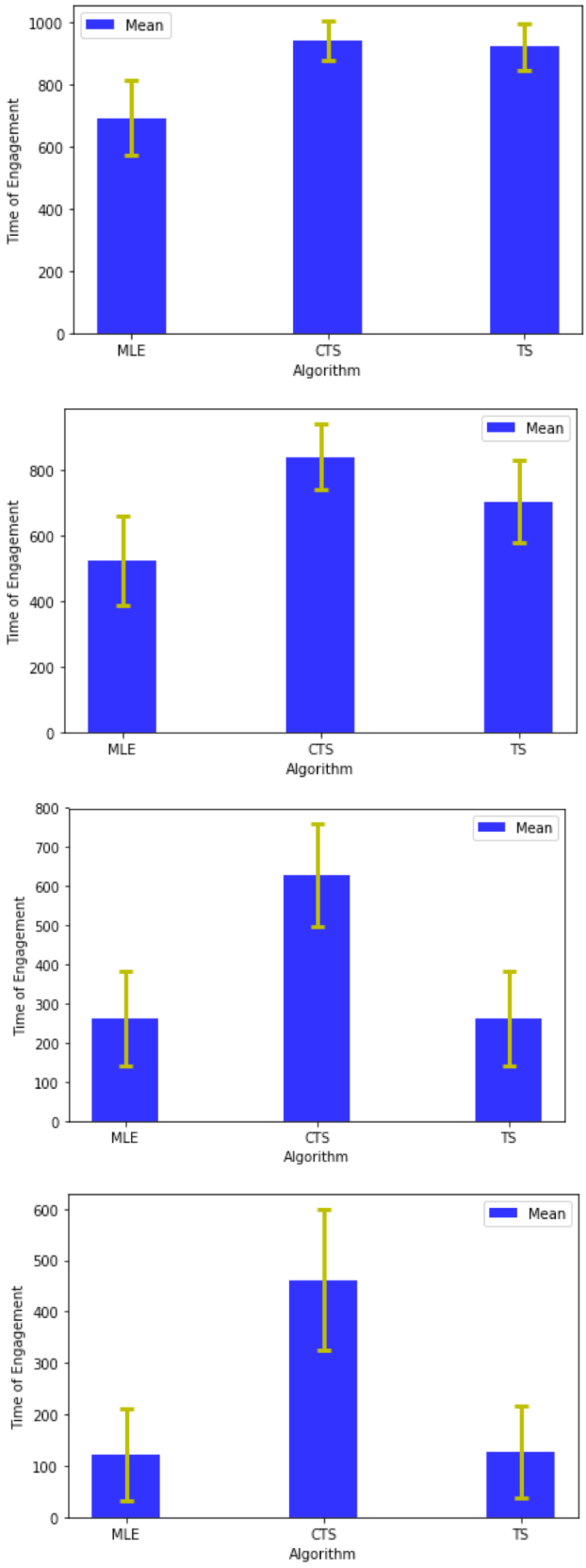

Figure 5 Mean time of engagement and 95% CI over 100 iterations of all algorithms as disengagement probability, p, changes from 1% to 10% to 50% to 100% (top to bottom). As the disengagement probability increases, the overall time of engagement decreases. Nevertheless, CTS considerably outperforms MLE and TS in all settings.

In Figure 5, we plot the mean and the 95% CI of time of engagement for different algorithms over 1000 randomly generated users as we change the disengagement probability from 1% to 100% (top to bottom). Recall that 1% leaving probability implies lower chances of disengagement, while 100% implies very high chances of disengagement. When $p = 1\%$, both TS and CTS perform very well. This is expected since TS and other classical bandit algorithms perform provably well in this setting. Exploring over differentiated products only expedites learning of the user preference and since there is a low chance of disengagement this exploration has relatively low risk. The Myopic policy performs relatively poorly because of incomplete learning as noted in Theorem 3. Even when there is little chance of disengagement, the myopic policy converges to a suboptimal product outside of the customer's relevance set and continues to recommend the product. This leads to eventual disengagement of the customer.

As we increase p from 1% to 10% (2^{nd} plot from the top), all algorithms perform worst in terms of the engagement time metric. Nevertheless, CTS starts to outperform the other two benchmark algorithms. This difference becomes considerable as we take the disengagement likelihood from 10% to 50%. The mean engagement time of CTS is more than 2.2 times that of TS showing considerable improvement in customer engagement. The improved performance can be attributed to optimal constraining of the exploration set to products that have a very high likelihood of being relevant for the next incoming customer. Since exploration is restricted to this set, chances of disengagement are relatively low which causes an improvement in the engagement time. This improvement increases further as p is increased to 100%. This is the case when the customer disengages as soon as an irrelevant product is recommended. CTS now outperforms other benchmarks by an even larger margin and the mean engagement time is as much as 4.4 times of the other benchmarks showing remarkable improvements that can be achieved on customer engagement by constrained exploration.

For managers using various personalized recommendation and communication channels such as email, web and others, this improvement in engagement is very important. For example, a recent report by Smith (2018) notes that an average worker receives as many as 121 emails on average per day. Furthermore, the average click rate for retail recommendation emails is as low as 2.5%. With more and more retailers likely to use email campaigns to promote products, high chances of disengagement becomes more and more likely. Making customers aware of the right product and understanding that there is a high penalty to be paid for recommending irrelevant recommendations becomes key.

7.2. Case Study: Movie Recommendations

We have so far established that the Constrained Bandit outperforms other benchmark algorithms in terms of engagement time in a synthetic data regime. In order to further test the algorithm in

a real world setting, we analyze the performance CTS with respect to the other benchmarks in a movie recommendation example with real data. Indeed, our modelling framework is motivated from a recommender systems setting where new customers disengage due to poor recommendations. In what follows, we discuss simple heuristics on estimating different problem parameters and the application of the benchmark algorithms and the proposed Constrained Bandit algorithm on the real world data set. We use movie ratings data collected by GroupLens Research. This data set is widely used in the academic community as a benchmark data set for various recommendation and CF algorithms (Harper and Konstan (2016)).

