The Personalization Paradox: Welfare Effects of Personalized Recommendations in Two-Sided Digital Markets*

Aaron P. Kaye†
Massachusetts Institute of Technology

November 14, 2024

Updated frequently, please click here for the latest version

Abstract

In many online markets, platforms engage in platform design by choosing product recommendation systems and selectively emphasizing certain product characteristics. I analyze the welfare effects of personalized recommendations in the context of the online market for hotel rooms using clickstream data from Expedia Group. This paper highlights a tradeoff between match quality and price competition. Personalized recommendations can improve consumer welfare through the "long-tail effect," where consumers find products that better match their tastes. However, sellers, facing demand from better-matched consumers, may be incentivized to increase prices. To understand the welfare effects of personalized recommendations, I develop a structural model of consumer demand, product recommendation systems, and hotel pricing behavior. The structural model accounts for the fact that prices impact demand directly through consumers' disutility of price and indirectly through positioning by the recommendation system. I find that ignoring seller price adjustments would cause considerable differences in the estimated impact of personalization. Without price adjustments, personalization would increase consumer surplus by 2.3% of total booking revenue (∼$0.9 billion). However, once sellers update prices, personalization would lead to a welfare loss, with consumer surplus decreasing by 5% of booking revenue (∼$2 billion).

JEL: D12, D83, L10, L13, L83, L86
Keywords: platform design, two-sided markets, e-commerce, consumer search

*First version November 15, 2023. I am extremely grateful to Ying Fan, Francine Lafontaine, Susan Athey, and Zach Brown for their invaluable comments, advice, and encouragement. I want to thank Joaquin Endara, Bruna Morais Guidetti, Ayush Kanodia, Will Mandelkorn, Brock Rowberry, Kannappan Sampath, Michael Sullivan, and Xuan Teng for helpful conversations, comments and suggestions. I also want to thank the seminar participants at the University of Michigan IO Seminar and Ross Business Economics Seminars for their valuable insights and suggestions.

†Aaron P. Kaye (apkaye@mit.edu): Massachusetts Institute of Technology, Sloan School of Management & Computer Science and Artificial Intelligence Laboratory.

1 Introduction

E-commerce has an increasingly prominent role in the economy and continues to reshape economic activity across industries. Consumers turn to online platforms for retail, travel, groceries, dining, medical care, and other goods and services. In the US, 275 million users shop through e-commerce platforms.[1] As of 2022, e-commerce accounted for 16.1% of total retail sales in the United States and was projected to grow to 21.9% by 2025.[2] In the global travel industry, online booking revenue reached $475 billion in 2022 and is projected to surpass $1 trillion in 2030.[3] A few large platforms often dominate each of these industries, design the marketplace, and connect consumers to third-party sellers. A central activity of these platforms is platform design, whereby platforms select product recommendation systems and selectively emphasize certain product attributes. These platforms leverage detailed personal data to inform product recommendations.[4]

This paper considers the welfare effects of personalized product recommendations in two-sided digital markets where platforms design the marketplace but third-party sellers determine prices. In such markets, the effect of personalized recommendations on consumer welfare, platform profit, and seller outcomes is unclear when we consider how personalization changes seller behavior. Platform design decisions impact both consumer decision-making and seller pricing incentives (Dinerstein et al., 2018). Recommendation systems present a tradeoff between match quality and price competition. Consumer welfare depends on the quality of the match between consumer tastes and the products that they ultimately choose, the search costs incurred to find products, and prices. Personalized recommendations may improve consumer welfare through the long-tail effect, whereby consumers match with products that more closely align with their individual tastes and they can find these products with less costly search. However, consumer welfare also depends on prices. Personalized recommendations could recommend products to consumers who are better matched and have higher willingness to pay, possibly incentivizing sellers to increase prices. In sum, under personalized recommendations, consumers could be matched to better products but face higher market prices.[5]

[1] https://www.statista.com/statistics/273957/number-of-digital-buyers-in-the-united-states/

[2] https://www.statista.com/statistics/379112/e-commerce-share-of-retail-sales-in-us/

[3] https://www.statista.com/statistics/1179020/online-travel-agent-market-size-worldwide/

[4] Large platforms now provide recommendation systems as a service, such as Microsoft Intelligent Recommendations and Amazon Personalize, making sophisticated recommendation systems possible for platforms of all sizes.

[5] It is important to distinguish this phenomenon from traditional price discrimination. For example, in first-degree price discrimination, sellers could individualize prices based on consumer-specific purchase tendencies or willingness to pay. The key tradeoff that I highlight in this paper relates to market prices, not individual prices. Personalized recommendations modify the product selection presented to consumers, which changes the equilibrium market prices faced by all consumers.

Against this backdrop, I ask: what are the welfare effects of personalized recommendations and other platform design policies when sellers can adjust prices and consumers update beliefs? The current empirical literature focusing on improving recommendations typically holds prices fixed or applies to situations without relevant prices. I build on this literature by explicitly modeling seller pricing decisions. I address this question in the context of the online accommodation industry, using clickstream data on hotel purchases from Expedia Group, a large online travel agency (OTA).

I first present empirical evidence that informs how I construct the structural model. Using data from an experiment conducted by Expedia, I show that position in search results, also called "slot", impacts demand even when recommendations are randomized. I also show that hidden product features are correlated with slots assigned by the default recommendation system. Together, these facts suggest that slots could impact demand through both search costs and rational expectations about the recommendation system. I later use a structural approach and find evidence for both mechanisms. The last empirical fact is that prices influence product positions. This fact is reflected on the seller side of my structural model, in which sellers account for how their choice of price impacts product rankings.

Next, I develop a structural model of demand, platform product recommendations, and hotel pricing behavior. I present the structural model working through each component, starting with demand. I estimate a rich demand model featuring costly consumer search based on Weitzman (1979), where consumers have rational expectations about product recommendations. Previous empirical literature modeling consumer search has often made simplifying assumptions about what information is revealed from search and consumer beliefs. However, these assumptions may be problematic for understanding the role of personalized recommendations. In my model, product features can be visible or hidden. This avoids issues highlighted by Abaluck et al. (2020), where assuming consumers are aware of hidden product features could bias results. I build on the models of Ursu et al. (2023), and Morozov et al. (2021), where consumers know a portion of the match quality term prior to search and learn the remainder from search, by introducing a data-driven approach, similar to nested logit, that splits the variance components of the match quality term. This model nests the full information demand model and optimal sequential search. Finally, I allow product recommendations to impact demand through search cost and rational expectations. Consumers may accurately believe that the products recommended by the platform have superior hidden features, a mechanism from which the empirical literature commonly abstracts. I allow consumers to have rational expectations and show that this setup fits the data better than a benchmark model where

position impacts only search cost. Failing to capture this issue would bias my estimates; if consumers, in part, search for products higher on the page because they tend to have better hidden features, then we would conclude from a model without hidden features that consumers have extremely high search costs.[6]

I estimate the demand model via maximum simulated likelihood. I use the optimal sequential search rules from Weitzman (1979) to construct the joint likelihood of clicking and booking decisions. I use variation in length of stay to separately identify utility parameters from search cost parameters. And I use consumers' repeated decisions (clicks and purchases) to identify heterogeneity in the utility and search cost parameters.

After modeling demand, I present the second component of the structural model, the platform model. Hotels aiming to maximize profits encounter an elasticity of demand influenced not only by consumer preferences but also by the platform's recommendation system. Specifically, a change in a product's price can shift its position in search results. The platform model aims to reverse-engineer Expedia's default recommendation system, capturing the relationship between price and product rankings. To account for the complexity of the default recommendations system, I use a "model of a model" approach from machine learning and cryptography literature.[7]

The third component of the structural model is the supply-side model of hotel pricing behavior, in which hotels set prices based on the time of stay and time of search to maximize expected profits, considering consumer preferences and the platform recommendation system. I use the opportunity cost of having a unit available to sell in the next period as the relevant marginal cost for hotels.[8] My supply-side model captures key features of the accommodation industry: at low occupancy, hotels have economies of scale, at high occupancy, hotels face increasing costs and capacity constraints. I use an instrumental variable (IV) approach to address the endogeneity concern of modeling costs as a function of quantity.

Then, with my structural model of demand, platform recommendations, and seller pricing behavior complete, I turn to developing recommendation systems to evaluate in counterfactual simulations. The Expedia data include observations from the (non-personalized) default recommen-

[6] I include a more detailed overview of these model features in the demand appendix

[7] This is also called "model extraction". There is extensive literature documenting approaches for reverse engineering black-box algorithms in a number of settings Papernot et al. (2017).

[8] Betancourt et al. (2022) focus on the airline industry with a dynamic pricing model. In their setup, the estimated marginal cost is the opportunity cost (option value) of having the unit in inventory in the next period. While I use the same interpretation of marginal cost, I do not model it as explicitly because of two practical constraints: First, they compare two competing airlines firms in a market, while I observe over 700 competing firms. Second, their data include capacity information, while I observe quantities.

dation system and the randomly ordered experimental data. To understand the welfare effects of personalized recommendation, I develop four increasingly personalized recommendation systems. For each recommendation system, I use an ensemble of 170 LambdaMARTs, a popular machine-learning algorithm for ranking problems, presented in Burges (2010). As with demand estimation, a challenge in training recommendation systems is that slot influences consumer choices but is also highly correlated with product features. I address this issue by using data from an experiment where Expedia randomly assigned slots. The least personalized recommender uses data only on product features. The next recommender includes additional data on the consumer queries. Consumers actively volunteer this information, such as length of stay and whether they are traveling with children. The next recommender incorporates personal data based on the consumer's location, distance to the destination, and time of search. The most personalized recommendation system includes data on consumer's past purchases, such as the average price and star rating of their previous purchases.

Next, I combine the structural model and personalized recommendations to evaluate counterfactuals. In the counterfactual simulations, I solve for the equilibrium induced by each of the four recommendation systems in three distinct phases: first, the platform updates the recommendation system; second, sellers update prices; and third, consumers update their beliefs about the recommendation system. My outcomes of interest are seller profits, quantity sold, platform revenue, and consumer surplus. By evaluating the four recommendation systems, this analysis helps us understand 1) the welfare effects of shifting from the default to personalized recommendation systems, and 2) the welfare implications of escalating levels of personalization.

I find that ignoring seller price adjustments would cause considerable differences in the estimated impact of personalization. In the counterfactual simulations, without price adjustments, personalization increases consumer surplus by 2.3% of total booking revenue. As a back-of-the-envelope calculation, if we scale this up by Expedia's total gross booking revenue in the same year, 2013, this is a $0.9 billion gain in consumer surplus. This finding is consistent with results from other papers that find gains in consumer surplus from improved recommendations. I find only small effects on quantity, revenue, and profits.

However, once sellers update prices, personalization ultimately leads to a welfare loss. Through higher markups, hotel profits increase by 4.9%, and quantity decreases by 4.5%. Gross booking revenue remains relatively unchanged. In the counterfactual with the most personalized recommendation system, consumer surplus declines by 5% of booking revenue (approximately $2 billion). This

amounts to a net welfare loss, as the decrease in consumer surplus is 190% of the increase in hotel profits.

My findings have important policy implications. Recent policies in the EU, such as the General Data Protection Regulation (GDPR), limit platforms' ability to record consumer data. The Digital Markets Act (DMA) focuses on gatekeeper platforms and includes provisions for algorithmic transparency. However, much of the regulatory attention to platforms focuses on self-preferencing, price discrimination, platform fees, and network size. This paper's results highlight an overlooked concern in e-commerce platform research and regulation: Better recommendation systems can reduce competition and ultimately harm consumer welfare. This is important to consider as e-commerce platforms' access to personal data grows and technological improvements allow platforms to deploy increasingly sophisticated recommendation systems.

This paper also has implications for managers. Consider a platform deliberating a tradeoff between profits and product match quality. With prices held fixed, the platform might be at a point where the tradeoff is obvious, where steering consumers to slightly more expensive products is unambiguously good for profits. However, this paper points out that evaluating this tradeoff is not so simple since changes in the recommendation system, in turn, change prices. It might, in fact, be more profitable for the platform to steer consumers to lower-priced goods.

1.1 Background Literature

This paper contributes to four strands of the literature. First, it contributes to the sizeable literature on how information frictions impact markets Stigler (1961), Akerlof (1970), and Diamond (1971) and how digitization reshapes economic activity Goldfarb and Tucker (2019). This paper is perhaps most related to Dinerstein et al. (2018), which explicitly considers a tradeoff between platform design and price competition on eBay. However, this paper differs from Dinerstein et al. (2018) on two dimensions: Dinerstein et al. (2018) focus on homogeneous instead of differentiated goods, and their counterfactual policy is a redesign of the display page instead of the recommendation system.

Second, this work contributes to the literature on feature emphasis in online platforms. One focus of this literature is price obfuscation, for example, through drip-pricing and junk fees (Ellison and Ellison, 2009; Blake et al., 2021).[9] The context of this paper is similar, as drip pricing also

[9]Drip pricing and junk fees are unavoidable parts of transaction prices that are hidden from consumers early in the search process. For example, a hotel might charge a resort fee that is omitted from the price displayed on landing pages.

impacts the accommodation market and is a focus of regulatory attention.[10] More broadly, prices are one product feature that platforms can make more or less costly for consumers to learn. Gardete and Hunter (2018) focus on the automobile market with data from Shift.com and study which features should go on landing pages and which can be moved to product pages. My paper is similar in that I allow consumers to learn about product features through search and to know the correlation between hidden and visible product features. This paper's analysis of consumer search is most related to Abaluck et al. (2020), which presents discrete choice methods for when consumers are not fully informed of product features. Their model enables the researcher to evaluate counterfactuals that change feature emphasis. This paper complements Abaluck et al. (2020) in that I develop a similar demand model of visible and hidden product features that additionally permits consideration of counterfactuals related to feature emphasis and the recommendation system.[11] My paper is different from Abaluck et al. (2020) since I also make use of click data and my counterfactual centers on the recommendation system.

