Product Ranking for Revenue Maximization with Multiple Purchases

Anonymous Author(s)
Affiliation
Address
email

Abstract

Product ranking is the core problem for revenue-maximizing online retailers. To design proper product ranking algorithms, various customer behavior models are proposed to characterize the customers' behaviors when they are provided a list of products. However, existing works assume that each customer purchases at most one product or will keep viewing the product list after purchasing a product, which does not agree with the common practice in real scenarios. We assume that each customer can purchase multiple products at will in this work. To model customers' willingness to view and purchase, we set a random attention span and purchase budget, which determines the maximal amount of products that he/she views and purchase, respectively. Under this setting, we first design an optimal ranking policy when the online retailer can precisely model customers' behaviors. Based on the policy, we further develop UCB-like algorithms with $\tilde{O}(\sqrt{T})$ regret that estimates customers' behaviors and maximize revenue simultaneously in online settings. Experiments on both synthetic and real-world datasets prove the effectiveness of the proposed algorithms.

1 Introduction

Online retailing has become increasingly popular over the last decades [17, 27]. The way of product ranking is the crux for online retailers because it determines the customers' shopping behaviors [17] and thus influences the retailers' revenue [20, 47]. For instance, the probability of customers' purchasing from a firm or clicking an advertisement is strongly related to the display order [8, 3, 32]. Therefore, it is crucial for online retailers to design proper product ranking algorithms for revenue maximization.

The customers' behavior, when faced with a list of products, is the basis of the problem [17]. Generally speaking, existing works assume that the customers view the list of products sequentially and select products according to a choice model [42, 41, 31, 37, 18, 25, 24, 36]. However, most of them assume that each customer purchases at most one product and stop browsing immediately afterward, which does not agree with the common practice since a customer may expect multiple purchases on a website or an app [33, 34, 16, 27]. Recently, several works consider a multiple-purchase setting while their targets are not revenue maximization [27, 15], or they assume that customers always continue viewing the product list after purchasing any product, which is not practical in real scenarios [39].

In this paper, we propose a more realistic setting to characterize the customer behaviors under multiple-purchase settings. Following [13, 17, 39], we assume that the customers view the product list sequentially and purchase a product when its utility exceeds a certain threshold. Similar to [17],

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

we assume the existence of the attention span for each customer, which determines the maximal amount of products that the customer is willing to view. More importantly, instead of forcing the customer to quit viewing once she purchases one product, we allow each customer to conduct multiple purchases at will and explore product ranking models for revenue maximization. Specifically, we constrain each customer with a purchase budget to determine the maximal amount of purchases. Both the attention span and the purchase budget are assumed to obey the geometric distribution [13, 39].

Under this setting, we study the optimal product ranking policy to maximize the online retailers' revenue. We first characterize the optimal ranking policy that achieves the maximal revenue when the online retailer knows the customer parameters, including the purchase probabilities for each product and the distribution of the attention span and purchase budget. Afterward, we investigate the problem in online learning frameworks where customer parameters are unknown to online retailers. In both non-contextual (*i.e.*, all customers share the same parameters) and contextual settings (*i.e.*, customers have personalized parameters), we provide UCB-like algorithms [1] that could fit customers' parameters and maximize revenue simultaneously. We further theoretically prove that the algorithms achieve $\tilde{O}(\sqrt{T})$ regrets in both settings. Finally, we conduct experiments to verify the effectiveness of the algorithms.

To conclude, our contributions are highlighted as follows:

- We propose a novel setting to deal with multiple-purchase activities of customers on online scenarios.
- We characterize the optimal rankling policy and further design UCB-like algorithms with $\tilde{O}(\sqrt{T})$ regret based on them in both non-contextual and contextual online settings.
- We conduct extensive experiments to prove the effectiveness of our method on the multiple-purchase setting.

2 Related works

Customers' behavior model Traditional choice models adopt the multinomial logic (MNL) [12, 22, 42] models to characterize customers' behaviors. However, these models can not describe the behaviors when the customers are provided with a list of products. As a result, several choice models are proposed recently. In detail, Abeliuk et al. [2] add a position bias term in the traditional MNL model. Gallego et al. [29], Aouad and Segev [4] suppose that the products are framed into a set of virtual web pages. Customers select a product, if any, from these pages following a general choice model (such as MNL). Flores et al. [28] propose the Sequential Multinomial Logit (SML), which assumed a two-stage MNL model and supposed that products are divided into two stages with prior knowledge. Liu et al. [40], Feldman and Segev [26], Gao et al. [30] further extend the setting to multiple stages. These works suppose that each stage contains multiple products while recently several works assume that a product is treated as a single stage [17, 13, 14]. These models are also closely related to the cascade model [12, 35, 37, 48, 19]. Gao et al. [31] further propose the general cascade click model. Following these works, we suppose that each stage contains only one product and the customers views the products sequentially.

In addition, several works consider the setting where customers may have multiple interactions with the platforms. However, most of them focus on the click model [15, 16, 33–35]. In addition, [27] maximizes the number of customers who engage with the site. Liang et al. [39] aimed to maximize the revenue in the multiple-purchase setting while they assume that customers always continue viewing the product list after purchasing any product, which is not practical in real scenarios.

Online learning and multi-armed bandit Online retailers need to design proper algorithms to model customers' behaviors and maximize the revenue at the same time, which is closely related to the multi-armed bandit framework [46, 44]. Many works [5, 9, 11, 10, 23] have been proposed to deal with demand learning or price experimentation problems with the help of the framework in the field of revenue management. Recently, more methods [39, 17, 13, 14, 6, 43] adopt similar techniques in the product ranking setting, which inspires the algorithm proposed in this paper.