7.2.1. Data and Disengagement In this section, we start by briefly describing the movie lens data used for the study. We further provide evidence of customer disengagement from the data set.

Data Description: As discussed earlier, for this study, we use publicly available movie ratings data from the MovieLens web site that has been collected by GroupLens Research. MovieLens is a movie recommendation website that provides personalized movie recommendations to users based on collaborative filtering. The data set contains over 20 million ratings applied to 27,000 movies by 138,000 users. We use a random sample of 100,00 ratings generated from 671 users over 9066 movies. Users rate movies over a scale of 1 to 5. Furthermore, each of the associated ratings also contain a time stamp of the time when the rating was generated. The average movie rating over the data set is 3.65.

Classifying disengagement and proof of disengagement: The first step in our analysis is to check for empirical evidence of disengagement in the ratings data. As in Section 2, we will show that disengagement is related to the relevance of recommendations made by the platform. In order to show this, we analyze the total number of ratings that users provide. In Figure 6, we plot the overall rating distribution of users over movies. Note that on average, users provide as many as 149 ratings. Furthermore, more than 50% of the users provided at least 71 ratings.

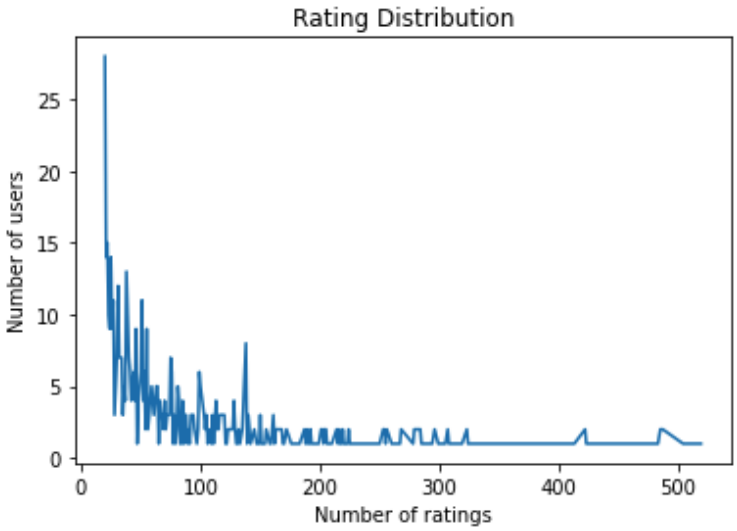

Figure 6 Empirical Ratings Distribution. More than 50 % of the users provide at least 72 ratings.

Clearly there is high variability and skewness in the number of ratings that users provide. Nevertheless, there can be many reasons for the differences in the number of ratings that users provide. We focus on two such causes: *satiation* and *irrelevance*.

Satiation happens when the user has interacted so many times with the platform that she decides to disengage from the platform. Nevertheless, disengagement due to satiation would happen to users who have provided many ratings on the platform (right tail of Figure 6) since these customers have looked at many recommendations. Since the left tail customers only provide very few ratings in comparison to the set of all movies (9000), satiation is less likely to be the cause of customer disengagement for the left tail customers.

Next we consider irrelevance of recommendations as a potential cause of disengagement. For example, a good movie recommendation would lead to the user coming back to the platform, rating the movie and engaging with the platform again for future recommendations. Vice versa, poor recommendations could lead to poor ratings. But subsequent poor recommendations could lead to customer disengagement which could be the reason for the lower number of ratings from a set of users.

In order to analyze this phenomenon further, we let all users who provide less that 27 ratings (bottom 15% users in terms of their ratings lying in the left tail) be *disengaged users*. We hypothesise that these users provided a low number of ratings because of irrelevant movie recommendations from the platform. In order to show evidence for the claim that users disengaged because of relevance of movies recommended (provided lesser ratings than other *engaged* users), we look at the average ratings that *disengaged* users provided in comparison to other *engaged* users. The average

rating of disengaged users is 3.56 (standard error of 0.10) in comparison to 3.67 (standard error of 0.04). Furthermore, one way ANOVA test (Welch (1951)) yields significant difference amongst the mean ratings of the two groups (F statistic of 29.23 and p value of 10^{-8}). This shows that users with lower engagement disliked their recommendations more than the engaged user.

Estimating latent user and movie features: In order to generate latent user and product features, we use low rank matrix factorization method (Ekstrand et al. (2011)). While we present results from an estimated low rank model of dimension 5, we note that the insights remain the same for higher dimension models as well. In order to generate the distribution of incoming users we let the $U \sim \mathcal{N}(\bar{u}, \bar{\Sigma})$ where $\bar{u}$ is the mean of latent user features and $\bar{\Sigma}$ is the covariance matrix of the latent features. Similarly V_i is the i^{th} product feature estimated from the low rank model.

Estimating tolerance threshold, ρ*:* Having characterized the set of disengaged users, we next describe the estimation procedure of ρ, the customer tolerance threshold. Recall that ρ defines the set of recommendations that user considers poor. Every time a poor recommendation is made, the user leaves the platform with probability p and disengages from the platform. Clearly the disengagement decision is driven by both, the user disengagement probability, p, and the user tolerance threshold, ρ. Consider any user, u_0, from the ratings data. For a given p and ρ, let t^{leave} denote the last rating of the user after which she disengaged from the platform and $a_1, ...a_{t_{leave}}$ be the recommendations made to the user until time t_{leave}. Then, the likelihood function function of the observation sequence is:

$$\mathcal{L}(p, \rho) = p(1-p)^{\left(t^{leave} - |a_i \in \mathcal{S}(u_0, \rho),, i=1,...t_{leave-1}|\right)},$$

where recall that $\mathcal{S}(u_0, \rho)$ defines the set of products that the user considers to be of good quality (above the user specific tolerance threshold). Note that since u_0 and V_i are known apriori (estimated from the low rank model), $\mathcal{S}(u_0, \rho)$ is also known a-priori for any given value of ρ. Hence, for any given value of p, user specific tolerance threshold, ρ, can be estimated to be the MLE estimator of $\mathcal{L}(p, \rho)$.