Third, this paper builds on the literature on platform design centered on position effects and recommendation systems. The significance of position effects is well documented, as evidenced by Ursu (2018) and Greminger (2022), who also use Expedia data. These observations are consistent with the common business strategy of auctioning top advertisement slots in search results.[12] Much of the research on the welfare effects of recommendation systems aims to blend demand methods with product recommendations and consider counterfactual utility-based recommendation systems; these include De los Santos and Koulayev (2016), Ursu (2018), Greminger (2022), and Compiani et al. (2021). In contrast to these works, my approach adopts techniques popular in industry and data science to generate ranking algorithms that one might expect to encounter on a platform that personalizes its recommendation system. Specifically, I use randomized data and an ensemble of LambdaMARTs.[13] In that sense, this paper relates to Donnelly et al. (2020), which evaluates personalized rankings using data from an e-commerce platform that randomly personalized recommendations for some consumers and presented nonpersonalized recommendations to others. I also contribute to this literature by accounting for the supply side. Works in this stream hold prices fixed under alternative hypothetical recommendation systems, whereas I endogenize seller pricing decisions. Incorporating

[10] https://www.ftc.gov/news-events/news/press-releases/2023/10/ftc-proposes-rule-ban-junk-fees

[11] Both of our papers also allow for product features that do not impact utility but are correlated with hidden features.

[12] Position effects are also important for sponsored search; Athey and Ellison (2011) present a model of bidding for sponsored link positions.

[13] Similar approaches were used by the contestants who won the Yahoo! Learning to Rank Challenge and the Personalized Expedia Hotel Searches contest.

the supply side appears to be essential since I find that personalization can improve consumer welfare if I hold prices fixed, consistent with the literature, but allowing prices to change yields a loss in consumer surplus.[14]

The fourth is the emerging literature on self-preferencing, which examines hybrid platforms that both operate the marketplace and compete within it. Notable contributions include Teng (2022), which analyzes the Apple App Store, and Lee and Musolff (2021), which examines Amazon's promoting itself as merchant over competing third-party sellers offering the same good.[15] Further, Lam (2021), Farronato et al. (2023) , and Reimers and Waldfogel (2023) investigate Amazon's practices of prioritizing its products over those of competitors. These papers use various parametric approaches to document the extent of self-preferencing and model recommendation system behavior in their respective settings. Lee and Musolff (2021), Teng (2022) and Lam (2021) then use these estimates to perform supply-side estimation of seller costs. This paper extends this line of research by introducing a method to reverse-engineer Expedia's default recommendations system. Instead of adopting a parametric methodology to model the platform recommendation system, I use a "model-of-model" technique from machine learning. This approach addresses potential concerns about misspecification associated with parametric representation of a sophisticated algorithm.

1.2 Outline

The rest of the paper is organized as follows. I cover the institutional background in section 2. In section 3, I present a stylized illustration of the tradeoff between match quality and price effects. I discuss the data and setting in section 4. In section 5, I present three empirical facts that inform the structural model. Section 6 introduces the structural model. Section 7 presents the estimation strategy and results for the structural model. In section 8, I develop personalized recommendation systems. Section section 9 presents the counterfactual simulations and results. Last, section 10 provides concluding remarks and discusses the next steps for this project.

[14] Agrawal et al. (2022) and Moehring (2023) also consider personalization and use industry-standard approaches to developing recommendation systems. Prices do not play a role in these papers, as they focus on consumer engagement—Agrawal et al. (2022) in educational technology and Moehring (2023) on r/News on Reddit.

[15] This self-preferencing is implemented through Amazon's Buy Box, which is the primary purchase option on a given product page. Self-preferencing through the Buy Box means that Amazon gives itself an advantage in terms of being selected as merchant over third-party sellers of the same good.

2 Institutional Background

In digital markets, platforms serving as online intermediaries lead to market behaviors that differ significantly from those in traditional, analog environments. A key factor in this difference is the role of information frictions, which are crucial in shaping consumer demand and market outcomes. In conventional economic models, demand is understood as being influenced by consumer preferences, product characteristics, and information frictions. These frictions affect cumulative demand faced by sellers, as the information available to consumers and the cost of acquiring new information directly impact their purchasing decisions.

E-commerce platforms have notably reduced these information frictions. However, they also uniquely influence these frictions through their platform design strategies. Two primary aspects of such design are recommendation systems and feature emphasis. Recommendation systems influence which products consumers are exposed to and the sequence in which they appear. Meanwhile, feature emphasis affects the visibility of specific product attributes by highlighting them in search results or placing them more discreetly on product-specific pages. This control over information flows allows platforms to act as gatekeepers, a role that has garnered significant policy and regulatory attention.

Platforms have incentives to improve their design since improvements can increase purchase volumes and help them respond to competitive pressure from other platforms. One avenue to improve platform design is to enhance the quality of recommendation systems through personalization. Platforms collect massive amounts of consumer data, including purchase histories and other browsing information, and can use these data to personalize product rankings. Addressing the problem of how best to recommend products is the focus of a growing body of literature, subject of data science competitions, and focus point for platforms. Platforms also face a tradeoff between recommending the products most relevant to consumers and recommending the products most profitable for the platforms themselves.

While this paper focuses on the accommodation industry, it addresses a familiar dynamic between sellers and e-commerce platforms in the increasingly digital economy. The platform chooses its platform design, including the recommendation system, but third-party sellers, in this case hotels, set prices. This setup is common among e-commerce platforms. In the accommodation industry, platforms operated by Expedia Group, Booking Holdings, and Airbnb curate listings by third-party sellers who choose prices. In the food and grocery delivery space, platforms, including Instacart, DoorDash, and Grubhub, choose their design, but restaurants and grocery stores choose prices.

StubHub and Ticketmaster act as intermediaries in the market for event tickets, yet third-party sellers choose prices. This dynamic also impacts hybrid platforms such as Amazon, which competes as a seller on its own platform but for which third-party sellers constitute almost 60% of its sales.[16]

The hotel industry is an ideal setting in which to study the welfare effects of platform design for several key reasons. First, the online travel booking and accommodations industry is inherently worth studying given its size and economic significance. For example, Expedia's economic footprint can be seen in its global gross booking revenues, which totaled $107.87 billion in 2019.[17] Second, this industry can provide insight into other major industries because of its parallels with broader e-commerce dynamics. This setting shares key characteristics with other popular e-commerce platforms: a few large platforms dominate the space, there are many differentiated goods in each market, and, as stated above, third-party sellers set prices. The third reason is data availability. It is rare for platforms to release clickstream data with this level of detail to the public. Fourth, these data also include details on an experiment where the product rankings were randomized. These randomized data are ideal for developing recommendation systems to evaluate counterfactuals. Last, this industry setting holds promise for addressing a recurring challenge in the search literature: it can be difficult to separately identify search costs from preferences, especially when slot and product features are collinear. In this setting, consumers arrive searching for stays of different numbers of nights. This introduces variation in the returns to search, which allows me to separately identify preferences from search costs.

3 Stylized Example

This section illustrates a simplified version of the trade-off between match quality (long-tail effect) and price competition. Figure 3.1 is a stylized illustration focusing on one niche hotel, "The Elvis Hotel", and two consumer types: The first type ( ) represents a typical online-shopper. The other type ( ) has a strong affinity and high willingness to pay for the Elvis-themed goods. The typical online-shopper is more common than the Elvis-type, and both types shop for hotels through an online platform. In this setting, the platform directs consumers to products based on its recommendation system, consumers make purchase decisions, and the hotel chooses a price. The hotel cannot price discriminate between consumer types, so it chooses one price based on the expected set of consumers

[16] https://www.aboutamazon.com/news/small-business/celebrating-a-record-breaking-holiday-season-for-amazon-with-customers-purchasing-more-items-than-ever-before-from-our-selling-partners

[17] In the period of study, 2013, Expedia reported $39.44 billion in gross booking revenue. https://www.statista.com/statistics/269386/gross-bookings-of-expedia/

that would see the hotel.

Figure 3.1: Illustration of Welfare Tradeoff: Match Quality vs Price



Note: For illustration purposes not based on data

The figure compares an environment with default (non-personalized) recommendations, and personalized recommendations. Under the *default recommendations*, the set of consumers steered to the hotel would be similar to the population distribution, with the common type ( ) outnumbering the Elvis-type ( ). Under the default recommendations, the hotel faces demand from a set of consumers mostly comprised of the common-type, and optimally chooses a price of \$120/night. Under *personalized recommendations*, the platform can identify each consumer's types with some accuracy, and recommend products accordingly. The figure shows the welfare gain from the long tail effect, as it is matching the Elvis-type consumers to the Elvis hotel. However, the hotel, now facing demand from a set of better matched consumers, would have an incentive to increase prices.

This stylized example abstracts away from many complexities of the hotel market; there is no formal model; it ignores entry and exit, consumer arrival to the platform, and the behavior of other hotels and other platforms. However, this example highlights the tradeoff between match quality and prices. In this simple example, the welfare effects of personalized recommendations are unclear; consumers are better matched to products but face higher market prices. We would need a formal model to conclude if the personalized recommendations resulted in a welfare gain or loss.

4 Data and Setting

My primary data source is clickstream data from Expedia Group. These data are publicly available on Kaggle.com and were initially released as part of a data science competition hosted through Kaggle and the International Conference on Data Mining (ICDM 2013) to improve Expedia's recommendation system with personalization. These data are popular among data scientists and an increasingly popular resource for researchers, as it is rare for platforms to publicly release such detailed clickstream data.[18]

The data cover searches from November 1, 2012, to June 30, 2013.[19] The data are at the search-impression level, with one observation corresponding to a consumer–product pair. They include 332,344 queries (consumer searches), with 9,917,530 product queries, covering 173 destination countries, and 136,886 unique properties. For each consumer query (a specific consumer's search), the data include details on up to the first 38 product listings, which products were clicked, and which products were purchased. They include characteristics of each hotel, location attractiveness scores, and information about each consumer's specific query and purchase history (summary statistics about past purchases), hotel availability, and prices on nine other OTAs.

The data are organized around hotel searches and impressions and divided into five categories: search criteria, static hotel characteristics, dynamic hotel characteristics, visitor information, and competing OTA information. For instance, search criteria might include the date and time of the search, destination ID, length of stay, number of adults/children/rooms, etc. Static hotel characteristics cover aspects such as hotel ID, country, star rating, user review score, and historical pricing, while dynamic features include the slot (position), promotion indicators, and headline price, among others.

Another important feature of these data is that they include a details from randomized controlled trial. For two-thirds of the data, consumers received results from Expedia's default recommendation system—so-called natural search results. For the other third, consumers received randomly ordered search results. Injecting this type of occasional experimental randomness into search results to train future versions of recommendation systems is a common practice among platforms. However, the results of these experiments are rarely made publicly available to researchers. For this paper, I use the naturally ordered results to estimate demand and the randomly ordered results to train

[18]These data have also been used by Ursu (2018), Abaluck et al. (2020), Greminger (2022), and Reimers and Waldfogel (2023).

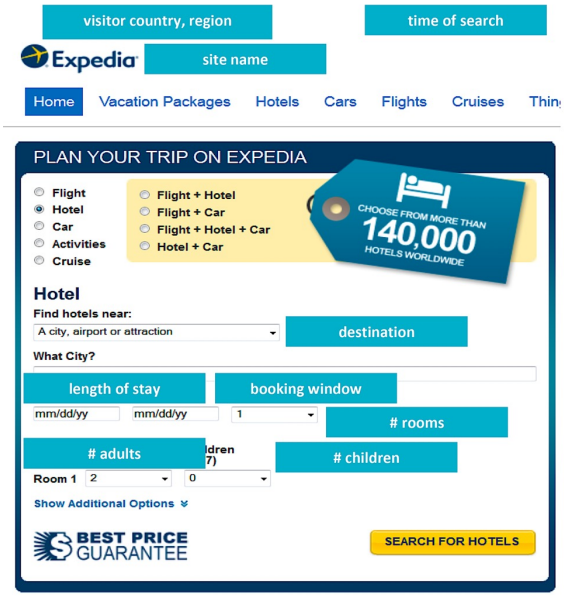
[19]The searches can be for stays as late as October 24, 2014.

(estimate) alternative recommendation systems that I use in the counterfactual analysis.

4.1 Consumer Search Process

Here, I outline the consumer search process and the associated data included and omitted from the Expedia data. We can think of the search process as including three phases: query, search, and purchase. During the query phase, consumers initially input specific search criteria such as location, dates, length of stay, and details about rooms, adults, and children. During the query phase, Expedia also records certain consumer-specific information, such as the country from which the search is made, the booking window (time between the date of search and the date of the stay), the time of the search, and information about the consumer's purchase history.

Figure 4.1: Query



Following the query, Expedia displays products on the landing page ordered into slots according to the recommendation system. On the landing page, consumers see the property star rating (class of hotel), customer review scores, approximate location information, whether the property is on sale through a promotion, and the headline price. The headline price is typically the average nightly price of the cheapest available room. Consumers also see a profile picture for each property, but this information is not included in the data.

In the search phase, consumers click on products to navigate to the product page, which reveals more information, including specific product information, room options, and additional pricing

Figure 4.2: Landing page, products ordered by recommendation system

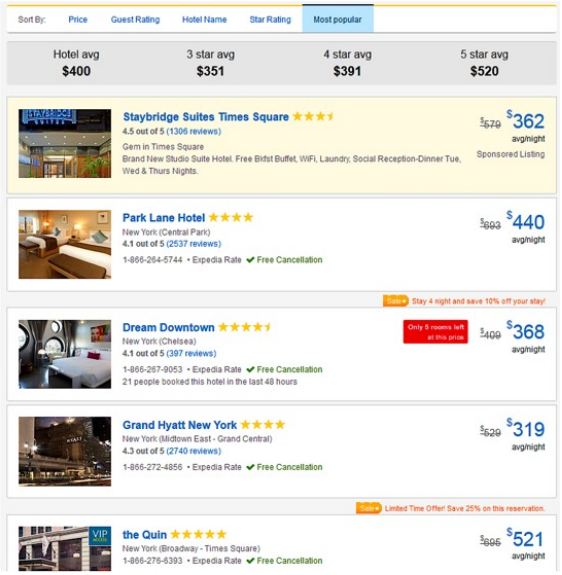
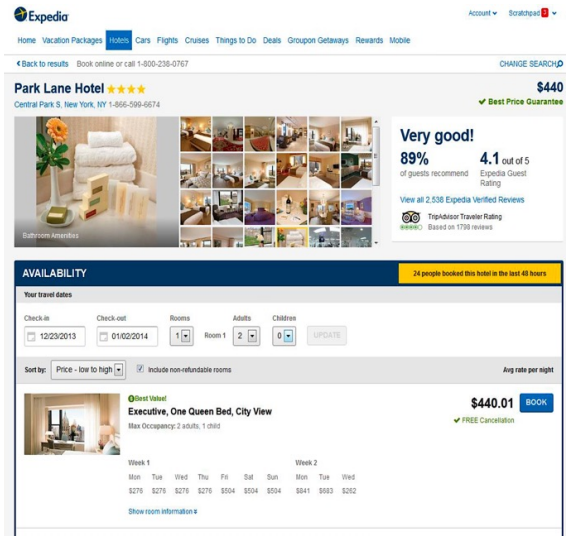


Figure 4.3: Click to product-specific page includes hidden product features



details. In most markets, the headline price on the landing page is the nightly rate of the cheapest available room. The Expedia data include two location desirability scores, which capture some of the hotel-specific information that consumers learn through clicks, as the landing page contains approximate but not specific locations.

