Online Search and Product Rankings: A Double Logit Approach*

Giovanni Compiani	Gregory Lewis	Sida Peng	Peichun Wang
Chicago Booth	Amazon	Microsoft	Unity

June 14, 2022

Abstract

We develop a flexible yet tractable model of consumer search and choice, and apply it to the problem of product rankings optimization by online retail platforms. In the model, products are characterized by a search index, which governs what consumers search; and a utility index, which governs which of the searched options is purchased. We show that this framework generalizes several commonly used search models. We then consider how platforms should assign products to search ranks. To optimize consumer surplus, platforms should facilitate product discovery by promoting "diamonds in the rough," products whose utility index exceeds their search index. By contrast, to maximize static revenues, the platform should favor high-margin products, creating a tension between the two objectives. We develop computationally tractable algorithms for estimating consumer preferences and optimizing rankings, and we provide approximate optimality guarantees in the latter case. When we apply our approach to data from Expedia, our suggested ranking achieves both higher consumer surplus and higher revenues than is achieved by the Expedia ranking, and also dominates ranking the products in order of utility.

Keywords: Consumer search, online platforms, product rankings.

*We are grateful for discussions with Bart Bronnenberg, Jean-Pierre Dubé, Pedro Gardete, Rafael Greminger, Stephan Seiler, Adam Smith, Raluca Ursu, Giorgos Zervas, as well as participants at the 2020 Marketing Science Conference, Netflix, UCL and Tilburg. This project was started when Giovanni was an intern at Microsoft Research New England.

1 Introduction

E-commerce is an important part of the economy, and becoming more so. As a result, platforms such as Amazon, Google and Expedia, play an increasingly crucial role in shaping consumer choices. One key tool at their disposal is product rankings, which have the ability to direct consumer attention and thus influence purchasing choices. A growing literature explores the impact that rankings have on choices (e.g., Yao and Mela (2011), Athey and Ellison (2011), Ghose et al. (2014), Chen and Yao (2017), Ursu (2018), Hodgson and Lewis (2020)). Since rankings are often found to be an important driver of consumer decisions, it is natural to investigate what is the best way to rank products on a platform. Moreover, to propose and evaluate rankings requires a unified model of consumer search and choice that can be taken as the ground truth for counterfactual analyses.

Given this, the current paper has three goals. First, we introduce a tractable and easy-to-understand model of search that generalizes several models in the economics and marketing literatures. Second, we use this model to propose algorithms for optimizing consumer surplus and platform revenue that offer formal approximation guarantees.[1] Third, we estimate our model on data from Expedia and show how our algorithms compare to competing rankings.

Our main idea is to view consumer search through the lens of two indices: a search index and a utility index. The search index governs what consumers search and therefore consider, whereas the utility index determines which of the considered products is actually bought. Rankings affect the search index, but not the utility index. This model is meant to capture the main features of the online purchasing funnel: viewing, clicking and purchasing. Viewing depends on the rankings — more prominent search results will be viewed more often. Clicks often depend on product characteristics and/or ad text. The viewing and clicking stages of the process are controlled by our search index.[2] Purchasing depends on the consumer's assessment of the product after clicking through and seeing the detailed information on the product page. In our model, this is captured by their learning the utility index.

We describe consumer behavior by an algorithm: consumers consider items in descending order of search index and terminate search if at any point the best product they have found thus far (i.e., the one with the highest utility index) has a higher utility index than the search index of the next product to be searched. They then choose the good that has the highest utility index among the goods they have searched — their consideration set. This is

[1] De los Santos and Koulayev (2017) propose a utility-based ranking that maximizes click-through rates.

[2] We do not separately model viewing and clicking because our data only contains click data, and not the eye tracking data necessary to disentangle the two.

exactly the same algorithm as in the canonical sequential search model of Weitzman (1979), but as we show, it also encompasses a variety of other search protocols. Building on a result by Choi et al. (2018), we provide a unified characterization of purchase behavior under this model, showing that the product ultimately chosen takes a maximin form: it is the product with the highest *effective index*, where the effective index of a product is the minimum of its search and utility indices.[3]

In order to study optimal rankings, we assume that platforms are able to increase and decrease the search indices of products beyond their baseline levels, by either *promoting* them (giving them prominence on the search page) or *burying* them in the search results. By deciding what to rank and where to rank it, platforms can thus affect consumer choices.

We start by considering the problem of choosing rankings to optimize consumer surplus. Knowing what is best for consumers is important both for platforms with long-run (growth) objectives as well as regulators who want to govern these platforms. To gain intuition, we consider a relaxation of the problem where the platform can continuously adjust the search indices subject to a budget constraint. We find that the platform should act to equate the ex-post "potential" of all products, where ex-post potential is defined as the difference between the mean utility and search indices *after the search indices have been adjusted to account for ranking*. Ex-ante, a high potential product is one with high utility relative to search index, a "diamond in the rough." The product is unlikely to be viewed and chosen by consumers unless the platform promotes it; such promotion helps consumers make better choices. Contrast this with a product that has both high mean utility and search indices, i.e. a product that is well-known to be high quality, such as a branded item. This product will often be chosen by consumers, and so the platform need not use its limited space promoting it.

An immediate but nontrivial implication of this analysis is that ranking products from highest to lowest utility is not optimal for consumer surplus, because high utility products need not have high potential. An example would be a product from a top brand that has both high salience and utility, and needs no promotion to be purchased. In this case, it would be preferable to rank a relatively unknown equally high-quality product ("diamond in the rough") at the top. Derakhshan et al. (2020) find a similar result based on a different model

[3]A growing theoretical literature in marketing studies consumer search for product attributes. Branco et al. (2012) characterize the optimal search strategy of a consumer gradually learning about a single product, and explore its implications for the firm's optimal pricing problem (see also Branco et al. (2016) for insights into the optimal information provision strategy). Ke et al. (2016) extend the analysis to the case where consumers face multiple differentiated products.

of search as well as simulation evidence. In contrast, existing work that does not explicitly model the consumer search process finds that ranking by utility is always optimal (Ghose et al. (2012)). Another implication of the framework is that a product that is highly salient but offers low utility ("click bait") should optimally be buried in the rankings, since that frees up space to promote more worthy products.

The optimal ranking is different when the platform is concerned with revenue maximization.[4] In a world in which the consumer will buy some product regardless of the selection, the platform should optimally offer the consumer no choice at all, and only display a single product, the one on which they earn the highest revenue.[5] However, realistically, consumers can shop elsewhere. This competitive force drives the platform towards aligning their rankings with consumer preferences, so as to make a sale. We show that the optimal product assortment includes all products that have a (weakly) higher product revenue than the average product revenue, where the averaging includes the zero-revenue outside option. Further, the revenue-optimizing ranking will typically be different from the utility-maximizing ranking (Ursu and Dzyabura (2020)).

The continuous relaxation of the problem which we have described so far gives useful intuition, but in practice the assignment of products to ranks is a discrete optimization problem, and the heuristics from the continuous case cannot be blindly followed. For example, it may not be optimal to place the product with the highest ex-ante potential at the top of the search rankings, since if it has a low enough search index it may attract very few purchasers even when top-ranked, which is a waste of that position.[6]

We thus offer a pair of similar algorithms for maximizing consumer surplus and revenue. They combine an exhaustive brute-force search of the best ranking for the K highest positions (where K is a small number) with a greedy algorithm for ranking the remaining products. When most clicks and purchases go to products ranked in the top K positions, we can prove that this algorithm is close to optimal. What makes this problem technically challenging and non-standard is interference effects among units — ranking one product more highly means that it is more likely to be purchased, and every other product less so — so that the analysis is not neatly separable across products. Simulation experiments verify that these

[4] Since the marginal cost of making a sale are essentially zero for many platforms (sellers are responsible for the physical delivery of the good), revenues coincide with profits.

[5] The revenue the platform earns from selling each product will depend on their individual agreements with each seller, but will typically be a fixed percentage of the sale price.

[6] Placing products with low search indices and high utilities (i.e., high ex-ante potential) products in top positions may also have the unfortunate effect of diminishing consumer trust in the search algorithm if the consumers never sample them. Good recommendations are those that will be followed.

solutions work extremely well in the simulated environment, achieving well over 99% of the available gains from rankings (where available gains are defined as the difference between the consumer surplus under the optimal and the worst ranking, and similarly for revenue).

The second part of the paper takes our approach to the wild. We use two datasets from Expedia which contain information about both clicks and purchases. In one dataset (the "training" data), rankings were assigned at random, whereas in the other (the "testing" data) they were assigned according to Expedia's algorithm (these datasets were also used in Ursu (2018)). Looking at the training data, we show that while there is positive correlation between click and purchase rates, many products have high click rates but poor purchase rates ("click-bait") and others have low click rates but are often purchased conditional on being clicked ("diamonds in the rough"). This motivates having two distinct indices to describe each product. We then estimate the relationship of these indices to product characteristics and rankings under a specification in which each index has an independent extreme value shock ("double logit"). Identification is aided by the fact that we see both what the consumers clicked on and what they bought, which implies a set of orderings for the search and utility indices. The likelihood of each of these orderings, in turn, can be written in closed form using the "exploded logit" formula, facilitating estimation. We fit the model only on the training data, to avoid concerns related to the endogeneity of product ranks. We then assess the fit of the model by predicting search patterns on the testing data; the model matches the sharp decay in click rates that we see under the Expedia rankings quite well.

Finally, we take these estimates and simulate consumer behavior, search duration, consumer surplus, and platform revenue under different ranking algorithms. We compare our ranking algorithm optimized for consumer surplus for $K = 3$ (i.e., we brute force the top 3 positions, then proceed greedily) to a straight utility ranking, an algorithm optimized for revenue, and the Expedia ranking itself. We find that our algorithm optimized for consumer surplus dominates the Expedia algorithm under our model, raising consumer surplus by 78¢ per consumer, while still raising revenue by 14¢ per customer. Ranking directly by utility is less effective, raising consumer surplus by around 60¢ relative to Expedia and revenue by 2¢. These numbers are at the search impression level; since only 3-4% of customers purchase, one should scale the figures up by 25-30 times to obtain numbers per purchasing consumer.

Additional Related Literature. In addition to the papers cited above, our work is related to the theoretical literature on consumer search, including the canonical models of satisficing (Simon, 1955), non-sequential search (Stigler, 1961) and sequential search with recall (Weitzman, 1979). We show that our framework subsumes several of these models as special cases. This means that, subject to the parametric assumptions embedded in the empirical specification, our estimates are robust to different search protocols that consumers may follow. Abaluck and Compiani (2020) also estimate consumer preferences without committing to a specific search model. A related, but distinct literature in marketing has focused on the determinants of consumers' consideration sets (see, among others, Roberts and Lattin (1991); Mehta et al. (2003); Honka et al. (2017)). Our search model can be viewed as one way to microfound the consumer's decision of which goods to consider that is amenable to studying optimal rankings.

Our work is also related to the literature on platform design and recommendation systems. While we focus on product rankings, other papers study the customization of e-mail communications to customers (e.g., Ansari and Mela (2003)), or the amount and type of information to display on the results page (e.g., Gardete and Hunter (2020) and Gu and Wang (2021)).

Next, a large literature in operations research studies the assortment optimization problem — i.e., deciding which products to stock, or which ones to show to consumers as they search (see Kök and Fisher (2007) for a survey, and Jagabathula and Rusmevichientong (2017) and Agrawal et al. (2019) for more recent contributions). Unlike most of this literature, we focus on optimizing not only revenue, but also consumer surplus, and on the decision of how to rank products, as opposed to just which ones to offer.

Finally, a literature in computer science studies optimal rankings in the contexts of sponsored search. These papers typically assume stylized models of search, such as the cascade model (Aggarwal et al., 2008; Kempe and Mahdian, 2008), which are often highly parameterized (e.g., Karmaker Santu et al. (2017)). In contrast, our approach is based on a microfounded model of consumer behavior but is more computationally intensive.

Paper Structure. The paper proceeds in three parts. First, we introduce the double index search model in Section 2. Then, in Section 3, we discuss optimal rankings. Finally, we apply the approach to the Expedia data in Section 4 and conclude in Section 5.