3 Preliminaries

Let $[N] \triangleq \{1, 2, \ldots, N\}$ be the set of all products owned by an online-retailer (he). The product k generates revenue r_k when a customer purchases it. Let $r_{\max} \triangleq \max_{k\in[N]} r_k$ be the maximal revenue for all products and $\boldsymbol{r} = \{r_k\}_{k=1}^N$. We use the notation $\boldsymbol{\sigma}_t \dashv [N]$ to denote the ranking policy $\boldsymbol{\sigma}_t$ on the products for the t-th customer (she). More precisely, $\forall k \in [N]$, $\sigma_{t,k} \in [N]$ represents the product displayed in the k-th position for her. Reversely, $\sigma_{t,k}^{-1}$ denotes the position of the k-th product in the ranking list. Furthermore, for any subset $Q \subseteq [N]$, we use $\sigma_{t,Q}^{-1}$ to denote the positions of the products in Q and $\sigma_{t,Q}^{-1} = \{\sigma_{t,k}^{-1} : k \in Q\}$. In addition, we use $\max(\cdot)$ to denote the maximal element in a set.

The customers' behavior model When the t-th customer (she) arrives, she views the list of products sequentially. The probability of product k being purchased by her is denoted as $\lambda_{t,k}$ and let $\boldsymbol{\lambda}_t = \{\lambda_{t,k}\}_{k=1}^N$. To characterize the customers' behaviors under multiple-purchase settings, we assume that the customer is endowed with the attention span V_t and purchase budget B_t, which determine the maximal amount of the products that she will view and purchase, respectively. In reality, the customer always have random attention span and purchase budget [17]. Therefore, following [13, 39], we assume $V_t \sim \mathrm{Geo}(1 - q_t)$ and $B_t \sim \mathrm{Geo}(1 - s_t)$ are both geometric distributions with parameter $1 - q_t$ and $1 - s_t$, respectively. As a result, $P(V_t = v) = q_t^{v-1}(1 - q_t)$ for any $v \geq 1$ and $P(B_t = b) = s_t^{b-1}(1 - s_t)$ for any $b \geq 1$.

With attention span V_t and purchase budget B_t, the customer keeps viewing the products until she views V_t or purchases B_t products. The top V_t products shown by the ranking policy σ_t is $A_t = \{\sigma_i : i \in [V_t]\}$. As a result, the probability that the customer purchases a subset of products $Q \subseteq A_t$ is given by

$$P^{\mathrm{buy}}(Q; V_t, B_t, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = \begin{cases} \left(\prod_{k\in Q} \lambda_{t,k}\right) \left(\prod_{k\in A_t\setminus Q} (1 - \lambda_{t,k})\right), & \text{if } |Q| < B_t. \\ \left(\prod_{k\in Q} \lambda_{t,k}\right) \left(\prod_{k\in A_t\setminus Q, \sigma_{t,k}^{-1} \leq \max(\sigma_{t,Q}^{-1})} (1 - \lambda_{t,k})\right), & \text{if } |Q| = B_t. \\ 0, & \text{if } |Q| > B_t. \end{cases} \tag{1}$$

Here the condition $\sigma_{t,k}^{-1} \leq \max(\sigma_{t,Q}^{-1})$ requires that the ranking policy σ_t displays the product k earlier than some products in the set Q. Specifically, the probability $P^{\mathrm{buy}}(Q; V_t, B_t, \boldsymbol{\lambda}, \boldsymbol{\sigma})$ is calculated based on the cardinality of Q. When $|Q| < B_t$, the customer does not meet her purchase budget and stops browsing the list after viewing V_t products. When $|Q| = B_t$, the customer stops browsing as long as she purchases the last product in the list Q. When $|Q| > B_t$, the probability becomes 0 since the amount of purchases $|Q|$ exceeds her purchase budget.

Remark. The setting of the purchase budget provides a rational way to characterize the customers' behaviors with multiple purchases and extends previous works that suppose customers would purchase at most one product [13, 17]. Furthermore, when $s_t \to 0$, the customers would tend to purchase at most one product and our setting degenerates to the setting proposed in [13]. Thus the previous setting can be considered as a special case of our setting.

Revenue optimization The online-retailer chooses a ranking policy $\boldsymbol{\sigma}_t \dashv [N]$ on the products for the t-th customer. For a customer with fixed purchase probabilities for each product $\boldsymbol{\lambda}_t$, attention span V_t, and purchase budget B_t, the expected revenue could be written as

$$R(V_t, B_t; \boldsymbol{\lambda}, \boldsymbol{\sigma}) = \sum_{Q\subseteq[V_t], |Q|\leq B_t} P^{\mathrm{buy}}(Q; V_t, B_t, \boldsymbol{\lambda}, \boldsymbol{\sigma}) \left(\sum_{i\in Q} r_{\sigma_i}\right). \tag{2}$$

In reality, the attention span V_t and purchase budget B_t are random and can not be estimated accurately by the online retailer. As a result, the retailer calculates the expected revenue for the t-th customer by sampling V_t from $\mathrm{Geo}(1 - q_t)$ and sampling B_t from $\mathrm{Geo}(1 - s_t)$.

$$\mathrm{Revenue}\left(\boldsymbol{\sigma}; \boldsymbol{\lambda}_t, q_t, s_t\right) = \mathbb{E}_{V_t\sim\mathrm{Geo}(1-q_t), B_t\sim\mathrm{Geo}(1-s_t)} \left[R(V_t, B_t; \boldsymbol{\lambda}_t, \boldsymbol{\sigma})\right]. \tag{3}$$

To maximize the revenue, the optimal ranking policy is given by

$$\boldsymbol{\sigma}_t^*(\boldsymbol{\lambda}_t, q_t, s_t) = \underset{\boldsymbol{\sigma} \dashv [N]}{\arg\max} \operatorname{Revenue}(\boldsymbol{\sigma}; \boldsymbol{\lambda}_t, q_t, s_t). \tag{4}$$

For simplicity, we use $\boldsymbol{\sigma}_t^*$ to denote $\boldsymbol{\sigma}_t^*(\boldsymbol{\lambda}_t, q_t, s_t)$.

Basic assumption We also need the following assumption to guarantee the tractability of the proposed problem. In reality, customers always have finite attention span, which indicates that q_t is always smaller than 1.