Finally, in the purchase phase, consumers purchase one of the clicked hotels or end their search (choose the outside option). At this point, Expedia records the gross booking revenue. The final transaction price can be higher than the headline price, as it includes taxes, fees, and upgrades. The differences between the headline and final transaction prices introduce some uncertainty about transaction prices.

4.2 Data Processing

Preparing the Expedia data for analysis requires several data processing steps. The Expedia data were released for a data science competition and are well-suited for training recommendation systems. However, a few data limitations present difficulties in conducting the type of demand estimation and counterfactual analysis in this paper. I include additional data processing details in A.2).

Market Definitions via K-means clustering. I define markets by groups of search terms. The Expedia data are de-identified, meaning I have hotel and search term identifiers but no keys.[20] I use k-means clustering to group together search terms by the similarity of their search results (details in Appendix A.2.1).

Final Transaction Price Prediction. A limitation of this dataset is that it records final transaction prices only when there is a purchase. When a result for a hotel is clicked but no purchase is made, the consumer may still discern the final transaction price, but this price is omitted from the data. Transaction prices are important for two reasons. First, they influence consumer search and purchase decisions. A consumer might, through a click, learn the final price, which informs their next search or purchase decisions. Second, for an accurate measure of consumer welfare, the final prices are essential, as these represent the actual expenditures by consumers. To address this missing data issue, I impute the percent difference between the headline price and the final per-night transaction prices using the hotel-length of stay median (details in Appendix A.2.2).[21] Figure 4.4, displays the impression level distribution of imputed hidden price difference. In the top market (by revenue), we see a median price difference of 18%, with a thick right tail. We see a bimodal distribution in the second-ranked market with mass points around 0% and 20%. In both cases, we see variation in the pattern of hidden prices both within and across markets.

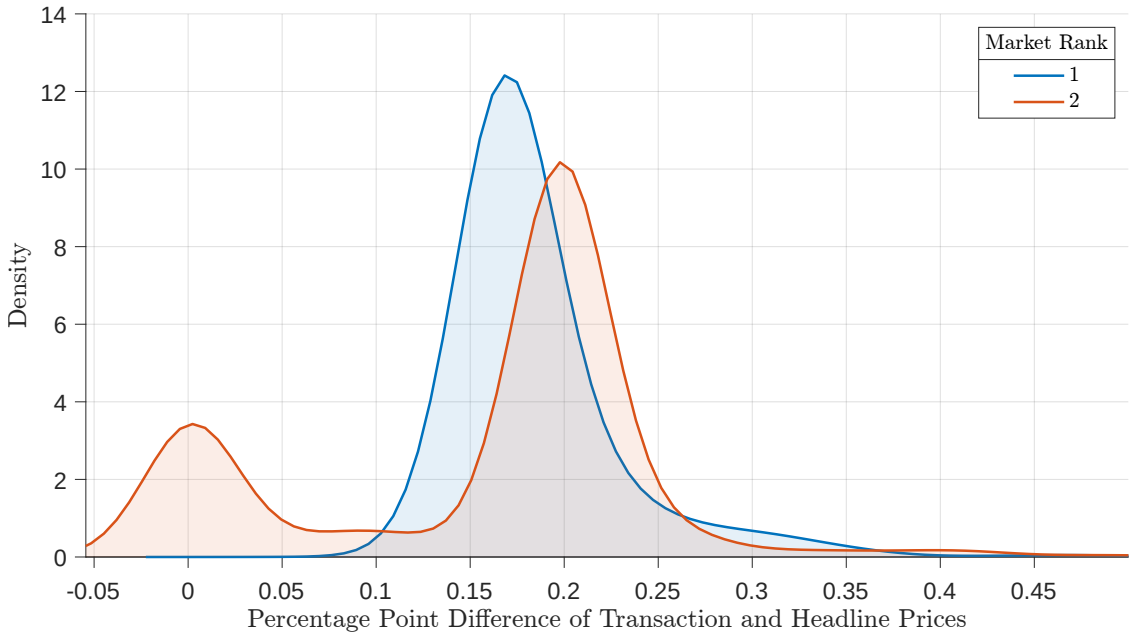
Click order prediction. The data includes indicators for clicks and purchases but does not include information about click orders. Less than 8% of consumer-queries have more than one click. I use a linear prediction model, detailed in Appendix A.2.3, to predict the click order.

Sample Selection. Three issues arise from the competition's data sampling method: 1) selection on clicks, 2) oversampling of transactions, and 3) ambiguity in the sample size. I address selection on clicks by using conditional likelihoods in demand estimation and selection weights on the supply side. I address the oversampling of transactions by using sample weights based on reported conversion rates in previous studies. I address the sample size ambiguity by comparing the cross-booking revenue from my data to publicly reported gross booking revenue from the same year. I detail each of the selection issues and solutions in Appendix A.3.

[20] For example, while an identifier might indicate "search term 52," there is no direct link to a specific term such as "Manhattan, NY."

[21] I use the hotel median for hotels with a limited number of transactions, while hotels with fewer than 3 observations are assigned the market-length of stay median. A potential concern with this methodology is that hotels could have modified their concealed pricing strategy during the study. I address this in Appendix A.2.2.

Figure 4.4: Impression Level Kernel Density of Hidden Price Differences by Market



4.3 Sample Restrictions

For the primary analysis, I focus on the largest market in terms of revenue. However, I develop the platform model and the recommendation systems I use in counterfactual using data from all five markets. In each of these machine learning applications, jointly estimating the model for each is helpful as since there could be cross-market learning spillovers. In the project's next phase, I intend to expand the analysis to additional markets.

5 Empirical Facts, Position Effects, Incentives

As in brick-and-mortar stores, where product placement on shelves (e.g., at eye level) influences consumer decision making, the positioning of products on digital "shelves" in slots on search result pages can influence consumer behavior and seller outcomes. The term "position effects" refers to the influence that position has on consumer behavior and seller outcomes, which is well established in the empirical literature (Ursu, 2018; Donnelly et al., 2020; Greminger, 2022), but its underlying mechanisms are still unclear.

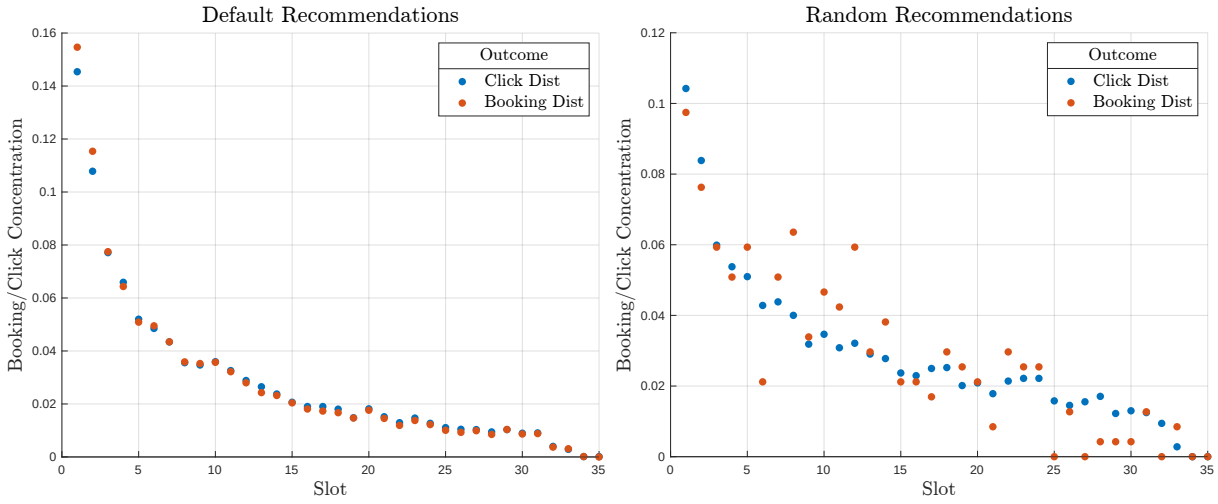
5.1 Three Empirical Facts

This section presents three empirical facts that inform how I construct the structural model.

Fact 1: Position impacts demand even when recommendations are random

The first fact documents position effects under default and random recommendations. Figure 5.1 plots click and purchase concentrations by slot under Expedia's default recommendation system (left) and under randomized recommendations (right). Blue points denote the percent of all bookings (under the given recommendation system) attributed to the given slot. Similarly, clicks denote the percent of all clicks in the given slot. The figures use concentration instead of levels to avoid misleading comparisons across recommendation systems, as observations with purchases were sampled at different rates for the default versus random recommendations.

Figure 5.1: Click and Booking Concentration by Slot and Recommendation System



If consumers were fully informed about products, we would expect to see position effects under the default recommendations (left) but not random recommendations (right). We might still expect to see position effects under default recommendations only due to the correlation between slot and desirable product features. However, since features and slots are uncorrelated in the random rankings, we would expect to see uniformly distributed clicks and purchases. Instead, the data show that clicks and purchases are concentrated in the slots higher on the page, implying that position effects depend on more than visible features. For example, search cost could depend on the slot. Consumers might also have beliefs (rational expectations) about the relationship between hidden product features and slots.

Fact 2: Hidden product features are correlated with slots

The second fact focuses on the relationship between hidden product features and product recommendations. The data include two location desirability scores. These scores can be considered hidden features since a general property location appears on the landing page, but the specific location appears on the product-specific page. Imagine, for example, a consumer searching for a beach vacation. They will see on the landing page that a property is near the beach but can only learn if it is a beachfront property after clicking on the property-specific page. Figure 5.2 plots relative location desirability scores by slot. The scores are demeaned on the consumer-query level since location scores can differ from market to market.

Figure 5.2: Hidden Features By Slot: Demeaned Location Desirability Scores

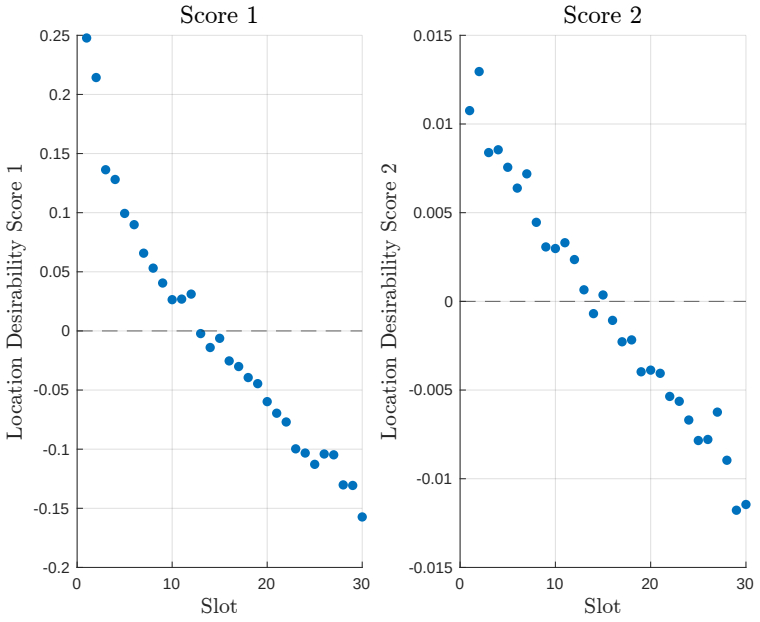


Figure 7.1 shows a correlation between slots and location desirability. This means that, on average, a product slotted higher on the page has higher location desirability scores than lower-ranked products in the same search. This correlation is unsurprising since the default recommendation system is likely a function of past consumer decisions, which partly depend on location desirability.

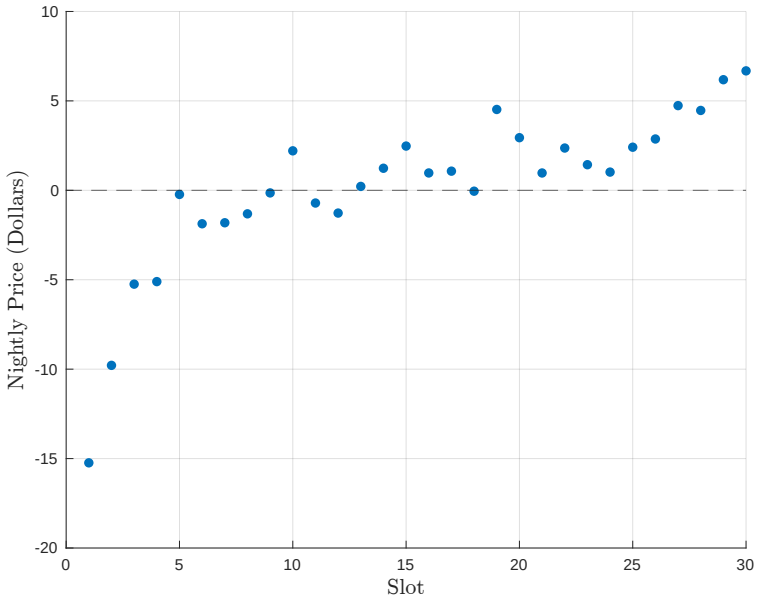
This correlation opens the possibility for a mechanism where consumers "trust the algorithm." In other words, consumers could have the accurate belief (rational expectations) that products positioned higher on the page have superior hidden features. A common approach in the empirical literature assumes that slot only impacts demand through search cost, abstracting from rational expectations about the recommendation system.

It is important to consider the rational expectations mechanism for two reasons. First, to separately identify search cost and utility parameters, practitioners often rely on experimental data with randomly ordered slots. However, with rational expectations, this could be an issue since, by design, consumers do not know they are receiving random recommendations and would still behave based on their beliefs about the default recommendation system. Second, failing to capture beliefs about the recommendation system could bias estimates of search cost; if consumers, in part, search for products higher on the page because they tend to have better-hidden features, then we would conclude from a model without rational expectations that consumers have extremely high search costs.