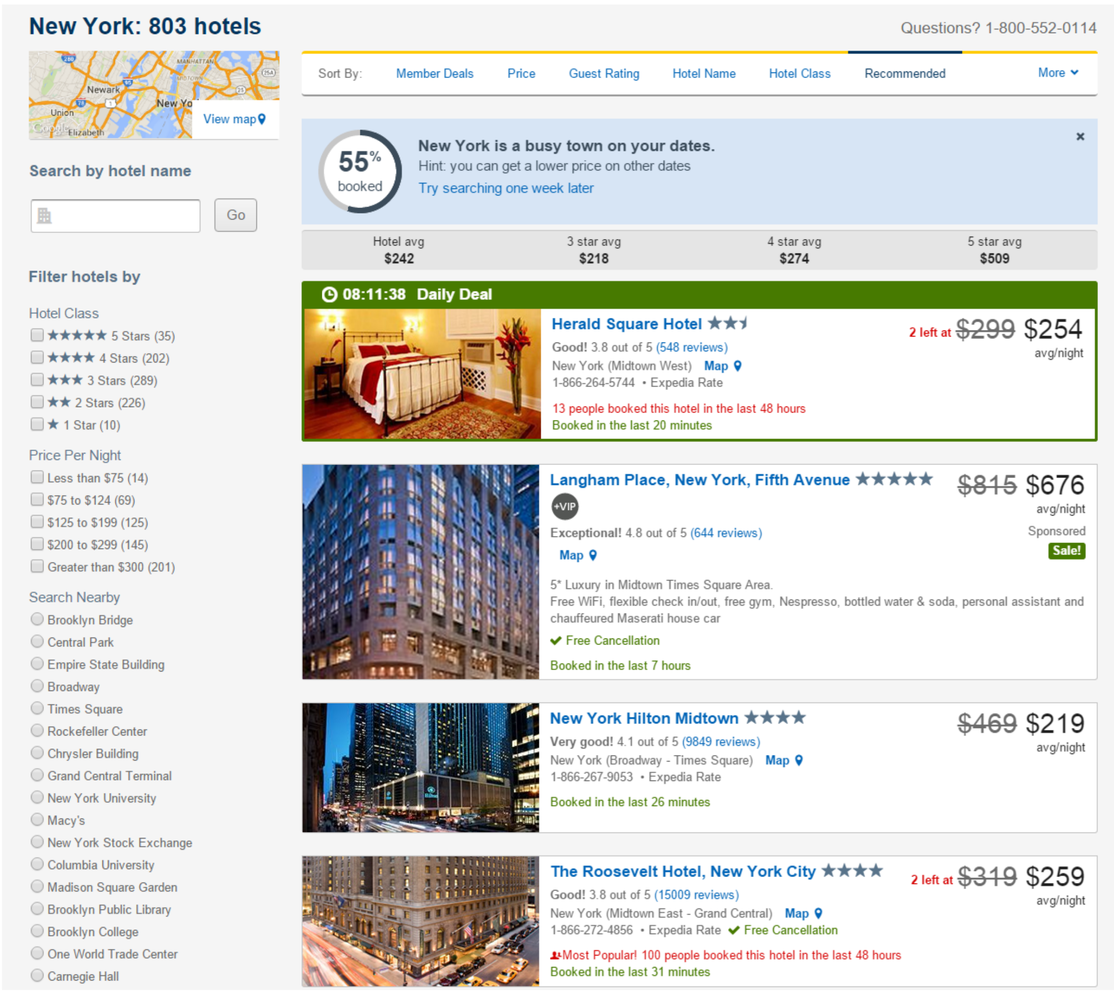


Figure 1: Search results page from Expedia

2 Model

2.1 Setting

We consider a setting in which consumers have unit demand for a good in a product category. They log on to an appropriate platform (e.g., Expedia), and enter a search query that describes that category (e.g., "hotel room in New York City on November 5th"). They are presented with a finite set of search results. These products are distinguishable by the *search characteristics* presented to the user.

Figure 1 shows an example of the search results on Expedia for the above hotel query. The search characteristics include the average price per night, photos, the number of reviews, and the number of rooms left. In the case of keyword search on Bing or Google, for example, the search characteristics would include whether a link is organic or sponsored, as well as the link text itself.

Based on what they see, the user may choose to either perform some other search operation (such as refining the query, or filtering the results); or abandon search; or click on one of the options. If they click, they are taken to a product page, where they may learn additional product characteristics (e.g., room amenities). They may then stop and purchase; or continue search; or abandon search without purchase. At the end of the process, they will either have picked the best option from those they considered (clicked on), or have chosen not to buy at all. We model this process using the following *double index model*.

2.2 Invididual behavior: A Double Index Model

A consumer has a need that may be met by purchasing a single product from a finite set of products $\mathcal{J} \equiv \{1, \ldots, J\}$ plus the outside option (denoted by 0). Each product is fully characterized by a search index, s_j, and a utility index u_j, and so the consumer faces the vector $(\mathbf{s}, \mathbf{u})$, where $\mathbf{s} = (s_0, s_1 \ldots s_J)$ and $\mathbf{u} = (u_0, u_1 \ldots u_J)$. At the outset of the search process, the consumer knows the search indices and the payoff of the outside option but the remaining payoffs $\{u_j\}_{j=1}^{J}$ are unknown.

Consumers learn u_j by searching for product j. Depending on the context, the u_j index could represent either the final consumption utility that the consumer derives from the good or (under risk neutrality) the expected utility based on the information on the detailed product description page. The search indices are to be interpreted as some combination of the visibility, salience and observable attractiveness of the product, i.e. they determine both product views and product clicks. In our application to hotels, organic product rankings are a key component of the search indices; in other settings, sponsored advertising may play an important role in generating views and clicks.

Let C be the set of products searched thus far, the *consideration set*. The consideration set initially contains only the outside option, but as products are searched, they are added to C. We assume that consumers employ the following search algorithm:

1. Search products in descending order of their search indices, terminating search whenever the highest available payoff $\max_{j \in C} u_j$ (weakly) exceeds the highest remaining search index $\max_{j \in \mathcal{J} \setminus C} s_j$.

2. Once search has terminated, choose the option with the highest utility index among those considered (possibly the outside option).

Since the outside option is always considered, we let $s_0 = \infty$. Further, we normalize its utility to zero, $u_0 = 0$. We offer a short example to illustrate the search process.

Example 1. *There are 3 products and the outside option, with payoffs $u_0 = 0$, $u_1 = 70$, $u_2 = 20$, and $u_3 = 40$, and search indices $s_0 = \infty$, $s_1 = 100$, $s_2 = 90$ and $s_3 = 60$ respectively. The consumer considers the outside option, with its payoff of zero. Then the consumer searches among the inside goods for the highest search index product. This is product 1. Since the payoff of 70 is less than the next highest search index of 90 (from product 2), they search product 2 and see a payoff of 20. Now, the next highest search index of 60 is less than the maximum available payoff of 70, so they stop search and purchase product 1.*

The decision to describe consumer behavior by an algorithm rather than as the solution to some optimization problem, though non-standard, has a long history in the search literature dating back to Simon (1955). The advantage for us is that the framework is general enough that many classic models proposed in the literature emerge as special cases. We explicitly provide a microfoundation based on the Weitzman (1979) model in section 2.4 below; for more examples, see Appendix A.1.

Our main result in this subsection is a lemma that characterizes the relationship between the search and utility indices and the product eventually purchased. This result is a minor modification of an existing result by Choi et al. (2018) for the Weitzman (1979) model, and we adopt their name for the result (see also Armstrong and Vickers (2015), Kleinberg et al. (2016) and Armstrong (2017)).

Lemma 1 (**Eventual Purchase**)**.** *A consumer facing a choice set consisting of $(\boldsymbol{s}, \boldsymbol{u})$, including an outside option with indices $(\infty, 0)$, will purchase a product $j \in \mathcal{J}^*$ where $\mathcal{J}^* = \arg\max_{j \in \mathcal{J}} v_j$, for $v_j = \min\{s_j, u_j\}$.*

Proof. See Appendix A.2. □

In words, it is the product with the highest minimum of search and utility indices that gets purchased. We will call this quantity $v_j = \min\{s_j, u_j\}$ the *effective index* of the product. Note that consumers only know u_j for the products they searched, but they still will end up choosing the good that maximizes the effective index across all products. One implication of

Lemma 1 is that products with high utility but low search indices will rarely be purchased. Further, the result provides a convenient way of calculating purchase probabilities.[7]

2.3 Aggregate Demand: A Double Logit Model

Having developed a model of individual search behavior, the next step in the analysis is to aggregate individual demand functions. We do this for two reasons. First, in order to analyze how platforms should optimally rank products, we have to be clear on what the platform knows and what the consumer knows. We will take the stance that the platform is better informed as to the mean utility offered by each product, as they observe many consumer choices and can thus infer which products offered the highest utility among those considered. On the other hand, we will assume that each consumer has idiosyncratic preferences that are private information. The distinction between mean utility and idiosyncratic shocks is only meaningful in the context of aggregate demand.

Second, we need to aggregate individual demands in order to define aggregate consumer surplus, which is the object to be optimized. For this task, we will use the familiar trick of assuming that the idiosyncratic shocks are drawn from the type-I extreme value distribution. In this case, since the model is built on two indices, the result is a double logit model. Further, we posit that the ranking of a product affects its search index (e.g., by making the product easier to find) but not its utility index. This is consistent with the findings of a growing empirical literature (e.g., Ursu (2018)).

The search and utility indices are assumed to take the following form:

$$\begin{aligned} s_{ij} &= \delta_j^S + f(r_j) + \varepsilon_{ij} \\ u_{ij} &= \delta_j^U + \varepsilon_{ij} \end{aligned} \tag{1}$$

where δ_j^S and δ_j^U are respectively the mean baseline search indices and mean utilities offered by each product, r_j is the rank of each product (a higher value indicates a more salient position), $f(r_j)$ is an increasing function, and ε_{ij} is distributed type–I extreme value. We normalize the mean utility of the outside option $\delta_0^U = 0$.

Consumers know all the components of the search indices at the time of search. They search to learn the payoffs offered by each product, which are unknown. Notice that the same logit

[7] A recent paper by Moraga-González et al. (2021) also uses this result to facilitate estimation of a search model.

error ε_{ij} enters both the search and the utility index, implying that consumers know the idiosyncratic part of their utility prior to search. This helps us in deriving a consumer surplus formula below, but we will relax it in the empirical section. What consumers learn through search is the mean utility δ_j^U. This is distinct from some other models in the marketing literature, in which mean utilities are known and consumers seek to learn their idiosyncratic match value (e.g., Kim et al. (2010) and Ursu (2018)). We would argue that whether consumers search to learn mean or idiosyncratic utility (or both) is context dependent. For example, in buying well-defined products, such as a new USB-C charger or a can opener on Amazon, consumers are mostly trying to learn quality (i.e., mean utility). In choosing a hotel — our application here — it seems plausible to us that consumers are trying to figure out both *"is this a good hotel?"* and *"is this a good hotel for me?"* during the search process (i.e., both mean and idiosyncratic utility). In this paper, we focus on the platform's ability to use product rankings to facilitate product discovery in contexts where the platform may have more information about mean utilities. Therefore, it is natural to model consumers as searching over the mean utilities.

In contrast, the platform is assumed to know both the mean baseline search index δ_j^S and mean utility δ_j^U, as well as the rankings function $f(\cdot)$. We think it is realistic to assume this given that platforms have access to rich data, including data from experiments, that will allow them to obtain good estimates of the quality and salience of each option. Indeed, in the empirical application, we propose one way to estimate these objects based on click and choice data, which are readily available to platforms.

Following Lemma 1, define the mean effective index, $\delta_j^V(r_j) \equiv \min\{\delta_j^S + f(r_j), \delta_j^U\}$. Define $\phi_j(r_j) \equiv \delta_j^U - \delta_j^S - f(r_j)$ to be the *potential* of product j — i.e., the difference between the mean utility and search indices. This potential is in part a measure of how much better the product is (on average) than it appears to be, though it is also affected by the rankings. Define the *ex-ante consumer surplus* to be the expected utility of a consumer on the platform, gross of any search costs, prior to the realization of their idiosyncratic logit shock.

Proposition 1 (Aggregate Demand and Consumer Surplus). *Let* $\mathbf{r} \equiv (r_1, \ldots, r_J)$*. The probability of a consumer choosing product* j *is given by*

$$P(\mathit{Choose}\ j) \equiv q_j(\mathbf{r}) = \frac{\exp \delta_j^V(r_j)}{1 + \sum_k \exp \delta_k^V(r_k)},$$

and the ex-ante consumer surplus is given by

$$CS \equiv C + \log\left(1 + \sum_j \exp \delta_j^V(r_j)\right) + \sum_{j:\phi_j(r_j)>0} q_j(\mathbf{r})\phi_j(r_j)$$

for C a constant.

Proof. See Appendix A.3. □

This proposition says, firstly, that the choice probabilities are determined by the mean effective indices according to the standard logit form. This is simply a consequence of Lemma 1 and the extreme-value distribution assumption. The consumer surplus, on the other hand, differs from the standard log-sum form, in that it has an additional term that binds only for products whose utility is higher than their search index ("diamonds in the rough").[8] The reason is that although *choices* are based on effective indices, consumer *surplus* is based on utility, and so whenever the utility exceeds the effective index this must be counted too (it can't be less than the effective index because of the min operator). This characterization of consumer surplus is a key building block in the analysis that follows.

Notice that in our double index model, there are no explicit search costs, and so consequently they do not appear in the ex-ante consumer surplus either. This assumption of zero search costs may be appropriate for the online environment, where the time costs of search are often small. But one may reasonably be concerned that these search costs are important in practice, and so in our application we will track the average number of clicks under different algorithms as an additional performance metric (in addition to consumer surplus and revenues). We also offer one potential microfoundation of the double index model that explicitly includes search costs in the next subsection.

2.4 Microfoundations: The Weitzman Model

The double logit model above abstracts from the consumer search problem, by describing their behavior by a heuristic. One might wonder if such behavior can be sustained as the solution of an optimal search problem. In this section, we show that it emerges from the

[8]This expression for consumer surplus arises in logit models whenever there is a distinction between anticipated and realized utility (Allcott, 2013; Train, 2015).

canonical sequential search model of Weitzman (1979).[9]

The primitives in the Weitzman (1979) model are payoff distributions and search costs. We begin with payoff distributions. Consumers obtain payoffs of:

$$u_{ij} = \delta_j^U + \varepsilon_{ij}$$

where the mean utility is unknown at the time of search but the idiosyncratic payoff is known. Consumers search to learn mean utilities. To map it more clearly to the empirical model below, it will be helpful to expand the mean utility as follows:

$$\delta_j^U = \beta^U x_j + \xi_j^U$$

where x_j are a set of product characteristics observed at the time of search, and ξ_j^U is a mean payoff component that is only learned by searching the product.

We assume that the mean utilities are conditionally mutually independent, where the conditioning is over the vector of all observed product characteristics. An implication is that the payoff realization for one product does not cause the consumer to update about the payoff distributions of the remaining products (for a model which relaxes this assumption, see Hodgson and Lewis (2020)). We complete the model by specifying that searching product j incurs some search cost c_j.

Then, it follows from Weitzman (1979) that consumers will optimally assign products search indices s_{ij} according to:

$$c_j = \int_{s_{ij}}^{\infty} (u - s_{ij}) f_{u_{ij}|x_j,\varepsilon_{ij}}(u)du \tag{2}$$

and search according to the exact heuristic we discuss above.

