Multi-page Menu Recommendation in Cascade Model with Externalities

Richa Dhingra, Hansraj Satyam Verma, Alexandre Reiffers-Masson, Veeraruna Kavitha

Multi-page Menu Recommendation in Cascade Model with Externalities

Richa Dhingra, Hansraj Satyam Verma, Alexandre Reiffers-Masson and Veeraruna Kavitha

***Abstract*—In this paper, we consider a variant of the *cascade model* of customer behavior, where the customer browses through a multi-page menu, scanning each page from top to the bottom predominantly. Each page is assigned items belonging to a specific class out of a set of such classes. He/she adopts the first most attractive content, which generates some revenue. We aim at maximizing the total revenue by finding an optimal index-based policy for ranking the content when the customer preferences and patience levels are known. When we have no prior information about the customer, we design the Online Greedy Algorithm (OGA) which we prove to be asymptotically converging to the optimal solution with probability one. We also provide high probability finite-time convergence bounds for the same.**

***Keywords*—recommender system, dynamic programming, online learning, perturbation analysis, cascade model, stochastic system, sponsored search**

I. Introduction

A Recommender System (RS) is an information filtering system, designed to predict or influence the behavior of the user based on the digital footprint left by him/her [9]. These systems have diverse applications, e.g., in Sponsored Search advertisement, product-recommenders for online-purchase portals etc. A recommendation list owner is interested in maximizing the click-through rate, the probability that one of the items will be clicked. The larger the click-through rate, the larger the expected revenue. Alternatively one might have different levels of preferences for click-through rates of different items.

There have been models [3],[10] based on the assumptions that the click-through rate of item i depends solely on the attraction probability of the item i and the location of the slot which it is assigned. Such models overlook the effects of other items shown on the same page. An item with lower revenue generation placed more prominently may detract the customer from another item which generates higher revenue but is assigned a lower position on the page. Moreover, an expensive item placed at a higher spot may have lower attraction probability for a customer as compared to a less expensive item assigned a lower spot. Thus, different items on a page appear to alter each others click-through rates.

In this paper, we aim at maximizing the revenue generated by a multi-page menu by taking into account this externality effect items have on each other. This menu consists of different classes of items on each page. The problem can be viewed as an allocation problem of the classes to the pages and of the items to the slots within those pages.

Our model: We focus on a model which recommends items based on user rating when certain groups/classes of products are present. We aim at finding the optimal ordering for a menu consisting of these classes of items using the click-log data of the customer. The menu consists of several pages, each page containing a certain class of items. The customer is interested in consuming/adopting one of the item on this menu. He scans each page from the top to bottom predominantly, however, can follow a random browsing pattern too. We term the customer's browsing pattern as general browsing. He can leave the menu mid-way if he is not able to find something of his interest or if he starts to get disinterested while browsing the list. The menu is such that items and the pages are displayed in a specific order and this order will impact the decision of a customer to adopt a given item. We model the behavior of the customers using parametric Markov chain. Since the click-data has an inherent position-bias effect, therefore the customer's attraction towards the list of items is not just dependent on the list of items shortlisted but also on the order in which that list is presented. This is due to the decay observed in the click through rate, with increase in rank.[1], [8] Additionally, we are finding a recommendation for the customer that helps the recommendation owner (or menu designer) maximize the utility he seeks through the decisions taken by the customer who is presented with it. This helps in offering services that meet the needs of both the players.

Organization and Main Results: The organization of the paper is as follows: In section II, we describe the model and the utility that the owner of the recommendation list is interested in. We also define Look Only Ahead browsing (LoA), i.e., where the user scans the items from top to down, in the context of our model. We provide a perturbation result which shows that there exists a subset of cases with general browsing where the optimal ordering policy for LoA browsing is optimal. In section III, we derive the optimal ranking for LoA browsing when the parameters of the model are known. In section IV, we provide an online learning algorithm (called OGA) which converges to the optimal policy by observing the partial feedback of the customer. The convergence is proved w.p. 1 and we also derive a high probability bound for finite time convergence. In section V, we provide extensive numerical experiments to illustrate the optimal ranking given in this paper. We also include plots illustrating the converging behaviour of OGA to the optimal ordering policy. Finally in section VI, we briefly conclude our work.

Related Work: The model used in this paper is a variation of the simple Cascade model of customer behaviour. In a simple Cascade model, the user is presented with a list of

K items ordered in a specific order. The user browses the menu starting from the first item to the last item in a top to down manner and selects the first attractive item and quits the session thereafter [6]. The authors in [5] focus on a variation of this model called the Slated cascade model. They provide an optimal ordering policy by considering externality effect. However, they considered a special case of user browsing pattern where every user scans any page from top to bottom (until stopping the scan).

We relax this restriction in our work. The user will predominantly scan from top to bottom, however, he can browse items randomly in any order on a given page. We also consider the existence of classes of items to be ranked on different pages. As a result, the model considered in this paper differs from the Slated Cascade model in [5]. In fact, the results of this paper can be considered as an extension of their work to a more general environment. Also, in this paper, we have suggested an algorithm that not only provides an ordering scheme for items but also helps in learning the key parameters that drive the behavior of the users (attraction probabilities and impatience level).

There are some timeline based models that study the influence on a customer's choices by other customers present in the social network as highlighted in [4], [7], [2]. There are other models like [1] that attempt to explain the position-bias effect present in the customer click logs and ways to model it. This is especially seen in the cascade models that tend to impact the customer's decisions.

II. Menu of Menus with general browsing

Consider a menu containing different pages. On each page, items belonging to one specific class are displayed. Let $\mathcal{C} := \{1, \cdots, C\}$ represent these classes. The class $c \in \mathcal{C}$ contain I_c number of items, given by $\mathcal{I}_c = \{1, \cdots, I_c\}$. The items are categorised into different classes based on their properties (e.g., physical properties like color, shape).

Menu Ordering Policy: Let $\Pi := (P_0, P_1, P_2, \cdots P_C)$ denote the set of permutation matrices (**menu ordering policy**) For a class $c \in \mathcal{C}$, P_c (with $c \geqslant 1$) represents the permutation which characterizes the ordering of items of class c on a page. P_0 corresponds to inter-page orderings, and assigns the classes of items to the pages. For simpler notations, we use index 0 to represent quantities related to inter-page entities. Let ΔP be the set of all tuples of permutation matrices.

Given a policy Π, we introduce some notations (characterized by this policy) which we will be using in this paper. P_0 explicitly denotes the permutation matrix $[[\tilde{p}_{cp}]]_{1 \leqslant p,c \leqslant C}$ used for assigning the classes of items to pages. Here, $\tilde{p}_{cp} = 1$ if class c has been assigned to page p and is equal to zero otherwise. $\tilde{s} = [\tilde{s}_p]_{1 \leqslant p \leqslant C}$ denotes the corresponding vector notation of the ordering of the menus where $\tilde{s}_p \in \mathcal{C}$ represents the class displayed on page p. Also, let vector $\tilde{r} = [\tilde{r}_c]_{1 \leqslant c \leqslant C}$ denote the mappings of the classes to the respective menu page i.e. the c^{th} component of $\tilde{r}$ denotes the page that class $c \in \mathcal{C}$ has been assigned to. For a fixed permutation Π, let $s_{p,k}$ be the item from class $\tilde{s}_p$ to be placed at position k within the page p. Let $r_{c,i}$ denote the position of item i within the page $p := \tilde{r}_c$ from class c. Finally, for any probability value q we will write $1 - q$ as $\bar{q}$.