Fact 3: Price is correlated with slot

This fact centers on the relationship between price and position on the page. Figure 7.1 plots the mean relative headline price difference by slot. For example, a value of -$15 for the first slot implies that, on average, for a given consumer query, the hotel in the first position has a headline price that is $15 cheaper per night than the average hotel in the search results.

Figure 5.3: Headline Price by Slot (Demeaned)



The pattern in these data suggests not only that products in higher slots tend to be lower-priced but also could indicate that Expedia's default recommendation system assigns slots as a function of price—a fact confirmed through my analysis with the platform model. This implies that sellers can potentially improve their slot positioning by lowering prices, and that counterfactual policies altering

recommendations could consequently affect prices as well. In light of the apparent relationship between prices and slots, I develop the supply side of the structural model to account for the fact that prices impact demand both directly, through consumers' disutility of price, and indirectly, through positioning by the recommendation system.

5.2 Implication of Empirical Facts

The first two facts inform the structure of my demand model and my choice of data. In the primary specification of my demand model, I allow slots to impact consumer decision-making through search costs and rational expectations. I also test these structural assumptions by benchmarking the demand model against a competing one, with the more conventional assumption that slot impacts demand only through search cost.

For my primary demand specification, I use the naturally ordered data instead of the randomized data. The naturally ordered data have two key advantages. First, because the demand model includes rational expectations about the recommendation system, the naturally ordered data better reflect consumers' equilibrium behavior, where recommendations align with consumer beliefs. In contrast, consumers are initially unaware when they receive randomized rankings, making the randomized data less suitable for capturing equilibrium behavior. Second, using the naturally ordered data allows me to construct observation weights that address sample selection issues. Additional sources provide information on Expedia's gross booking revenue and typical click-through and purchase rates in OTAs and e-commerce settings, which I can use to inform these observation weights. No such sources are available to make sample selection adjustments for the randomly ordered data. I discuss the sample weights further in Section A.3.

The third fact informs the supply-side of the model and counterfactual simulations. Since price impacts position, sellers current pricing strategy depends on the default recommendation systems. A change in recommendation systems would, changes the relationship between prices and recommendations, which changes pricing incentives.

6 Structural Model

To understand the welfare effects of personalized recommendations, I develop a structural model of consumer demand, product recommendation systems, and hotel pricing behavior. The demand side consists of an optimal sequential search model where consumers have beliefs about the joint

distribution of product features and recommendations, form consideration sets through clicks, and make a final purchase decision from their consideration set. For the product recommendation model, I use a "model-of-a-model" machine-learning approach to reverse-engineer Expedia's default recommendation system. Combining the results from the demand and recommendation system models allows me to construct the supply side of the model, where capacity-constrained hotels consider how changes in price impact their position on the page in search results.

6.1 Demand

In this section, I describe the individual demand model, an optimal sequential search model based on Weitzman (1979), where consumers have beliefs about the joint distribution of product features and recommendations, form consideration sets through clicks, and make a final purchase decision from their consideration set. The search model has three consumer–product-specific components: indirect utility, search cost, and reservation utility. I use these to construct the final utility and the likelihoods required for estimation. This model requires three basic subcomponents: utility, search cost, and reservation utility. The utility and search cost estimates follow directly from the model parameters and observable, while the reservation utility can be expressed using a value function.

6.1.1 Demand Timing

Consumers arrive to the platform exogenously with queries for a specific market and length of stay. The demand model captures consumer search and purchase decisions. Note that I model consumer behavior once consumers are on the platform (including their choice of the outside options), not the decision to search in the first place.[22] Consumers can click, make a purchase, or choose the outside option. The outside option is to end the search without making a purchase.

6.1.2 Final Utility

Consider consumer i conducting a search at time t, for a stay at time t', with a length of stay x_{it}^{nights}.[23][24] We can express the consumer's final utility from their search and purchase decisions as:

$$U_{it} = x_{it}^{\text{nights}} u_{ijt}^{\text{choice}} - \sum_{j \in S_{it}} c_{ijt} \tag{6.1}$$

[22] Hortaçsu et al. (2021) develops a demand estimation approach that incorporates consumer arrival and applies it to air travel demand.

[23] The model endogenizes search and purchase decisions but takes consumer arrivals as exogenous.

[24] In most cases, I dropped the t' subscript for clarity, as each consumer search identifies t'.

where x_{it}^{nights} is the number of nights, u_{ijt}^{choice} is the per-night utility of consumer i's choice, and I subtract the incurred search cost of each product that consumer i added to their consideration set. While utility and reservation utility depend on the length of stay, the search costs that consumers face do not. I will return to this fact in estimation, as this difference allows me to separately identify utility and search costs parameters.

6.1.3 Benchmark Indirect Utility

As a benchmark, it is helpful to consider the full-information demand setup and then highlight how the search model differs. In the full-information demand setup, I assume that the consumer i's utility for product j has two additively separable components:

$$u_{ij} = \delta_{ij} + \varepsilon_{ij} \tag{6.2}$$

where δ_{ij} denotes the part of utility observed by the researcher, and ϵ_{ij} represents the portion of utility known to consumers but not observed by the researcher. In full-information demand models, consumers know ϵ_{ij} for each product. In the typical search models used in empirical work, consumers know δ_{ij} and pay a search cost to learn ϵ_{ij}. A few papers have consumers know part of ϵ_{ij} prior to search, and learn part of ϵ_{ij} after search.

6.1.4 Indirect Utility Visible and Hidden Product Features

In the context of e-commerce platforms such as Expedia, the assumption that consumers know product features prior to search can be overly strict. Platforms choose their feature emphasis, which determines which product features appear on the landing page and which appear on product-specific pages. Incorrectly assuming that consumers are perfectly informed about product features would bias parameter estimates Abaluck et al. (2020). For example, assuming that consumers are perfectly informed about prices could lead to underestimates of price parameters, as consumers would appear to not react to price differences among unsearched products.

This paper presents a formalized decomposition of indirect utility, distinguishing between "visible" and "hidden" components. Reparametrizing the indirect utility function, we have

$$u_{ij} = \underbrace{\delta_{ij}^{v} + \varepsilon_{ij}^{v}}_{\text{Visible}} + \underbrace{\delta_{ijt}^{h} + \varepsilon_{ij}^{h}}_{\text{Hidden}} \tag{6.3}$$

where δ^v_{ijt} denotes the part of utility observed by the researcher and known to the consumer before search. δ^h_{ijt} is the part of utility observed by the researcher and known to the consumer only after search. Similarly, ϵ^v_{ij} represents the portion of utility known to the consumer prior to search but not observed by the researcher. ϵ^h_{ij} is the portion of utility not observed by the researcher and known to the consumer only after costly search. I detail the structure of the two match quality terms below.

6.1.5 Visible and Hidden Variance Components of the Match Quality Term

In the demand model, I distinguish between visible and hidden product features. The match quality term follows a similar structure, with visible and hidden components, but with an added parameter λ that determines how much of the match quality term is known before search and how much is learned from search along with the hidden product features. We can express the sum of the terms as

$$\epsilon_{ijt} = \lambda\varepsilon^v_{ijt} + \varepsilon^h_{ijt}(\lambda) \tag{6.4}$$

where ϵ_{ijt} is consumer i's match quality for product j at time t and follows an i.i.d. type-1 extreme value distribution. ε^v_{ijt} is the match quality known before search and follows an i.i.d. type-1 extreme value distribution and is multiplied by $\lambda \in (0, 1)$. ε^h_{ijt} follows a Cardell(λ) distribution , whose characteristic function depends on λ.

To achieve this structure, I use a novel application of the properties of the variance components of the type-1 extreme value distribution established in Cardell (1997).[25]. I also use recent advances by Galichon (2022), which proves a relationship between the Cardell distribution and stable distribution. For more details, see Appendix C.1.

6.1.6 Indirect Utility

Rewriting nightly utility to include the lambda terms, we have the expressions

$$u_{ijt} = \delta^v_{ijt} + \delta^h_{ijt} + \lambda\varepsilon^v_{ijt} + \varepsilon^h_{ijt}(\lambda)$$

The value of the outside option is

$$u_{i0t} = \varepsilon_{i0t} \tag{6.5}$$

[25]These properties are often used to construct nested logit models, such as Berry (1994), which includes nest-level and the item-level variance components of the error term.

Section 7.1.1 details the primary specification of δ^v_{ijt} and δ^h_{ijt}. Relating this to aspects of platform design, the observable product features that appear on the landing page enter utility through δ^v_{ij}, and the product features relegated to the product pages belong to δ^h_{ijt}. Similarly, consumers may intuit a portion of the match quality, for example, from product photos or prior searches, which enter through ε^v_{ij}. The hidden portion of the match quality term is ε^h_{ij}. Consumers know the utility of the outside option, u_{i0t}, prior to search.

6.1.7 Search Cost

I assume that the consumer is not fully informed about the hidden components of utility, δ^h_{ijt} and ε^h_{ij}, and must pay a search cost c_{ijt} to learn them.

$$c_{ijt} = f(\theta_i, \text{slot}^{\text{appear}}_{ijt}) \tag{6.6}$$

where c_{ijt} is consumer i's search cost for product j. θ_i is the set of consumer-specific search cost parameters, and $\text{slot}^{\text{appear}}_{ijt}$ denotes the position on the page for product j in search it. Advertisements for opaque offers, on occasion, displace products in the slot. Section 7 details the functional form of search costs, which allows for heterogeneous search costs and flexibly captures the relationship between c_{ijt} and $\text{slot}^{\text{appear}}_{ijt}$.

6.1.8 Reservation Utility in Optimal Sequential Search

Suppose consumer i has already clicked on $r-1$ products. Their maximum utility among the clicked options is $u^*_{i(r-1)} = \max_{k=0}^{r-1}\{u_{ik}\}$. To save on notation, I drop the $r-1$ subscript and let u^*_i refer to the maximum utility among the clicked options at any stage in the search process. The consumer's expected marginal benefit from searching for (in this case, clicking on) item r is given by Weitzman (1979) as:

$$B_{ir}(u^*_i) = \int_{u^*_i}^{\infty} (u_{ir} - u^*_i) f_{u_{ir}}(u_{ir}|\Omega_{it}) du_{ir} \tag{6.7}$$

where $f_{u_{ir}}(u_{ir}|\Omega_{it})$ is the probability density of u_{ir} conditional on information Ω_{it}. In the general case, search continues as long as there is a unsearched product where expected benefit exceeds the cost.

$$B_{ir}(u^*_i) > c_{ij} \tag{6.8}$$

For each target product, there is an indifference point where $B_{ir}(u^*_i) = c_{ij}$ such that the

consumer is indifferent between receiving r_{ijt} with certainty and continuing to search. Under optimal sequential search, consumers search in order of reservation utility. We can define reservation utility, r_{ijt}, for consumer i, product j in search t as the value that satisfies the following equality:

$$c_{ijt} = \int_{r_{ijt}}^{\infty} (u_{ijt} - r_{ijt}) f_{u_{ijt}}(u_{ijt}|\Omega_{it}) du_{ijt} \tag{6.9}$$

where r_{ijt} is the level of per-night utility that would make consumer i indifferent between receiving r_{ijt} with certainty or paying search cost c_{ijt} to learn u_{ijt} given information set Ω_{it}. Since consumers know the visible part of utility and learn both δ^h_{ijt} and $\varepsilon^h_{ijt}(\lambda)$, the reservation utility depends on the consumer's beliefs about and the distribution of $\delta^h_{ijt} + \varepsilon^h_{ijt}(\lambda)$. We can rewrite this condition as

$$c_{ijt} = \delta^v_{ijt} + \lambda\varepsilon^v_{ijt} + \int_{r_{ijt}}^{\infty} (\delta^h_{ijt} + \varepsilon^h_{ijt}(\lambda) - r_{ijt} - \delta^v_{ijt} - \lambda\varepsilon^v_{ijt}) f_{u_{ijt}}(\delta^h_{ijt} + \varepsilon^h_{ijt}(\lambda)|\Omega_{it}) d(\delta^h_{ijt} + \varepsilon^h_{ijt}(\lambda)) \tag{6.10}$$

A common approach follows Kim et al. (2010), where each utility component except the error term is assumed to be known before search, and search reveals the match quality term. In that setup, $f(u_{ir}|x_{ij})$ depends only on the distribution of the match quality term. Alternatively, in my model, since consumers also learn δ^h_{ijt}, which depends on product features, $f(u_{ir}|x_{ij})$ also depends on the distribution of δ^h_{ijt}.

With some additional algebra, I can rewrite this equation to express reservation utility as

$$r_{ijt} = \delta^v_{ijt} + \lambda\varepsilon^v_{ijt} + E_i[\delta^h_{ijt}|\Omega_{it}] + \zeta_{ijt} \tag{6.11}$$

where δ^v_{ijt} and $\lambda\varepsilon^v_{ijt}$ are the visible portion of utility, $E_i[\delta^h_{ijt}|\Omega_{it}]$ is consumer i's expectation of δ^h_{ijt} conditional on their information set Ω_{it}, and ζ_{ijt} is the portion of the reservation utility that does not have a closed-form expression that allows r_{ijt} to satisfy the equality in equation (6.10). We can write ζ_{ijt} as a value function

$$\zeta_{ijt} = V(c_{ijt}, \theta^h_i, \lambda, x^{\text{nights}}_{it}|\Phi_{it}, \Omega_{it}) \tag{6.12}$$

where the state variables are c_{ijt}, the search cost, θ^h_i, consumer i's utility parameters for hidden product features, λ, and x^{nights}_{it}, the length of stay. Φ_{it} is the distribution hidden utility, and Ω_{it} is the consumer's information set.

In estimation, I solve r_{ijt} numerically since r_{ijt} does not have a closed-form expression.

6.1.9 Structural Assumptions

The model requires some structural assumptions, which I document here. As stated earlier, the researcher decides which product features are hidden and which are visible. Additionally, since I treat some product features as hidden, I need structural assumptions about consumers' beliefs about hidden product features. I assume that consumers have rational expectations of hidden utility, with two related components that impact reservation utilities, $E_i[\delta^h_{ijt}|\Omega_{it}]$ and ζ_{ijt}.

For $E_i[\delta^h_{ijt}|\Omega_{it}]$, I assume that consumers form rational expectations of δ^h_{ijt}, conditional on their information set. For the non-price components of δ^h_{ijt} consumers' information set includes the star rating, $slot^{\text{rank}}_{ijt}$, and if the product is on promotion. For the price component of δ^h_{ijt}, the final transaction price, consumers know the headline price and the median percent difference between the headline and final prices.[26] For the ζ_{ijt} component of utility, I assume that consumers know the distribution of $\delta^h_{ijt} + \varepsilon^h_{ijt}(\lambda) - E_i[\delta^h_{ijt}|\Omega_{it}]$.