Next, we would like to show that these search indices take the double logit form. Towards this, assume further that the conditional distribution of ξ_j^U belongs to a location family: $\xi_j^U = \gamma x_j + \tilde{\xi}_j^U$ for some parameter vector γ and a mean-zero random variable $\tilde{\xi}_j^U$ with $\tilde{\xi}_j^U \perp x_j$. Then, the conditional mean is linear in the product characteristics: $E[\xi_j^U|x_j] = \gamma x_j$. From this, we can interpret γ as capturing the relationship between product characteristics observed on the search page, and those that are only observed after clicking through (e.g., price could act as a signal of quality).

[9]Papers that have estimated this model empirically include Kim et al. (2010), Honka and Chintagunta (2017), and Kim et al. (2017).

We can then simplify the right hand side of the above equation as follows:

$$\int_{s_{ij}}^{\infty}(u-s_{ij})f_{u_{ij}|x_j,\varepsilon_{ij}}(u)du = \int_{s_{ij}}^{\infty}(u-s_{ij})f_{\xi_j^U|x_j}(u-\beta^U x_j-\varepsilon_{ij})du \quad (3)$$

$$= \int_{s_{ij}}^{\infty}(u-s_{ij})f_{\tilde{\xi}_j^U}(u-\beta^U x_j-\gamma x_j-\varepsilon_{ij})du \quad (4)$$

$$= \int_{s_{ij}-\beta^U x_j-\gamma x_j-\varepsilon_{ij}}^{\infty}(y+\beta^U x_j+\gamma x_j+\varepsilon_{ij}-s_{ij})f_{\tilde{\xi}_j^U}(y)dy \quad (5)$$

where the first equality follows from the relationship between the conditional densities of u_j and ξ_j, the second from the location family assumption, and the third by a change of variable ($y = u - \beta^U x_j - \gamma x_j - \varepsilon_{ij}$). Let ρ_j be the solution of $c_j = \int_{\rho_j}^{\infty}(y-\rho_j)f_{\tilde{\xi}_j^U}(y)dy$, and let $\beta^S = \beta^U + \gamma$. Then if we let the search index take the form $s_{ij} = \beta^U x_j + \gamma x_j + \rho_j + \varepsilon_{ij} = \beta^S x_j + \rho_j + \varepsilon_{ij}$, we may subtitute into (5), and verify that s_{ij} indeed satisfies (2).

We make one final assumption to reach the double logit form. Suppose that search costs are determined by rankings, and decreasing in rank (i.e., high ranked products have low search costs). Then it follows that we can write the thresholds ρ_j as $\rho_j = f(r_j)$ for some unknown increasing function $f(\cdot)$. Putting this all together, we have $s_{ij} = \delta_j^S + f(r_j) + \varepsilon_{ij}$ with $\delta_j^S = \beta^S x_j$, as in the double logit model above.

3 Optimal Rankings

Online platforms can influence what is bought through their product rankings, which have powerful effects on product views and search indices. With this in mind, we now turn to the problem of optimizing those rankings, either for consumer surplus or platform revenue. In this section, we will consider two main objective functions for the platform: maximizing consumer surplus (an appropriate target for a platform trying to maximize the size of its user base), and maximizing revenue.[10] For any given search query, platforms can decide how to order the results that they return. Assuming that each search query corresponds to a fixed set of relevant products $\mathcal{J}$, the problem is then which of those products to rank and how to rank them. The platform may choose not to present all products to the consumer. This is equivalent to allowing the platform to set a product's rank to be 0, with $f(0) = -\infty$, so that this product is never considered. However, we rule out "gaps" in the ranking: if there

[10]We believe that the algorithms we develop extend to the case where the platform maximizes a convex combination of consumer surplus and revenue, but we have not formally analyzed this.

is a product ranked in position L, then positions $1 \ldots L-1$ must be filled. The analysis in this section is conditional on a search query and thus can seamlessly incorporate additional information that the platform might have on the consumer, delivering personalized rankings.

3.1 Optimizing Consumer Surplus

We want to match products to ranks in such a way as to maximize consumer surplus. This matching problem is discrete, and therefore not amenable to standard techniques. So we begin instead with a relaxation of the problem in which ranks can be assigned continuously, subject to a budget constraint that may hold with inequality (since the platform can choose not to list some products at all, leaving some "surplus" ranks). Let $f(\cdot)$ now be defined on the reals, with $f' \geq 0$ and let $CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)$ denote the consumer surplus as a function of the vectors of rankings $\mathbf{r}$ and indices $\delta^{\mathbf{S}}, \delta^{\mathbf{U}}$. Then, the problem can be written as:

$$\begin{aligned} &\max_{r_1 \ldots r_J} CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right) \\ &\text{subject to} \sum_{j\,:\,j \text{ is listed}} r_j \leq \frac{J(J+1)}{2} \end{aligned}$$

The optimal solution has the property that the partial derivative of the consumer surplus is equal for all products that are ranked: $\partial CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right) / \partial r_j = \lambda \ \forall \ j \in \mathcal{L}$ where $\mathcal{L}$ is the set of products listed and when all products are listed, λ is the Lagrange multiplier on the budget constraint. If it is feasible to choose ranks so that $\delta_j^S + f(r_j) \geq \delta_j^U \ \forall \ j$ (i.e., every product has a higher search index than their utility index), it is also optimal for consumer surplus. The reason is that then the mean effective indices, $\min\left\{\delta_j^S, \delta_j^U\right\}$, are ordered in the same way as mean utility, so that the highest utility products are most often purchased.

However, in most cases, it will not be possible to promote all products enough to achieve this, and some products will need to be prioritized. To see which products benefit most from higher rankings, take the derivative of consumer surplus with respect to r_j:

$$\frac{\partial CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)}{\partial r_j} = \begin{cases} q_j f'(r_j) \left(\phi_j(r_j) - \sum_{k:\phi_k(r_k)>0} q_k \phi_k(r_k)\right) & , \ \delta_j^S + f(r_j) < \delta_j^U \\ 0 & , \ \delta_j^S + f(r_j) \geq \delta_j^U \end{cases}$$

where again q_j denotes the choice probability of product j. We derive this expression in Appendix A.4. The intuition for it is in two parts. When $\delta_j^S + f(r_j) \geq \delta_j^U$, so that the effective index of j is determined by the utility, marginally improving the ranking of the product (i.e., increasing r_j) will not change the choice probabilities, and hence has no effect on consumer surplus. On the other hand, when $\delta_j^S + f(r_j) < \delta_j^U$, improving a product's ranking will increase its choice probability on the margin. Whether this is good or bad for consumers depends on the sign of the expression in parentheses, which relates the potential of product j to a weighted sum of all products with positive potential. If j offers positive potential, then it may be worth promoting this product, but it depends on the potential of the other products, since there is a finite amount of promotion available to the platform.

If $f'(\cdot)$ is large enough (i.e., rankings impact search to a sufficient degree) and if there is no outside option, i.e. $\sum_{k \leq J} q_k = 1$, it is possible to equate all the derivatives at zero, which is optimal. This is achieved by assigning better rankings to products with high baseline potential $\delta_j^U - \delta_j^S$ at the expense of products with low baseline potential. So the optimal ranking is based on equating product potential, rather than ordering by utility. The reason for this is that when consumers have to buy something, then only the relative effective indices between products matter. Equating potentials sets the effective indices equal to mean utilities, in which case consumers behave as though they were perfectly informed, thus maximizing consumer surplus. Instead, with an outside option, it is necessary to balance potential and choice probabilities (since not all consumers will buy). Consumer surplus is maximized in this case by equating all the positive derivatives.

The argument so far has treated rankings as continuous. However, in practice $f(\cdot)$ is bounded, and each rank is discrete and associated with a fixed jump in its impact on a product's search index. It may no longer be possible to equate potentials, e.g., some high utility products may have such low search indices that even with favorable rankings they will be ignored by consumers, and from the point of view of the platform, this is "wasted" promotion. In view of this, an algorithm that respects the discrete nature of the problem is needed.

The OPT-K algorithm The discrete ranking problem is combinatorial in the number of positions, and therefore demands a computationally tractable algorithm. We propose the following algorithm, which we label OPT-K: instead of ranking all J products, let us instead consider the simpler problem of assigning products to the first K or fewer positions to maximize the consumer surplus from that assignment, which we notate as CS^K, i.e.

$$CS^K = C + \log\left(1 + \sum_{j:r_j \geq J-K+1} \exp \delta_j^V(r_j)\right) + \sum_{j:\phi_j(r_j)>0, r_j \geq J-K+1} q_j(\mathbf{r})\phi_j(r_j),$$

where r_j now represents the discrete rank of each product and is defined as $J - position_j + 1$ (so again, higher is better). Maximizing CS^K by brute force requires only $\sum_{k=1}^{K} J!/(J-k)!$ evaluations.[11] Let the ranking that maximizes CS^K be r^K.

Proposition 2 (Approximate Optimality)**.** *If* $\exists\, K < J$ *such that for any feasible ranking (i)* $\delta_j^S + f(r_j) < -1, \forall\ r_j < J - K + 1$ *and (ii)* $\sum_{j:r_j<J-K+1} -(\delta_j^S + f(r_j)) \exp(\delta_j^S + f(r_j)) < \nu$*, then the optimal ranking for all* J *products can be approximated by the optimal ranking of the first* K *products with the following error bound:*

$$CS(\boldsymbol{r}^K) \geq CS(\boldsymbol{r}^*) - \nu\left(\max_j \delta_j^U + \frac{2-\nu}{1-\nu}\right),$$

where $\boldsymbol{r}^K \in \arg\max CS^K$ *and* $\boldsymbol{r}^* \in \arg\max CS$*.*

Proof. See Appendix A.5. □

Proposition 2 says that if all products placed after position K (i.e., those with low ranks) are unlikely to be bought because their search indices become very low, then optimizing product assignments for the first K positions will deliver close to optimal consumer surplus. To interpret assumptions (i) and (ii) in the statement of Proposition 2, notice that if (i) and (ii) hold simultaneously, it must be that $\sum_{j:r_j<J-K+1} \exp(\delta_j^S + f(r_j)) < \nu$ for any feasible ranking, since $-(\delta_j^S + f(r_j)) > 1$. This in turn implies $\sum_{j:r_j<J-K+1} q_j < \nu$. Therefore, the assumptions can be viewed as requiring the combined choice probabilities of the products placed after position K to be small enough.

What makes the result of Proposition 2 tricky to prove is that the decision of how to rank any one product affects the choice probabilities for *all other products*. By doing an exhaustive search over the top K positions, we can guarantee approximate optimality whenever the remaining positions "bury" the products placed there sufficiently so that they are rarely bought. Many online environments have the property that the top ranked products get the

[11] All possible assignments of the J products to the first l positions take $J!/(J-k)!$ evaluations, and since the algorithm may assign up to position K, we must sum over all assignments that have $k = 1$ to $l = K$.

vast majority of clicks, in which case there may be a reasonably small K for which the ν bound is reasonably tight.

Ranking the remaining $J-K$ products The OPT-K algorithm does not prescribe how to rank the remaining $J-K$ products. In practice, there may be substantial gains from ranking all the products. We now propose a practical greedy algorithm for the remaining products and call it the OPT-K+Greedy (OPTKG) algorithm.

The greedy algorithm is only needed after the OPT-K algorithm ranks all first K positions. That is, if the OPT-K algorithm determines only $L < K$ products are needed to maximize consumer surplus, then the OPTKG algorithm also terminates. But if needed, for the remaining positions, the greedy algorithm begins with the highest remaining rank and myopically assigns the best product to each position holding fixed all the products that have already been ranked. The algorithm terminates when either all positions are assigned or for some rank it is best not to assign any product to that rank. Algorithm 1 formally presents the OPTKG algorithm for consumer surplus optimization, labeled OPTKG-CS.

Algorithm 1: OPT-K+Greedy Algorithm for CS (OPTKG-CS)

```
Result: Assign a unique rank position {K+1, K+2, ..., K+k}, ∀k ≤ J−K to k
        unique products
Initialization: Let f_i denote the effect of position i's ranking on the search index.
 Set N contains all J−K unranked products.
for position i from K+1 to J do
    Calculate CS_ij = CS({r_j = f^{-1}(f_i), r_{{1,2,...,J}\N}}, δ^S, δ^U) for each product j ∈ N;
    Calculate CS_i0 = CS({r_{{1,2,...,J}\N}}, δ^S, δ^U);
    Assign position i to product j such that CS_ij ∈ arg max_{l∈N∪{0}} CS_il;
    if j = 0 then
        Break;
    else
        Update N = N \ j;
    end
end
```

3.2 Revenue Maximization

The platform might also consider matching products to ranks as to maximize the platform's revenues. We begin again with a relaxation of the problem in which ranks can be assigned continuously, and then consider the discrete problem.

Let π_j be the revenue that the platform earns from selling product j. We focus on revenue maximization since marginal costs are essentially zero for many platforms and thus revenues correspond to profits. However, an analogous argument applies to the case where marginal costs are nonzero and the platform maximizes profits. The platform's revenue maximization problem can be written as:

$$\max_{r_1 \ldots r_J} \sum_j q_j(\mathbf{r})\pi_j$$
$$\text{subject to} \sum_{j\,:\,j \text{ is listed}} r_j \leq \frac{J(J+1)}{2},$$

We let $\pi(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}})$ denote the objective function above, i.e. the expected (per-customer) revenue. The optimal solution has the property that the partial derivative of expected revenues is equal for all products that are ranked: $\partial\pi\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)/\partial r_j = \partial\pi\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)/\partial r_k \; \forall\, j, k$. However, unlike the consumer surplus, even if it is feasible to choose ranks so that $\delta_j^S + f(r_j) \geq \delta_j^U \; \forall\, j$ (i.e., every product has higher search than utility index), it might not be optimal for revenues because consumer utilities are not necessarily positively correlated with the revenue from each product. Thus, a short-term revenue maximizing platform has an incentive to distort ranks (from a consumer surplus perspective) even with unlimited ranking power. To get more intuition, take the derivative of platform revenue with respect to r_j (we derive this in Appendix A.6):

$$\frac{\partial\pi\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)}{\partial r_j} = \begin{cases} q_j f'(r_j)\left(\pi_j - \sum_{k \leq J} q_k \pi_k\right) & , \; \delta_j^S + f(r_j) < \delta_j^U \\ 0 & , \; \delta_j^S + f(r_j) \geq \delta_j^U \end{cases}$$

The expression resembles that of the consumer surplus derivative, with the revenue margins π_j taking the place of the potentials ϕ_j. But because the revenues are *fixed* even as the rankings are adjusted, the optimal solution is quite different. When there is no outside option, the platform should only display the highest-margin good. The intuition is clear: if the consumer will buy something regardless, steer them in the direction of highest revenues.