Assumption 3.1. There exists a known constant $\epsilon_q > 0$ such that for all $t > 0$, $q_t \leq 1 - \epsilon_q$.

4 Optimal ranking policy given customers' characteristics

In this section, we introduce the optimal ranking policy when the online retailer exactly knows the customers' parameters, including the purchase probability for each product $\boldsymbol{\lambda}$, random attention span (parametrized with q), and random purchase budget (parametrized with s).

We drop the subscript t in this section for simplicity. The optimal ranking strategy stems from the following theorem.

Theorem 4.1. *Under Assumption 3.1, for any $1 \leq i < N$, the following holds for the optimal ranking strategy $\boldsymbol{\sigma}^*(\boldsymbol{\lambda}, q, s)$.*

$$\frac{\lambda_{\sigma_i^*} r_{\sigma_i^*}}{1 - q + q(1-s)\lambda_{\sigma_i^*}} \geq \frac{\lambda_{\sigma_{i+1}^*} r_{\sigma_{i+1}^*}}{1 - q + q(1-s)\lambda_{\sigma_{i+1}^*}}. \tag{5}$$

The expected revenue of a single product σ_i^* is $\lambda_{\sigma_i^*} r_{\sigma_i^*}$, which grows as the purchase probability and the revenue generated by the product increases. Yet purchasing a product reduces the purchase probability of the following products, and thus the expected revenue of other products decreases as σ_i^* increases. Furthermore, when $s \to 0$, the term in Equation (5) become the same as that in [13, Theorem 1]. As a result, Theorem 4.1 consistently extends the results on the single purchase setting [13] to the multiple purchase setting.

As a result, according to Theorem 4.1, the online retailer can provide the optimal ranking policy $\boldsymbol{\sigma}^*$ by sorting the products in a descending order with value $(\lambda_k r_k) \,/\, (1 - q + q(1-s)\lambda_k)$ for the k-th product. The time complexity is $O(N \log N)$.

5 Online learning of the ranking policy

In this section, we consider a more realistic scenario where the online retailer has no prior knowledge about customers' characteristics. At each timestamp, a customer arrives and the online retailer can only provide proper product ranking policies based on the feedbacks of previous customers. As a result, it is essential for the retailer to design online learning algorithms to model customers' behaviors and maximize the revenue in the meantime. We consider two settings, *i.e.*, the non-contextual setting where all customers share the same parameters (Section 5.1) and the contextual setting where customers have personalized behaviors (Section 5.2). Before introducing the details of the algorithms, we first formally define the notations to characterize the customers' behaviors.

Notations for customers' behaviors Let Φ_t be the random variable that denotes the amount of products viewed by the t-th customer and $\boldsymbol{\Gamma}_t = \{\gamma_{t,k}\}_{k=1}^{\Phi_t}$ be the purchase activities. Specifically, $\gamma_{t,k} = 1$ if and only if the product at the k-th position (*i.e.*, $\sigma_{t,k}$) is purchased.

Let random variable $\eta_{t,k}$ denote whether the customer continues viewing products after she views the k-th product but does not buy it and $\boldsymbol{\eta}_t = \{\eta_{t,k}\}_{k=1}^{\Phi_t}$. Let random variable $\mu_{t,k}$ denote whether the customer continues viewing products if she buys the k-th product and $\boldsymbol{\mu}_t = \{\mu_{t,k}\}_{k=1}^{\Phi_t}$. So $\eta_{t,k}$ is observed only when $\gamma_{t,k} = 0$ and $\mu_{t,k}$ is observed only when $\gamma_{t,k} = 1$.

Let $\mathrm{Ber}(\cdot)$ denote the Bernoulli distribution and we have the following result on the distribution of these random variables.

Lemma 5.1. *$\gamma_{t,k}$ are i.i.d. random variables and $\gamma_{t,k} \sim \mathrm{Ber}\left(\lambda_{\sigma_{t,k}}\right)$. $\eta_{t,k}$ are i.i.d. random variables and $\eta_{t,k} \sim \mathrm{Ber}(q_t)$. $\mu_{t,k}$ are i.i.d. random variables and $\mu_{t,k} \sim \mathrm{Ber}(q_t s_t)$.*

Lemma 5.1 is the basis of the algorithms introduced later that estimate parameters $\boldsymbol{\lambda}$, q, and s from historical data. In addition, compared with s_t, it is easier to estimate $q_t s_t$ from data and we denote it as $w_t \triangleq q_t s_t$.

5.1 Non-contextual online setting

In this subsection, we assume that all customers share the same parameters, including the purchase probabilities on different products and the parameters on the distribution of the attention span and purchase budget. To be specific, we assume that there exist $\boldsymbol{\lambda}, q, s$ such that $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}$, $q_t = q$, and $s_t = s$.

A) Estimation of parameters

To estimate the parameters at timestep t, we leverage the observed behavior of customers before timestep t. We first calculate the following statistics that can help provide unbiased estimators of the customer parameters.

In detail, let $C_{t,k}$ be the total number of times that the product k is observed and $c_{t,k}$ be the number of times the product k is purchased, *i.e.*, $C_{t,k} \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\sigma_{t,i} = k\right]$ and $c_{t,k} \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\sigma_{t,i} = k\right]\mathbb{I}\left[\gamma_{t,i} = 1\right]$. In addition, Let D_t^q be times that customers view a product and not purchases it and d_t^q be the times that customers continue viewing the list after viewing a product without purchasing it, *i.e.*, $D_t^q \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\gamma_{t,i} = 0\right]$ and $d_t^q \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\gamma_{t,i} = 0\right]\eta_{t,i}$. Furthermore, let D_t^w be times that customers purchase a product and d_t^w be the times that customers continue viewing the list after purchasing a product, *i.e.*, $D_t^w \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\gamma_{t,i} = 1\right]$ and $d_t^w \triangleq \sum_{u=1}^{t}\sum_{i=1}^{\Phi_t} \mathbb{I}\left[\gamma_{t,i} = 1\right]\mu_{t,i}$.