In Figure 7, we plot the empirical distribution of ρ, for all disengaged users. Note that ρ estimation is robust to different values of leaving probability, p. That is, for any value of $p \in (0, .75]$, we observe that the ρ distribution does not change. Most of the disengaged users have a maximum relevance threshold of 1. That is, they will not consider disengagement if the recommendations provided are within 1 unit rating from their optimal movie recommendation. Furthermore, very few disengaged users have high ρ, which implies that they have high relevance expectations from the movie recommendations made by the platform.

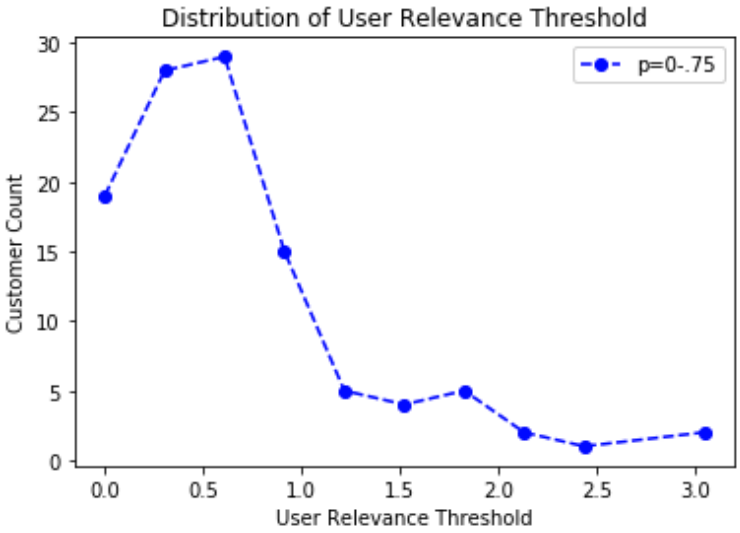

Figure 7 Empirical distribution of ρ, the tolerance threshold of a user. Note that the ρ distribution is calculated by calculating the MLE estimate of ρ for a fixed leaving probability, p.

As discussed before we compare different algorithms on the engagement time metric. In our simulation experiment, we simulate 1,000 randomly generated customer arrivals and fix the maximum time of stay (T) to be 100. 200 movies are randomly selected and are considered for recommendation for the each customer arrival. We use MLE and TS algorithms as benchmark algorithms and compare their performance against the proposed Constrained Thompson Sampling (CTS) algorithm.

7.2.2. Time of engagement Time of engagement captures the total time for which customers remained engaged with the recommendation platform. In Figure 8, we plot the mean and 95% CI of mean time of engagement with different benchmark algorithms. We change the probability of disengagement from 1% (highly unlikely to disengage) to 100% (highly likely to disengage) and see similar trends as in the synthetic data study. When users are unlikely to disengage (p=1%), all algorithms perform very well. As the disengagement probability increases the engagement time across algorithms decreases drastically. Nevertheless, CTS starts to outperform other algorithms. The mean engagement time of CTS is 1.26x, 1.66x, 1.8x of the best benchmark algorithm as p goes from 10% to 50% to 100%. Hence, CTS decisively outperforms other benchmark algorithms in the engagement time metric.

We perform similar experiments with lesser number of products and note that CTS continues to outperform the benchmark algorithms when customers are likely to disengage from the platform.