With an outside option, the platform has to balance the probability that the consumer buys anything at all with the incentive to push the highest-margin products. The sign of the derivative depends on the term $\pi_j - \sum_{k \leq J} q_k \pi_k$. This is the revenue from j less the choice-probability weighted revenues from all other products (which includes the zero-margin outside option). So when most consumers don't purchase anything (i.e., the weighted revenue is close to zero), this term will be positive, and all products will be ranked, with high-margin and high-effective index products getting the top spots. But if consumers who are presented with the full product assortment are likely to purchase something, the weighted revenue will exceed the revenue offered by some of the products in the assortment, and the platform can improve revenues by excluding those low-margin products. Platforms whose customers are unlikely to shop elsewhere (or not buy at all) can therefore afford to choose a product assortment that consists mostly of high-margin products, while platforms with less loyal customers cannot.

The OPT-K algorithm We propose the analogue OPT-K algorithm for revenue maximization: Instead of ranking all J products, we only consider assignments to the first K positions to maximize revenues from those K or fewer products. Define

$$\pi^K = \sum_{j:r_j \geq J-K+1} q_j(\mathbf{r})\pi_j.$$

Maximizing π^K by brute force again requires only $\sum_{l=1}^{K} \binom{J}{l}$ evaluations. Let the ranking that maximizes π^K be r^K.

Proposition 3 (Approximate Optimality)**.** *If* $\exists\, K < J$ *such that for any feasible ranking (i)* $\delta_j^S + f(r_j) < -1, \forall\, r_j < J-K+1$ *and (ii)* $\sum_{j:r_j<J-K+1} -(\delta_j^S + f(r_j)) \exp(\delta_j^S + f(r_j)) < \nu$*, then the optimal ranking for all J products can be approximated by the optimal ranking of the first K products with the following error bound:*

$$\pi(\boldsymbol{r}^K) \geq \pi(\boldsymbol{r}^*) - \nu \max_j \pi_j,$$

where $\boldsymbol{r}^K \in \arg\max \pi^K$ *and* $\boldsymbol{r}^* \in \arg\max \pi$.

Proof. See Appendix A.7. □

Analogously to Proposition 2, Proposition 3 says that if products placed after position K are unlikely to be bought, then optimizing product assignments for the first K positions can be sufficiently close to the optimal ranking for all products. As before, it may be useful in practice to rank the remaining $J-K$ products. In the same way as earlier, we define a greedy algorithm OPTKG that maximizes revenues over all possible assignments of the first K products, and then greedily assigns each of the remaining products by checking which assignment generates the greatest improvement in revenues over the prior assignment, terminating if non-assignment is ever the best option.

3.3 Simulations

While Propositions 2 and 3 give theoretical guarantees for the performance of the OPT-K algorithm, in practice the algorithm's performance depends on the number of products J and the choice of K, the distributions of utilities and revenues, as well as whether the conditions of the propositions are met. In this section, we illustrate, via simulations, the performance of the OPT-K algorithm under a wide range of conditions. We also discuss in practice how to optimize rankings for the remaining $J - K$ products as well as the runtime of various OPT-K algorithms.

Simulation environment We simulate $J = 5$ products to be assigned to ranks $r \in \{5, 4, ..., 1\}$, where the effect of ranking on the mean search index exponentially decays, i.e. $f(r) = A \cdot exp(r - 6)$ for $A \in \{5, 10\}$. We limit the number of products to 5 to retain the ability to brute force and find the actual best and worst assignments as benchmark. Mean search and utility indices of each product are drawn from i.i.d. normal distributions, i.e. $\delta_j^S \sim i.i.d.N(0, 10)$ and $\delta_j^U \sim i.i.d.N(\mu, 10)$ for $\mu \in \{-5, -2, 2, 5\}$. For each combination of (A, μ), we simulate 1,000 times, for a total of 8,000 simulations. For each simulation draw, we first find the maximum and minimum consumer surplus CS_{max} and CS_{min} by enumerating all $\sum_{l=1}^{J} \binom{J}{l}$ possible rankings, and then report the consumer surplus under the OPT-K algorithm as $\frac{CS - CS_{min}}{CS_{max} - CS_{min}}$.

Results Table 1 presents the performance results of the OPT-K algorithm for both consumer surplus and platform revenues. In each simulation, we first run the OPT-K algorithm. To evaluate the overall performance, for the remaining positions, we compare the OPTKG algorithm to either not assigning any product to the remaining $J - K$ positions or a random assignment.[12] We repeat for $K = \{1, 2, ..., 5\}$. Note that the OPT-1+Greedy algorithm is equivalent to running Algorithm 1 for all J products, and the OPT-5 algorithm is equivalent to brute force for all $J = 5$ products.

[12] To calculate consumer surplus or platform revenues under the random assignment, we enumerate all possible assignments for the remaining $J - K$ positions and average the consumer surplus or platform revenues associated with each assignment.

Table 1: OPT-K performance

$J=5$; $8 \times 1,000$ simulations	Consumer Surplus			Platform Revenues		
Positions $K+1,...J$	None	Random	Greedy	None	Random	Greedy
$K=1$	96.5%	95.3%	99.3%	97.0%	84.7%	99.4%
$K=2$	99.5%	97.9%	99.9%	99.6%	94.3%	99.9%
$K=3$	99.9%	99.3%	100%	100%	97.2%	100%
$K=4$	100%	99.9%	100%	100%	98.3%	100%
$K=5$	100%	100%	100%	100%	100%	100%

Note: Simulation results of OPT-K algorithm performance for consumer surplus and platform revenues with $J=5$ products. Positions $K+1,...,J$ assign: (i) no products, (ii) randomly ordered products, or (iii) products according to OPTKG. Performance is first normalized by $\frac{Q-Q_{min}}{Q_{max}-Q_{min}}$, where $Q=\{CS,\pi\}$ and Q_{min} and Q_{max} are obtained via enumeration, and then averaged across 8 combinations of parameters $(A,\mu) \times$ 1,000 simulations per each set of parameters.

With $J=5$ products, OPT-1, by picking the best product for the first position, achieves more than 96% of maximum consumer surplus or platform revenues when assigning no products to any positions after the first. By using the greedy algorithm to assign the remaining slots, OPT1G achieves more than 99% in both consumer surplus and platform revenues. Random assignments for remaining positions obtain lower consumer surplus and platform revenues, at 95.3% and 84.7%, respectively, as displaying bad products (either in utility or revenue) can distract consumers away from good products. When K is small, or equivalently $J-K$ is large, the OPTKG algorithm achieves meaningful improvements over no or random assignment of the remaining products, suggesting that adding a greedy assignment step may be important in practice. Strikingly, the OPT2G algorithm achieves more than 99.9% of the optimal consumer surplus and platform revenues across all these simulations, despite brute forcing the assignment of only 2 products (40% of the products available).

We now turn to the computational cost of OPTKG. The number of evaluations required to execute OPT-K is $\sum_{l=1}^{K}\binom{J}{l}$, which scales up quickly with both K and J. On the other hand, the greedy algorithm is cheap: If OPT-K assigns products to all K positions, then the OPTKG algorithm would add at most an additional cost of $\frac{(J-K)(J-K+3)}{2}$ evaluations,[13] depending on how many products end up being assigned.

Table 2 compares different OPT-K algorithms in terms of their run times. We increase the number of products to $J=25$, which is more realistic in practice: it roughly represents the

[13] Assignments of all J products would require $(J-K+1)+(J-K)+...+2=\frac{(J-K)(J-K+3)}{2}$ evaluations.

Table 2: OPT-K runtime (in seconds)

$J = 25$; 8×10 simulations	Consumer Surplus		Platform Revenues	
Positions $K + 1, ...J$	None	Greedy	None	Greedy
$K = 1$	1.2e-4	9.2e-4	2.7e-4	9.5e-4
$K = 2$	4.8e-3	5.6e-3	5.4e-3	5.8e-3
$K = 3$	0.11	0.11	0.12	0.12
$K = 4$	3.2	3.2	3.6	3.6
$K = 5$	70	70	64	65

Note: Simulation results of OPT-K algorithm runtime for consumer surplus and platform revenues in seconds with $J = 25$ products. Positions $K + 1, ..., J$ assign: (i) no products or (ii) products according to OPTKG. Runtime are averaged across 8 combinations of parameters $(A, \mu) \times 10$ simulations per each set of parameters.

two full pages of search results on a typical platform that are most relevant to the consumers. With an otherwise similar simulation environment, we run various OPT-K algorithms for 10 times for each set of parameters. While runtime increase quickly with K, the OPT-K algorithm remains computationally feasible for small K when $J = 25$. The additional cost of the OPTKG algorithm is small and does not meaningfully scale with K. Runtime for consumer surplus and platform revenues are similar. We conclude that in our simulations the OPTKG algorithm is both computationally feasible and close to optimal for small K.

4 Empirical Application

We apply our method to study the customer search and choice data from a hotel booking platform (Expedia). Using our double logit demand estimates, we compare our optimal ranking algorithm to both a utility-based ranking and the Expedia ranking, evaluating how they each perform with respect to revenue, consumer surplus and number of searches.

4.1 Data and Descriptive Evidence

Our data comes from Expedia and is made available through Kaggle.com, an online platform where data miners can use datasets to take part in competitions posted by companies. We refer the reader to Ursu (2018) for a comprehensive discussion of the data; here, we focus on the features that are most directly relevant to our analysis. The Expedia data is composed of two datasets, a training dataset and a testing dataset. Both data contain customer search and choice records from Nov 2012 to Jun 2013 among 34 different markets. There are 120,883 search impressions in the training data and 276,644 impressions in the testing data. Both training and testing data contain only impressions where customers searched at least once and each impression displays 5 to 38 different hotels. There are 124,561 hotels in the training data and 130,136 hotels in the testing data. We observe their rating, price, country, review score, whether it belongs to a chain or not, and a location score. Further, we observe whether the hotel is being clicked on and whether the hotel is booked.

The main difference between the two datasets is that in the training data, hotels are ranked randomly, whereas in the testing data, hotels are ranked according to the (proprietary) Expedia algoritm. The advantage of the training data is that we can identify the effect of rankings on the customer search index without having to worry about endogeneity of ranks. Although we observe whether a customer searched a hotel or not, we do not observe the order of search. In the estimation, this will require us to integrate out along all possible permutation of search paths. Table 3 shows summary statistics for hotel characteristics in the training and testing samples. One can see that the hotel characteristics have very similar distributions in the two samples, except that prices are slightly lower in the testing sample. Table 4 shows summary statistics on consumer behavior. First, consumers face fairly large choice sets, consisting of around 25 hotels on average. In spite of this, consumers only search slightly more than one hotel on average, suggesting that rankings are likely to play an important role in driving final choices. There is, however, heterogeneity in search, with some consumers clicking on several hotels. Note that, in order to keep the integration along search paths tractable, we drop impressions where customers searched more than five times, which corresponds to around 0.5% of the overall sample.

Next, we provide some descriptive evidence to motivate our model. Figure 2 shows the relationship between the number of clicks and the number of bookings for the same hotel in the training data.[14] While hotels that are clicked more often also tend to be booked more, there is also substantial independent variation in the two variables. This suggests that search and choice patterns are driven by two distinct mechanisms, which motivates our double index model. In other words, a model featuring a single index, such as a standard discrete choice model, is not likely to fit the data well.

Second, we look at the ranking algorithm used by Expedia in the testing data. Figure 3 relates a hotel's average position in the *testing data* with the number of clicks (left panel) and bookings (right panel) in the *training data*. We would expect that hotels that are more

[14]For the descriptive evidence, we focus on the top 50 most often displayed hotels in the training data.

Table 3: Summary statistics: Hotel characteristics

	Observations	Mean	Median	SD	Min	Max
Training Sample:						
Star rating	2,922,728	3.17	3.00	1.06	0	5
Review score	2,922,728	3.69	4.00	1.17	0	5
Chain	2,922,728	0.60	1.00	0.49	0	1
Location score	2,922,728	2.88	2.83	1.55	0	6.98
Price	2,922,728	164.90	128.00	145.20	0	5,000
Promotion	2,922,728	0.20	0	0.40	0	1
Testing Sample:						
Star rating	6,947,458	3.18	3.00	1.04	0	5
Review score	6,947,458	3.81	4.00	1.01	0	5
Chain	6,947,458	0.65	1.00	0.48	0	1
Location score	6,947,458	2.87	2.77	1.52	0	6.98
Price	6,947,458	149.30	120.00	112.50	0	5,000
Promotion	6,947,458	0.22	0	0.42	0	1

Note: The table shows summary statistics of hotels. An observation is a hotel-impression, so that hotels are weighted by their appearance in search results.