With these statistics, the parameters $\boldsymbol{\lambda}$, q, and $w = qs$ are estimated as follows.

$$\hat{\lambda}_{t,k} \triangleq c_{t,k}/C_{t,k}, \quad \hat{q}_t \triangleq d_t^q/D_t^q, \quad \hat{w}_t \triangleq d_t^w/D_t^w. \tag{6}$$

According to the Hoeffding's inequality, we provide the following proposition that gives the error bound of our parameter estimation approach.

Proposition 5.2. *For any $t \geq 1$, with probability at least $1 - 2(N+2)t^{-2}$, the following holds:*

$$\left|\hat{\lambda}_{t,k} - \lambda_k\right| \leq \sqrt{\frac{\log t}{C_{t,k}}}, \forall k \in [N], \quad |\hat{q}_t - q| \leq \sqrt{\frac{\log t}{D_t^q}}, \quad \textit{and } |\hat{w}_t - qs| \leq \sqrt{\frac{\log t}{D_t^w}}. \tag{7}$$

B) Algorithm

Following the classic *optimism in the face of uncertainty principle* [1], we design a UCB-like algorithm that learns customer behaviors and maximizes revenue simultaneously. In detail, according to Proposition 5.2, we learn the optimistic estimators of the parameters given as follows

$$\tilde{\lambda}_{t,k} \triangleq \min\left\{1, \hat{\lambda}_{t,k} + \sqrt{\frac{\log t}{C_{t,k}}}\right\}, \tilde{q}_t \triangleq \min\left\{1 - \epsilon_q, \hat{q}_t + \sqrt{\frac{\log t}{D_t^q}}\right\}, \tilde{w}_t \triangleq \min\left\{\tilde{q}_t, \hat{w}_t + \sqrt{\frac{\log t}{D_t^w}}\right\}. \tag{8}$$

As shown in Algorithm 1, when the t-th customer arrives, the online retailer displays the products based on Theorem 4.1 with $\tilde{\boldsymbol{\lambda}}_{t-1}$, $\tilde{q}_{t-1}$, and $\tilde{w}_{t-1}/\tilde{q}_{t-1}$ shown in Step 4. Afterward, the online retailer updates the statistics according to customers' feedbacks as shown in Steps 5 and 6. The optimistic estimators are then calculated in Step 7.

Algorithm 1: Non-contextual online policy

Input: Products revenue $\boldsymbol{r}$ and hyperparameter ϵ_q.
Initialization: $\tilde{\lambda}_{0,k} = 1$ for $k \in [N]$, $\tilde{q}_0 = 1 - \epsilon_q$, $\tilde{w}_0 = 1 - \epsilon_q$.
for *t=1:T* **do**
  Let $\boldsymbol{\sigma}_t$ be the optimal offline policy from Theorem 4.1 with $\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}_{t-1}$, $q = \tilde{q}_{t-1}$, and $s = \tilde{w}_{t-1}/\tilde{q}_{t-1}$.
  Offer ranking policy $\boldsymbol{\sigma}_t$ and observe Φ_t, $\boldsymbol{\Gamma}_t$, $\boldsymbol{\eta}_t$, $\boldsymbol{\mu}_t$.
  Update statistics $C_{t,k}$, $c_{t,k}$, D_t^q, d_t^q, D_t^w, and d_t^w.
  Calculate $\tilde{\lambda}_{t,k}$, $\tilde{q}_t$, and $\tilde{w}_t$ by Equations (6) and (8).
end

C) Regret analysis

To theoretically analyze the performance of Algorithm 1, we first introduce the regret, which measures the total difference between the maximal revenue achieved by $\boldsymbol{\sigma}^*$ and the cumulative reward by an online algorithm after T rounds.

$$\mathrm{Reg}_T \triangleq \sum_{t=1}^{T} \mathrm{Revenue}\left(\boldsymbol{\sigma}^*; \boldsymbol{\lambda}, q, s\right) - \mathrm{Revenue}\left(\boldsymbol{\sigma}_t; \boldsymbol{\lambda}, q, s\right). \tag{9}$$

Reg_T is a random variable and the randomness comes from the uncertainty in customers' behaviors. As a result, we focus on $\mathbb{E}\left[\mathrm{Reg}_T\right]$, which is a common routine in bandit literature [45]. The performance of the algorithm is given by the following theorem.

Theorem 5.3. *Under Assumption 3.1 (with parameter ϵ_q), for any $T > 1$, the regret achieved by Algorithm 1 is bounded by*

$$\mathbb{E}\left[\mathrm{Reg}_T\right] \leq C \cdot \frac{r_{\max}}{\epsilon_q} N \sqrt{T \log T}, \tag{10}$$

where C is a absolute constant (independent of problem parameters) and $r_{\max} = \max_{k \in [N]} r_k$.

Remark. We analyze different parameters' impacts on the regret bound in Equation (10). Firstly, the regret grows at $\tilde{O}(\sqrt{T})$, which is a standard result in online learning algorithms [45]. Secondly, the regret bound linearly related to N, which is similar to the results in cascade bandit models [48] and revenue manamge literature [17]. Finally, the bound depends on $r_{\max}/\epsilon_q$, which measures the maximal expected revenue the online retailer could achieve from each customer.

5.2 Contextual online setting

We consider a more realistic scenario where the customers' behaviors are various and depend on their own features. In detail, let $x_t \in \mathbb{R}^{m_x}$ be the feature of the t-th customer with dimension m_x and $y_{t,k} \in \mathbb{R}^{m_y}$ be the joint feature of the t-th customer and k-th product with dimension m_y. $y_{t,k}$ can be the concatenation of customer features and product features or fusion through a transformation such as matrix multiplication. In this paper, we adopt concatenation for simplicity. Let $\|\cdot\|$ denote the ℓ_2 norm. Following [17], we consider the classic linear bandit setting [7, 21, 38] and the ground-truth data-generating process is given by the following assumptions.