General Browsing: The customer starts the browsing session with the content at position 1 on page 1. If the user gets attracted to an item, he/she buys it and a revenue is generated. $\beta_{c,i}$ denotes the attraction probability of an item i belonging to class c and $w_{c,i}$ denotes the revenue generated when it is bought. Upon observing the item at position k on page p, the customer takes a sequence of 3 decisions:
(i) With probability $\alpha_{p,k} := \beta_{\tilde{s}_p, s_{p,k}}$, the customer will buy the item and generate revenue $\omega_{p,k} := w_{\tilde{s}_p, s_{p,k}}$. When doing so, the customer is automatically logged out of the session.
(ii) If $k < I_{\tilde{s}_p}$ and the customer does not buy item at position k, then, with probability γ_p, he stops scanning the menu and exits the process.
(iii) If the customer decides to continue scanning (w.p. $\overline{\gamma}_p = 1 - \gamma_p$), then the customer predominantly scans the item at the next level, but has small probability of going back and forth on the list displayed in any page. Let $m^p_{k,l}$ be the probability that the customer scans item at l-th level after scanning the item at k-th level on p-th page.

To simplify the notations and analysis [1], we make the following assumptions with respect to the browsing behaviour:
• We assume that a customer can only browse randomly within a page, but cannot go back and forth across the pages.
• We also assume that $m^p_{I_{\tilde{s}_p},j} = 0$ for all $j \leqslant I_{\tilde{s}_p}$. i.e. if a customer reaches the end of any page without buying an item, then with probability γ_0 he will quit the menu and with probability $\bar{\gamma}_0 = 1 - \gamma_0$ he will directly go to the next page. The customer will not go to some other level within the same page once he is in the last level. This assumption also implies that if the customer reaches the last item of the last page of the menu ($(p,k) = (C, I_{\tilde{s}_C})$) without purchasing any item, then he automatically quits the session.
• We fix $m^p_{k,k} = 0$ for any $k = 1 \ldots I_{\tilde{s}_p}$. It is a reasonable assumption that if a customer does not buy an item on the current level and does not intend to quit either, then he will not come back to the current level in the immediate next-step.
Let $M^p_{\epsilon^p} = M^p_{\epsilon^p}(\Pi) := [[m^p_{k,l}]]_{1 \leqslant k,l \leqslant I_{\tilde{s}_p}}$ represent the $I_{\tilde{s}_p} \times I_{\tilde{s}_p}$ matrix corresponding to the class $\tilde{s}_p$ displayed on page p. With a high probability, the user scans the next item. i.e., we assume that $m^p_{k,k+1} = 1 - \epsilon^p_k$, while the rest of the probabilities $\sum_{k \neq l, l+1} m^p_{k,l} = \epsilon^p_k$. Let $\epsilon := \max_{p,k} \epsilon^p_k$.

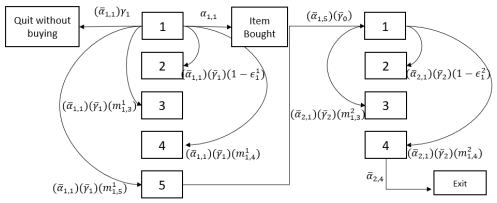

Fig. 1: Example, 2-Page menu with 2 classes (5 & 4 items), $\bar{\alpha}_{p,k}$-probability of not buying item at level k of page p, after inspection.

When a menu is designed using the ordering pol-

[1] One can easily consider back and forth behaviour across the pages, and take $m^p_{I_{\tilde{s}_p},j} (j \leqslant I_{\tilde{s}_p})$ to be non-zero. An analysis equivalent to the one given in this paper will hold.

icy Π, the overall browsing process including the quitting possibilities can be summarized through the matrix $\mathbf{M}^{\epsilon}(\Pi) = [[\tilde{m}^{p,p'}_{k,k'}]]_{1\leqslant p,p'\leqslant C, 1\leqslant k\leqslant I_{\tilde{s}_p}, 1\leqslant k'\leqslant I_{\tilde{s}'_p}}$ of dimension $\bar{I}\times\bar{I}$, where $\bar{I} := \left(\sum_{p=1}^{C} I_{\tilde{s}_p}\right)$, given by:

$$\mathbf{M}^{\epsilon}(\Pi) = \begin{bmatrix} \bar{\gamma}_1 M^1_\epsilon & \bar{\gamma}_0\mathbf{T}_1 & \mathbf{O}_{1,3} & \cdots & \mathbf{O}_{1,C} & \mathbf{O}_{1,C} \\ \mathbf{O}_{2,1} & \bar{\gamma}_2 M^2_\epsilon & \bar{\gamma}_0\mathbf{T}_2 & \cdots & \mathbf{O}_{2,C-1} & \mathbf{O}_{2,C} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{O}_{C-1,1} & \cdots & \cdots & \cdots & \bar{\gamma}_{C-1} M^{C-1}_\epsilon & \bar{\gamma}_0\mathbf{T}_{C-1} \\ \mathbf{O}_{C,1} & \cdots & \mathbf{O}_{C,3} & \cdots & \cdots & \bar{\gamma}_C M^C_\epsilon \end{bmatrix}$$

where $\mathbf{O}_{\mathbf{p},\mathbf{p}'}$ is the zero matrix with dimension $I_{\tilde{s}_p}\times I_{\tilde{s}_{p'}}$, and $\mathbf{T}_p$ is the $I_{\tilde{s}_p}\times I_{\tilde{s}_{p+1}}$ dimensional matrix representing transitions from page p to the next one ($\mathbf{e}_1$, is all zero vector, except for first component which is one)

$$\mathbf{T}_p = \begin{bmatrix} \mathbf{O}_{I_{\tilde{s}_p}-1\times I_{\tilde{s}_{p+1}}} \\ \mathbf{e}_1 \end{bmatrix}.$$

We derive an optimal menu design with the aim of maximising the overall expected revenue generated for cases with small values of ϵ.

Look Only Ahead (LoA) Browsing: We define $\mathbf{M}^0(\Pi)$ to represent a specialized browsing pattern, in which the customer only scans the next item down the list, i.e., when $\epsilon = 0$. In this case, the matrices $\{M^p_0\}_p$ are upper triangular matrices such that $m^p_{k,l} = 1$ if and only if $l = k+1$. In the next section, we show that for small values of ϵ, the optimizers for general browsing setting converge to the the optimizers of LoA browsing setting.