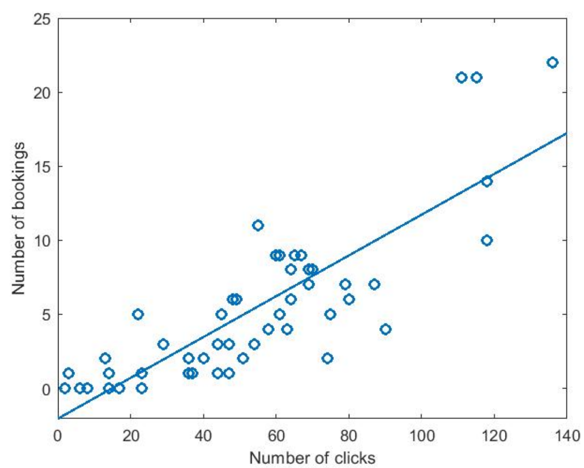


Figure 2: Clicks and bookings

Note: The figure shows the relationship between number of clicks and number of bookings for the top 50 most displayed hotels in the training data. Each dot corresponds to a hotel, and the best linear fit is plotted.

Table 4: Summary statistics: Consumer behavior

	Observations	Mean	Median	SD	Min	Max
Training Sample:						
Number of hotels in choice set	120,883	24.18	28.00	9.19	5	38
Number of searches	120,883	1.10	1.00	0.43	1.00	5
Indicator for purchase	120,883	0.13	0	0.34	0	1
Testing Sample:						
Number of hotels in choice set	276,644	25.11	30.00	9.08	5	38
Number of searches	276,644	1.08	1.00	0.38	1	5
Indicator for purchase	276,644	0.94	1.00	0.24	0	1

Note: The table shows summary statistics for consumer behavior at the search impression level.

often clicked and purchased in the training data (where ranks are assigned randomly) are the ones that Expedia would choose to rank more favorably in its own algorithm. This is indeed the case.

Finally, Figure 4 shows how the probability of clicking and booking a hotel varies with the average hotel position across the two datasets. As expected, better positions are associated with higher click probabilities on average in both data sets (left panel). However, the slope of the relationship is steeper in the testing data. This is consistent with the idea that Expedia is optimizing its rankings, so that the hotels ranked in the first few positions are relatively more attractive than random hotels. Notice also that the probability of searching a hotel in position 30 or above (i.e., lower rank) declines to almost zero in the testing data, but is still relatively high in the training data. This suggests that when rankings are not optimized, consumers end up having to search further. Turning to bookings, the right panel of Figure 4 shows similar patterns: the booking rate declines more rapidly as a function of rank in the testing than the training data, and low ranked hotels are sometimes booked in the training data, but almost never in the testing data.[15]

[15] We plot the probability of booking on different axes here because the testing data over-samples searches that terminate in booking, so that plotting them on the same axes may be misleading.

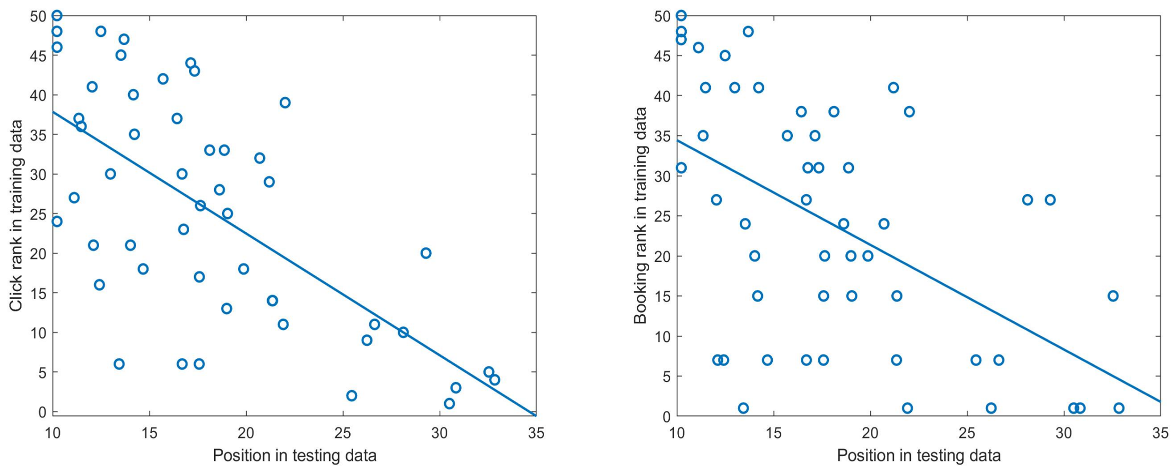


Figure 3: Clicks and bookings in training data as a function of average Expedia position

Note: The figure shows the relationship between the average position of a hotel in the testing data and its click rank in the training data (left panel) and its booking rank in the training data (right panel). Each dot corresponds to a hotel.

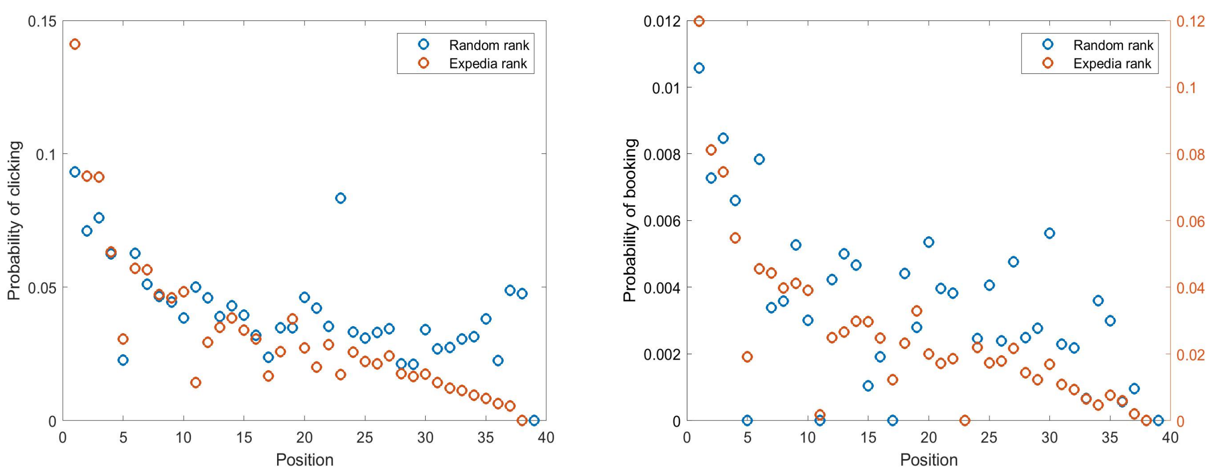


Figure 4: Probability of clicking and booking as a function of position

Note: The figure shows the relationship between the position of a hotel and the probability of it being clicked (left panel) and being booked (right panel). Each dot corresponds to a position, with blue dots referring to the training data and red dots referring to the testing data. In the right panel, the booking probability for the training and testing data are on separate axes, as the testing data oversamples searches that conclude in booking.

4.2 Empirical "Double Logit" Model

For the purpose of estimation, we parameterize the utility and search indices in search impression t as follows:

$$\begin{aligned} s_{ijt} &= \beta^S x_{jt} + \gamma(r_{jt}) + \xi_j^S + \varepsilon_{ijt} + \tilde{\varepsilon}_{ijt}^S \\ u_{ijt} &= \beta^U x_{jt} + \xi_j^U + \varepsilon_{ijt} + \tilde{\varepsilon}_{ijt}^U \end{aligned} \tag{6}$$

where x_{jt} denotes observed (to the researcher) characteristics of hotel j, r_{jt} denotes the rank of hotel j on the web page, and ξ_j^U, ξ_j^S are hotel-specific attributes that affect utility and search, respectively, but are not captured by the data. Consistent with the theoretical model, we let a common unobserved term, ε_{ijt}, enter both the search and the utility index. We take ε_{ijt} to be distributed i.i.d. $N(0, \sigma^2)$ where σ is a parameter to estimate.[16] We assume that consumers observe x_{jt}, r_{jt}, ξ_j^S and ε_{ijt} for all j, and decide whether or not to learn ξ_j^U by searching hotel j. This is consistent with the Weitzman microfoundation in Section 2.4. In addition, we include idiosyncratic shocks, $\tilde{\varepsilon}_{ijt}^U$ and $\tilde{\varepsilon}_{ijt}^S$, which are taken to be distributed i.i.d. type–I extreme value. These independent shocks help smooth out the likelihood function and lead to convenient closed form expressions, as described below. We follow Ursu (2018) in assuming that the observed hotel characteristics (including price) are exogenous. Note that rankings are exogenous by construction in the training data since they are randomized.

We estimate the model via maximum likelihood. In order to write the likelihood of each consumer's observed click and purchase outcomes, we proceed in two steps. First, we exploit a convenient property of the type–I extreme value distribution to obtain a closed form for the outcome probability for any given search sequence. Then, since the search sequence is not observed in the data, we sum over all possible sequences to obtain the probability of the observed outcome.

To illustrate, consider a simple example with two hotels and an outside option and suppose that the data tells us that consumer i searches both hotels and books hotel 1. This set of outcomes is consistent with two search sequences:

- Search hotel 1, then 2, then book 1 (sequence A)

[16] We make this distributional assumption since it requires fewer samples to approximate the distribution of ε_{ijt} in estimation, and hence is less computationally demanding. This is in contrast to the theoretical model, which assumed a type-I extreme value distribution for this term. Still, we find that the theory-based optimal rankings perform well (see, e.g., Figure 7) and thus conclude that this discrepancy between the theoretical and empirical model is not a first-order issue in our setting.

- Search hotel 2, then 1, then book 1 (sequence B)

Dropping the t subscripts for simplicity, the only possible ordering of the search and utility indices associated with sequence A is

$$s_{i1} \geq s_{i2} \geq u_{i1} \geq u_{i2}, u_{i0} \tag{7}$$

This is because: (i) $s_{i1} \geq s_{i2} \geq u_{i0}$, given that both hotels are searched and 1 is searched before 2; (ii) $s_{i2} \geq u_{i1}$, since 2 is searched after u_{i1} is revealed; and (iii) $u_{i1} \geq u_{i2}, u_{i0}$, since 1 is chosen. Under the maintained assumptions and for fixed $(\varepsilon_{i1}, \varepsilon_{i2})$, the probability of the ordering in (7) is equal to

$$\frac{\exp\left(\delta_{i1}^{S}\right)}{1+\exp\left(\delta_{i1}^{S}\right)+\exp\left(\delta_{i2}^{S}\right)+\exp\left(\delta_{i1}^{U}\right)+\exp\left(\delta_{i2}^{U}\right)} \frac{\exp\left(\delta_{i2}^{S}\right)}{1+\exp\left(\delta_{i2}^{S}\right)+\exp\left(\delta_{i1}^{U}\right)+\exp\left(\delta_{i2}^{U}\right)} \frac{\exp\left(\delta_{i1}^{U}\right)}{1+\exp\left(\delta_{i1}^{U}\right)+\exp\left(\delta_{i2}^{U}\right)} \tag{8}$$

where we let $\delta_{ij}^{S} = \beta^{S} x_{j}^{S} + \gamma(r_j) + \xi_{j}^{S} + \varepsilon_{ij}$ and $\delta_{ij}^{U} = \beta^{U} x_j + \xi_{j}^{U} + \varepsilon_{ij}$ for $j = 1, 2$. The expression in (8), sometimes referred to as the "exploded logit" model, is convenient in that it allows us to write the likelihood in a way that does not rely heavily on computationally costly simulation methods. Specifically, while we still need to numerically integrate over the distribution of $(\varepsilon_{i1}, \varepsilon_{i2})$, we do not need to do so for the idiosyncratic shocks $\left(\tilde{\varepsilon}_{i1}^{U}, \tilde{\varepsilon}_{i2}^{U}, \tilde{\varepsilon}_{i1}^{S}, \tilde{\varepsilon}_{i2}^{S}\right)$. Similarly, for sequence B, there are three possible orderings:

$$s_{i2} \geq s_{i1} \geq u_{i1} \geq u_{i2}, u_{i0}$$
$$u_{i1} \geq s_{i2} \geq s_{i1} \geq u_{i2}, u_{i0}$$
$$s_{i2} \geq u_{i1} \geq s_{i1} \geq u_{i2}, u_{i0}$$

We can write the probability of each of these orderings using the exploded logit formula as in (8); the sum across the three orderings then gives us the outcome probability associated with sequence B. Next, summing the outcome probabilities associated with the two sequences we obtain the likelihood of the observed outcome for consumer i. A similar logic applies to more complicated outcomes involving more than two hotels and thus we are able to write the probability of the data in closed form.

Finally, note that the data only covers consumers who clicked on at least one hotel. To account for this sample selection, we divide the likelihood by the probability of clicking on at least one hotel (which again can be written in closed form using the exploded logit formula). This is the (conditional) likelihood that we maximize. More formally, the log-likelihood is given by:

$$\ell(data;\theta) = \sum_t \log\left(\frac{\sum_{\text{ord}_t} \int P(\text{ord}_t; x_t, \theta) dF_\varepsilon}{P(\text{click at least one hotel}; x_t, \theta)}\right) \tag{9}$$

where $x_t = (x_{1t}, \ldots, x_{Jt})$, ord_t indexes the different possible orderings of search and utility indices for impression t, F_ε denotes the distribution of $(\varepsilon_1, \ldots, \varepsilon_J)$, and $P(\text{ord}_t; x_t, \theta)$ denotes the probability of ordering ord_t based on the exploded logit formula (e.g., the expression in (8)).

We now provide some intuition for the type of variation in the data that allows us to identify the model (Appendix A.8 contains a more formal identification argument). The correlation between hotel characteristics and the probability that hotels are clicked identifies the parameters in the search indices. Similarly, the correlation between hotel characteristics and the probability that hotels are chosen, conditional on being clicked on, identifies the parameters in the utility indices. Notice that it is necessary to have a model of search in order to consistently estimate the choice model, since the options in the consideration sets are endogenously determined, and the idiosyncratic errors of options in the consideration set will generally not be iid extreme value. In other words, simply estimating a logit model on the observed consideration sets is likely to lead to biased results.