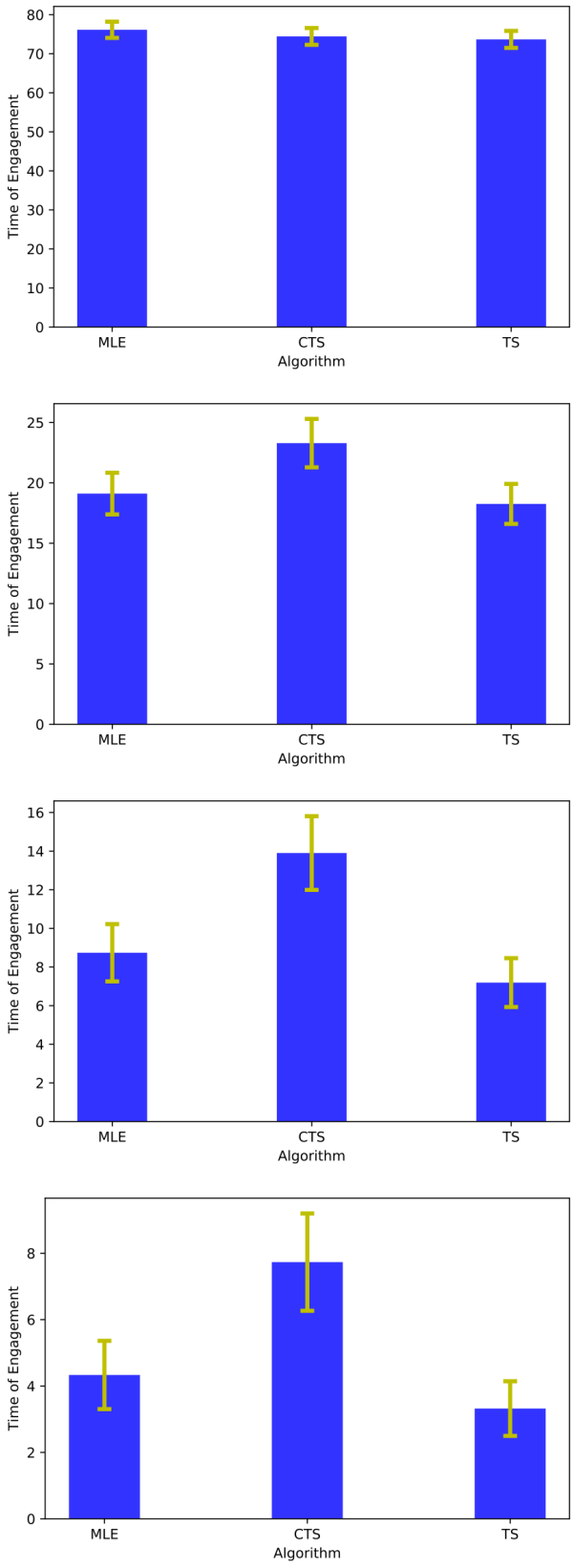

Figure 8 Mean time of engagement and 95% CI over 1000 iterations over MovieLens data of all algorithms as disengagement probability, p, changes from 1% to 10% to 50% to 100% (top to bottom). As the disengagement probability increases, the overall time of engagement decreases. Nevertheless, CTS considerably outperforms MLE and TS in all settings.

8. Conclusion

In this paper we consider the problem of optimally learning user preferences by sequential recommendations when customers are likely to disengage from the platform due to irrelevant recommendations. First, we show empirical evidence of user disengagement based on relevance of platform recommendations using email ad data from a major airline partner. Next, we propose a disengagement model in which a customer's disengagement decision is dependent on the relevance of the recommendations made. We prove that this learning setting is considerably challenging by proving that existing methods are bound to fail when customers are likely to disengage. We propose a learning algorithm that constraints exploration and prove that it can keep a large fraction of customers engaged with the system. We perform numerical experiments on both synthetic and real data sets to further demonstrate the superior performance of the proposed algorithm.

References

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *TOIS*, 22(1):5–53, 2004.

Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260. ACM, 2002.

Guy Bresler, George H Chen, and Devavrat Shah. A latent source model for online collaborative filtering. In *NIPS*, pages 3347–3355, 2014.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *SIGIR*, pages 539–548. ACM, 2016.

Aditya Gopalan, Odalric-Ambrym Maillard, and Mohammadi Zaki. Low-rank bandits with latent mixtures. *arXiv preprint arXiv:1609.01508*, 2016.

Carol F Surprenant and Michael R Solomon. Predictability and personalization in the service encounter. *the Journal of Marketing*, pages 86–96, 1987.

BPS Murthi and Sumit Sarkar. The role of the management sciences in research on personalization. *Management Science*, 49(10):1344–1362, 2003.

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing & Service Operations Management*, 18(1):15–33, 2015.

Emre M Demirezen and Subodha Kumar. Optimization of recommender systems based on inventory. *Production and Operations Management*, 25(4):593–608, 2016.

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

Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39, 2017.

N Bora Keskin and Assaf Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167, 2014.

Arnoud V den Boer and Bert Zwart. Simultaneously learning and optimizing using controlled variance pricing. *Management science*, 60(3):770–783, 2013.

Adel Javanmard and Hamid Nazerzadeh. Dynamic pricing in high-dimensions. *arXiv preprint arXiv:1609.07574*, 2016.

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. A near-optimal exploration-exploitation approach for assortment selection. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 599–600. ACM, 2016.

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. Thompson sampling for the mnl-bandit. *arXiv preprint arXiv:1706.00977*, 2017.

Karin A Venetis and Pervez N Ghauri. Service quality and customer retention: building long-term relationships. *European Journal of marketing*, 38(11/12):1577–1598, 2004.

Jana Lay-Hwa Bowden. The process of customer engagement: A conceptual framework. *Journal of Marketing Theory and Practice*, 17(1):63–74, 2009.

Rui Sousa and Chris Voss. The impacts of e-service quality on customer behaviour in multi-channel e-services. *Total Quality Management & Business Excellence*, 23(7-8):789–806, 2012.

Mark M Davis and Thomas E Vollmann. A framework for relating waiting time and customer satisfaction in a service operation. *Journal of Services Marketing*, 4(1):61–69, 1990.

Yina Lu, Andrés Musalem, Marcelo Olivares, and Ariel Schilkrut. Measuring the effect of queues on customer purchases. *Management Science*, 59(8):1743–1763, 2013.

Yash Kanoria, Ilan Lobel, and Jiaqi Lu. Managing customer churn via service mode control. 2018.

Marc Nerlove and Kenneth J Arrow. Optimal advertising policy under dynamic conditions. *Economica*, pages 129–142, 1962.

Sam Aflaki and Ioana Popescu. Managing retention in service relationships. *Management Science*, 60(2): 415–433, 2013.

Gavan J Fitzsimons and Donald R Lehmann. Reactance to recommendations: When unsolicited advice yields contrary responses. *Marketing Science*, 23(1):82–94, 2004.

Ramesh Johari and Sven Schmit. Learning with abandonment. *arXiv preprint arXiv:1802.08718*, 2018.

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

Kris Jhonson Ferreira and Shreyas Sekar. Learning to rank an assortment of products. *MS&OM Conference*, 2018.

Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. *arXiv preprint arXiv:1610.04491*, 2016.

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, pages 2312–2320, 2011.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

Yudong Chen and Yuejie Chi. Harnessing structures in big data via guaranteed low-rank matrix estimation. *arXiv preprint arXiv:1802.08397*, 2018.