Expected revenue for given policy Π: Let $R^p_\epsilon(k;\Pi)$ be the conditional expected revenue, under ordering policy Π, when user starts browsing from position k on page p. The expected conditional revenue derived by the system for a given user before quitting with/without purchase equals:

$$R^p_\epsilon(k;\Pi) = \alpha_{p,k}\omega_{p,k} + \bar{\alpha}_{p,k}\sum_{p',k'}\tilde{m}^{p,p'}_{k,k'}R^{p'}_\epsilon(k';\Pi) \qquad (1)$$
$$1\leqslant p\leqslant C, 1\leqslant k\leqslant I_{\tilde{s}_p}.$$

Define $f(\Pi,\epsilon) = R^1_\epsilon(1;\Pi)$. The menu designer is interested in the optimal menu ordering policy (Π^*) defined by:

$$f^*(\epsilon) := \max_{\Pi\in\Delta P} f(\Pi,\epsilon) = f(\Pi^*,\epsilon) \qquad (2)$$

For different values of ϵ, we define the set of optimizers as:

$$\Pi^*(\epsilon) \quad := \quad \arg\max_{\Pi} f(\Pi,\epsilon). \qquad (3)$$

Theorem 1. [Perturbation] There exists $\bar{\epsilon} > 0$, such that the set of optimizers for any ϵ-system is same as that for LoA system, i.e., $\Pi^*(\epsilon) = \Pi^*(0)$ for all $\epsilon\leqslant\bar{\epsilon}$. $\diamondsuit$

Proof: Appendix (VII-A). ■

In view of the above theorem, we now derive an optimal menu using optimal menu(s) of LoA system, i.e., from $\Pi^*(0)$.

III. Menu of Menus with LoA

We have the same setup as defined in the previous section with extra assumption that $\epsilon = 0$. Let $f(\Pi,0)$ be the revenue generated by the menu when the items are laid out on the menu according to ordering policy $\Pi = (P_0, P_1, \cdots, P_C)$ with classes sorted on different pages as per order P_0 and each class c on the page is ordered as per the order P_c. Then, mathematically, we have :

$$f(\Pi,0) = \sum_{c=1}^{C} W_c(\Pi), \text{ where,} \qquad (4)$$

$$W_c(\Pi) := \sum_{i\in\mathcal{I}_c} q_{c,i}(\Pi) w_{c,i}, \qquad (5)$$

$$q_{c,i}(\Pi) := \bar{\gamma}_0^{(\tilde{r}_c-1)}\prod_{p=1}^{\tilde{r}_c-1}\kappa_{\tilde{s}_p}\bar{\gamma}_p^{I_{\tilde{s}_p}-1} \beta_{c,i}\prod_{k=1}^{r_{c,i}-1}\bar{\gamma}_{\tilde{r}_c}\bar{\beta}_{c,s_{\tilde{r}_c,k}}, \qquad (6)$$

with $\kappa_c := \prod_{i\in\mathcal{I}_c}\bar{\beta}_{c,i}$. Here, $q_{c,i}(\Pi)$ can be interpreted as the probability that the customer will buy item i belonging to class c, when the classes are ordered according to permutation matrix P_0 and the items for the class c are ordered according to $P_c, c > 0$. The menu designer is interested in finding the optimal permutation Π^* that maximize the overall expected revenue obtained i.e.

$$f^*(0) = \max_{\Pi} f(0,\Pi).$$

The model setting considered in the [5, Lemma 1] can be seen as a special case of the Look Only Ahead (LoA) browsing where the authors devise an ordering scheme based on the score or index value given through the function $h(.)$ (defined below) for a menu containing only one page and having items from one particular class.

Definition III.1. For a given class $c\in\mathcal{C}$ assigned to page p where the quitting probability is γ_p, we say that item i_1 is bigger than item i_2, according to the rule h (represented by, $i_1 >^{\Theta}_h i_2$) if $h(i_1;\Theta)\geqslant h(i_2;\Theta)$. Here, $h(i;\Theta)$ is defined as:

$$h(i;\Theta) := \frac{\beta_{c,i}w_{c,i}}{1-\bar{\gamma}_p\bar{\beta}_{c,i}}. \qquad (7)$$

where $\beta_{c,i}$ and $w_{c,i}$ are the attraction probabilities and price of item i respectively, note the ordering depends upon the parameters, $\Theta := \{\{\beta_{c,i}, w_{c,i}\}_l, \gamma_p\}$.

We refer to ordering scheme given based the score/index given by the above function as the h-index ordering policy. In the following theorem, we further extend the scope of [5, Lemma 1] to a more generalized menu by trying to construct an optimal menu of menus for LoA system using an h-index based ordering scheme.

Theorem 2. Assume $\gamma_p = \gamma$ for all p. Then, the optimal menu of menus that solves (2) for LoA system is given by the following:

(i) Let $[i^*_1, i^*_2\cdots, i^*_{I_c}]$ be the optimal order of the items of a class c, then

$$i^*_1 >^{\Theta_c}_h i^*_2 >^{\Theta_c}_h, \cdots, >^{\Theta_c}_h i^*_{I_c}, \text{with} \qquad (8)$$
$$\Theta_c := \{\{\beta_{c,i}, w_{c,i}\}_i, \gamma\}.$$

(ii) Additionally, the optimal average revenue for this class c is given by:

$$W_c^* = \sum_{j=1}^{I_c} \beta_{c,i_j^*} w_{c,i_j^*} \left[\prod_{l=1}^{j-1} \bar{\beta}_{c,i_l^*} \bar{\gamma} \right]. \tag{9}$$

(iii) Let $[c_1^*, c_2^* \cdots, c_I^*]$ be the optimal allocation of classes to pages, then

$$c_1^* >_h^\Theta c_2^* >_h^\Theta, \cdots, >_h^\Theta c_I^*, \text{ with } \Theta := \{\{\beta_c, w_c\}_c, \gamma_0\}, \tag{10}$$

$$w_c := \frac{W_c^*}{(1 - \bar{\kappa}_c \bar{\gamma}^{I_c - 1})}, \ \beta_c := 1 - \bar{\kappa}_c \bar{\gamma}^{I_c - 1}. \tag{11}$$

(iv) The optimal expected revenue for the entire menu equals:

$$f^*(0) = \sum_{k=1}^{C} w_{c_k^*} \beta_{c_k^*} \prod_{j=1}^{k-1} [\bar{\beta}_{c_j^*} \bar{\gamma}_0]. \quad \diamondsuit \tag{12}$$

Proof: Appendix (VII-B). ■

IV. Learning the Optimal Ranking

In the previous section, we study the optimal ranking (menus) in Menu of Menus with LoA browsing and their properties. In reality, the menu designer will not know the customer behaviour related parameters, β and γ. They would only have knowledge about which items belong to which classes. In this section, we are interested in obtaining the optimal ordering policy alongside learning the above required parameters. Towards this, we provide a greedy algorithm and illustrate its convergence to the optimal performance.