4.3 Results

We fit our double index model to the *training dataset* by maximum likelihood. Table 5 presents the results. The first two columns of the table present coefficients and standard errors for the search index, while the last two columns present coefficients and standard errors for the utility index. First, notice that, as expected, the rank position coefficients in the search index are monotonically decreasing. Two positions — positions 5 and 11 — are an exception to this pattern. This is consistent with the fact that Expedia places "opaque offers," i.e., offers in which the consumer does not know the name of the hotel before making a purchase, in these positions (see Ursu (2018) for more on this point). All other coefficients estimates seem reasonable. For example, price coefficients are negative in both the search and the utility indices which suggest that higher prices not only reduce the probability of booking conditional on searching, but also deter customers from searching in the first place. Similarly, higher review scores positively impact both search and booking. The estimates of the coefficients in the utility index are also in line with those in Ursu (2018); in particular, we estimate a price coefficient of -0.2722, which is very close to the estimate of -0.2867 in

Ursu (2018) (in both models, prices are measured in hundreds of dollars). Finally, note that the parameter σ is estimated to be close to zero (albeit with a large standard error), which suggests that the correlation between the unobserved shocks in the search indices and those in the utility indices is small.

4.4 Model Fit

Table 6 shows the average odds of clicking on a hotel for different positions. Comparing the first two rows, we can see that the model does a good job at matching the patterns in the training data, especially for the first few positions. The model tends to overestimate the odds of search for positions 7-10, but the overall fit is quite good. In the third and fourth row, we assess the fit of the model out of sample, namely on the testing dataset where ranks are not randomized but rather optimized by Expedia. Again, the patterns observed in the data are broadly matched by our model.

Regarding purchase rates, the model fits well, predicting a booking rate in the training data of 14%, slightly higher than the 13% we see in Table 4. On the other hand, the model substantially under-predicts purchase rates in the testing data. This is to be expected given that impressions leading to a transaction were over-sampled in the testing data to a larger degree relative to the training data (Ursu, 2018).

4.5 Counterfactual Analysis

Given the estimated model, we perform a range of counterfactual exercises to compare our proposed algorithm to competing algorithms. Specifically, we draw 1,000 customers (i.e., choice sets) at random from the testing data. For each customer, we use the model and the estimated search and utility index parameters to compute the expected consumer welfare, number of searches and revenue under four rankings.[17] The four rankings are: (i) "OPT3G-CS," our approximately optimal algorithm for maximizing consumer surplus with an exhaustive search over the top 3 positions and a greedy algorithm for rest of the positions; (ii) "OPT3G-Rev," our approximately optimal algorithm for maximizing revenue; (iii)

[17] We approximate the expectation by Monte Carlo simulation: for each consumer, we draw 200,000 vectors of the independent idiosyncratic shocks GC: We also draw the common ϵ_{ijt} normal shocks, right? to the search and utility indices for all products, determine which product they purchase, and their consumer surplus, and then output the average across all draws. We do this because there is no closed-form expression for consumer surplus when the draws are independent (in contrast, when they are perfectly correlated, there is an expression given by Proposition 1).

Table 5: MLE estimates of double logit model

	Search index		Utility index	
	Coef	SE	Coef	SE
Position 1	1.0578	0.0435		
Position 2	0.7900	0.0452		
Position 3	0.5936	0.0465		
Position 4	0.4684	0.0476		
Position 5	0.1601	0.2727		
Position 6	0.3289	0.0489		
Position 7	0.4025	0.0497		
Position 8	0.1918	0.0545		
Position 9	0.2249	0.0564		
Position 10	0.1615	0.0611		
Position 11	-5.5639	1.0058		
Position 12	0.1690	0.0647		
Star rating	0.2413	0.0132	0.0193	0.0234
Review score	0.0725	0.0096	0.1077	0.0201
Chain	0.1149	0.0262	0.3555	0.0502
Location score	0.1029	0.0109	0.0111	0.0171
Price ($100)	-0.2869	0.0138	-0.2722	0.0295
Promotion	0.2583	0.0291	0.0356	0.0686
Constant	-8.1309	0.0374	-5.1602	0.0282
σ	0.0002	0.3618		

Note: The table shows the estimates of the double logit model. The first two columns report the coefficients and standard errors for the search index parameters, whereas the last two refer to the utility index parameters. σ determines the correlation between the shocks in the search and the utility indices. Hotel fixed effects are included in both the search and the utility index, but are not reported.

"Utility," the algorithm that simply ranks goods in descending order of their utility indices; and finally (iv) the Expedia ranking.

Table 7 shows the average changes in each of these metrics for the OPT3G-CS, OPT3G-Rev and utility rankings relative to the Expedia algorithm. OPT3G-CS yields an average gain in consumer surplus of almost $0.80 per customer relative to Expedia. Remember that this is an average over all consumers, *including consumers who do not even search*, since our model allows for non-search as an option. The simulated conversion rate over this population is in the range of 3-4%, implying a gain in consumer surplus for consumers who purchase of over $20 (similarly, the revenue numbers may be scaled by 25-30 times, so e.g. OPT3G-Rev

Table 6: Odds of clicking for different positions

Positions	1	2	3	4	5	6	7	8	9	10
Data - Training	1.0000	0.7843	0.6751	0.5380	0.0063	0.4634	0.3969	0.3218	0.2863	0.2706
Estimates - Training	1.0000	0.7599	0.6395	0.5510	0.0058	0.4887	0.4327	0.3837	0.3591	0.3294
Data - Testing	1.0000	0.6643	0.4996	0.3886	0.0037	0.3115	0.2519	0.2118	0.1755	0.1523
Estimates - Testing	1.0000	0.7211	0.5861	0.4752	0.0197	0.4340	0.3787	0.3350	0.3114	0.2621

Note: The table shows the odds of clicking for positions 1-10 relative to the odds for position 1. The first row reports the odds in the training data, whereas the second row reports the odds predicted by the model for the same data. The third and fourth rows repeat this for the testing data.

Table 7: Average changes relative to the Expedia ranking

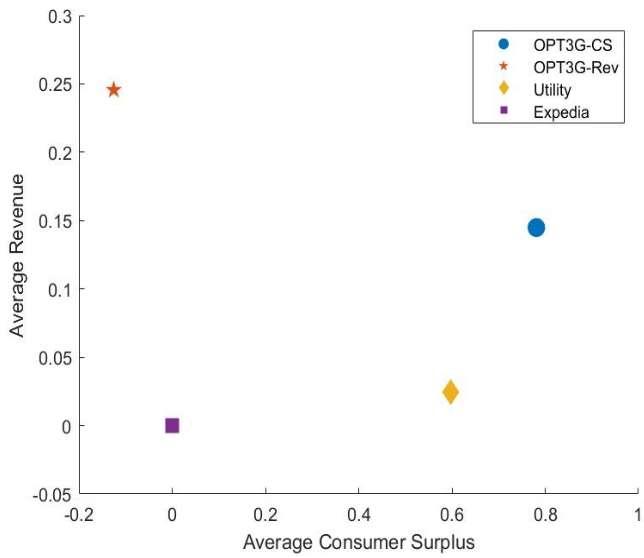
	OPT3G-CS	OPT3G-Rev	Utility
Δ Consumer Surplus ($)	0.7810	-0.1253	0.5970
Δ Search Count	0.0019	-0.0052	-0.0028
Δ Revenue ($)	0.1449	0.2456	0.0244

delivers in additional revenue of over $6 per purchase).

One might wonder whether achieving this higher surplus requires more search on the part of consumers, as in principle, higher search costs could offset the gains from finding a better match. We find that, on average, consumers engage in only 0.0019 additional searches under OPT3G-CS relative to Expedia. While our model does not provide an estimate of search costs, this very small difference suggests that the additional search costs implied by our algorithm are negligible relative to the utility gains. The magnitudes of the consumer welfare numbers should be interpreted with caution since, in the Expedia dataset, impressions leading to a transaction were oversampled, though it is unclear whether this is only true of the testing data (94% purchase rate, versus 13% in the training data), or both. Still, the comparison is informative about the relative performance of the different ranking algorithms.

Next, Figure 5 shows that there is a trade-off between revenue and consumer surplus. Among the four algorithms, OPT3G-CS sacrifices some revenue in order to maximize expected consumer surplus, whereas OPT3G-Rev achieves higher average revenues at the cost of much lower consumer surplus. Notice, however, that OPT3G-CS dominates the Expedia and the

Figure 5: Trade-off between consumer surplus and revenue maximization

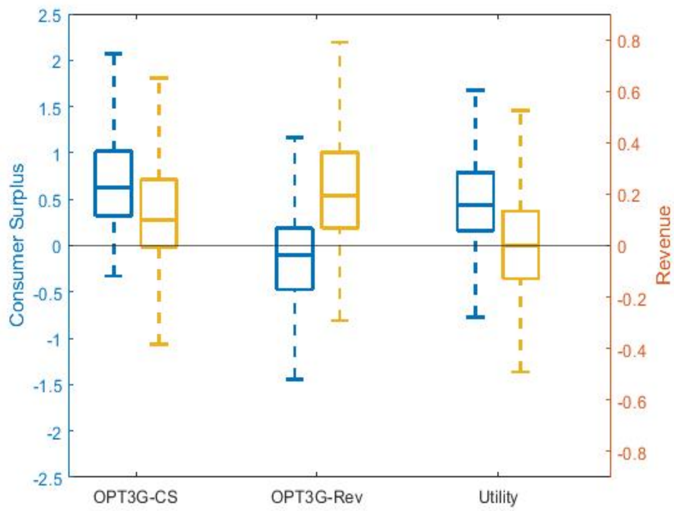


Note: The figure shows the relationship between the average consumer surplus and the average revenue for the OPT3G-CS, OPT3G-Rev and Utility algorithms relative to the Expedia algorithm.

utility rankings in terms of both consumer surplus and revenue. In this sense, the Expedia and the utility rankings are within the Pareto frontier.

Going beyond averages, Figure 6 shows the entire distribution of consumer surplus and revenue for the OPT3G-CS, OPT3G-Rev and utility algorithms relative to the Expedia ranking. One can see that the results for the averages continue to hold when we look at the full distributions of customers.

Figure 6: Changes in consumer surplus, number of searches and revenue relative to Expedia ranking



Note: The figure compares the OPT-3, OPT3G-Rev and Utility algorithms to that used by Expedia. For each of them, the blue box plots (on the left) show the distribution of changes in consumer surplus and the yellow box plots (On the right) show the distribution of changes in revenues. Each box marks the 25th, 50th and 75th percentiles, the dotted lines indicate the 5th to 95th percentile range, and the solid lines show the outliers.

In order to further shed light on how the algorithms perform for any given customer, Table 8 reports the fraction of customers for whom each of the four algorithms maximizes expected surplus (relative to the remaining three). As expected, OPT3G-CS maximizes consumer surplus for the vast majority of choice sets. In comparison, the Expedia and the utility rankings are optimal only for around 10% of customers each. OPT3G-Rev comes in last, which is not surprising since it targets a different objective function.

Note that there are at least two reasons why OPT3G-CS does not always maximize consumer surplus. First, OPT-K is an *approximately* optimal algorithm. As such, it is possible that other algorithms would sometimes dominate it. Second, our algorithm is (approximately) optimal for the case in which the idiosyncratic error terms in the utility and search indices are perfectly correlated, but in estimation we assumed that a component of the shocks is iid across the two indices. This discrepancy could also cause OPT-K to not achieve the optimum under the estimated model. This being said, the fact that OPT3G-CS still maximizes consumer surplus in more than three quarters of choice sets is reassuring and suggests that we are not too far from the optimal algorithm.

The results so far show that our OPT3G algorithms achieve desirable outcomes relative to

Table 8: Breakdown of customers by which algorithm maximizes expected CS

	OPT3G-CS	OPT3G-Rev	Utility	Expedia
% of customers	76.60%	1.30%	11.20%	11.80%

Note: For each algorithm, the table reports the fraction of customers in the testing data for whom the algorithm yields a higher consumer surplus than the remaining three.

two competing rankings, the Expedia and the utility ranking. However, they do not say anything about the performance of OPT3G in comparison with any of the (many) other possible algorithms. In order to shed some light on this, we draw 1,000 rankings at random, and apply them to a random subset of 1,000 customers (i.e., choice sets) from the testing data. Specifically, we draw a random ranking of all the products that ever appear in a choice set. Then for any particular choice set — a subset of all products — we list them in the order given by the ranking. We compare the consumer surplus they achieve to that from OPT3G, as well as other competing algorithms. The left panel of Figure 7 shows the results. We can see that both our OPT3G algorithms as well as the utility and Expedia algorithms do better than the vast majority of random rankings. In particular, the OPT3G-CS algorithm achieves an average consumer surplus that is several standard deviations higher than the bulk of the distribution for random rankings, which illustrates the value of our algorithm. The right panel of Figure 7 complements the analysis by looking at average revenue. Again, the four algorithms all tend to dominate the majority of random rankings, but now OPT3G-Rev performs better than the others, including OPT3G-CS, illustrating the trade-off between optimizing revenues and maximizing consumer surplus.

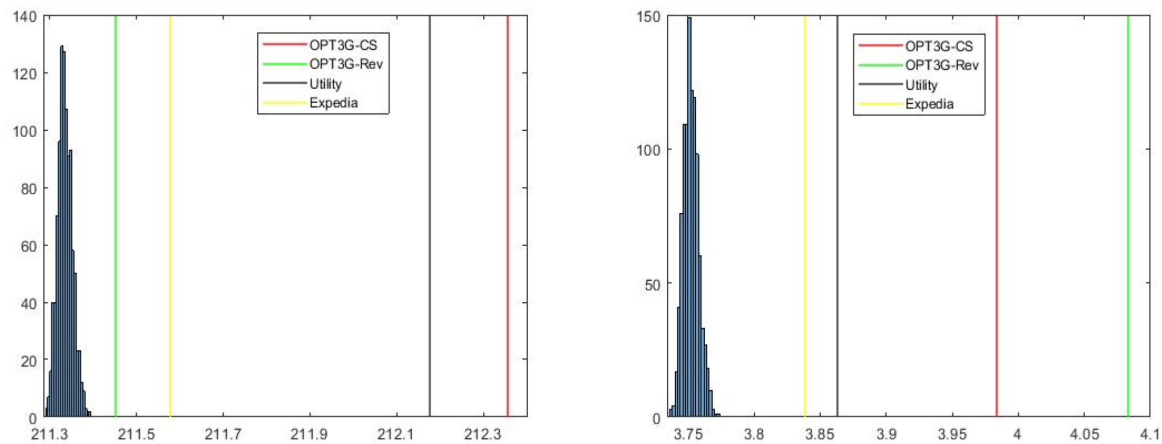


Figure 7: Comparison with random rankings

Note: The left panel shows the distribution of average consumer surplus for 1,000 random rankings (the blue histogram) as well as the four algorithms described in the text. The average is taken over choice sets and utility shocks. The right panel shows the same for average revenue.