Craig Smith. 90 interesting email statistics and facts, 2018. URL `https://expandedramblings.com/index.php/email-statistics/`.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.

Bernard Lewis Welch. On the comparison of several mean values: an alternative approach. *Biometrika*, 38 (3/4):330–336, 1951.

Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2):81–173, 2011.

Appendix

A. Lemmas

B. Proofs

B.1. Proofs of Section 4

Proof of Lemma 1 The proof follows directly from Corollary 2 of Lattimore and Szepesvari (2016) (See Appendix C) where we replace $\mathcal{A}$ with the set of all products, θ by the unknown feature vector u_0, and x with the features of the products in the product set. □

Proof of Theorem 1: In order to show the linearity of regret for any policy, consider the following example. Let $U_0 \in \mathbb{R}^2$ be such that,

$$U_0 = \begin{cases} u_1([1,0]) & \text{w.p. } 1/2, \\ u_2([0,1]) & \text{w.p. } 1/2. \end{cases}$$

Furthermore, let $V_1 = [1,0]$ and $V_2 = [0,1]$ be the set of feasible products that can be recommended. Finally, let $\rho \in (0,1)$, and p be 1. Clearly V_1 is the optimal product for user vector, [1,0], and V_2 for user vector, [0,1]. Since the tolerance threshold is positive and strictly less than 1, we have that if the decision maker recommends, V_1 to u_2 or V_2 to u_1, the customer disengages from the platform right away. By (1), we have that

$$\begin{aligned} R(T,\rho) = \mathbb{E}_{U_0 \sim \mathcal{P}} \left[\sum_{t=1}^{T} r_t(U_0) \right] &= \frac{1}{2} \left(\sum_{t=1}^{T} r_t(u_1) \right) + \frac{1}{2} \left(\sum_{t=1}^{T} r_t(u_2) \right) \\ &= \frac{1}{2} \left(0.\mathbb{1}\{a_1 = 1\} + T.\mathbb{1}\{a_1 = 2\} \right) + \frac{1}{2} \left(0.\mathbb{1}\{a_1 = 2\} + T.\mathbb{1}\{a_1 = 1\} \right) \\ &= \frac{1}{2} \left(T.\mathbb{1}\{a_1 = 2\} \right) + \frac{1}{2} \left(T.\mathbb{1}\{a_1 = 1\} \right) \\ &= \frac{T}{2} \end{aligned}$$

where the first equality follows since choosing the right recommendation in the first round leads to the identification of the optimal product. Nevertheless, recommending a sub-quality product leads to instant customer disengagement and leads to linear regret. Hence, any non-anticipating policy would result in a linear regret. □

Proof of Theorem 2: We will prove the above statement by showing that whenever $|S(u_0)| < d$, any consistent policy, π, recommends products outside of the customer's feasibility set infinitely often. Customer disengagement thus follows directly since there is a positive probability, p, of customer leaving the platform whenever a product outside the customer's feasibility set is offered.

In order to show that a consistent policy shows products outside the feasibility set infinitely often, we will use Corollary 1. More specifically, we will construct a counter example such that no consistent policy will satisfy the condition of the Corollary unless it exits the set of feasible

products infinitely many times.

Let us assume by contradiction that there exists a policy π that is consistent and offers products inside the feasible set infinitely often. This implies that there exists $\bar{t}$ such that $\forall t > \bar{t}$, $a_t \in \mathcal{S}$. Now consider a setting where there are d products in total ($n = d$) and the feature vector of the i^{th} product is the i^{th} basis vector. That is,

$$V_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, V_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, V_3, ..., V_d = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Further let u_o, the unknown consumer feature vector, and γ, the tolerance threshold parameter be such that WLOG, $\mathcal{S}(u_0) = \{2, 3...d\}$ (follows by 2). That is, only the first product is outside of the feasible set. Also let,

$$R_n^\pi = \begin{bmatrix} T_1^\pi(n) & 0 & \dots \\ \vdots & \ddots & \\ 0 & & T_d^\pi(n) \end{bmatrix}$$

where

$$T_j(n) = \sum_{t=1}^{t=n} \mathbb{1}\{a_t^\pi = j\}$$

$T_j(n)$ is the total number of times the j^{th} product is offered until time n under policy π.

Next consider the following:

$$\begin{aligned} \limsup_{n\to\infty} \log(n)||e_1||^2_{X_n^{-1}} &= lim\ sup_{n\to\infty} log(n) e_1^T X_n^{-1} e_1 \\ &= \limsup_{n\to\infty} \log(n) e_1^T \left[\sum_{t=1}^{t=n} a_t a_t^T\right]^{-1} e_1 \\ &= \limsup_{n\to\infty} \log(n) e_1^T \left[R_n\right]^{-1} e_1 \\ &\geq \limsup_{n\to\infty} \log(n) \left(\frac{1}{T_1(n)}\right) \\ &\geq \limsup_{n\to\infty} \log(n) \left(\frac{1}{T_1(\bar{t})}\right) \\ &= \infty \end{aligned}$$

Where the second to last inequality follows by the fact that $\forall t > \bar{t}$, π recommends products inside the feasible set, $\mathcal{S}$, which does not contain product 1. Furthermore,

$$T_1(\bar{t}) = T_1(\bar{t}+1) = T_1(\bar{t}+2) = ... = \lim_{n\to\infty} T_1(\bar{t}+n).$$

For any finite Δ_{V_1}, we have that

$$\limsup_{n\to\infty} \log(n)||e_1||^2_{G_n^{-1}} \geq \frac{\Delta_x}{2}.$$

which implies that $\exists V_i$ in the action space such that the condition of Corollary 1 is not satisfied. Hence we have show that there exists no consistent policy that recommends products inside of the feasible set of products infinitely often.