In the Online Greedy Algorithm (OGA), we suggest an iterative scheme to update the estimates $[\hat{\beta}_{c,i}]$'s, $[\hat{\gamma}_p]$'s and $\hat{\gamma}_0$, which then are used to choose a revenue optimal menu (for LoA browsing). One of the strengths of the OGA is that, whatever is the succession of the menus that the different customers are facing, the OGA always learns the parameters of the model ($[\beta_{c,i}]_i$, $[\gamma_p]_p$ and γ_0) and therefore the optimal menu. This is true at least asymptotically as proved by Theorem 3. However, this scheme does not guarantee that the convergence of all the estimators will be quick enough to ensure sufficient accuracy (required for choosing the correct revenue optimal menu) for initial (or finite number of) customers. Proposition 4 provides a result concerning the finite-time behaviour of the algorithm.

Let $t \in \mathbb{N}_+$ represent the number of customers browsing the menu. Let $\Pi(t)$ be the ordering policy used to design the menu shown to the t^{th} customer. Let $\theta_{c,i}(t) \in \{0,1\}$ be the random variable associated with the event that the t^{th} customer has seen item i from class c, and let $\zeta_{c,i}(t) \in \{0,1\}$ be the random variable indicating whether or not the customer bought that item. For a given ordering policy $\Pi(t)$ and the associated vectors $\tilde{s}(t) = [\tilde{s}_1(t), \ldots, \tilde{s}_I(t)]$, $\theta_{c,i}(t)$ is drawn according to a Bernoulli distribution with mean equal to $q_{c,i}(\Pi)/\beta_{c,i}$ (see (6)). Moreover, $\zeta_{c,i}(t)$ is drawn from a Bernoulli distribution with parameter $\beta_{c,i}$.

For each i, let $v_{c,i}(t) := \sum_{t'=0}^{t} \theta_{c,i}(t')$ be the number of customers that were shown item i among the first t customers, and $\eta_p(t) := \sum_{t'=0}^{t} (1 - \zeta_{\tilde{s}_p, s_{p,1}}(t'))$ be the number of customers that did not purchase the first item displayed in the menu, for any $t \geqslant 0$. Similarly, let $\eta_0(t) := \sum_{t'=0}^{t} (1 - \zeta_{\tilde{s}_1, s_{1,I_{\tilde{s}_1}}}(t'))$ be the number of times a customer did not purchase the item shown at the last level of page 1. Recall that $\tilde{s}_1, s_{1,I_{\tilde{s}_1}}$ is the item displayed in the last level of page 1. We assume here that one can observe the items (among the menu) seen by the customer. Define step size sequences $\{a(t) = \frac{1}{t+1}\} \subset (0, \infty)$.

Algorithm Explanation: We initialize the algorithm with a price based policy (i.e., arranging items within class in decreasing order of their prices and then arranging the classes in the decreasing order of w_c). Observe that $\zeta_{\tilde{s}_p, s_{p,k}}(t)$ denotes whether the t^{th} customer bought the item in level k at page p. Similarly, $\theta_{\tilde{s}_p, s_{p,k}}(t)$ denotes whether the t^{th} customer bought the item in level k at page p. Thus, given a customer bought an item at page p_q in level k_q, (13) and (14) update $\zeta_{c,i}$ to 1 for the item picked, and keep it zero for the other items. Moreover, all the items before page p_q are observed. The first Indicator function in (15) updates $\theta_{c,i}$ to 1 for all the items observed before page p_q. The second Indicator function updates $\theta_{c,i}$ for the items observed on page p_q before level k_q, including k_q.

β Estimation: Given that we have the estimate $\hat{\beta}_{c,i}(t)$ and item i was seen at time $t+1$ (i.e. $\theta_{c,i}(t+1) = 1$), $\hat{\beta}_{c,i}(t+1)$ is updated using (16). Note that $\hat{\beta}_{c,i}(t) = \frac{\sum_{t' \leqslant t} \zeta_{c,i}(t')}{v_{c,i}(t)}$, where $v_{c,i}(t)$ gives the number of observations of the related item i belonging to class c and $\sum_{t' \leqslant t} \zeta_{c,i}(t') \leqslant v_{c,i}(t)$. So, the update is just an iterative form of law of large numbers.

γ estimation: For each page p, we keep an estimate of the continuation probability γ_p. This is done by keeping track of the number of times we do not purchase the item at the first level of the page and continue to see the second item. $\eta_p(t)$ gives the number of estimates for γ_p, while the estimate $\hat{\gamma}_p$ can be obtained using $\sum_{t' \leqslant t} \theta_{\tilde{s}_p, s_{p,2}}(t')$ (number of customers that have seen items in position 2 on page p) and $\eta_p(t)$, i.e., $\bar{\hat{\gamma}}_p(t) = 1 - \hat{\gamma}_p(t)$ where $\bar{\hat{\gamma}}_p(t) = \frac{\sum_{t' \leqslant t} \theta_{\tilde{s}_p, s_{p,2}}(t')}{\eta_p(t)}$. Thus (17) is the usual way of rewriting the sample mean estimator for $\bar{\hat{\gamma}}_p$, (which is the probability of continuing to browse for page p) using a stochastic approximation based scheme. Iteration (18) can be explained in a similar way.

Online Greedy Ranking (OGA) *Initialization:* A tuple of doubly stochastic matrices $\Pi(0) = (P_0, P_1, \cdots, P_C)$. *For each new customer* $t = 1, 2, \ldots, n$

1) Use the ordering policy $\Pi(t)$.

2) Observe the quitting level of the customer as (p_q, k_q) and his last action $\zeta_{\tilde{s}^{\bar{r}}}(t)$. Compute:

$$\zeta_{\tilde{s}_{p_q}, s_{p_q,k_q}}(t) = 1, \text{if purchased} \tag{13}$$

$$\zeta_{c,i}(t) = 0, \ \forall c, i \text{ s.t. } i \neq s_{p_q,k_q} \tag{14}$$

$$\theta_{\tilde{s}_p, s_{p,k}}(t) = \mathbb{1}_{p<p_q} + \mathbb{1}_{k \leqslant k_q, p=p_q} \tag{15}$$

3) *Compute:*

$$\hat{\beta}_{c,i}(t+1) = \hat{\beta}_{c,i}(t) + a(v_{c,i}(t+1)) \times \theta_{c,i}(t+1)\left(\zeta_{c,i}(t+1) - \hat{\beta}_{c,i}(t)\right) \quad (16)$$

$$\bar{\hat{\gamma}}_p(t+1) = \bar{\hat{\gamma}}_p(t) + a(\eta_p(t)) \times \left(\theta_{\tilde{s}_p, s_{p,2}}(t) - \bar{\hat{\gamma}}_p(t)\right) \quad (17)$$

$$\hat{\gamma}_p(t+1) = 1 - \bar{\hat{\gamma}}_p(t+1)$$

$$\bar{\hat{\gamma}}_0(t+1) = \bar{\hat{\gamma}}_0(t) + a(\eta_0(t)) \times \left(\theta_{\tilde{s}_2, s_{2,1}}(t) - \bar{\hat{\gamma}}_0(t)\right) \quad (18)$$

$$\hat{\gamma}_0(t+1) = 1 - \bar{\hat{\gamma}}_0(t+1)$$

5) For each class c, update $P_0(t+1)$ where for each (p,c), set $\tilde{p}_{cp}(t+1) = \delta_{r_{c_p^*},p}$, with $\{c_p^*\}$ are as defined in the theorem 2. using $\hat{\beta}(t+1)$ and $\hat{\gamma}(t+1)$.