Assumption 5.1. There exists constant $\beta_\lambda \in \mathbb{R}^{m_y}$, $\beta_q \in \mathbb{R}^{m_x}$, and $\beta_s \in \mathbb{R}^{m_x}$, such that

$$\lambda_{t,k} = y_{t,k}^\top \beta_\lambda, \forall k \in [N], \quad q_t = x_t^\top \beta_q, \quad \text{and} \quad s_t = x_t^\top \beta_s. \tag{11}$$

Assumption 5.2. For any $t > 0$ and $k \in [N]$, $\|x_t\| \leq 1$ and $\|y_{t,k}\| \leq 1$. In addition, there exists a constant $U > 0$ such that $\|\beta_\lambda\|, \|\beta_q\|, \|\beta_s\| \leq U$.

Remark. These two assumptions are common in revenue maximization and linear bandits [1, 45]. Assumption 5.1 can be satisfied with kernel functions for complex data-generating processes and Assumption 5.2 can be achieved by normalization. Thus the assumptions are rational and easily attainable in real applications.

A) Estimation of parameters

For the estimation of $\boldsymbol{\lambda}_t$ and q_t, we use the ridge regression method to estimate β_λ and β_q. Given the observation of the t-th customer's behaviors, the estimation $\hat{\beta}_{t,\lambda}$, $\hat{\beta}_{t,q}$ is given by

$$\hat{\beta}_{t,\lambda} = \operatorname*{argmin}_{\beta \in \mathbb{R}^{m_y}} \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \left(y_{u,\sigma_{u,k}}^\top \beta - \gamma_{u,k} \right)^2 + \alpha_\lambda \|\beta\|_2^2 = \Sigma_{t,\lambda}^{-1} \rho_{t,\lambda}, \tag{12}$$

$$\hat{\beta}_{t,q} = \operatorname*{argmin}_{\beta \in \mathbb{R}^{m_x}} \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 0] \left(x_u^\top \beta - \eta_{u,k} \right)^2 + \alpha_q \|\beta\|_2^2 = \Sigma_{t,q}^{-1} \rho_{t,q}. \tag{13}$$

Here $\Sigma_{t,\lambda} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} y_{u,\sigma_{u,k}} y_{u,\sigma_{u,k}}^\top + \alpha_\lambda \mathbf{I}_{m_y}$, $\rho_{t,\lambda} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \gamma_{u,k} y_{u,\sigma_{u,k}}$, $\Sigma_{t,q} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 0] x_u x_u^\top + \alpha_q \mathbf{I}_{m_x}$, and $\rho_{t,q} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 0] \eta_{u,k} x_u$. $\mathbf{I}_{m_x}$ and $\mathbf{I}_{m_y}$ are the identity matrices with dimension m_x and m_y, respectively.

For the estimation of s_t, similar to Section 5.1, we estimate $w_t = q_t s_t$ instead. Note that $w_t = q_t s_t = x_t^\top \beta_q \beta_s^\top x_t = \mathrm{vec}(x_t x_t^\top)^\top \mathrm{vec}(\beta_q \beta_s^\top)$, where $\mathrm{vec}(\cdot)$ denotes the vectorization operation that maps a matrix to a vector. As a result, define $z_t \triangleq \mathrm{vec}(x_t x_t^\top)$ and $\beta_w \triangleq \mathrm{vec}(\beta_q \beta_s^\top)$ and we can get $w_t = z_t^\top \beta_w$. Similarly, we estimate β_w by conducting ridge regression as follows

$$\hat{\beta}_{t,w} = \operatorname*{argmin}_{\beta \in \mathbb{R}^{m_x^2}} \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 1] \left(z_u^\top \beta - \mu_{u,k} \right)^2 + \alpha_w \|\beta\|_2^2 = \Sigma_{t,w}^{-1} \rho_{t,w}. \tag{14}$$

Here $\Sigma_{t,w} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 1] z_u z_u^\top + \alpha_w \mathbf{I}_{m_x^2}$ and $\rho_{t,w} = \sum_{u=1}^{t} \sum_{k=1}^{\Phi_u} \mathbb{I}[\gamma_{u,k} = 1] \mu_{u,k} z_u$. $\mathbf{I}_{m_x^2}$ is the identity matrix with dimension m_x^2.

Let $\|\beta\|_A = \sqrt{\beta^\top A \beta}$ for any positive definite matrix A. Proposition 5.4 gives the estimation error bound.

Proposition 5.4. *Under Assumptions 5.1 and 5.2 (with parameter U), for any $t \geq 1$, with probability at least $1 - 3t^{-2}$, the following holds:*

$$\left\| \hat{\beta}_{t,\lambda} - \beta_\lambda \right\|_{\Sigma_{t,\lambda}} \leq \tau(t, m_y, \alpha_\lambda), \left\| \hat{\beta}_{t,q} - \beta_q \right\|_{\Sigma_{t,q}} \leq \tau(t, m_x, \alpha_q), \textit{and } \left\| \hat{\beta}_{t,w} - \beta_w \right\|_{\Sigma_{t,w}} \leq \tau(t, m_x^2, \alpha_w). \tag{15}$$

Here $\tau(t, m, \alpha) = 1/2\sqrt{m \log(1 + tN/(m\alpha)) + 4 \log t} + \alpha^{1/2}(U+1)^2$.