Furthermore, this implies that as long as the probability measure of $\{u : \mathcal{S}(u) < d\}$ is positive under the distribution $\mathcal{P}$, not only will the the cumulative regret $R(T)$ be linear but also, customers will leave the platform with probability 1 as we scale the time horizon and hence will eventually disengage from the platform. $\square$

Proof of Theorem 3: We prove the above result in two parts. In the first part we show a lower bound on choosing a product which is optimal for $\bar{u}$ but is not optimal for the incoming feature. In the next part we show a lower bound on the probability that the myopic policy continues to select the same sub optimal product. Since the myopic policy never recommends the optimal arm for the incoming customer, this results in an overall lower bound on the regret rate of the myopic policy.
Step 1 (Lower bound on selecting an initial suboptimal product): In this step, we show a lower bound on the probability that a random incoming user feature's optimal product is different from the optimal product of the prior mean of user feature vector, $\bar{u}$. That is, the first step of the myopic policy chooses a product that is different from the optimal product of the current customer. More specifically, we are interested in characterizing the probability of the following event:

$$\{u : \arg\max_i V_i^T u \neq \arg\max_i V_i^T \bar{u}\}$$

Let $\tilde{i}$ denote the optimal product corresponding to feature vector, $\bar{u}$, and consider any $i \neq \tilde{i}$, then

$$\mathbb{P}(V_i^T U_0 - V_{\tilde{i}}^T \bar{u} \geq 0) = \mathbb{P}(V_i^T U_0 - V_i^T \bar{u} \geq V_{\tilde{i}}^T \bar{u} - V_i^T \bar{u}) = \mathbb{P}(V_i^T (U_0 - \bar{u}) \geq \bar{u} \left(V_{\tilde{i}}^T - V_i^T\right)$$

Now note again that $V_i^T (U_0 - \bar{u}) \sim \mathcal{N}(0, V_i^T \Sigma V_i)$. Hence, using the lower bounding tail bound of normal distributions we have that for any standard normal random variable Z, $\mathbb{P}(Z - \mu > t) \geq \frac{e^{\frac{-t^2}{2}}}{2\sqrt{2\pi}t}$. Hence,

$$\mathbb{P}\left(V_i^T (U_0 - \hat{u}) \geq V_{\tilde{i}}^T \bar{u} - V_i^T \bar{u}\right) = \mathbb{P}\left(\frac{V_i^T (U_0 - \hat{u})}{V_i^T \Sigma V_i} \geq \frac{V_{\tilde{i}}^T \bar{u} - V_i^T \bar{u}}{V_i^T \Sigma V_i}\right) \geq \frac{e^{\frac{-\Delta_{max}^2}{2}}}{2\sqrt{2\pi}\Delta_{max}}$$

Step 2 (Upper bound on the probability of switching from the current product to a different product during the later periods:) In this step we want to upper bound the probability that the myopic policy switches from the current product to the other product. Note that from Step 1, we have already characterized the lower bounded the probability of starting with a sub-optimal product.

Of course, since we are dynamically updating the estimated latent customer feature vector, the probability of switching depends on the realization of ϵ_t, the idiosyncratic noise term that governs the customer response.

Recall that $\tilde{i}$ denotes the optimal apriori product. Let $E_i^t = \{V_{\tilde{i}}^T \hat{u}_t - V_i^T \hat{u}_t > 0\}$ and $\Delta_i = V_{\tilde{i}}^T U_0 - V_i^T U_0$. E_i^t denotes the event that the myopic optimal is indeed better than the i^{th} product in the product assortment at time t. Similarly, Δ_i is a random variable that denotes the difference between the utilities of a customer from the myopic optimal product $\tilde{i}$ and some other product, i. Finally, let G^t be the event that the myopic policy continues to choose $\tilde{i}$ until time t. Then,

$$\mathbb{P}(G^t) = 1 - \mathbb{P}((G^t)^c) = 1 - \mathbb{P}\left((\cap_{i=1..K, i\neq i^*} \cap_{j=1..t} E_i^j)^c\right) \tag{6}$$

$$= 1 - \mathbb{P}\left(\cup_{i=1..K, i\neq i^*} \cup_{j=1..t} (E_i^j)^c\right) \tag{7}$$

$$\geq 1 - \sum_{j=1..t} \sum_{i=1..K, i\neq i^*} \mathbb{P}((E_i^j)^c) \tag{8}$$

We will upper bound the probability of the initial myopic optimal product, i^* not being the optimal myopic product for any time t. In order to do this we will analyze the probability of the event $(E_i^t)^c$. First note that,

$$\begin{aligned} V_i^T \hat{u}_t - V_{\tilde{i}}^T \hat{u}_t &= V_i^T \hat{u}_t - V_{\tilde{i}}^T \hat{u}_t - V_{\tilde{i}}^T U_0 + V_{\tilde{i}}^T U_0 - V_i^T U_0 + V_i^T U_0 \\ &= (V_i - V_{\tilde{i}})^T (\hat{u}_t - U_0) - \Delta_i \\ &\leq ||V_i - V_{\tilde{i}}||_{\bar{X}_t^{-1}} ||\hat{u}_t - U_0||_{\bar{X}_t^{-1}} - \Delta_i \\ &\leq ||V_i - V_{\tilde{i}}||_{\bar{X}_t^{-1}} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right) - \Delta_i \end{aligned}$$

where

$$\bar{X}_t = (I\lambda + \sum_{s=1}^{s=t} a_t a_t^T)$$

and the last inequality follows by Lemma 3 (Appendix C). Recall that λ is the regularization parameter for the L2-regularized regression and a_t is the corresponding recommended product at time t. The above result holds with probability at least 1-δ as long as $||U_0||_2 \leq S$. By Lemma 2, we have that $||U_0||_2 \leq S$ with probability at least $1 - 2de^{-\left(\frac{S - \sum_{i=1}^{i=d} \mu_i}{2d\sigma}\right)^2}$. Hence the above result holds with probability at least $(1-\delta)\left(1 - 2de^{-\left(\frac{S - \sum_{i=1}^{i=d} \mu_i}{2d\sigma}\right)^2}\right)$

Next, we will upper bound $||V_i - V_{i^*}||_{\bar{X}_t^{-1}}$.