In the next theorem, we show that the above greedy algorithm converges to the optimal menu where the menu contains a single page. This result can be easily extended to prove the convergence of OGA for a menu containing multiple pages. The proof of the extension is avoided to keep the explanations simple.

Theorem 3. Let $\mathcal{I}$ denote the item set with $0 < \beta_i < 1$ for all $i \in \mathcal{I}$ and assume $0 < \gamma < 1$. Let P^* be the permutation matrix corresponding to the optimal menu order. Then, we have the following:

(i) With probability one, there exists a customer index t^* such that$P(t) = P^*$, for all $t \geqslant t^*$.

(ii) Further, the derived time-averaged revenue $\overline{W}(t)$, converges to optimal expected revenue with probability one: $\overline{W}(t) \to W^*$, where,

$$\overline{W}(t) := \frac{1}{t}\sum_{t' \leqslant t}\sum_{k=1}^{n} \zeta_{s_k(t')}(t') w_{s_k(t')}. \qquad \diamondsuit$$

Proof: Appendix (VII-C). ■

Remark: The above theorem highlights the strength of OGA. It gives assurance that OGA will eventually learn all the customer parameters and converge to the optimal menu, irrespective of the noise present in the samples collected for parametric learning.

We are now interested in knowing the number of estimates required to achieve convergence with a given probability. For a given $\alpha > 0$, the following proposition proves the existence of a time index t_0 (or customer index) beyond which the algorithm converges to the optimal solution with at least probability $1 - \alpha$.

Proposition 4. For every $\alpha > 0$, there exists $t_0 < \infty$, such that the OGA converges to the optimal menu, for all $t \geqslant t_0$, with probability $1 - \alpha$. Here t_0 is solution of:

$$[2 - (1 - d_i + d_i \exp(-\frac{d^2/2}{K^2}))^t - (1 - q + q\exp(-\frac{d^2/2}{K^2}))^t]_0 \geqslant 1 - \alpha,$$

with $[x]_0 = \max\{0, x\}$, $d = \min_{i,j \in \mathcal{I}; i \neq j} \mid h(i;\Theta) - h(j;\Theta) \mid$ and $q = 1 - \max_{i \in \mathcal{I}} \beta_i$. In other words, for every $\alpha > 0$, there exists a finite $t_0 > 0$ such that:

$$P[P(t) = P^*, \text{for all } t \geqslant t_0] \geqslant 1 - \alpha. \qquad \diamondsuit$$

Proof: Appendix (VII-D). ■

V. Numerical Experiments

A. Perturbation Analysis

In this section, we compare the optimal expected revenue for a given system and the expected revenue obtained by the optimal ordering of the equivalent LoA system. We compute both the Monte-Carlo estimate and the theoretical value for the expected revenue and plot them. For the Monte-Carlo estimates, we averaged the revenue obtained over 1000 customers. The theoretical value was obtained using numerical computations. For a given ϵ and policy Π, the corresponding revenue can be obtained by solving for $R^1_\epsilon(1;\Pi)$ in (1). The theoretical benchmark value was obtained by going through each possible permutation of menus and picking the policy which gives the highest revenue. The theoretical value for h-index menu was obtained by using the h-index policy (say Π_h) for $\epsilon = 0$ (LoA system) to calculate $R^1_\epsilon(1;\Pi_h)$ for different values of ϵ. The graphs in Fig. 2. show the plot for expected revenue for a system with three classes having 4, 3 and 3 items. We have also included the plot for the expected revenue on using a price-based menu.

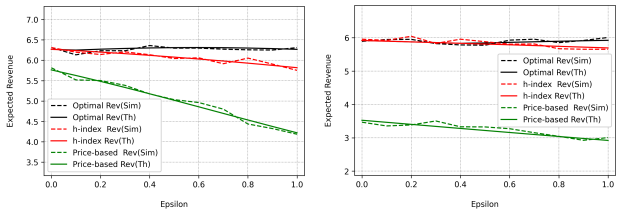

Fig. 2: $C = 3$; $\gamma_0 = 0.5$; $\gamma = 0.1$ and 0.5 for the first and second graphs respectively

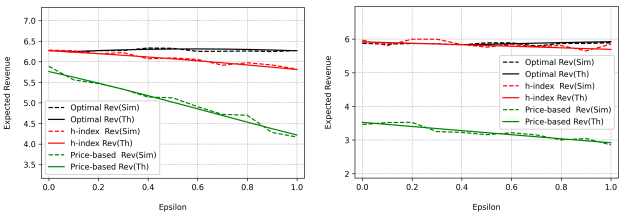

Fig. 3: $C = 3$; $\gamma_0 = 0.5$; $\{\gamma_p\} = \{0.1, 0.2, 0.3\}$ and $\{0.5, 0.6, 0.7\}$ for the first and second graphs respectively

We take $\gamma_0 = 0.5$ and consider two instances of this system with different levels of user patience within the

pages, specifically $\gamma = 0.1$ (in the left sub-figure of Fig. 2) and 0.5 (in right sub-figure). The attraction probabilities and prices in both are fixed and are inversely proportional. We observe that as the quitting probability γ increases, the h-index menu gives better approximations for the optimal revenue for larger range of ϵ (in right sub-figure). As we decrease the quitting probability, the menu obtained by h-index gives good approximations for lower range of ϵ. The percentage revenue loss at $\epsilon = 1$ for first and second graphs in Fig. 2. is 7.448% and 3.9339% respectively. The green plot shows the performance of a price-based menu, which is significantly inferior.

We also observe the performance of h-index menu in a system with different γ_p for different pages, in particular with increasing values of γ_p over the pages. Fig. 3 shows the performance of the same h-index menus as in Fig. 2 (by taking $\gamma = 0.1$ and 0.5), except that now we have the quitting probabilities $\gamma_p = 0.1, 0.2$ and 0.3 for the first system (in the left sub-figure of Fig. 3) and $\gamma_p = 0.5, 0.6$ and 0.7 for the second system (in the right sub-figure of Fig. 3). The percentage revenue loss at $\epsilon = 1$ for first and second graphs in Fig. 3. is 7.5488% and 3.9534% respectively. We observe that small changes of γ_p across different pages do not severely affect the performance of h-index menu. Price based menu is once again significantly inferior. Fig. 4 (left sub-figure) portrays another example