5 Conclusion

Engagement with the virtual world seems likely to increase over time. How and where platforms choose to direct consumer attention is a key component of the online world, shaping the choices that consumers make. The model presented here offers one way of formalizing this relationship between platforms and consumers, positing that platforms can affect a search index that determines what consumers choose to view, though it cannot directly affect the consumption utility from content. Using this model, we can show that when a platform wants to optimize for consumer surplus, it should aid search discovery, by promoting products that are better than the customer would have believed based on their observable characteristics, i.e. by surfacing products that would have otherwise been overlooked.

One major contribution of the paper is the double index model itself, which translates a lot of folklore and intuition about online markets into the familiar econometric language of discrete choice. The specialization to the double logit, which we used in our application, allowed us to use exploded logit formulas, facilitating easy estimation. But it is not without loss of generality. While we have shown that the search protocol in Weitzman (1979) can be used to microfound the model, it would be interesting to explore which other protocols are consistent with it. Different protocols might lead to more complicated expressions for the search and utility indices, possibly requiring a more flexible (nonparametric) specification.

A second direction would be to allow for additional heterogeneity in preferences. For example, if consumers are ex-ante differentiated in their taste for characteristics as in the random coefficients model of Berry et al. (1995), then search rankings should anticipate this heterogeneity, and provide a diverse set of product options. This intuition is not captured here, in line with existing empirical work using the same data (Ursu (2018)).

Finally, one might wonder whether the tension between maximizing revenues and optimizing consumer surplus is a consequence of the fact that the model is static. In a dynamic model where Expedia takes into account the effect of its rankings not just on today's, but also on future outcomes, it's possible that the two objectives might be more aligned. For instance, if consumer surplus is a good predictor of the likelihood of customer retention, then maximizing consumer surplus may also be optimal from a revenue perspective in the long run. Since the data does not track consumers over time, we are unable to explore this question here, but it would be an interesting avenue for future research.

References

Jason Abaluck and Giovanni Compiani. A method to estimate discrete choice models that is robust to consumer search. Working Paper, National Bureau of Economic Research, 2020.

Gagan Aggarwal, Jon Feldman, Shanmugavelayutham Muthukrishnan, and Martin Pál. Sponsored search auctions with markovian users. In *International Workshop on Internet and Network Economics*, pages 621–628. Springer, 2008.

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. Mnl-bandit: A dynamic learning approach to assortment selection. *Operations Research*, 67(5):1453–1485, 2019.

Hunt Allcott. The welfare effects of misperceived product costs: Data and calibrations from the automobile market. *American Economic Journal: Economic Policy*, 5(3):30–66, 2013.

Asim Ansari and Carl F Mela. E-customization. *Journal of marketing research*, 40(2): 131–145, 2003.

Mark Armstrong. Ordered consumer search. *Journal of the European Economic Association*, 15(5):989–1024, 2017.

Mark Armstrong and John Vickers. Which demand systems can be generated by discrete choice? *Journal of Economic Theory*, 158:293–307, 2015.

Susan Athey and Glenn Ellison. Position auctions with consumer search. *Quarterly Journal of Economics*, 126(3):1213–1270, 2011.

Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica*, pages 841–890, 1995.

Fernando Branco, Monic Sun, and J Miguel Villas-Boas. Optimal search for product information. *Management Science*, 58(11):2037–2056, 2012.

Fernando Branco, Monic Sun, and J Miguel Villas-Boas. Too much information? information provision and search costs. *Marketing Science*, 35(4):605–618, 2016.

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

Michael Choi, Anovia Yifan Dai, and Kyungmin Kim. Consumer search and price competition. *Econometrica*, 86(4):1257–1281, 2018.

Babur De los Santos and Sergei Koulayev. Optimizing click-through in online rankings with endogenous search refinement. *Marketing Science*, 36(4):542–564, 2017.

Mahsa Derakhshan, Negin Golrezaei, Vahideh Manshadi, and Vahab Mirrokni. Product ranking on online platforms. In *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, page 459, 2020.

Pedro Gardete and Megan Hunter. Guiding consumers through lemons and peaches: An analysis of the effects of search design activities. Stanford University Graduate School of Business Research Paper, 2020.

Anindya Ghose, Panagiotis G Ipeirotis, and Beibei Li. Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. *Marketing Science*, 31(3):493–520, 2012.

Anindya Ghose, Panagiotis G Ipeirotis, and Beibei Li. Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science*, 60(7):1632–1654, 2014.

Michelle Sovinsky Goeree. Limited information and advertising in the US personal computer industry. *Econometrica*, 76(5):1017–1074, 2008.

Chris Gu and Yike Wang. Consumer online search with partially revealed information. *Management Science*, 2021.

Charles Hodgson and Gregory Lewis. You can lead a horse to water: Spatial learning and path dependence in consumer search. Cowles Foundation Discussion Paper 2246, 2020.

Elisabeth Honka and Pradeep Chintagunta. Simultaneous or sequential? search strategies in the us auto insurance industry. *Marketing Science*, 36(1):21–42, 2017.

Elisabeth Honka, Ali Hortaçsu, and Maria Ana Vitorino. Advertising, consumer awareness, and choice: Evidence from the us banking industry. *The RAND Journal of Economics*, 48(3):611–646, 2017.

Srikanth Jagabathula and Paat Rusmevichientong. A nonparametric joint assortment and price choice model. *Management Science*, 63(9):3128–3145, 2017.

Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 475–484, 2017.

T Tony Ke, Zuo-Jun Max Shen, and J Miguel Villas-Boas. Search for information on multiple products. *Management Science*, 62(12):3576–3603, 2016.

David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*, pages 585–596. Springer, 2008.

Jun B Kim, Paulo Albuquerque, and Bart J Bronnenberg. Online demand under limited consumer search. *Marketing science*, 29(6):1001–1023, 2010.

Jun B Kim, Paulo Albuquerque, and Bart J Bronnenberg. The probit choice model under sequential search with an application to online retailing. *Management Science*, 63(11): 3911–3929, 2017.

Robert Kleinberg, Bo Waggoner, and E Glen Weyl. Descending price optimally coordinates search. *arXiv preprint arXiv:1603.07682*, 2016.

A Gürhan Kök and Marshall L Fisher. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research*, 55(6):1001–1021, 2007.

Nitin Mehta, Surendra Rajiv, and Kannan Srinivasan. Price uncertainty and consumer search: A structural model of consideration set formation. *Marketing science*, 22(1):58–84, 2003.

José L Moraga-González, Zsolt Sándor, and Matthijs R Wildenbeest. Consumer search and prices in the automobile market. 2021.

John H Roberts and James M Lattin. Development and testing of a model of consideration set composition. *Journal of Marketing Research*, 28(4):429–440, 1991.

Herbert A Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1):99–118, 1955.

Kenneth A Small and Harvey S Rosen. Applied welfare economics with discrete choice models. *Econometrica*, pages 105–130, 1981.

George J. Stigler. The economics of information. *Journal of Political Economy*, 69(3): 213—225, 1961.

Kenneth Train. Welfare calculations in discrete choice models when anticipated and experienced attributes differ: A guide with examples. *Journal of Choice Modelling*, 16:15–22, 2015.

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.

Raluca M Ursu and Daria Dzyabura. Retailers' product location problem with consumer search. *Quantitative Marketing and Economics*, pages 125–154, 2020.

Martin L Weitzman. Optimal search for the best alternative. *Econometrica*, pages 641–654, 1979.

Song Yao and Carl F Mela. A dynamic model of sponsored search advertising. *Marketing Science*, 30(3):447–468, 2011.

A Appendix

A.1 Additional Microfoundations

As noted earlier, the double index model is sufficiently flexible to nest many models of search. The Weitzman (1979) model is discussed in the main text. We offer another set of examples here:

Example 2 (Non-Sequential Search). *Consider the non-sequential search model of Stigler (1961), in which products are ex-ante identical with payoffs drawn independently from F^U supported on $[\underline{u}, \bar{u}]$, and consumers must choose how many products n to sample, paying a cost $c(n)$ that is convex in the number of products sampled. Let n^* be the optimal sample size, and set $s_j = \bar{u}$ for a randomly chosen n^* of the products, and $s_j = \underline{u}$ for the remainder.*

Example 3 (Random Consideration). *Consider the random consideration model used in Goeree (2008). Each product is searched with probability p_j, and then the best option in the consideration set is chosen. Then this is equivalent to our model where s_j takes value $\bar{u}$ with probability p_j and value $\underline{u}$ otherwise (where $\underline{u}$ and $\bar{u}$ are defined as above).*

The algorithm also nests as special cases some other algorithmic models of behavior.

Example 4 (Satisficing). *Consider a model in which agents "satisfice" in the sense of Simon (1955), searching ex-ante identical products at random and stopping at the first alternative that offers them a desired utility level. Let u^* be that target level and let $s_j = u^*$ for all items.*

Example 5 (Cascade Model). *Consider the cascade model of sponsored search, introduced by Kempe and Mahdian (2008).*[18] *Users consider advertisements starting from the top-ranked position (position 1) down. They click each ad with probability q_j and continue down ("cascade") to consider clicking the next-ranked ad with probability p_j regardless of whether they clicked the previous ad or not. We can generate this behavior from our model by positing that search indices are Bernoulli, with $s_1 = \bar{u} + 1$ with probability q_1 and $\underline{u}$ otherwise, and s_j $(j \geq 2)$ distributed according to:*

$$s_j = \begin{cases} \bar{u} + 1/j & \text{with probability } q_j \prod_{k=1}^{j-1} p_k \\ \underline{u} & \text{otherwise} \end{cases}$$

[18] Aggarwal et al. (2008) contemporaneously developed a similar model. The consumers in Athey and Ellison (2011) behave similarly, but their model is micro-founded: consumer behavior is a best response to the way products are ordered following a position auction.

Example 6. *(Price Thresholding) Consider the nonparametric joint assortment and price choice model of Jagabathula and Rusmevichientong (2017), in which consumers choose the best product available that has a price (weakly) below some threshold* $\underline{p}$. *Set* $s_j = \bar{u}$ *if* $p_j \leq \underline{p}$ *and* $s_j = \underline{u}$ *otherwise.*

A.2 Proof of Lemma 1

Proof. Let $j^* \in \mathcal{J}^* = \arg\max_{j \in \mathcal{J}} v_j$. Towards a contradiction assume $j' \notin \mathcal{J}^*$ is purchased. Then either $s_{j'} < v_{j^*}$ or $u_{j'} < v_{j^*}$. If $s_{j'} < v_{j^*}$, then product j^* is searched before j' since $s_{j^*} \geq v_{j^*} > s_{j'}$, and the consumer stops before searching j', since $u_{j^*} \geq v_{j^*} > s_{j'}$. On the other hand, if $u_{j'} < v_{j^*}$, then the consumer either searches j^* first, or searches j' first but does not stop until they search j^* because $s_{j^*} \geq v_{j^*} > u_{j'}$. In either case, the consumer does not purchase j' since $u_{j^*} \geq v_{j^*} > u_{j'}$. So product j' cannot be purchased, generating a contradiction. □

A.3 Proof of Proposition 1

Proof. Applying Lemma 1, the choice probabilities take the standard logit form:

$$P(\text{Choose } j) = \frac{\exp \delta_j^V}{1 + \sum_k \exp \delta_k^V},$$

since j is purchased if and only if it has the highest v_{ij} which happens with probability proportional to $\exp \delta_j^V$.

Ex-ante consumer surplus (i.e. prior to the realization of the common logit shock) is given by:

$$\begin{aligned}
&\sum_j P(\text{Choose } j) E[u_{ij} | \text{Choose } j] \\
&= \sum_j P(\text{Choose } j) E[v_{ij} | \text{Choose } j] + \sum_j P(\text{Choose } j) E[u_{ij} - v_{ij} | \text{Choose } j] \\
&= \sum_j P(\text{Choose } j) E[v_{ij} | \text{Choose } j] + \sum_j P(\text{Choose } j) E[\delta_j^U + \varepsilon_{ij} - \delta_j^V - \varepsilon_{ij} | \text{Choose } j] \\
&= \sum_j P(\text{Choose } j) E[v_{ij} | \text{Choose } j] + \sum_j P(\text{Choose } j) (\delta_j^U - \delta_j^V) \\
&= C + \log \left(1 + \sum_j \exp \delta_j^V \right) + \sum_{j: \phi_j > 0} q_j \phi_j,
\end{aligned}$$

where the second line follows by linearity of conditional expectations, and the final line uses the consumer surplus formula due to Small and Rosen (1981). Notice that $\delta_j^U - \delta_j^V > 0 \; \forall j$ since the effective index is a minimum of the search and utility indices.

□

A.4 Derivative of Consumer Surplus

Recall that consumer surplus is given by:

$$CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right) = C + \log\left(1 + \sum_j \exp \delta_j^V(r_j)\right) + \sum_{j:\phi_j(r_j)>0} q_j(\boldsymbol{r})\phi_j(r_j),$$

where $\delta_j^V(r_j) = \min\{\delta_j^S + f(r_j), \delta_j^U\}$ is the mean effective index, $q_j(\boldsymbol{r}) = \frac{\exp \delta_j^V(r_j)}{1+\sum_k \exp \delta_k^V(r_k)}$ is the market share of product j and depends on all other products' rank positions, and $\phi_j(r_j) = \delta_j^U - \delta_j^S - f(r_j)$ is product j's potential.