Also note that,

$$\begin{aligned} ||V_i - V_{i^*}||_{\bar{X}_t^{-1}} &= \sqrt{(V_i - V_{i^*})^T \bar{X}_t^{-1} (V_i - V_{i^*})} = \sqrt{(V_i - V_{i^*})^T \left(I\lambda + \sum_{s=1}^{s=t} a_t a_t^T\right)^{-1} (V_i - V_{i^*})} \\ &\leq \sqrt{(V_i - V_{i^*})^T \frac{1}{\lambda_{min}\left(I\lambda + \sum_{s=1}^{s=t} a_t a_t^T\right)} (V_i - V_{i^*})} \\ &\leq \sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \end{aligned}$$

Hence, with probability at least 1 -δ

$$||V_i - V_{i^*}||_{\bar{X}_t^{-1}} ||\hat{U}_t - U||_{\bar{X}_t^{-1}} \leq \sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right)$$

Note that,

$$\begin{aligned}
\mathbb{P}\left((E_i^t)^c\right) &= \mathbb{P}\left(V_{i*}^T \hat{u}_t - V_i^T \hat{u}_t \leq 0\right) = \mathbb{P}\left((V_i - V_{i^*})^T (\hat{u}_t - U_0) - \Delta_i \geq 0\right) = \mathbb{P}\left((V_i - V_{i^*})^T (\hat{u}_t - U_0) \geq \Delta_i\right) \\
&\leq \mathbb{P}\left(||V_i - V_{i^*}||_{\bar{X}_t^{-1}} ||\hat{u}_t - U_0||_{\bar{X}_t^{-1}} \geq \Delta_i\right) \\
&\leq \mathbb{P}\left(\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right) \geq \Delta_i \right) \\
&\leq \mathbb{P}\left(\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right) - (V_{i^*} - V_i)^T \hat{u} \geq \Delta_i - (V_{i^*} - V_i)^T \hat{u} \right) \\
&\leq \mathbb{P}\left(\frac{\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right) - (V_{i^*} - V_i)^T \hat{u}}{(V_{i^*} - V_i)^T \Sigma (V_{i^*} - V_i)} \geq \frac{\Delta_i - (V_{i^*} - V_i)^T \hat{u}}{(V_{i^*} - V_i)^T \Sigma (V_{i^*} - V_i)} \right) \\
&\leq \mathbb{P}\left(\frac{\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S \right) - (V_{i^*} - V_i)^T \hat{u}}{(V_{i^*} - V_i)^T \Sigma (V_{i^*} - V_i)} \geq Z \right) = \mathbb{P}\left(\bar{B}_i \geq Z\right) \\
&= 1 - \mathbb{P}\left(\bar{B}_i \leq Z\right) \\
&\leq 1 - \left(\frac{1}{\bar{B}_i} - \frac{1}{\bar{B}_i^2}\right) \frac{e^{-\bar{B}_i^2}}{\sqrt{2\pi}}
\end{aligned}$$

where $Z \sim \mathcal{N}(0,1)$ and recall by definition that,

$$\bar{B}_i = \frac{\sqrt{\lambda (V_i - V_{i^*})^T (V_i - V_{i^*})} \left(R\sqrt{dlog\left(\frac{1+TL^2}{\delta}\right)} + \lambda^{1/2} S \right) - (V_{i^*} - V_i)^T \hat{u}}{(V_{i^*} - V_i)^T \Sigma (V_{i^*} - V_i)}$$

The last inequality follows from the lower bounding tail bound of standard normal distributions. Next, let

$$\bar{B}_{min} = \min_{i=1,..,n} \bar{B}_i$$

And note by assumption, that

$$\left(\frac{1}{\bar{B}_{min}} - \frac{1}{\bar{B}_{min}^2}\right) \frac{e^{-\bar{B}_{min}^2}}{\sqrt{2\pi}} \geq 1 - \frac{1}{2KT}$$

Then, by (6), we have that

$$\mathbb{P}(G^t) \geq 1 - \sum_{j=1..t} \sum_{i=1..K, i \neq i^*} \mathbb{P}((E_i^j)^c) \tag{9}$$

$$\geq 1 - \sum_{j=1..t} \sum_{i=1..K, i \neq i^*} \left(1 - \left(\frac{1}{\bar{C}_i} - \frac{1}{\bar{B}_i^2}\right) \frac{e^{-\bar{C}_i^2}}{\sqrt{2\pi}}\right) \tag{10}$$

$$\geq 1 - \sum_{j=1..t} \sum_{i=1..K, i \neq i^*} \left(1 - 1 + \frac{1}{2KT}\right) \tag{11}$$

$$\geq 1 - \sum_{j=1..t} \sum_{i=1..K, i \neq i^*} \left(\frac{1}{2KT}\right) \tag{12}$$

$$\geq \frac{1}{2} \tag{13}$$

Finally, combining Step (1) and Step (2), we know that Step (1) bounds the probability of choosing a sub optimal arm and Step (2) lower bounds the probability of continuing to pull the same arm repeatedly. Hence, with probability at least $(1-\delta)\left(\frac{e^{\frac{-\Delta_{max}^2}{2}}}{2\sqrt{2\pi}\Delta_{max}}\right)\left(1 - 2de^{-\left(\frac{S-\sum_{i=1}^{i=d}\mu_i}{2d\sigma}\right)^2}\right)$ a myopic policy based on L2 regularized OLS regression would get stuck at a sub optimal arm. □