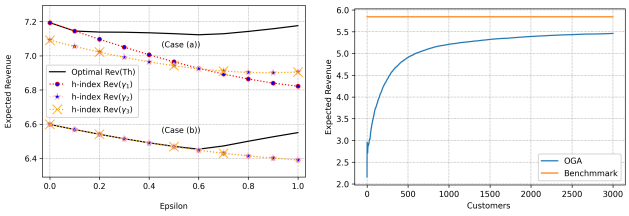

Fig. 4: (left) $\gamma_0 = 0.3$; (right) $\gamma = 0.4, \gamma_0 = 0.5$

with $C = 3$, and same number of items as before in each class. It contains the plots of theoretical values and not the Monte-Carlo Estimates. We take two different sets of γ_p. Three h-index menus are generated for each of the two cases, by using γ_1, γ_2 and γ_3 to create the menu. In this example, the level of user patience changes more significantly along the pages. For case (a), $\gamma_p = \{0.2, 0.4, 0.7\}$. The percentage revenue loss at $\epsilon = 1$ is $5.059\%, 3.839\%$ and 3.8588% for the three created menus. However, the average percentage revenue loss is $2.336\%, 2.577\%$ and 2.585% respectively. For case (b), $\gamma_p = \{0.4, 0.7, 0.9\}$. The percentage revenue loss at $\epsilon = 1$ is 2.474% while the average percentage revenue loss is 0.594% and both values change negligibly for other values of γ_p. We observe that for lower levels of user patience (case (b)), all three h-index policies perform optimally for a significant range of ϵ. For case (a), h-index policy corresponding to γ_1 performs optimally for a smaller range of ϵ. h-index policies corresponding to γ_2 and γ_3 do not perform optimally for any ϵ. We observe that as user patience gets higher, h-index policy may perform sub-optimally. However, such scenarios are less likely to be observed.

B. *Finite-Time Convergence Rate of OGA*

The graph in Fig. 4 (right sub-figure) shows a sample path of the time averaged revenue for the Online-Greedy algorithm in LoA system. We take $C = 3$ and number of customers $= 3000$. The 3 classes have $7, 7$ and 6 items respectively. The initial policy is taken to be a price-based policy and over-time as the parameter estimates get better, we start converging to the h-optimal policy. The graph portrays the result proved in theorem 3 regarding the convergence of the policy given by OGA to the optimal policy. The expected revenue received by implementing OGA seems to asymptotically converge to the benchmark value. Even from the point of view of finite-time analysis, it can be observed that after 3000 customers, the percentage revenue loss is 7.017% when using the policy obtained by OGA.

VI. Conclusion

In this paper, we extend the results of [5] where authors focus on a Slated cascade model. We drop the assumption that customer will scan the items from top to bottom as assumed in [5] and introduce a variation by considering classes of items. We find the optimal ordering policy for LoA browsing and prove its optimality even for instances of browsing which are not LoA. We propose a learning algorithm that learns the model parameters efficiently and derive mathematical results regarding its convergence to the optimal ordering policy. Moreover, experiments show that the h-index policy performs well in general browsing scenarios especially when user patience is not high, which is highly likely.

References

[1] Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: Proceedings of the 2008 international conference on web search and data mining (2008)

[2] Dhounchak, R., Kavitha, V., Altman, E.: A viral timeline branching process to study a social network. In: 2017 29th International Teletraffic Congress (ITC 29). vol. 3. IEEE (2017)

[3] Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American economic review **97**(1) (2007)

[4] Hargreaves, E., Agosti, C., Menasché, D., Neglia, G., Reiffers-Masson, A., Altman, E.: Biases in the facebook news feed: a case study on the italian elections. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE (2018)

[5] Kempe, D., Mahdian, M.: A cascade model for externalities in sponsored search. In: International Workshop on Internet and Network Economics. Springer (2008)

[6] Kveton, B., Szepesvari, C., Wen, Z., Ashkan, A.: Cascading bandits: Learning to rank in the cascade model. In: International Conference on Machine Learning (2015)

[7] Reiffers-Masson, A., Hargreaves, E., Altman, E., Caarls, W., Menasché, D.S.: Timelines are publisher-driven caches: Analyzing and shaping timeline networks. ACM SIGMETRICS Performance Evaluation Review **44**(3) (2017)

[8] Reiffers-Masson, A., Hayel, Y., Altman, E., Martel, G.: A generalized fractional program for maximizing content popularity in online social networks. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE (2018)

[9] Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Recommender systems handbook. Springer (2011)

[10] Varian, H.R.: Position auctions. international Journal of industrial Organization 25.6 (2007)

VII. Appendix

A. Proof of theorem 1

Consider $\Pi \in \Delta P$. We begin by showing the continuity of $f(\Pi, \epsilon)$. Rewriting the system of equations in (1) in the vector form, we have:

$$(I - \mathbf{BM}^{\epsilon}(\Pi)) \begin{bmatrix} R^1_{\epsilon}(1;\Pi) \\ R^1_{\epsilon}(2;\Pi) \\ \vdots \\ R^C_{\epsilon}(I_{\tilde{s}_C};\Pi) \end{bmatrix} = \begin{bmatrix} \alpha_{1,1}\omega_{1,1} \\ \alpha_{1,2}\omega_{1,2} \\ \vdots \\ \alpha_{C,I_{\tilde{s}_C}}\omega_{C,I_{\tilde{s}_C}} \end{bmatrix}$$

where $\mathbf{B}$ is an $\overline{I} \times \overline{I}$ diagonal matrix, having entries $(1-\alpha_{p,k})$ for $1 \leqslant p \leqslant C, 1 \leqslant k \leqslant I_{\tilde{s}_p}$. For the sake of understanding, the explicit form of $\mathbf{M}^{\epsilon}(\Pi)$ where Π assigns a class having 4 items on page 1 and a class of 3 items on page 2 is given below: .

$$\begin{bmatrix} 0 & \bar{\gamma}_1\bar{\epsilon}^1_1 & \bar{\gamma}_1 m^1_{1,3} & \bar{\gamma}_1 m^1_{1,4} & 0 & 0 & 0 \\ \bar{\gamma}_1 m^1_{2,1} & 0 & \bar{\gamma}_1\bar{\epsilon}^1_2 & \bar{\gamma}_1 m^1_{2,4} & 0 & 0 & 0 \\ \bar{\gamma}_1 m^1_{3,1} & \bar{\gamma}_1 m^1_{3,2} & 0 & \bar{\gamma}_1\bar{\epsilon}^1_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{\gamma}_0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{\gamma}_2\bar{\epsilon}^2_1 & \bar{\gamma}_2 m^2_{1,3} \\ 0 & 0 & 0 & 0 & \bar{\gamma}_2 m^2_{2,1} & 0 & \bar{\gamma}_2\bar{\epsilon}^2_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

To establish the continuity of $R^p_{\epsilon}(k;\Pi)$, we first need to show that $(I - \mathbf{BM}^{\epsilon}(\Pi))^{-1}$ exists. Towards this, consider the continuity of $(I - \mathbf{BM}^{\epsilon}(\Pi))^{-1}$ at $\epsilon = 0$ and observe that $\mathbf{BM}^0(\Pi)$ has non-zero elements only in positions $(i, i+1)$ with $1 \leqslant i \leqslant \overline{I} - 1$ and hence $(I - \mathbf{BM}^0(\Pi))^{-1}$ exists as $Det((I - \mathbf{BM}^0(\Pi))^{-1}) \neq 0$. As $\epsilon \to 0$, $\mathbf{BM}^{\epsilon}(\Pi) \to \mathbf{BM}^0(\Pi)$ for any given Π. As the determinant is a continuous function, there exists an $\hat{\epsilon}$ such that $(I - \mathbf{BM}^{\epsilon}(\Pi))^{-1}$ exists $\forall \epsilon \in [0, \hat{\epsilon}]$ and is continuous in ϵ. Since there are finitely many Π, one can choose a common $\hat{\epsilon}$ for all Π. Thus, $R^p_{\epsilon}(k;\Pi)$ is continuous $\forall \epsilon \in [0, \hat{\epsilon}]$ for every possible p, k, Π. Specifically, $R^1_{\epsilon}(1;\Pi)$ is continuous, i.e., $f(\Pi, \epsilon)$ is continuous $\forall \epsilon \in [0, \hat{\epsilon}]$, for all Π.