6.1.10 Mechanisms for Position Effects

Now that we have expressions for utility, search cost, and reservation utility, we can discuss how product positioning in the search results impacts demand. The standard approach in the empirical literature imposes a structural assumption that position impacts demand only through search cost. The demand model allows for three mechanisms:

Mechanism 1 Search cost c_{ijt}: Position on the page impacts search cost. This is captured by including $slot^{\text{appear}}_{ijt}$ in the search cost function.

Mechanism 2 Expectation of δ^h_{ijt}: Consumers have accurate beliefs (rational expectations) about the relationship between position and mean hidden utility. This can be captured through $E_i[\delta^h_{ijt}|\Omega_{it}]$ by including the slot in Ω_{it}.

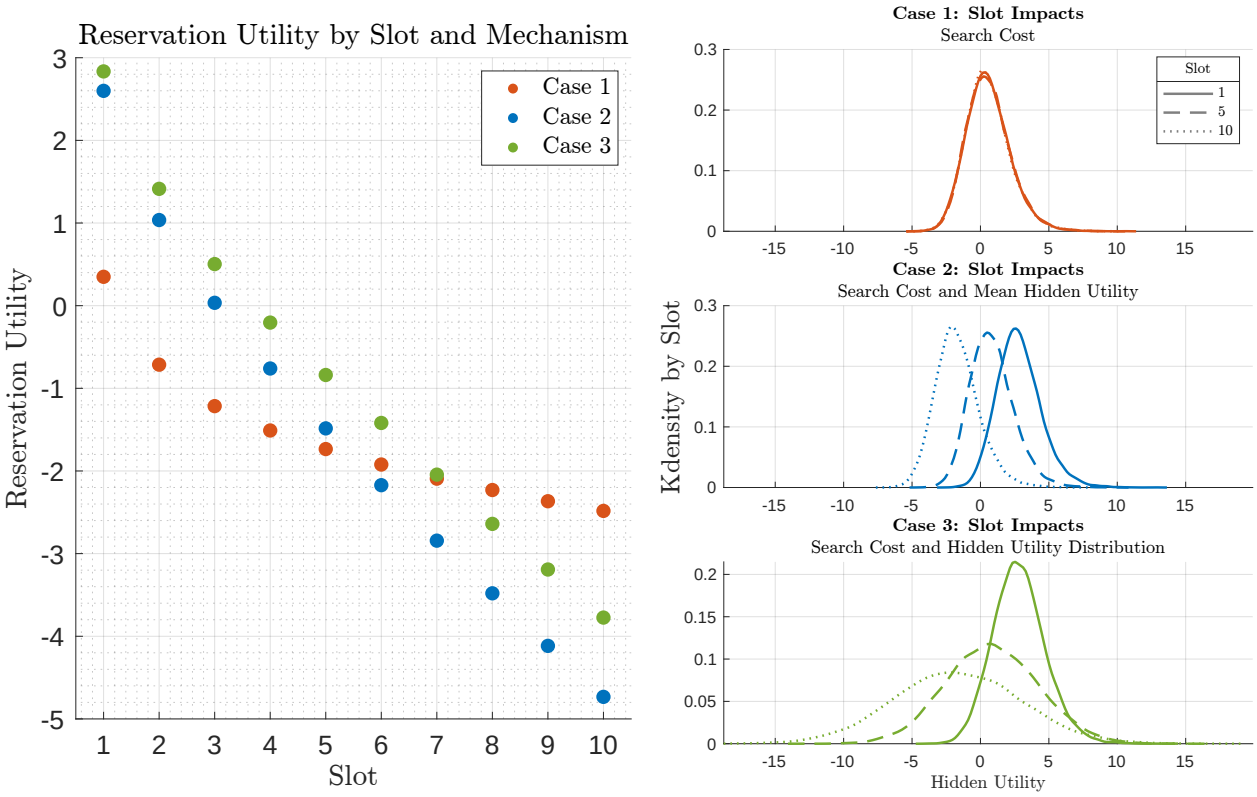
Mechanism 3 Higher-order beliefs: Consumers have beliefs about the relationship between position and the distribution of hidden utility. This can be captured by including the position as an additional state variable in the value function of ζ_{ijt}.

Mechanism 1 is standard in the empirical literature. However, mechanisms 2 and 3 require some additional explanation. Regarding mechanism 2, a consumer might expect that products higher versus lower on the page have different hidden features. In section 5, we see the correlation between

[26] $E_i[price_{ijt}|\Omega_{it}] = (1+\tau_m)price^{headline}_{ijt}$ where τ_m is the market-level median % hidden price difference.

slot and hidden product features. Another way to think about this is that consumers "trust the algorithm", perceiving products higher on the page as more promising. Regarding mechanism 3, higher-order beliefs, a simplification is that consumers believe the variance of the hidden features to be correlated with slot.[27]

Figure 6.1: Simulation Results of Position Effect Mechanisms



It is important to consider these mechanisms to accurately measure consumer welfare and estimate the correct substitution patterns. Incorrectly excluding one of the mechanisms, may cause in biased results. Consider an example where the true data generating process includes mechanisms 1 and 2, so slot would impact demand both through search cost and rational expectations, but the model only included the search cost mechanism. Observed clicks and purchases purchases would be concentrated near the top of the page because of search cost and rational expectations, but the model would only be able to explain the pattern in the data through search cost. As a results we would overestimate search cost.

These mechanisms also highlight a limitation of A/B tests and randomized data. If consumers have accurate beliefs about the relationship between slot and hidden utility, there still exists an

[27]For this iteration of the paper, I do not include this mechanism in the empirical model; however, in work in progress, it will be added.

endogeneity concern with the randomly ordered data. The randomization does address collinearity between features and search costs, but consumers do not know they are in a random treatment group. As a result, their search and purchase decisions would still depend on their beliefs about the recommendation system, i.e., mechanisms 2 and 3.

Accounting for each of these mechanisms requires making different structural assumptions. To validate the assumptions used in the estimation, I re-estimate the demand under alternative assumptions and compare the in- and out-of-sample likelihoods. The results of this exercise are presented in section 7.1.7.

6.1.11 Characteristics Impacting Reservation Utility

The model also accommodates characteristics that impact demand through reservation utility but do not affect search cost or utility. For example, a promotion on a property, visible on the landing page, certainly impacts utility through the lower price. However, once we control for the price difference, the promotion itself would not change the desirability of the property unless we have a model where consumers derive pleasure from finding "a deal." Nevertheless, promotions may be correlated with hidden product features and thus enter the reservation utility as part of Ω_{it} in $E_i[\delta^h_{ijt}|\Omega_{it}]$.

Similarly, as advertisements or opaque offers may occasionally displace products on the page, I have two variables to keep track of the position on the page: $slot^{appear}_{ijt}$, which denotes where the product appears on the page, and $slot^{rank}_{ijt}$, which denotes the product ranking among non-advertisement products. While $slot^{appear}_{ijt}$ enters the search cost, $slot^{rank}_{ijt}$ enters Ω_{it}, which impacts reservation utility but not utility or the search cost. For example, a product ranked fifth might be displaced by an advertisement in the fifth slot and appear in the sixth slot.

6.2 Platform Model of Product Recommendation Systems

This section outlines the platform model, which relates to Expedia's default recommendation system. The platform model and estimates are a necessary component of the structural model to estimate marginal costs on the supply side (hotels) and to provide a baseline for the counterfactual simulations. As discussed in section 5, product rankings (slots) play a pivotal role in consumer decision-making. Expedia's recommendation system assigns slots based on query, consumer, and product features, including price. Consequently, firms aiming to maximize profits encounter an elasticity of demand influenced not only by consumer preferences but also by the platform's design. Specifically, a change in a product's price can shift its position in the search results when consumers look for hotels.

The platform model aims to reflect Expedia's default recommendation system accurately. The goal is to generate product recommendations based on a set of queries that match the ranking probabilities of the actual recommendation system and also to capture how a price adjustment for a product alters its likely position in the search results. To achieve this, I use a "model-of-a-model" approach from machine learning to reverse-engineer Expedia's recommendations system.

Expanding on the recommendation system's mechanics, we can write the recommendation systems set up in the format of a demand model with indirect utility but instead thinking of u^r_{ijts} as product j's relevance score for slot s in consumer i's searching at time t:

$$u^r_{ijts} = \omega_s \psi_{ijt} + \epsilon_{ijt} \tag{6.13}$$

where u^r_{ijts} denotes the slot-s relevance score of product j for consumer i's query at time t. $\psi_{ijt} = f(x^r_{ijt})$ defines the deterministic portion of the relevance score, which depends on x^r_{ijt}, a set of consumer, product, and query features. The deterministic score is scaled by ω_s for each slot s. The relevance score also includes some experimental noise, ϵ_{ijt}, which follows a type-1 extreme value distribution. The scale term ω_s is slot-specific since the underlying recommendation system may be relatively more deterministic for some slots than for others. Another way to think of this is that ψ_{ijt} is a seller's (hotel's) expected relevance score conditional on x^r_{ijt}, the information available to the hotel, and ε_{ijt} is the error term.[28]

While this formulation bears similarities to the setup of traditional demand models, there are notable differences. First, an algorithm determines the product recommendations, so the objective of the platform model is to back out the preferences of a single, sophisticated machine. Second, in a demand model, one might expect data on revealed preferences to take the form of clicks and purchases. In the platform model data, the revealed preferences of the algorithm are the complete list of first-page rankings.

In modeling the platform's product recommendation system, it is important to consider that the underlying recommendation system can be a complicated black-box. E-commerce platforms devote significant resources to developing their recommendation systems, using historical data and learning-to-rank methods including neural networks (Ranknet), collaborative filtering (matrix factorization), and gradient-boosted machines (LambdaMART). These methods introduce nonlinearities and high-

[28] Sometimes platforms add noise to the rankings. This can provide useful variation for training future recommendation systems and can prevent price-undercutting strategies in which firms move their price to one cent lower than a competing firm's to move up the ranking.

dimensional interactions, making a straightforward parametric approach prone to misspecification.

Instead of a parametric approach, such as rank-ordered logit, I use a "model-of-a-model" approach from machine learning, also known as model extraction. A growing body of work in the computer science, machine-learning, and cryptography literature demonstrates cases where black-box machine-learning algorithms can be reverse-engineered by training a new model on data generated from queries to the black-box model and the black-box model's results (Papernot et al. (2017); Orekondy et al. (2020)).

In section 7.2.2, I outline the estimation procedure to generate the platform model. The estimation procedure involves two steps. In step one, I estimate the function $\psi_{ijt} = f(x^r_{ijt})$ using LambdaMART, a machine learning algorithm used for ranking Burges (2010). Next, I make out-of-fold predictions, $\hat{\psi}_{ijt}$ and estimate the scale terms ω_s for each slot using conditional logit.

6.3 Supply-Side Model of Hotel Pricing

This section describes the supply side, where capacity-constrained hotels set price schedules and consider how changes in price impact position on the page in search results. I identify hotels in the data based on a property identifier. I treat each hotel as operating independently since I do not observe the hotels' ownership structure.

I focus on the hotels' pricing decisions. However, it is worth noting that there are several decisions hotels can make. They decide prices, can activate promotions on Expedia, and decide whether to sell through Expedia or other platforms such as Booking. They can also choose what percent of the final transaction price to hide from the search results. On the supply side, I hold these decisions fixed, with firms selecting only prices.[29] With these limitations in mind, I model the seller as setting prices to maximize expected profits:

$$\underset{p_{jtt'}}{\operatorname{argmax}} E\left[((1-\varphi)p_{jtt'} - c_{jtt'})q_{jtt'} \mid \Omega_{jtt'}\right] \tag{6.14}$$

where $p_{jtt'}$ is the price for room–night j, staying period t, and searching in period t'. φ is the percentage of revenue that goes to the platform and taxes, assuming that both taxes and platform fees are a percent of gross booking revenue. $c_{jtt'}$ denotes the average variable cost. $q_{jtt'}$ is the expected quantity purchased through Expedia; and $\Omega_{jtt'}$ is a hotel's information set, including the own costs, demand elasticities, the features and availability of other products in the same market,

[29] I conduct the modeling in terms of final price and keep the ratio of hidden-to-visible prices fixed.

and market size. For market size, I assume that the arrival rate of consumers to Expedia is known to firms.

6.3.1 Opportunity Cost Interpretation of Average Variable Cost

The firm's problem depends on the average variable cost, $c_{jtt'}$. For this model, $c_{jtt'}$ is a reduced-form object; it is helpful to discuss its interpretation. The hotel sets prices and faces capacity constraints but risks the room–night remaining vacant if it does not sell it by the time of the stay ($t' \geq t$). The interpretation of $c_{jtt'}$ is as the opportunity cost of having room–night jt available to sell in period $t' + 1$.[30]

6.3.2 Ancillary Revenue

Another aspect of the hotel industry worth noting is that hotels make additional profits post-booking. Hotels' additional goods and services such as room service, bars, and restaurants generate ancillary revenue. In the extreme, we might think about hotel-casinos, where the rooms themselves can be a loss leader, with profits coming from the casino. These ancillary revenue streams are embedded within the $c_{jtt'}$ value.

6.3.3 Economies of Scale and Capacity Constraints

In the accommodation industry, average variable costs depend on quantity. Hotels may face economies of scale at low quantities, as adding another guest may not require additional staffing. At high quantities, hotels may face increasing marginal costs, for example, from needing to pay overtime to meet staffing requirements. Further, hotels face capacity constraints, which imply increasing opportunity cost as quantity approaches the capacity constraint. I capture these features by allowing average and marginal costs to depend on quantity and quantity squared. Equations 6.15 and 6.16 specify the marginal and average variable costs functions in relation to quantity. These costs exclude any of the large fixed costs typical of the hotel industry.

[30] Betancourt et al. (2022) details a similar dynamic game for the airline industry. A few limitations prevent me from fully modeling the dynamic game. For example, Betancourt et al. (2022) focuses on two competing airlines and observes quantity and capacity. In contrast, I observe a random sample of quantity but not the capacity constraint, and I have hundreds of competing hotels in the market.