C. Supplementary Results

LEMMA 2. *Let $X \in \mathbb{R}^d \sim \mathcal{N}(\mu, \sigma^2 I)$ be a multivariate normal random variable with mean vector $\mu \in \mathbb{R}^d$. Let $S \in \mathbb{R}^d$ be such that $S \geq \sum_{i=1}^{i=d} \mu_i$. Then,*

$$\mathbb{P}(||X||_1 \leq S) \geq 1 - 2de^{-\left(\frac{S-\sum_{i=1}^{i=d}\mu_i}{2d\sigma}\right)^2}$$

Proof: First note that,

$$||X||_1 = \sum_{i=1}^{i=d} |X_i| = \sum_{i=1}^{i=d} \sigma\left(\frac{|X_i - \mu_i + \mu_i|}{\sigma}\right) \leq \sum_{i=1}^{i=d} \sigma\left(\frac{|X_i - \mu_i|}{\sigma} + \frac{\mu_i}{\sigma}\right)$$

Then, we have that

$$\begin{aligned}
\mathbb{P}(||X||_1 > S) &\leq \mathbb{P}\left(\sum_{i=1}^{i=d} \sigma\left(\frac{|X_i - \mu_i|}{\sigma} + \frac{\mu_i}{\sigma}\right) \geq S\right) \leq \mathbb{P}\left(\sum_{i=1}^{i=d} \frac{|X_i - \mu_i|}{\sigma} \geq \frac{S - \sum_{i=1}^{i=d}\mu_i}{\sigma}\right) \\
&\leq \mathbb{P}\left(\sum_{i=1}^{i=d} |Z| \geq \frac{S - \sum_{i=1}^{i=d}\mu_i}{\sigma}\right) \leq d\left(\mathbb{P}\left(|Z| \geq \frac{S - \sum_{i=1}^{i=d}\mu_i}{d\sigma}\right)\right) \\
&= d\left(\mathbb{P}\left(Z \geq \frac{S - \sum_{i=1}^{i=d}\mu_i}{d\sigma}\right) + \mathbb{P}\left(Z \leq -\frac{S - \sum_{i=1}^{i=d}\mu_i}{d\sigma}\right)\right) \\
&= 2d\left(\mathbb{P}\left(Z \geq \frac{S - \sum_{i=1}^{i=d}\mu_i}{d\sigma}\right)\right) \\
&\leq 2de^{-\left(\frac{S-\sum_{i=1}^{i=d}\mu_i}{2d\sigma}\right)^2}
\end{aligned}$$

where $Z \in \mathbb{R}^1 \sim \mathcal{N}(0,1)$ and the first set of inequalities follow by the pigeon-hole principle and the union bound. The last inequality follows by the tail probability of standard normal random variables. The result follows easily. □

COROLLARY 1 **(from Lattimore and Szepesvari (2016))**. *Let π be a consistent policy, $\theta \in R^d$ such that there is a unique optimal arm in $\mathcal{A}$. Then*

$$\limsup_{n\to\infty} log(n)||x||^2_{\bar{G}_t^{-1}} \leq \frac{\Delta_x}{2}$$

and also $\limsup_{n\to\infty} \frac{R_\theta^\pi(n)}{\log(n)} \geq c(\mathcal{A},\theta)$. *where $c(\mathcal{A},\theta)$ is the solution to the following optimization problem:*

$$\inf_{\alpha\in[0,\infty)^{\mathcal{A}}} \sum_{x\in\mathcal{A}^-} \alpha(x)\Delta_x \text{ subject to}$$

$$||x||^2_{H^{-1}(\alpha)} \leq \frac{\Delta_x^2}{2}, \forall x \in \mathcal{A}^{-1}$$

where $H(\alpha) = \sum_{x\in\mathcal{A}} \alpha(x) x x^T$

LEMMA 3 **(Theorem 2 in Abbasi-Yadkori et al. (2011))**. *Let $\mathcal{F}_{t=0}$ be a filtration. Let $\eta_{t=1}^{\infty}$ be a real-valued stochastic process such that η_t is $\mathcal{F}_t$ measurable and η_t is conditionally R-sub-Gaussian for some $R \geq 0$ i.e.*

$$\forall\lambda \in \mathbb{R}, E[e^{\lambda\eta}|\mathcal{F}_{t-1}] \leq exp\left(\frac{\lambda^2 R^2}{2}\right)$$

Let $X_{t=1}^{t\to\infty}$ be an R^d valued stochastic process such that X_t is $\mathcal{F}_{t1}$-measurable. Assume that V is a dXd positive definite matrix. For any $t \geq 0$, define

$$V_t = V + \sum_{s=1}^{s=T} X_s X_s^T, S_t = \sum_{s=1}^{s=t} \eta_s X_s$$

Then for any $\delta > 0$, the following holds with probability at least 1-δ for all $t \geq 0$.

$$||S_t||_{\bar{V}_t^{-1}} \leq 2R^2 log\left(\frac{det(\bar{V}_t)^{1/2} det(\lambda I)^{-1/2}}{\delta}\right) \tag{14}$$

Now let $V = I\lambda$, $\lambda > 0$, define $Y_t = X_t^T + \eta_t$ and assume that $||\theta_*||_2 \leq S$. Then, for any $\delta > 0$, with probability at least 1-δ, for all $t \geq 0$, θ^* lies in the set

$$\mathcal{C}_t : \left\{\theta \in \mathbb{R}^d : ||\hat{\theta}_t - \theta||_{\bar{V}_t} \leq \left(R\sqrt{dlog\left(\frac{1+tL^2}{\delta}\right)} + \lambda^{1/2} S\right)\right\} \tag{15}$$