Finally, let $\Pi^*_0 \in \Pi^*(0)$, be any optimizer at $\epsilon = 0$. Then,

$$f(\Pi^*_0, 0) - \max_{\Pi \in \Delta P, \Pi \notin \Pi^*(0)} f(\Pi, 0) > 0.$$

Define the following function of ϵ,

$$g(\epsilon) := f(\Pi^*_0, \epsilon) - \max_{\Pi \in \Delta P, \Pi \notin \Pi^*(0)} f(\Pi, \epsilon). \tag{19}$$

We know that $f(\Pi^*_0, \epsilon)$ and $f(\Pi, \epsilon)$ are both continuous for $\epsilon \in [0, \hat{\epsilon}]$, and max function applied over a continuous function is continuous. Therefore, $g(\epsilon)$ is continuous $\forall \epsilon \in [0, \hat{\epsilon}]$. Using $g(0) > 0$ and the continuity of g, there exists another $\bar{\epsilon} \leqslant \hat{\epsilon}$ such that $g(\epsilon) \geqslant 0$ for all $\epsilon \leqslant \bar{\epsilon}$.

Thus from (19), the policy Π^*_0 is also optimal for system with any $0 \leqslant \epsilon \leqslant \bar{\epsilon}$. ♢

B. Proof of theorem 2

For any policy, $\Pi = (P_0, P_1, \ldots, P_C)$, let:

$$V(\Pi) \quad = \quad f(\Pi, 0) = \sum_{c=1}^{C} W_c(\Pi) \tag{20}$$

and let the conditional revenue for any class c, given ordering P_c, be denoted as:

$$G_c(P_c) = \sum_{i \in \mathcal{I}_c} \beta_{c,i} w_{c,i} \prod_{k=1}^{r_{c,i}-1} (1-\gamma)(1 - \beta_{c,s_{\tilde{r}_c,k}})$$

Consider an optimal policy for any class c,

$$P^*_c \in \arg\max_{P_c} G_c(P_c), \text{ and let } W^*_c := \max_{P_c} G_c(P_c). \tag{21}$$

We first prove the following:

$$V(\Pi) \leqslant V([P_0, P^*_1, \cdots, P^*_C]) \tag{22}$$

Using (5) and (20), we have:

$$V(\Pi) \quad = \quad \sum_{i \in \mathcal{I}_c} q_{c,i}(\Pi) w_{c,i}$$

Substituting the value of $q_{c,i}$ from (6) and taking $\gamma_p = \gamma$ for all pages, we get the following:

$$V(\Pi) \quad = \quad \sum_{c=1}^{C} \sum_{i \in \mathcal{I}_c} w_{c,i} \bar{\gamma}_0^{(\tilde{r}_c - 1)} \textstyle\prod_{p=1}^{\tilde{r}_c - 1} \kappa_{\tilde{s}_p} \bar{\gamma}^{I_{\tilde{s}_p} - 1} \beta_{c,i} \prod_{k=1}^{r_{c,i}-1} (1-\gamma)(1 - \beta_{c,s_{\tilde{r}_c,k}})$$

$$\Rightarrow V(\Pi) = \sum_{c=1}^{C} \bar{\gamma}_0^{(\tilde{r}_c - 1)} \textstyle\prod_{p=1}^{\tilde{r}_c - 1} \kappa_{\tilde{s}_p} (1-\gamma)^{I_{\tilde{s}_p} - 1} \; G_c(P_c),$$

From the above equation, it can be easily observed that for any $\Pi = (P_0, P_1, \ldots, P_C)$, (22) holds true. We emphasize that the assumption $\gamma_p = \gamma$ for all pages p is needed for the above inequality to hold true. As the set of all permutations is finite, We can directly take the max on both sides in (22) to get:

$$\max_{\Pi} V(\Pi) \quad \leqslant \quad \max_{P_0} V([P_0, P^*_1, \cdots, P^*_C]) \tag{23}$$

$$\Rightarrow \max_{\Pi} V(\Pi) \quad = \quad \max_{P_0} V([P_0, P^*_1, \cdots, P^*_C]) \tag{24}$$

(24) is true because $\Pi^* := [P^*_0, P^*_1, \cdots, P^*_C]$ can achieve the equality where P^*_0 optimizes the right hand side. Finally, [5, Lemma 1] can be used to solve the optimization problem (21) for each c. This proves part (i) and (ii) of the theorem. Further $V^* := \max_{\Pi} V(\Pi)$ equals

$$V^* \quad = \quad \max_{P_0} \sum_{c=1}^{C} (1-\gamma_0)^{(\tilde{r}_c - 1)} \textstyle\prod_{p=1}^{\tilde{r}_c - 1} \kappa_{\tilde{s}_p} \bar{\gamma}^{I_{\tilde{s}_p} - 1} W^*_c,$$

which can be rewritten in the following way, that facilitates using [5, Lemma 1] again now across the pages,

$$V^* \quad = \quad \max_{P_0} \sum_{c=1}^{C} w_c \beta_c \prod_{p=1}^{\tilde{r}_c - 1} [(1 - \beta_{\tilde{s}_p})(1 - \gamma_0)], \tag{25}$$

where the consolidated terms $\{w_c, \beta_c\}$ are defined in equation (11) in the theorem. Thus, part (iii) and (iv) of the theorem follow by applying [5, Lemma 1] to (25). ♢

C. Proof of theorem 3

Since we only have a single page, we use all the previously defined notations without the subscript c, for simplicity. For any item i, let p_i be the minimum probability of a typical customer visiting item i, among all the possible permutation matrices, i.e.,

$$p_i \quad := \quad \min_{P\in\mathcal{P}} q_i(P)/\beta_i, \tag{26}$$

where $q_i(P)$ is equivalent to (7). Clearly $p_i > 0$ for all i, since for all i, $0 < \beta_i < 1$, $0 < \gamma < 1$ and the number of permutations $|\mathcal{P}|$ is finite. One can lower bound $v_i(n)$ by (coupled) binomial random variable $Bin(n, p_i)$ which converges to ∞ as $n \to \infty$ with probability one (w.p.1). Therefore, $v_i(n) \to \infty$ as $n \to \infty$ w.p.1 and hence by strong law of large numbers, all the beta estimators converge to their respective true values w.p.1. The same is the case with estimator of γ. The h-index ordering given by (7) are defined using continuous functions of $[\beta_l]_l$ and γ, hence there exists a neighbourhood of true values $(\gamma, [\beta_l]_l)$, for which the menu chosen using h-index rule (7) in *Update* step of OGA equals that given by the optimal one of the theorem 2. This proves part (i) of the theorem. Part (ii) immediately follows from part (i), once again by law of large numbers (applied to iterates after t*). ♢

D. Proof of proposition 4

Approach: The proof of the theorem is a sophisticated application of the Hoeffding's inequality after bounding the distribution mean of the number of observations/samples for all the items from below.