Cost Functions

$$\text{average cost:} \quad c_{jtt'} = mc^{\text{base}}_{jtt'} + \frac{1}{2}\gamma_{1jtt'}q_{jtt'} + \frac{1}{3}\gamma_{2jtt'}q^2_{jtt'} \tag{6.15}$$

$$\text{marginal cost:} \quad mc_{jtt'} = mc^{\text{base}}_{jtt'} + \gamma_{1jtt'}q_{jtt'} + \gamma_{2jtt'}q^2_{jtt'} \tag{6.16}$$

where $mc^{\text{base}}_{jtt'}$ is the variable cost associated with providing one additional unit of accommodation before considering any effects from economies of scale or increasing costs due to higher occupancy levels. This represents the base per-unit cost of accommodating a guest for hotel j, for a stay at time t, and at search time t'. $\gamma_{1jtt'}$ is a negative relationship between marginal cost and quantity that captures economies of scale. $\gamma_{2jtt'}$ is a positive term on quantity squared that captures increasing costs and serves as a soft capacity constraint.

This approach is similar to that in Farronato and Fradkin (2022), which models hotel capacity constraints with a hockey stick–type function, with flat cost for low quantity and then linearly increasing cost above 85% occupancy. I do not observe capacity; however, the polynomial specification in terms of quantity should be able to capture the inflection point where marginal costs increase. In the primary specification, γ_1 and γ_2 are star-rating specific. However, with enough data, one could specify a firm-specific γ_1 and γ_2.

6.3.4 Seller First-Order Condition

We can take the derivative of the seller's problem with respect to $p_{jtt'}$ to obtain the profit-maximizing first-order condition:

$$\frac{mc_{jtt'}}{(1-\varphi)} = p_{jtt'} + \left(\frac{\partial q_{jtt'}}{\partial p_{jtt'}}\right)^{-1} q_{jtt'} \tag{6.17}$$

I do not observe the percent of revenue that goes to taxes and fees, φ, so I express the marginal cost as a ratio of $(1-\varphi)$. This is not an issue for estimation as long as φ remains fixed.

If we allow marginal cost to depend on quantity, the numerator of the left-hand side of the problem becomes $mc_{jtt'} = mc^{\text{base}}_{jtt'} + \frac{\partial c_{jtt'}}{\partial q_{jtt'}}q_{jtt'}$. In section 7.3, I discuss estimating costs. Since costs depend on quantity, I need an additional model of cost and instruments for quantity and quantity squared.

7 Estimation and Results

This section discusses the estimation procedure and results for the empirical model presented in the previous section. I estimate the demand and platform models separately. I then use their combined results to estimate the supply-side model.

7.1 Demand Estimation and Results

I use the optimal sequential search rules from Weitzman (1979) and logit smoothing techniques covered in Train (2009) and proposed in McFadden (1989) to construct the joint likelihood of clicking and booking decisions.

7.1.1 Utility Specification

Writing out the primary per-night utility specification, we have:[31]

$$\text{inside option: } u_{ijtt'} = \underbrace{\beta_i^v x_{jtt'}^v + \xi_t^{\text{month}} + \xi_t^{\text{day}}}_{\delta_{ijt}^v} \underbrace{-e^{\rho_i} \overbrace{p_{jtt'}}^{\text{final price}} + \beta_i^h x_j^h}_{\delta_{ijt}^h} + \underbrace{\lambda \varepsilon_{ijtt'}^v + \varepsilon_{ijtt'}^h(\lambda)}_{\text{match quality} \sim EV1} \tag{7.1}$$

$$\text{outside option: } u_{i0tt'} = \alpha_0 + \varepsilon_{i0tt'} \tag{7.2}$$

where x_j^v are the visible product features, p_{jt} is the final transaction price (including taxes and fees), and x_j^h are the hidden product features. The price coefficient, e^{ρ_i}, follows a log normal distribution. x_j^v includes indicators for star-rating, an interaction on brand and star rating, a linear spline of review score, indicators for missing values, and consumer segment information. The consumer segments are quantiles based on the booking window of the stay, the time of the search (morning, working hours, evening, and weekend or weekday), and the length of stay. The hidden features include splines on both location desirability scores and a missing indicator for location desirability score 2. The time effects ξ_t^{month} and ξ_t^{day} control for the market–month and market–day of the week of the stay, respectively. The final price p_{jt} is assumed to be hidden, as consumers see the headline price on the landing page.

The utility specification includes random coefficients on the indicators for star rating, brand, and price. The price coefficient is correlated with search costs. I estimate the elements of the Cholesky decomposition of the random coefficient covariance matrix.

[31] For the rest of this section, I again drop the t' subscript.

To save on notation, I use the following expression for within-estimation utility:

$$u_{ijt}^{[s]} = \delta_{ijt}^{v[s]} + \delta_{ijt}^{h[s]} + \lambda\varepsilon_{ijt}^{v[s]} + \varepsilon_{ijt}^{h[s]}(\lambda) \tag{7.3}$$

This notation adds a superscript $[s]$ that indexes the set of draws, so $u_{ijt}^{[s]}$ means that this is simulated per-night utility for consumer i, product j, for stay at time t, (searching at time t'), and simulated draws s. The draws are scrambled-Halton draws for utility parameters, the random coefficients, and the error terms.

The hidden match quality term, $\varepsilon_{ijt}^{h[s]}(\lambda)$, depends on the parameters λ. Each iteration of the estimation loop requires producing a new $\varepsilon_{ijt}^{h[s]}(\lambda)$. Prior to estimation, I take Halton draws that do not change during estimation, then use the approximate Cardell distribution to obtain $\varepsilon_{ijt}^{h[s]}(\lambda) = \text{ICDF}(\lambda, d_{ijt}^{[s]})$.

7.1.2 Search Cost

The search cost follows from the model parameters and $\text{slot}_{ijt}^{\text{appear}}$.

$$c_{ijt}^{[s]} = \log\left(1 + \exp\left(\kappa_i^{[s]} + \sum_{k\in K}\tau_k\left(\log\left(\text{slot}_{ijt}^{\text{appear}}\right) - \gamma_k\right)_+\right)\right) \tag{7.4}$$

The log-exponential functional form above guarantees positive search costs. $\kappa_i^{[s]}$ is normally distributed with mean κ and correlated with the price coefficient ρ_i. The position on the page $\text{slot}_{ijt}^{\text{appear}}$ enters search cost with a spline function. Using splines on slot allows me to flexibly capture the relationship between search cost and slot. This has the added benefit of making the results relatively robust to the functional form assumption for search cost.

7.1.3 Heterogeneous Preferences and Search Costs

The demand model allows for a rich set of random coefficients on utility and search cost. The primary specification includes random coefficients on price, the inside option, star rating (1–5), and search cost. Additionally, the primary specification includes correlated random coefficients on search cost and price.

The star ratings indicate the hotel class.[32] The random coefficient on each star rating serves a similar purpose to nests in a nested logit, where consumers have correlated tastes for hotels within

[32] https://www.expedia.com/Hotel-Star-Rating-Information

the same class.[33] An extension of the model would allow for correlation among the star-rating coefficients.

The random coefficient on price and search cost allows different consumers to have different search costs and different price sensitivities. Allowing for price–search cost correlation is sensible, as one interpretation of the search cost is that it is partially the opportunity cost of time, and the price parameter captures the opportunity cost of money. For example, a high income consumer might have a high opportunity cost of time and a low opportunity of money.

7.1.4 Reservation Utility

Next, the reservation utility consists of four elements:

$$r_{ijt}^{[s]} = \delta_{ijt}^{v[s]} + E[\delta_{ijt}^{h[s]}|\Omega_{it}] + \zeta_{ijt}^{[s]} + \lambda\varepsilon_{ijt}^{v[s]} \tag{7.5}$$

Utility from visible features, $\delta_{ijt}^{v[s]}$, the visible match quality, $\lambda\varepsilon_{ijt}^{v[s]}$, the expected utility from hidden features $E[\delta_{ijt}^{h[s]}|\Omega_{it}]$, and $\zeta_{ijt}^{[s]}$, which corresponds to the portion of reservation utility that satisfies equation 6.9. $\delta_{ijt}^{v[s]}$ and $\lambda\varepsilon_{ijt}^{v[s]}$ can be calculated directly from the model parameters, consumer–product–draw features and random draws. $E[\delta_{ijt}^{h[s]}|\Omega_{it}]$ and $\zeta_{ijt}^{[s]}$ require additional processing.

Expected Hidden Utility One approach to calculating $E[\delta_{ijt}^{h[s]}|\Omega_{it}]$ is to estimate a linear regression with $\delta_{ijt}^{h[s]}$ as the left-hand side and the relevant variables from Ω_{it} on the right-hand side and then predict the values of $\delta_{ijt}^{h[s]}$. This would be computationally burdensome, as $\delta_{ijt}^{h[s]}$ depends on the model parameters and so this estimation and prediction would need to be repeated with each evolution of the objective function. Alternatively, to save estimation time, I use linearity of expectations to express expected hidden utility $E[\delta_{ijt}^{h[s]}|\Omega_{it}]$ as a function of model parameters and expected hidden features:

$$E[\delta_{ijt}^{h[s]}|\Omega_{it}] = -e^{\rho_i^{[s]}} E[p_{ijt}|\Omega_{it}] + \beta_{ijt}^{h[s]} E[x_{ijt}^{h[s]}|\Omega_{it}] \tag{7.6}$$

Since the expected features can be estimated directly from data (with rational expectations), I estimate $E[p_{ijt}|\Omega_{it}]$ and $E[x_{ijt}^{h[s]}|\Omega_{it}]$ outside the estimation loop. The expected final price, $E[p_{ijt}|\Omega_{it}]$, is the headline price of the hotel multiplied by the median hidden price percentage for the market.

[33] Train (2009) discusses the similarity between a model with a random coefficient on a categorical variable and a nested logit with a category nest.

For the features, $E[x_{ijt}^{h[s]}|\Omega_{it}]$, consumers know the star rating, whether the hotel is on promotion, and a spline of the logged $\text{slot}_{ijt}^{\text{rank}}$.

Value Function Approximation The final component of reservation utility, $\zeta_{ijt}^{[s]}$, does not have an analytic expression but can be solved numerically, as I know that reservation utilities satisfy the equality in equation 6.9. Numerically solving for each $\zeta_{ijt}^{[s]}$ for every evaluation of the objective function would obviously be computationally infeasible. Instead, researchers solve for $\zeta_{ijt}^{[s]}$ numerically, on a fine grid of state variables, and then use curve fitting to estimate $\zeta_{ijt}^{[s]}$ not exactly at the grid points. This is a point on which my approach differs from common approaches in the search literature. Kim et al. (2010) establish a commonly used approach where $\zeta_{ijt}^{[s]}$ can be solved numerically prior to estimation on an arbitrarily fine grid (this approach relies on the assumption that 1) consumers know product features, and only learn match quality from search).[34] However, since consumers learn about product features and the λ term, $\zeta_{ijt}^{[s]}$ depends on too many parameters for $\zeta_{ijt}^{[s]}$ to be feasibly solved outside the estimation loop. Instead, I move the value function approximation that yields $\zeta_{ijt}^{[s]}$ inside the estimation loop and use a grid interpolation approach,[35] following a common approach used in economics to approximate value functions.

To do this, I include an inner loop, where I numerically solve for the ζ component on a grid of state variables then fit a spline interpolation object to the grid. ζ depends on all the hidden utility parameters, search cost, price coefficients, λ, length of stay, and the random coefficients on hidden features and price. In the primary specification, the grid includes 1,692 point.

Position Variables The position variables, $\text{slot}_{ijt}^{\text{rank}}$ and $\text{slot}_{ijt}^{\text{appear}}$, impact reservation utility differently. $\text{slot}_{ijt}^{\text{rank}}$ is used to shift the expectation of hidden features through $E[\delta_{ijt}^{h[s]}|\Omega_{it}]$. In the current setup, $\zeta_{ijt}^{[s]}$ varies with $\text{slot}_{ijt}^{\text{appear}}$ through search costs. As an extension, it would be possible to account for higher-order consumer beliefs about the relationship between product rankings and hidden features by including $\text{slot}_{ijt}^{\text{rank}}$ as a state variable in the grid interpolation object. Of course, this comes with a practical tradeoff of requiring more points to solve for ζ numerically.

Optimal Sequential Search Setup

The optimal sequential search model from Weitzman (1979) establishes four rules: order, continuation, stopping, and choice. These four conditions can collectively identify the ordered search and purchase

[34] Other examples that use this approach include Chen and Yao (2017) and Ursu (2018).
[35] Examples of grid interpolation appear here: https://uk.mathworks.com/help/matlab/ref/griddedinterpolant.html.

decisions of consumers.

In this section, I go through each rule in turn, consolidate the rules as applying to clicks or purchases, and then use logit smoothing to convert these rules into joint likelihoods.

In this setup, it is helpful to define an index of consumer actions, m, where m^*_{it} refers to the last action (which is always a purchase or choice of the outside option) and each $m < m^*_{it}$ refers to a click. We can use this index to identify observed consideration sets $S_{it}(m)$, which refer to the first $m-1$ items clicked on by consumer i, and the outside option. Going through each rule in turn, we have:

Order Rule: Consumers search in descending order of reservation utility. This means that, at any stage in the search process, the next-clicked item must have a higher reservation utility than that of all of the not-clicked and not-yet-clicked items. Writing out the condition, we have

$$\forall m < m^*_{it}, \quad r_{ijt} \geq r_{ikt} \; \forall k \notin S_{it}(m) \tag{7.7}$$

where m is the m-th step of the search process and m^*_{it} is the number of clicks for consumer search it. $S_{it}(m)$ is consumer i's ordered consideration set of $m-1$ already searched items and the outside options.

Continuation Rule Search continues if any unsearched items have a higher reservation utility than the best option in the consideration set. Formally,

$$\forall m < m^*_{it}, \exists k^* \notin S_{it}(m) \colon r_{ik^*t} \geq u_{ikt} \quad \forall k \in S_{it}(m) \tag{7.8}$$

Stopping Rule Search stops if the utility from the best option so far (including the outside option) is greater than the reservation utilities of all the remaining unsearched options.

$$\exists k^* \in S_{it}(m^*_{it}) \colon u_{ik^*t} \geq r_{ikt} \quad \forall k \notin S_{it}(m^*_{it}) \tag{7.9}$$

where $S_{it}(m^*_{it}) = S_{it}$ is the consumer's complete ordered-consideration set.

Choice Rule Once search ends, the consumer chooses the product with the highest utility in the consideration set.

$$u_{ijt} \geq u_{ikt} \quad \forall k \in S_{it}(m^*_{it}) \tag{7.10}$$

While optimal sequential search has four conditions, in my setting, there are only two types of actions: clicks and purchases (including choosing the outside option). The order and continuation rules apply to clicks, and the stopping and choice rules apply to purchases. Combining the rules, we

have the following two consolidated conditions.