B) Algorithm

Given the estimation of β_λ, β_q and β_w we estimate $\lambda_{t,k}$, q_t and w_t as follows.

$$\begin{aligned} \tilde{\lambda}_{t,k} &= \mathrm{Proj}_{[0,1]} \left(y_{t,k}^\top \hat{\beta}_{t-1,\lambda} + \tau(t-1, m_y, \alpha_\lambda) \|y_{t,k}\|_{\Sigma_{t-1,\lambda}^{-1}} \right) \\ \tilde{q}_t &= \mathrm{Proj}_{[0,1-\epsilon_q]} \left(x_t^\top \hat{\beta}_{t-1,q} + \tau(t-1, m_x, \alpha_q) \|x_t\|_{\Sigma_{t-1,q}^{-1}} \right) \\ \tilde{w}_t &= \mathrm{Proj}_{[0,\tilde{q}_t]} \left(z_t^\top \hat{\beta}_{t-1,w} + \tau(t-1, m_x^2, \alpha_w) \|z_t\|_{\Sigma_{t-1,w}^{-1}} \right) \end{aligned} \tag{16}$$

As shown in Algorithm 2, when the t-th customer arrives, the online retailer observes the customer features x_t and $y_{t,k}$ for $k \in [N]$ before learns the intermediate feature z_t and calculate $\tilde{\lambda}_{t,k}$, $\tilde{q}_t$, and $\tilde{w}_t$ according to Equation (16). Then the ranking policy $\boldsymbol{\sigma}_t$ is given as shown in Step 6. Then the statistics are calculated in Step 6 and 7 and finally the parameters $\hat{\beta}_{t,\lambda}$, $\hat{\beta}_{t,q}$, and $\hat{\beta}_{t,w}$ are estimated in Step 9 for the next iteration.

C) Regret analysis

We analyze the performance of Algorithm 2 by investigating its regret bound. For a sequence of T customers with features $\boldsymbol{X}_T = \{x_t\}_{t=1}^T$ and $\boldsymbol{Y}_T = \{\{y_{t,k}\}_{k=1}^N\}_{t=1}^T$, the regret is given by

$$\mathrm{Reg}_T(\boldsymbol{X}_T, \boldsymbol{Y}_T) = \sum_{t=1}^{T} \mathrm{Revenue}\left(\boldsymbol{\sigma}_t^*; \boldsymbol{\lambda}_t, q_t, s_t\right) - \mathrm{Revenue}\left(\boldsymbol{\sigma}_t; \boldsymbol{\lambda}_t, q_t, s_t\right). \tag{17}$$

Algorithm 2: Contextual online policy

Input: Products revenue $\boldsymbol{r}$ and hyperparameter ϵ_q, α_λ, α_q, and α_w.
Initialization: $\hat{\beta}_{0,\lambda} = 0$, $\hat{\beta}_{0,q} = 0$, and $\hat{\beta}_{0,w} = 0$.
for *t=1:T* **do**
  Observe customer features x_t and $y_{t,k}$ for $k \in [N]$. Let $z_t = \mathrm{vec}(x_t x_t^\top)$.
  Calculate $\tilde{\lambda}_{t,k}$, $\tilde{q}_t$, and $\tilde{w}_t$ according to Equation (16).
  Let $\boldsymbol{\sigma}_t$ be the optimal ranking policy from Theorem 4.1 with $\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}_t$, $q = \tilde{q}_t$, and $s = \tilde{w}_t/\tilde{q}_t$.
  Offer ranking policy $\boldsymbol{\sigma}_t$ and observe Φ_t, $\boldsymbol{\Gamma}_t$, $\boldsymbol{\eta}_t$, $\boldsymbol{\mu}_t$.
  Calculate statistics $\Sigma_{t,\lambda}$, $\rho_{t,\lambda}$, $\Sigma_{t,q}$, $\rho_{t,q}$, $\Sigma_{t,w}$, and $\rho_{t,w}$.
  Calculate estimated parameters $\hat{\beta}_{t,\lambda}$, $\hat{\beta}_{t,q}$, and $\hat{\beta}_{t,w}$ according to Equations (12), (13) and (14).
end

Similar to the non-contextual setting, $\mathrm{Reg}_T(\boldsymbol{X}_T, \boldsymbol{Y}_T)$ is a random variable and we focus on $\mathbb{E}[\mathrm{Reg}_T(\boldsymbol{X}_T, \boldsymbol{Y}_T)]$. As a result, the performance of the algorithm is given by the following theorem.

Theorem 5.5. *Under Assumptions 3.1 (with parameter ϵ_q), 5.1, and 5.2 (with parameter U), for any $T > 1$, the regret achieved by Algorithm 2 is bounded by*

$$\mathbb{E}[\mathrm{Reg}_T(\boldsymbol{X}_T, \boldsymbol{Y}_T)] \leq C \cdot \frac{r_{\max}}{\epsilon_q} \left(\chi(T, m_y, \alpha_\lambda) + \chi(T, m_x, \alpha_q) + \chi(T, m_x^2, \alpha_w) \right). \tag{18}$$

Here C is a absolute constant (independent of problem parameters), $r_{\max} = \max_{k\in[N]} r_k$, and

$$\chi(T, m, \alpha) = \left(\frac{1}{2}\sqrt{m \log\left(1 + \frac{TN}{m\alpha}\right) + 4\log T} + \alpha^{1/2}(U+1)^2 \right) \sqrt{\frac{TN^2 m \log\left(1 + \frac{TN}{m\alpha}\right)}{\alpha \log(1+\alpha^{-1})}}. \tag{19}$$

Remark. We analyze different parameters' impacts on the regret bound in Equation (19). Firstly, the regret grows at $\tilde{O}(\sqrt{T})$, a standard result in online learning algorithms [45]. Secondly, the regret depends $\tilde{O}(N)$, which shares a similar result with the revenue management literature [17]. Thirdly, similar to Theorem 5.3, the bound depends linearly on $r_{\max}/\epsilon_q$ that measures the maximal expected revenue the online retailer could achieve from each customer. Fourthly, the bound depends on the dimensions of features (*i.e.*, m_x and m_y) by $\tilde{O}(\sqrt{m_y})$ and $\tilde{O}(m_x)$ because the feature dimension for the estimation of $w_t = q_t s_t$ is m_x^2 and then the result is consistent with [17, 48]. Finally, the bound depends on $O(U^2 + U)$ since $\|\beta_w\| \leq U^2$ and $\|\beta_q\|, \|\beta_\lambda\| \leq U$, which agrees with [17].