We denote $f(w, \beta, \gamma) = \frac{w\beta}{1-(1-\gamma)(1-\beta)}$ which is a Lipschitz function where we let K be the associated Lipschitz constant. We define $\overline{M}_{\beta_i}(v_i(t)) = \beta_i - \hat{\beta}_i(t)$ and $\overline{M}_\gamma(\eta(t)) = \gamma - \hat{\gamma}(t)$ as the noise observed while estimating the parameters for customer preference and patience level respectively.
For every $i \in \mathcal{I}$, the following implies that:

$$\begin{aligned} & \mid f(w_i, \beta_i, \gamma) - f(w_i, \hat{\beta}_i(t), \hat{\gamma}(t)) \mid \\ = \quad & \mid f(w_i, \beta_i, \gamma) \\ & -f(w_i, \beta_i - \overline{M}_{\beta_i}(v_i(t)), \gamma - \overline{M}_\gamma(\eta(t))) \mid \\ \leqslant \quad & K(\mid \overline{M}_{\beta_i}(v_i(t)) \mid + \mid \overline{M}_\gamma(\eta(t)) \mid = g_i, \text{ say.} \end{aligned}$$

Now, we have to recall the following fact: For any $x_1, y_1, x_2, y_2 \in \mathbb{R}$, we have $x_1 + x_2, y_1 + y_2$ sharing the same ordering relation as x_1, y_1 i.e. $x_1 \geqslant y_1 \implies x_1 + x_2 \geqslant y_1 + y_2$ (or $x_1 \leqslant y_1 \implies x_1 + x_2 \leqslant y_1 + y_2$) if $\mid x_2 - y_2 \mid \leqslant \mid x_1 - y_1 \mid$. We do not prove this fact due to its simplicity.

We aim to show that for all the items of the set $\mathcal{I}$, if the difference in the estimators for the $h(.)$ are bounded, then the menu ordering obtained through the estimators coincides with the optimal ordering. Therefore, if for every i, $g_i < d$, then the optimal order is preserved. Consider the sequences of noises $\{\overline{M}_{\beta_i}(v_i(t))\}_{t\geqslant 1}$ and $\{\overline{M}_\gamma(\eta(t))\}_{t\geqslant 1}$. We aim to prove that there exists t_o such that with at least probability $1-\alpha$, for every $t > t_o$ we have the sequence of estimators $\{\hat{\beta}_i(t)\}_{t\geqslant t_0}$ and $\{\hat{\gamma}_{t\geqslant t_0}\}$ preserving the optimal ordering.

$$P[\cap_i g_i \leqslant d] \geqslant 1 - \alpha \tag{27}$$

for all $t \geqslant t_o$. If $\alpha_i \leqslant \frac{\alpha}{I}$, with $P[g_i \leqslant d] \geqslant 1 - \alpha_i$, then by Frechet inequality, we know that:

$$P[\cap_i g_i \leqslant d] \geqslant [\sum_i (1 - \alpha_i) - (I - 1)]_0 \geqslant \ [1 - \alpha]_0,$$

with $[x]_0 = \max\{0, x\}$. By using Frechet inequality, we have reduced the problem to finding $t_0 > 1$, such that for all $t > t_0$:

$$P[g_i \leqslant d] \geqslant 1 - \frac{\alpha}{I}. \tag{28}$$

Now, we use Hoeffding's inequality to bound the above probability for a fixed value of $v_i(t)$. After taking expectation on both the sides and bounding the mean of $v_i(t)$ by d_i from below, we get:

$$P[K \mid \overline{M}_{\beta_i}(v_i(t)) \mid \leqslant d_1] \geqslant 1 - (1 - d_i + d_i \exp(-\frac{2d_1^2}{K^2}))^t.$$

By Hoeffding's Inequality, we have:

$$P(|\overline{X} - E[\overline{X}]| \geqslant t) \leqslant exp\left(\frac{-2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Now, applying the Hoeffding on $v_i(t)$ first, we have:

$$\begin{aligned} & P[K \mid \overline{M}_{\beta_i}(v_i(t)) \mid \leqslant d_1] \\ & \qquad \geqslant \quad 1 - exp\left(\frac{-2(v_i(t))^2t^2}{\sum_{i=1}^{v_i(t)}(1-0)^2}\right) \\ & \qquad = \quad 1 - 2exp\left(-2(v_i(t))(d_1/K)^2\right) \end{aligned}$$

Taking expectation on both the sides, we get that:

$$\begin{aligned} & P[K \mid \overline{M}_{\beta_i}(v_i(t)) \mid \leqslant d_1] \\ & \qquad \geqslant \quad 1 - 2 * E[exp\left(-2t^2 v_i(t)\right)] \\ & \qquad = \quad 1 - 2 * \sum_{j=0}^{n} exp\left(-2t^2 j\right) (d_i^j)(1 - d_i)^{n-j} \\ & \qquad = \quad 1 - 2 * \sum_{j=0}^{n} ((d_i * exp(-2t^2))^j (1 - d_i)^{n-j}) \\ & \qquad = \quad 1 - 2 * (d_i * exp(-2(d_1/K)^2) + 1 - d_i)^t \end{aligned}$$

Note that for $d_1 + d_2 = d$, we have :

$$\begin{aligned} & P[K(\mid \overline{M}_{\beta_i}(v_i(t)) \mid + \mid \overline{M}_\gamma(\eta(t)) \mid)] \leqslant d] \\ = \quad & P[K(\mid \overline{M}_{\beta_i}(v_i(t)) \mid)] \leqslant d_1, K(\mid \overline{M}_\gamma(\eta(t)) \mid)] \leqslant d_2] \\ \geqslant \quad & [[P[K(\mid \overline{M}_{\beta_i}(v_i(t)) \mid) \leqslant d_1] \\ & \qquad + P[K(\mid \overline{M}_\gamma(\eta(t)) \mid)] \leqslant d_2] - 1]_0 \end{aligned}$$

$$\geqslant [1-(1-p_i+d_i \exp(-\tfrac{2d_1^2}{K^2}))^t + 1-(1-q+q \exp(-\tfrac{2d_2^2}{K^2}))^t]_0.$$

(by Hoeffding's inequality)

where $q = 1 - max_i \beta_i$. Therefore by taking $d_1 = d_2 = \frac{d}{2}$, and by defining t such that

$$[2 - (1 - p_i + d_i \exp(-\frac{d^2/2}{K^2}))^t - (1 - q + q \exp(-\frac{d^2/2}{K^2}))^t]_0 \geqslant 1 - \alpha.$$

we can conclude that (27) is satisfied. $\diamondsuit$