Click Rule

The click rule combines the order and continuation conditions.

Formally, we can combine them as:

$$\forall m < m^*_{it}, \quad \underbrace{\left(r_{ijt} \geq r_{ik't} \quad \forall k' \notin S_{it}(m)\right)}_{\text{order rule}} \wedge \underbrace{\left(\exists k^* \notin S_{it}(m) \colon r_{ik^*t} \geq u_{ikt} \quad \forall k \in S_{it}(m)\right)}_{\text{continuation rule}} \tag{7.11}$$

For the m-th clicked item, it must have the highest reservation utility among the not-(yet-)searched items and also have a higher reservation utility than the items already in the consideration set. If the item has a lower reservation utility than a different product, then that product would be clicked instead. If the product has a lower reservation utility than a product already in the consideration set, then the search would stop.

We can further simplify the click rule as follows:

$$\forall m \leq m^*_{it}, (r_{ijt} \geq u_{ikt} \quad \forall k \in S_{it}(m)) \wedge \left(r_{ijt} \geq r_{ik't} \quad \forall k' \notin S_{it}(m)\right) \tag{7.12}$$

Purchase Rule

The purchase rule combines the stopping and choice conditions. For each consumer, the purchase rule applies only to the last action.

$$\underbrace{\left(\exists k^* \in S_{it}(m^*_{it}) \colon u_{ik^*t} \geq r_{ikt} \quad \forall k \notin S_{it}(m^*_{it})\right)}_{\text{stopping rule}} \wedge \underbrace{\left(u_{ijt} \geq u_{ikt} \quad \forall k \in S_{it}(m^*_{it})\right)}_{\text{choice rule}} \tag{7.13}$$

This is equivalent to saying that the chosen item (in step m^*_{it}) action satisfies the purchase rule if the chosen product has a higher utility than that of all other items in the consideration set and a higher utility than the reservation utilities of the not-searched items.

Joint Likelihood Construction and Logit Smoothing

Now that we have the consolidated click rules and purchase rules, we can construct the joint likelihood of sequential search and purchase decisions. One approach would be to use an accept–reject (AR) simulator. In the AR simulator, for each consumer-draw ($i[s]$), record a one if each click rule and the purchase rule are satisfied; then, take the average over simulations to obtain the joint

likelihood. However, as noted in Ursu (2018), the dimensionality of ordered sets makes this type of AR simulation impractical; for example, with just ten products, there are over 60 million possible ordered consideration sets and choices. An alternative approach is to use logit smoothing, following Train (2009).[36]

As stated above, I am integrating over the match quality terms, ε^h and ε^v, with scrambled Halton sequences. In logit smoothing, I conduct the modeling as if there is a type-1 extreme value term, scaled by a smoothing parameter ω, associated with each click and choice condition. I can then obtain a logit-smoothed expression for the click and purchase conditions.

Logit-Smoothed Click-Condition

$$P_{it}^{\text{click[s]}} = \prod_{m \in S_{it}} \left(\frac{\exp\left(\frac{x_{it}^{\text{nights}} r_{imt}^{[s]}}{\omega}\right)}{\sum_{k \in S_{it}(m)} \exp\left(\frac{x_{it}^{\text{nights}} u_{ikt}^{[s]}}{\omega}\right) + \sum_{k' \notin S_{it}(m)} \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)} \right) \tag{7.14}$$

where $P_{it}^{\text{click[s]}}$ is the smoothed likelihood that all $m_{it}^* - 1$ clicks for consumer i satisfy the click conditions. ω denotes the smoothing parameter, and x_{it}^{nights} is the length of stay.

Logit-Smoothed Purchase Condition

$$P_{it}^{\text{choice[s]}} = \frac{\exp\left(\frac{x_{it}^{\text{nights}} u_{ijt}^{[s]}}{\omega}\right)}{\sum_{k \in S_{it}} \exp\left(\frac{x_{it}^{\text{nights}} u_{ikt}^{[s]}}{\omega}\right) + \sum_{k' \notin S_{it}} \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)} \tag{7.15}$$

where $P_{it}^{\text{click[s]}}$ is the smoothed likelihood that consumer i's purchase decision satisfies the purchase conditions.

Joint Likelihood

We can now combine these conditions to obtain the logit-smoothed joint likelihood of search and purchase decisions.

[36] Logit smoothing is a popular approach to non–smooth objective functions. It is also a useful tool for search models for example Honka (2014), Ursu (2018), and Honka and Chintagunta (2017) use various logit-smoothing Logit smoothing is also a well-established technique in computer science for smoothing loss functions, in which ω is referred to as the temperature parameter (Platt, 2000)

$$\mathrm{P}_{it}^{[s]} = \overbrace{\prod_{m \in S_{it}} \underbrace{\left(\frac{\exp\left(\frac{x_{it}^{\text{nights}} r_{imt}^{[s]}}{\omega}\right)}{\sum_{k \in S_{it}(m)} \exp\left(\frac{x_{it}^{\text{nights}} u_{ikt}^{[s]}}{\omega}\right) + \sum_{k' \notin S_{it}(m)} \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)} \right)}_{\text{click condition for } m\text{-th click}}}^{\text{click condition(s)}} \times \overbrace{\left(\frac{\exp\left(\frac{x_{it}^{\text{nights}} u_{ijt}^{[s]}}{\omega}\right)}{\sum_{k \in S_{it}} \exp\left(\frac{x_{it}^{\text{nights}} u_{ikt}^{[s]}}{\omega}\right) + \sum_{k' \notin S_{it}} \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)} \right)}^{\text{purchase condition}} \tag{7.16}$$

where $P_{it}^{[s]}$ is the joint likelihood of the observed search and purchase decisions for consumer i in search t within the set of simulated draws s. Averaging $P_{it}^{[s]}$ over simulations yields the likelihood:

$$P_{it}^{\text{joint}} = \frac{1}{D} \sum_{s=1}^{D} P_{it}^{[s]} \tag{7.17}$$

where D denotes the number of simulations (draws).

7.1.5 Sample Selection Adjustments

As discussed in section A.3, there are two types of sample selection. These data include only observations with at least one click, and observations with a purchase are oversampled relative to searches without a purchase. Not adjusting for these sampling issues would lead to biased parameter estimates. I adjust for the first (any click) by using conditional likelihoods. I adjust for the second issue by using observation weights.

Conditioning on Any Click

The sample includes only data from searches in which consumers clicked at least one of the options. I adjust for this selection by conditioning the likelihoods on clicking at least one item. This requires calculating the likelihood of clicking at least one item. At the consumer it–simulation $[s]$, level we can express this as the likelihood of at least one reservation utility being greater than the utility of the outside option.

Smoothed likelihood of any clicks. The smoothed likelihood of making any clicks for consumer i at time t for simulation s is given by

$$P_{it}^{\text{any click[s]}} = \frac{\sum_k \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)}{\exp\left(\frac{x_{it}^{\text{nights}} u_{i0t}^{[s]}}{\omega}\right) + \sum_k \exp\left(\frac{x_{it}^{\text{nights}} r_{ik't}^{[s]}}{\omega}\right)} \tag{7.18}$$

It is possible to estimate $P_{it}^{\text{any click[s]}}$ without smoothing; however, as it is used to condition the joint likelihood $P_{it}^{[s]}$, which is smoothed, I use the same smoothing approach for $P_{it}^{\text{any click[s]}}$, which avoids conditional likelihoods above 1.

Conditional likelihood. Using the definition of conditional probability, we can write the conditional likelihood of a consumer's ordered search and purchase conditional on clicking at least one hotel as

$$P_{it|\text{any click}}^{\text{joint}} = \frac{\frac{1}{D}\sum_{s=1}^{D} P_{it}^{[s]}}{\frac{1}{D}\sum_{s=1}^{D} P_{it}^{\text{any click[s]}}} \tag{7.19}$$

where $P_{it}^{[s]}$ is the unconditional joint likelihood of search and purchase from 7.16 and $P_{it}^{\text{any click[s]}}$ is the likelihood of any clicks from 7.18.

Sample Weights

To address the different sample rate for observations with versus without a purchase, I use sample weights to achieve a target purchase rate (conditional on at least one click) of 16.66%. I calculate the weights using data from the top 5 markets. The relative weight depends on the observed purchase decisions

$$w_i = \begin{cases} w^{\text{in}}, & \text{if consumer } i \text{ chose an inside good} \\ w^{\text{out}}, & \text{otherwise} \end{cases} \tag{7.20}$$

where w_i is the weight for consumer i (note that I do not observe consumer IDs across searches). In the primary specification, w^{in} is normalized to one and $w^{\text{out}} = 56.63$.

Ignoring the sampling issue would lead to biased parameters. The direction of some parameters is not obvious, but a simple way of thinking about this problem is the following. In the observational data, there is approximately a 90% conversion rate. Without weights, the inside option would seem highly desirable when, in reality, consumers rarely make a purchase. Ignoring the weighting would also cause other concerns, as match quality terms and random coefficients would also influence the

decision to make the first click.

Weighted Log Simulated Likelihood

Applying sample weights and using the conditional likelihoods for each observation yields the consumer–search-level weighted likelihood:

$$wll_i = w_i \log\left(\frac{1}{D}\sum_{s=1}^{D} P_{it}^{[s]}\right) - w_i \log\left(\frac{1}{D}\sum_{s=1}^{D} P_{it}^{\text{any click[s]}}\right) \tag{7.21}$$

Summing across consumers yields the logit-smoothed log simulated likelihood.

$$SLL = \sum_i wll_i \tag{7.22}$$

Algorithm 1 in Appendix E.0.1 summarizes the demand estimation procedure.

7.1.6 Informal Identification

Since this is a maximum simulated likelihood estimation, to some extent, everything identifies everything. However, it is helpful to discuss the intuition for parameter identification in terms of the optimal sequential search rules and notable variation in the data. Table 7.1 summarizes the key sources of identification, and I discuss further details below.

Length of stay separately identifies search costs and utility parameters. The challenge of separately identifying search cost and utility parameters in search models is well-documented. For example, slot affects search costs but is often highly correlated with product features. Koulayev (2014) notes the challenge of distinguishing high search costs from low tastes.[37] My approach addresses this issue by leveraging variation in length of stay. Consumers searching for longer stays would be consuming more of the good and paying a multiple of the prices. This means returns to search depend on length of stay. However, length of stay is, presumably, independent of search costs. More formally, two consumers with identical utility and search cost parameters but different lengths of stay would have the same per-night utilities, but different reservation utilities. To my knowledge, this is the first paper to take advantage of length stay to address these identification issues.

[37] See Ursu et al. (2023) for an overview other challenges, such as cases where parameters that suggest both returns to search are and search costs are high, produce results similar to those where parameters suggest that both returns to search and search costs are low.

Table 7.1: Informal Identification of Demand Parameters

	Sequential Search Conditions				Notable Variation		
Parameters	Order	Continuation	Stopping	Choice	Nights	Diversion	Displacement
Utility Parameters							
Consumer Segments: δ_{it}		✓	✓	✓†	✓	✓	✓
Time Effects: ξ_{it}^{month},ξ_{it}^{day}		✓	✓	✓†	✓	✓	✓
Mean: ρ, β^{v}, β^{h}	✓	✓	✓	✓	✓	✓	✓
Heterogeneous: Σ_u	✓*	✓*	✓*	✓*	✓	✓	✓
Visible Error Scale: λ	✓	✓	✓		✓	✓	✓
Search Cost Parameters							
Mean: κ, τ_k	✓	✓	✓		✓	✓	✓
Heterogeneous: Σ_κ	✓*	✓$*$	✓*		✓	✓	✓

Note: Checkmarks with an asterisk (✓*) indicate parameters that are identified by repeated decisions within consumer (e.g., clicks and purchase). Checkmarks with a dagger (✓†) indicate parameters that are identified by selecting an inside good versus the outside option, but not from the choice of one inside good over another. "Nights" refers to length of stay. "Diversion" refers to substitution patters from variation in product features and availability. "Displacement" refers the variation in positions caused by advertisements/opaque offers.

Diversion and displacement. Other variation also helps me separately identify utility and search costs parameters, including diversion (similar to the diversion ratio), where I observe different search and purchase decisions under different hotel availability, product rankings and prices. Additionally, in some searches, opaque offers appear and displace the positioning of some hotels; this means that in some scenarios, $slot_{ijt}^{rank} \neq slot_{ijt}^{appear}$.

Repeated within-consumer decisions. Although I do not observe multiple search sessions for each consumer, I still observe repeated decisions within a search session. A consumer's clicks and purchase within a session help to identify random coefficients. For example, if consumers click only on hotels of the same star rating, that is indicative of the random coefficients on star rating. Similarly, the correlation of prices within a consumer's consideration set informs the random coefficient on price.

Market–time effects. While it might be reasonable to assume that the hotels can be modeled in feature space, a key feature of the accommodation industry is that prices move with time-varying demand. For example, in a college town, prices and demand increase during sporting events and graduations. I include market–month and market–day-of-week effects. This assumes that the time-varying shifts in demand occur at the market level and are not hotel–time specific. As an

extension, I could use narrower time effects, for example, market–week instead of market–month.

Feature space. I model utility for hotels in feature space, relying on the assumption that the rich set of product features captures the hotel-specific utilities. There are over 700 products in the top market, and many of them appear rarely, making a product fixed effect approach impractical.

7.1.7 Demand Results

The primary specification of the model uses a 90% sample of observations from the top market, with observations that were subject to Expedia's default recommendation system.[38] Table 7.2 presents the parameter estimates.[39] The results are consistent with intuition. The λ parameter takes on a value of 0.28, suggesting that consumers know part of the match quality prior to search but learn most from the search. Four- and five-star hotels have higher mean utility than lower-rated hotels. Search cost is monotonically increasing in page position (this is not a constraint). I include additional results on search costs and implied differences in reservation utility by slot in F.

[38] The remaining 10% are used to evaluate the out-of-sample performance of the model.