6 Experiments

In this section, we conduct experiments on synthetic data to verify the performances of the online ranking policies. More experimental results on both synthetic and real-world datasets are in appendix.

Data-generating processes For the non-contextual setting, we consider $N = 50$ and 300 products and $T = 100,000$ customers. The revenue for each product r_k is uniformly distributed in $[0, 1]$ and the purchase probabilities for each product are uniformly sampled from $[0, 0.2]$. We set $q = 0.8$ and $s = 0.2$ and 0.5 to test the performance of different models on characterizing customers' behaviors with multiple purchases.

For the contextual setting, we consider $N = 50$ and 300 products and $T = 30,000$ customers. The revenue r_k is also uniformly distributed in $[0, 1]$. Both the dimension of customer features and products features are set to 10 (i.e., $m_x = 10$). The joint feature $y_{t,k}$ (the feature of the t-th customer and k-th product) is a concatenate of the corresponding customer and product feature so that $m_y = 20$. The customer features and product features are both uniformly sampled from $[0, 1/\sqrt{m_x}]$ to satisfy the normalization constraint in Assumption 5.2. Afterward, the coefficients β_λ, β_q, and β_s in Equation (11) are uniformly sampled from $[0, 1]$. To ensure $\lambda_{t,k}$, q_t, s_t have similar range in the contextual setting, we normalize the coefficients such that the maximal elements in $\lambda_{t,k}$ and q_t (denoted as $\lambda_{\max}$ and $q_{\max}$) are 0.2 and 0.8, respectively. Similarly, by normalizing β_s, we set the maximal element in s_t (denoted as $s_{\max}$) to 0.2 and 0.5.

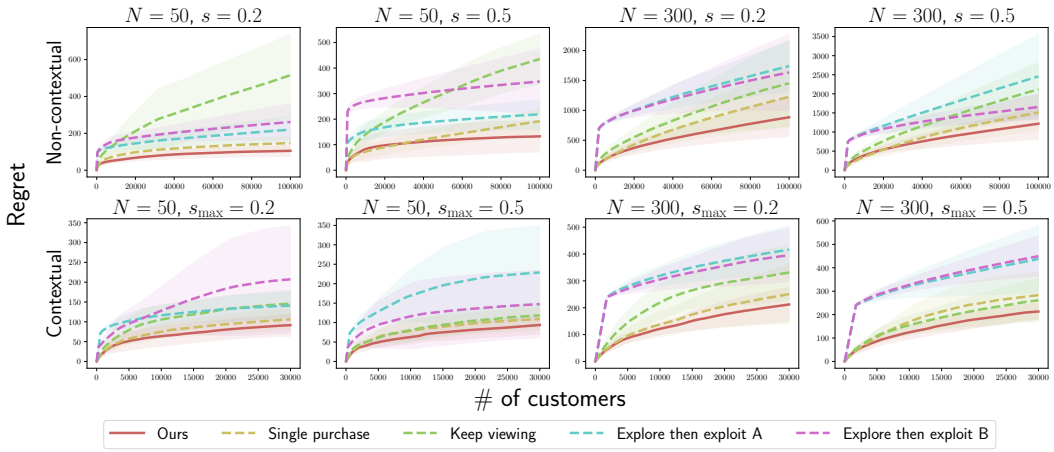

Figure 1: Experimental results on synthetic data. We plot the regret curves for different settings by varying $N = 50, 300$ and $s/s_{\max} = 0.2, 0.5$ in both non-contextual and contextual scenarios.

Baselines We adopt the following baselines. Firstly, Cao and Sun [13] considered the single purchase setting, which is the simplified version of our multiple purchase setting. We denote it as **Single purchase**. Secondly, we adopt the method in [39] (denoted as **Keep viewing**), which assumes that customers always keep viewing the ranking list after purchasing products. Furthermore, we implement two explore-then-exploit algorithms (denoted as **Explore then exploit A** and **Explore then exploit B**, respectively) to verify the advantages of our method that could balance exploration and exploitation.

Results and analysis We implement the experiments for 10 distinct simulations by resampling customers' behaviors while their basic characteristics remain unchanged. Specifically, in both the non-contextual and contextual settings, the purchase probabilities (*i.e.*, $\boldsymbol{\lambda}_t$) and parameters on the span attention and purchase budget (*i.e.*, q_t and s_t) are the same. However, the same customer in distinct simulations may purchase different products due to randomness.

We then evaluate our method and baselines and present the regret via calculating the regret (according to Equations (9) and (17)). The results are shown in Figure 1 and our method outperforms all baselines in both settings with various parameters. On the one hand, the Single purchase and Keep viewing baseline considers different customers' behaviors settings and is not directly appliable here for the more realistic multiple purchase setting. On the other hand, compared with the explore-then-exploit based algorithms, our method integrates the advantages from traditional UCB-like algorithms, leading to the better exploration-exploitation trade-off for smaller regret. Furthermore, as a accumulation of regret of each customer, the reported overall regret grows as the number of customers increases. Basically, the gap between our method and Single purchase decreases as N and $s/s_{\max}$ decrease since the expected number of purchase is approaching to 1. Thus as an extension of the single purchase setting, the multiple purchase setting is more realistic and challenging. Moreover, the growth gradient of our method is smoother than its counterparts, suggesting better scalability on customer numbers and applicability to real application scenarios. Reports and discussions of more results including the expected revenue and the ratio over the expected revenue of the optimal ranking policy are in Appendix.

7 Conclusion

To conclude, we propose a novel setting to deal with multiple-purchase activities of customers on online scenarios. We characterize the optimal rankling policy and further design UCB-like algorithms with $\tilde{O}(\sqrt{T})$ regret based on them in both non-contextual and contextual online settings. We conduct extensive experiments to prove the effectiveness of our method.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

[2] Andrés Abeliuk, Gerardo Berbeglia, Manuel Cebrian, and Pascal Van Hentenryck. Assortment optimization under a multinomial logit model with position bias and social influence. *4OR*, 14 (1):57–75, 2016.