When $\delta_j^S + f(r_j) \geq \delta_j^U$, the mean effective index is determined by the utility and does not depend on the ranking, i.e. $\frac{\partial \delta_j^V(r_j)}{\partial r_j} = \frac{\partial \delta_j^U}{\partial r_j} = 0$, and product j has non-positive potential, i.e. $\phi_j(r_j) \leq 0$ and thus does not enter into the second term in the consumer surplus. Therefore the derivative of consumer surplus with respect to product j's ranking is zero.

When $\delta_j^S + f(r_j) < \delta_j^U$, the derivative of consumer surplus with respect to product j's ranking is given by:

$$\begin{aligned}
\frac{\partial CS\left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}}\right)}{\partial r_j} &= \frac{\partial \log\left(1 + \sum_k \exp \delta_k^V(r_k)\right)}{\partial r_j} + \frac{\partial q_j \phi_j(r_j)}{\partial r_j} - \frac{\partial \sum_{k\neq j:\phi_k(r_k)>0} q_k \phi_k(r_k)}{\partial r_j} \\
&= \frac{\exp \delta_j^V(r_j) f'(r_j)}{1 + \sum_k \exp \delta_k^V(r_k)} + \phi_j(r_j)\left(\frac{\exp \delta_j^V(r_j) f'(r_j)}{1 + \sum_k \exp \delta_k^V(r_k)} - \frac{\exp \delta_j^V(r_j)^2 f'(r_j)}{(1 + \sum_k \exp \delta_k^V(r_k))^2}\right) \\
&\quad - q_j f'(r_j) - \sum_{k\neq j:\phi_k(r_k)>0} \phi_k(r_k) \frac{\exp \delta_k^V(r_k) \exp \delta_j^V(r_j) f'(r_j)}{(1 + \sum_l \exp \delta_l^V(r_l))^2} \\
&= q_j f'(r_j) + \phi_j(r_j)\left(q_j f'(r_j) - q_j^2 f'(r_j)\right) - q_j f'(r_j) - \sum_{k\neq j:\phi_k(r_k)>0} \phi_k(r_k) q_k q_j f'(r_j) \\
&= q_j f'(r_j)\left(\phi_j(r_j) - \sum_{k:\phi_k(r_k)>0} q_k \phi_k(r_k)\right)
\end{aligned} \tag{10}$$

A.5 Proof of Proposition 2

Proof. For some $K < J$, such that (i) $\delta_j^S + f(r_j) < -1, \forall r_j < J-K+1$ and (ii) $\sum_{j:r_j<J-K+1} -(\delta_j^S + f(r_j)) \exp(\delta_j^S + f(r_j)) < \nu$, and any ranking $\boldsymbol{r}$, first decompose the ex-ante consumer surplus and then re-arrange:

$$
\begin{aligned}
CS(\boldsymbol{r}) &= C + \log\left(1 + \sum_j \exp \delta_j^V\right) + \sum_{j:\phi_j>0} q_j \phi_j \\
&= C + \log\left(1 + \sum_{j:r_j \geq J-K+1} \exp \delta_j^V\right) + \log\left(1 + \frac{\sum_{j:r_j<J-K+1} q_j}{1 - \sum_{j:r_j<J-K+1} q_j}\right) + \\
&\quad \sum_{j:\phi_j>0, r_j \geq J-K+1} q_j \phi_j + \sum_{j:\phi_j>0, r_j<J-K+1} q_j \phi_j \\
&\leq C + \log\left(1 + \sum_{j:r_j \geq J-K+1} \exp \delta_j^V\right) + \frac{\nu}{1-\nu} + \\
&\quad \sum_{j:\phi_j>0, r_j \geq J-K+1} q_j \phi_j + \sum_{j:\phi_j>0, r_j<J-K+1} (\delta_j^U - \delta_j^S - f(r_j)) \exp(\delta_j^S + f(r_j)) \qquad (11) \\
&\leq CS^K(\boldsymbol{r}) + \frac{\nu}{1-\nu} + \sum_{j:\phi_j>0, r_j<J-K+1} \delta_j^U \exp(\delta_j^S + f(r_j)) + \sum_{j:\phi_j>0, r_j<J-K+1} (-\delta_j^S - f(r_j)) \exp(\delta_j^S + f(r_j)) \\
&\leq CS^K(\boldsymbol{r}) + \frac{\nu}{1-\nu} + \max_j \delta_j^U \cdot \sum_{j:\phi_j>0, r_j<J-K+1} \exp(\delta_j^S + f(r_j)) + \sum_{j:\phi_j>0, r_j<J-K+1} (-\delta_j^S - f(r_j)) \exp(\delta_j^S + f(r_j)) \\
&\leq CS^K(\boldsymbol{r}) + \frac{\nu}{1-\nu} + \max_j \delta_j^U \cdot \sum_{j:r_j<J-K+1} \exp(\delta_j^S + f(r_j)) + \sum_{j:r_j<J-K+1} (-\delta_j^S - f(r_j)) \exp(\delta_j^S + f(r_j)) \\
&\leq CS^K(\boldsymbol{r}) + \frac{\nu}{1-\nu} + \nu \max_j \delta_j^U + \nu \\
&\leq CS^K(\boldsymbol{r}) + \nu \left(\max_j \delta_j^U + \frac{2-\nu}{1-\nu}\right),
\end{aligned}
$$

where $CS^K(\boldsymbol{r}) = C + \log\left(1 + \sum_{j:r_j \geq J-K+1} \exp \delta_j^V\right) + \sum_{j:\phi_j>0, r_j \geq J-K+1} q_j \phi_j$. The second line follows on noting that $\log(a+b) = log(a) + log(1 + b/a)$ and then dividing and multiplying by the logit denominator in the second term to convert to choice probabilities. The third line follows on noting that $\log\left(1 + \frac{\sum_{r_j<J-K+1} q_j}{1-\sum_{r_j<J-K+1} q_j}\right) \leq \frac{\sum_{r_j<J-K+1} q_j}{1-\sum_{r_j<J-K+1} q_j} \leq \frac{\nu}{1-\nu}$. The following lines make use of the bounds either assumed or implied by the assumptions.

We are left to show $CS^K(\boldsymbol{r^*}) \leq CS(\boldsymbol{r^K})$, where $\boldsymbol{r}^K \in \arg\max CS^K$ and $\boldsymbol{r}^* \in \arg\max CS$. We show that $\boldsymbol{r}^K$ only considers assignments to K or fewer positions. We prove this by contradiction: Suppose the optimizer of CS^K assigned products to positions beyond K. Then by dropping those products, the choice probabilities for products in the top K positions must strictly increase, and all other terms comprising CS^K remain unchanged, implying that dropping these products is an improvement - a contradiction. Therefore, we have $CS^K(\boldsymbol{r}^*) \leq CS^K(\boldsymbol{r}^K) = CS(\boldsymbol{r}^K)$.

Then since (11) holds for any ranking, including the optimal, we have:

$$\begin{aligned} CS(\boldsymbol{r}^*) &\leq CS^K(\boldsymbol{r}^*) + \nu \left(\max_j \delta_j^U + \frac{2-\nu}{1-\nu} \right) \\ &\leq CS(\boldsymbol{r}^K) + \nu \left(\max_j \delta_j^U + \frac{2-\nu}{1-\nu} \right). \end{aligned}$$

Re-arranging terms we have:

$$CS(\boldsymbol{r}^K) \geq CS(\boldsymbol{r}^*) - \nu \left(\max_j \delta_j^U + \frac{2-\nu}{1-\nu} \right)$$

. □

A.6 Derivative of Platform Revenues

Recall that the platform's revenue function is given by:

$$\pi \left(\mathbf{r}, \delta^{\mathbf{S}}, \delta^{\mathbf{U}} \right) = \sum_j q_j(\boldsymbol{r}) \pi_j,$$

where $q_j(\boldsymbol{r}) = \frac{\exp \delta_j^V(r_j)}{1+\sum_k \exp \delta_k^V(r_k)}$ is the market share of product j and π_j is its revenue.

When $\delta_j^S + f(r_j) \geq \delta_j^U$, the mean effective index is determined by the utility and does not depend on the ranking, i.e. $\frac{\partial \delta_j^V(r_j)}{\partial r_j} = \frac{\partial \delta_j^U}{\partial r_j} = 0$. Therefore the derivative of revenues with respect to product j's ranking is zero.

When $\delta_j^S + f(r_j) < \delta_j^U$, the derivative of platform revenues with respect to product j's ranking is given by:

$$\begin{aligned}
\frac{\partial \pi\left(\mathbf{r},\delta^{\mathbf{S}},\delta^{\mathbf{U}}\right)}{\partial r_j} &= \frac{\partial q_j \pi_j}{\partial r_j} + \frac{\partial \sum_{k\neq j} q_k \pi_k}{\partial r_j} \\
&= \pi_j \left(\frac{\exp \delta_j^V(r_j) f'(r_j)}{1+\sum_k \exp \delta_k^V(r_k)} - \frac{\exp \delta_j^V(r_j)^2 f'(r_j)}{(1+\sum_k \exp \delta_k^V(r_k))^2} \right) - \sum_{k\neq j} \pi_k \frac{\exp \delta_k^V(r_k) \exp \delta_j^V(r_j) f'(r_j)}{(1+\sum_l \exp \delta_l^V(r_l))^2} \\
&= \pi_j \left(q_j f'(r_j) - q_j^2 f'(r_j) \right) - \sum_{k\neq j} \pi_k q_k q_j f'(r_j) \\
&= q_j f'(r_j) \left(\pi_j - \sum_k q_k \pi_k \right)
\end{aligned} \tag{12}$$

A.7 Proof of Proposition 3

Proof. For any ranking $\boldsymbol{r}$, because $\sum_{j:r_j<J-K+1} q_j < \nu$, we have:

$$\begin{aligned}
\pi(\boldsymbol{r}) &= \sum_{j:r_j\geq J-K+1} q_j \pi_j + \sum_{j:r_j<J-K+1} q_j \pi_j \\
&\leq \pi^K(\boldsymbol{r}) + \nu \max_j \pi_j
\end{aligned} \tag{13}$$

The same argument from consumer surplus applies so we have $\pi^K(\boldsymbol{r}^*) \leq \pi^K(\boldsymbol{r}^K) = \pi(\boldsymbol{r}^K)$, where $\boldsymbol{r}^K \in \arg\max \pi^K$ and $\boldsymbol{r}^* \in \arg\max \pi$. As a result, we have:

$$\pi(\boldsymbol{r}^*) \leq \pi(\boldsymbol{r}^K) + \nu \max_j \pi_j$$

□

A.8 Identification

In this section, we argue that the double logit model is identified as long as a small number of moments are observed. One caveat is that one of those moments is the probability that a user clicks nothing on the page, which we do not see in the present dataset. This is not to say that we don't think that the double logit model is identified from the present data, but it would require a different argument than the one we present here. In addition, we abstract from correlation between the error terms in the utility and search indices, consistent with our empirical findings.

The following moments are assumed to be observed in the data: (i) the probability that nothing is clicked, p_0 (not present in the current data); (ii) the probability that j is clicked first, $\{p_j\}_{j\in\mathcal{J}}$ (observed in current data); (iii) the probability that j is clicked first and then search is terminated without purchase, $\{p_j^0\}_{j\in\mathcal{J}}$ (observed in current data). For simplicity, we assume that the probabilities in (i) and (iii) are strictly positive.

We first derive expressions for these moments. Let $\delta_j \equiv \beta x_j + \xi_j$ and $\widetilde{\delta}_j \equiv \widetilde{\beta}\widetilde{x}_j + \widetilde{\xi}_j$ be the mean utility and search indices respectively. The decision to click option j first (or not click at all) is entirely analogous to the choice in any other discrete choice model, and has a familiar functional form:

$$p_j = \frac{e^{\widetilde{\delta}_j}}{1 + \sum_k e^{\widetilde{\delta}_k}} \tag{14}$$

Having decided to click on j, terminating search without purchase indicates that the payoff to the outside option exceeds the payoff to j and the search indices of the remaining options, i.e., the event occurs if and only if $s_{i,j} > u_{i,0} > \max\{u_{i,j}, \max_{k\neq j} s_{i,k}\}$. The probability of this event can be expressed using the "exploded logit" formula as:

$$p_j^0 = \frac{e^{\widetilde{\delta}_j}}{1 + e^{\delta_j} + \sum_k e^{\widetilde{\delta}_k}} \frac{1}{1 + e^{\delta_j} + \sum_{k\neq j} e^{\widetilde{\delta}_k}} \tag{15}$$

Now we can use a familiar trick in (14) to solve for the mean search indices $\{\widetilde{\delta}_j\}_{j\in\mathcal{J}}$:

$$\widetilde{\delta}_j = \log\left(\frac{p_j}{p_0}\right)$$

With those known, equation (15) can be re-arranged:

$$\left(1 + e^{\delta_j} + \sum_k e^{\widetilde{\delta}_k}\right)\left(1 + e^{\delta_j} + \sum_{k\neq j} e^{\widetilde{\delta}_k}\right) - \frac{e^{\widetilde{\delta}_j}}{p_j^0} = 0$$

This is a quadratic equation in e^{δ_j}, where all the remaining terms are known. And since the coefficients on both the degree two and degree one terms in the quadratic are both positive, from the usual formula for the solutions of a quadratic equation there can be only one non-negative solution (a solution must exist if the model is correct). Solving these quadratics for each j and then taking logs identifies all the utility indices $\{\delta_j\}_{j\in\mathcal{J}}$. From there, the parameters $(\widetilde{\beta}, \beta)$ are identified as the solution to a linear system under a rank condition.