[3] Ashish Agarwal, Kartik Hosanagar, and Michael D Smith. Location, location, location: An analysis of profitability of position in online advertising markets. *Journal of marketing research*, 48(6):1057–1073, 2011.

[4] Ali Aouad and Danny Segev. Display optimization for vertically differentiated locations under multinomial logit preferences. *Management Science*, 67(6):3519–3550, 2021.

[5] Victor F Araman and René Caldentey. Dynamic pricing for nonperishable products with demand learning. *Operations research*, 57(5):1169–1188, 2009.

[6] Arash Asadpour, Rad Niazadeh, Amin Saberi, and Ali Shameli. Ranking an assortment of products via sequential submodular optimization. *arXiv preprint arXiv:2002.09458*, 2020.

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

[8] Michael R Baye, J Rupert J Gatti, Paul Kattuman, and John Morgan. Clicks, discontinuities, and firm demand online. *Journal of Economics & Management Strategy*, 18(4):935–975, 2009.

[9] Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.

[10] Omar Besbes and Assaf Zeevi. Blind network revenue management. *Operations research*, 60 (6):1537–1550, 2012.

[11] Josef Broder and Paat Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012.

[12] Juan José Miranda Bront, Isabel Méndez-Díaz, and Gustavo Vulcano. A column generation algorithm for choice-based network revenue management. *Operations research*, 57(3):769–784, 2009.

[13] Junyu Cao and Wei Sun. Dynamic learning of sequential choice bandit problem under marketing fatigue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3264–3271, 2019.

[14] Junyu Cao, Wei Sun, and Zuo-Jun Max Shen. Sequential choice bandits: Learning with marketing fatigue. *Available at SSRN 3355211*, 2019.

[15] Junyu Cao, Wei Sun, Zuo-Jun Max Shen, and Markus Ettl. Fatigue-aware bandits for dependent click models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3341–3348, 2020.

[16] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10, 2009.

[17] Ningyuan Chen, Anran Li, and Shuoguang Yang. Revenue maximization and learning in products ranking. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 316–317, 2021.

[18] Yuxin Chen and Song Yao. Sequential search with refinement: Model and application with click-stream data. *Management Science*, 63(12):4345–4365, 2017.

[19] Wang Chi Cheung, Vincent Tan, and Zixin Zhong. A thompson sampling algorithm for cascading bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 438–447. PMLR, 2019.

[20] Roberta Comunian, Caroline Chapain, and Nick Clifton. Location, location, location: exploring the complex relationship between creative industries and place. *Creative Industries Journal*, 3 (1):5–10, 2010.

[21] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

[22] James M Davis, Guillermo Gallego, and Huseyin Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.

[23] Arnoud V Den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18, 2015.

[24] Mahsa Derakhshan, Negin Golrezaei, Vahideh Manshadi, and Vahab Mirrokni. Product ranking on online platforms. *Management Science*, 2022.

[25] Daria Dzyabura and John R Hauser. Recommending products when consumers learn their preference weights. *Marketing Science*, 38(3):417–441, 2019.

[26] Jacob Feldman and Danny Segev. Improved approximation schemes for mnl-driven sequential assortment optimization. *Available at SSRN 3440645*, 2019.

[27] Kris J Ferreira, Sunanda Parthasarathy, and Shreyas Sekar. Learning to rank an assortment of products. *Management Science*, 2021.

[28] Alvaro Flores, Gerardo Berbeglia, and Pascal Van Hentenryck. Assortment optimization under the sequential multinomial logit model. *European Journal of Operational Research*, 273(3): 1052–1064, 2019.

[29] Guillermo Gallego, Anran Li, Van-Anh Truong, and Xinshang Wang. Approximation algorithms for product framing and pricing. *Operations Research*, 68(1):134–160, 2020.

[30] Pin Gao, Yuhang Ma, Ningyuan Chen, Guillermo Gallego, Anran Li, Paat Rusmevichientong, and Huseyin Topaloglu. Assortment optimization and pricing under the multinomial logit model with impatient customers: Sequential recommendation and selection. *Operations research*, 69 (5):1509–1532, 2021.

[31] Xiangyu Gao, Stefanus Jasin, Sajjad Najafi, and Huanan Zhang. Joint learning and optimization for multi-product pricing (and ranking) under a general cascade click model. *Management Science*, 2022.

[32] Anindya Ghose and Sha Yang. An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management science*, 55(10):1605–1622, 2009.

[33] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, pages 11–20, 2009.

[34] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*, pages 124–131, 2009.

[35] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Dcm bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning*, pages 1215–1224. PMLR, 2016.

[36] Sergei Koulayev. Search for differentiated products: identification and estimation. *The RAND Journal of Economics*, 45(3):553–575, 2014.

[37] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776. PMLR, 2015.

[38] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[39] Yuan Liang, Chunlin Huang, Xiuguo Bao, and Ke Xu. Sequential dynamic event recommendation in event-based social networks: An upper confidence bound approach. *Information Sciences*, 542:1–23, 2021.

[40] Nan Liu, Yuhang Ma, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with sequential offerings. *INFORMS Journal on Computing*, 32(3):835–853, 2020.

[41] Qing Liu and Neeraj Arora. Efficient choice designs for a consider-then-choose model. *Marketing Science*, 30(2):321–338, 2011.

[42] Siddharth Mahajan and Garrett Van Ryzin. Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49(3):334–351, 2001.

[43] Rad Niazadeh, Negin Golrezaei, Joshua Wang, Fransisca Susan, and Ashwinkumar Badanidiyuru. Online learning via offline greedy: Applications in market design and optimization. 2020.

[44] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[45] Aleksandrs Slivkins. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

[46] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[47] Raluca M Ursu. The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science*, 37(4):530–552, 2018.

[48] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. *arXiv preprint arXiv:1603.05359*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See details in the appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See details in the appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Assumption 3.1, 5.1, and 5.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See details in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Details can be found in the appendix
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details can be found in the appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See details in the appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]