[39] standard errors are still a work in progress, as bootstrapping this model is computationally expensive

Table 7.2: Demand Parameter Estimates

Utility Parameters

Variable	(1)
Outside option	1.90
Price ($100s) ρ	-1.76
Match quality split λ	0.28
Visible Features	
3 star	0.30
4 star	0.54
5 star	0.48
Non-star	0.31
2 star brand	-0.16
3 star brand	-0.28
4 star brand	0.03
5 star brand	0.29
Prop. review score	
Spline 1: score 1–3	-0.51
Spline 2: score 3–5	0.04
Mi. dummy	-1.40
Hidden Features	
Location score 1	
Spline 1	0.52
Spline 2	-0.51
Spline 3	0.05
Spline 4	2.61
Location score 2	
Spline 1	0.27
Spline 2	1.50
Spline 3	0.37
Mi. dummy	1.64

Estimation Details

Observations	2,262
Weighted obs.	13,444
Halton draws	400
Smoothing term ω	0.2
Grid points	1,692
Log likelihood	-85,028

Search Cost Parameters

Variable	(1)
Constant	-1.10
Log Slot Appear	
Spline 1	0.11
Spline 2	0.21
Spline 3	0.37
Spline 4	0.08

Random Coefficients

Parameter	(1)
σ^2_{price}	0.729
$\sigma^2_{\text{inside option}}$	0.095
$\sigma^2_{\text{1 \& 2 star}}$	0.080
$\sigma^2_{\text{3 star}}$	0.009
$\sigma^2_{\text{4 star}}$	0.015
$\sigma^2_{\text{5 star}}$	0.000
$\sigma^2_{\text{search cost}}$	0.279
$\sigma^2_{\text{price-search cost}}$	-0.444

Additional Controls

Day of week	✓
Month	✓
Time before stay	✓
Length of stay	✓
Search time	✓
Search on weekends	✓

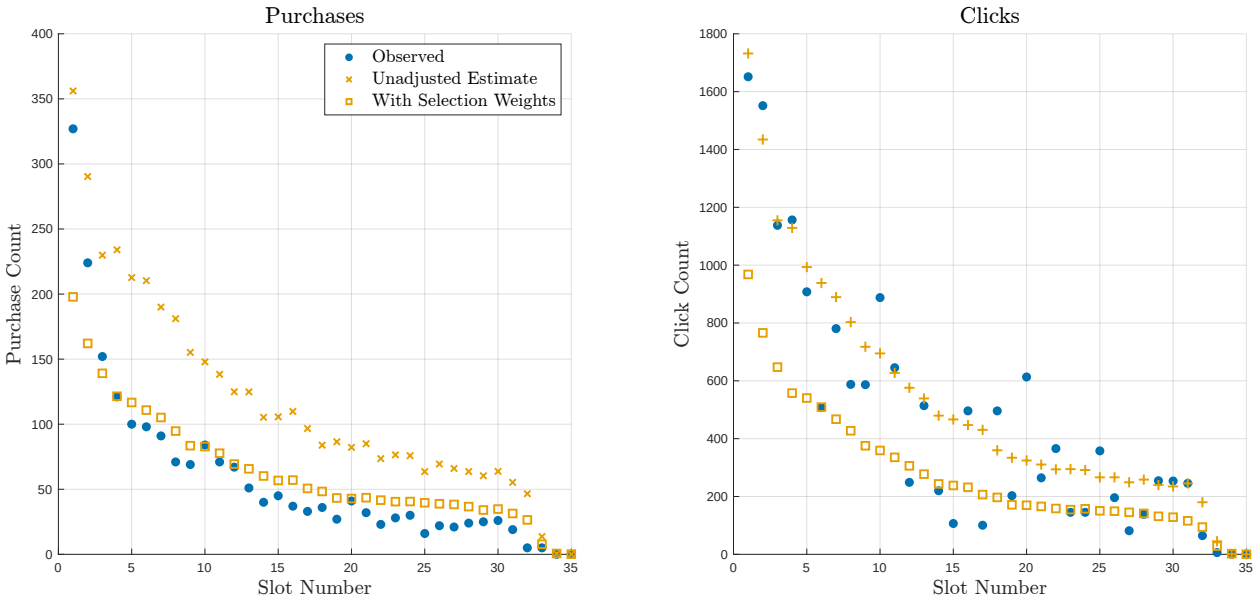
Notes: Likelihood is the logit-smoothed likelihood for joint search and purchase decisions. Splines are linear B-spline. Variance of random coefficients estimated using Cholesky decomposition.

Model Fit

Figure 7.1 plots observed purchases (left) and clicks (right) by position on the page. It compares the observed amounts to the expected quantities implied by the parameter estimates from the demand model.

Both figures show the expected pattern of observed and predicted counts of purchases and

Figure 7.1: Demand Fit: Predicted vs Observed Quantity and Clicks



Note: Base on the in-sample data from the top market.

clicks decreasing as we move down the page. The selection weights adjustment appears to bring the estimated counts closer to the observed counts, but there is still a noticeable gap, especially at lower slot numbers, where the models underestimate the position effects, resulting in lower predicted counts of purchases and clicks compared to the observed levels. This suggests that, while the model has some predictive accuracy, there is room for improvement.

Position Effect Mechanisms: Search Cost and Rational Expectations

Here, I briefly return to the discussion on the mechanisms driving search cost. To test the structural assumption that position impacts demand through search cost and expectation of hidden features, I reestimate the demand using alternative structural assumptions. Table 7.3 shows the results, comparing the primary specification results to those of one where position impacts demand only through search cost. The primary specification, which allows position effects to be driven by both search costs and beliefs, outperforms the search cost–only model both in and out of sample.

Both models have the same number of underlying parameters. The only difference is the structural assumptions, so we can directly compare the log-likelihoods. If we were testing a model with alternative parameters, then we would need a measure, such as the Akaike or Bayesian information criterion (AIC or BIC), that includes a penalization for additional parameters.

Table 7.3: Position Effect Mechanism Results

	Position Effect Structural Assumption	
	Search Cost	**Search Cost & Beliefs**
Log Likelihood (In Sample)	-85,567	-85,028
Log Likelihood (Out of Sample)	-13,936	-13,914

Note: Logit-smoothed joint likelihoods of search and purchase conditional on at least one click. Includes sampling weights based on conversion rates. 2262 in-sample (training) observations, 251 out-of-sample (testing) observations.

7.2 Platform Model: Product Recommendation Model Estimation and Results

This section describes the estimation procedure for the platform model. I estimate the platform model with the naturally ordered Expedia data, using the natural rankings as the outcome variable and the product, query, and consumer features as the explanatory variables. The estimation procedure consists of two steps. The first step uses a "model-of-a-model" approach from machine learning to estimate the deterministic portion of the relevance scores. The second step is to estimate a set of slot-specific conditional logits, which scale the deterministic portion of the relevance scores from the first step.

In section 8, I provide a more detailed overview of learning-to-rank methods. In that section, I develop personalized rankers explicitly for ranking, whereas in this section, although I use a learning to ranking method, I do so to model the behavior of Expedia's default recommendation system.

I use a "model-of-a-model" approach from machine learning. This connects to the the growing body of work in the computer science, machine-learning, and cryptography literature that demonstrates cases where black-box machine-learning algorithms can be reverse-engineered by training a new model on data generated from queries to the black-box model and the black-box model's results (Tramer et al., 2016; Papernot et al., 2017; Oh et al., 2018; Hu and Pang, 2021).

7.2.1 Platform Model Step 1: Model Extraction via LambdaMART

The first step of the platform model is the model extraction step which applies LambdaMART, a machine learning algorithm used for ranking problems, detailed in Burges (2010), to the naturally ordered data. The name "LambdaMART" is derived from the fact that it is a combination of "Lambda" (referring to the gradient boosting approach it uses, which computes lambda-like quantities) and "MART" (Multiple Additive Regression Trees). LambdaMART has proven to be one of the more effective ranking algorithms, and is a popular choice of algorithm in data science competitions and

industry. Microsoft uses LambdaMART as the underlying algorithm in Bing's search engine.[40] An ensemble of LambdaMART rankers won Track 1 of the Yahoo! Learning to Rank Challenge (Chapelle and Chang (2011)), and an ensemble rankers including LambdaMART also won the Personalize Expedia Hotel Searches – ICDM 2013 competition, the source of my data.

Pseudo-Relevance Score

Learning-to-rank models rely on a relevance score, so I convert the product rankings to scores between zero and five, where a five is the top slot, and 0 is the last product listed.

$$rel_{ijt} = 5 - \frac{1 + max_k(slot_{ikt}^{rank}) - slot_{ijt}^{rank}}{max_k(slot_{ikt}^{rank})} \tag{7.23}$$

Where rel_{ijt} is the relevance score for product j in consumer i's query at time t. $slot_{ijt}^{rank}$ is the position on the page, adjusting for advertisements and opaque offers.[41] A product in the first slot receives a relevance score of five, while a product in the last slot, $max_k(slot_{ikt}^{rank})$, would receive a relevance score of zero.

Input Data

The input data for the first stage of the platform model include the training observations for the top five markets. I incorporate a range of input variables and interactions. These include product features including headline price, promotion indicator, hidden price percentages, star rating, review score, brand indicators, location scores, search query affinity, the log of the historical price and indicators for missing values. I also include product specific data from eight competing OTAs. At the query level, I account for market id, and submarket id, time of search, booking window, weekend searches, if the stay includes a weekend, and specific site indicators. The query level variables do not vary withing a search, so I include these variables as interactions with the product features variables listed above.

Step 1 Estimation

In this setup, the features are the query information. The outcome of interest is the product ranking. The model includes one constraint. Relevance scores are constrained to be monotonic in price. This

[40] https://www.microsoft.com/en-us/research/blog/ranknet-a-ranking-retrospective/

[41] If a product is in the sixth slot, but the fifth slot is occupied by an opaque offer, then $slot_{ijt}^{rank}$ would be five, since it is ranked fifth by the recommendation system. This assumes advertisements and opaque offers are determined separately, which is reasonable since they occupy the same position on the page.

constraint prevents situations where a firm can achieve a higher slot by increasing its prices, which can lead to positive own-price elasticities.

Aside from the adjustments to the relevance score and monotonicity in price, the estimation proceeds much like a standard learning to rank problem. I use normalized discounted cumulative gain (NDCG) as the loss function. I use cross-validation to select the optimal number of trees. The hyperparameters are shrinkage, interaction depth, and the out-of-bag fraction. I use .8 as a pilot out-of-bag fraction and Bayesian Optimization to select the shrinkage and interaction depth. Once I have the hyperparameters and number of trees, I fit the model on eight separate folds of the data. I estimate the model on distinct folds since the platform model relies on out-of-fold predictions. I then evaluate the fit of each model on a held out test data set.

Step 1 Results

Table 7.4 presents the out-of-sample performance for the first step of the platform model. Each row corresponds to a separate model, including a random benchmark, a model trained on the entire training data, each fold-specific model, and an ensemble from averaging the predictions of each fold-specific model. The columns correspond to loss functions; lower numbers mean better model performance. The two loss functions are NDCG, which is the loss function I used in model training, and concordant pair loss (Conc), which is the percent of pairwise pairs the model incorrectly ranks. Each fold-specific model has similar out-of-sample performance and performs well, correctly predicting rankings 72 percent of the time and performing better in predicting top-ranked products.[42] It is also worth pointing out that there is room for improvement, as the ensemble model (bottom row) performs notably better than each fold-specific model.

Figure 7.2 visualizes the distribution of predicted rankings from the fold-specific models (treating their predictions as deterministic). The horizontal axis is the predicted slot, with 1 being the top-ranked product. The vertical axis is the observed slot. The dark diagonal means the predictions are in roughly the correct position. The darker region in the top left also illustrates that these models accurately predict the top products. This is important as most purchases and clicks occur in this top region of pages.

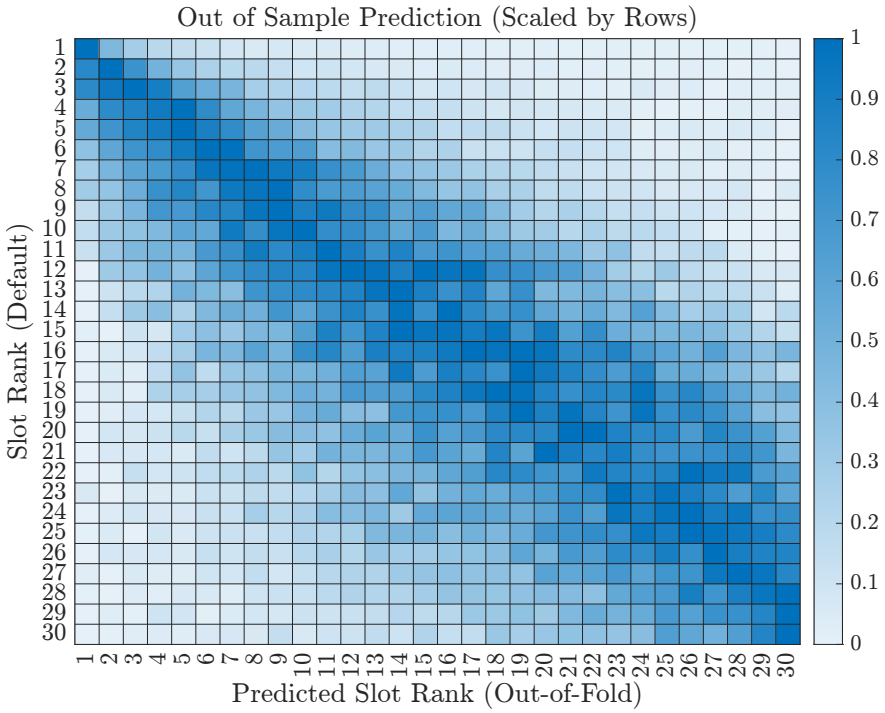
[42]The loss function for the model is the normalized discounted cumulative gain (NDCG), which can be challenging to interpret. I present the pairwise matching accuracy here since it is easier to understand.

Table 7.4: Comparison of Model Results

Measure	Model	NDCG Loss	Conc Loss
1	Random Benchmark	0.175	0.506
2	LambdaMART (NDCG): Full	0.060	0.276
3	LambdaMART (NDCG): Fold 1	0.061	0.277
4	LambdaMART (NDCG): Fold 2	0.061	0.277
5	LambdaMART (NDCG): Fold 3	0.061	0.276
6	LambdaMART (NDCG): Fold 4	0.061	0.277
7	LambdaMART (NDCG): Fold 5	0.061	0.278
8	LambdaMART (NDCG): Fold 6	0.061	0.276
9	LambdaMART (NDCG): Fold 7	0.060	0.277
10	LambdaMART (NDCG): Fold 8	0.061	0.277
11	LambdaMART (NDCG): Ens	0.059	0.272

Note: NDCG is normalized discounted cumulative gain loss. Conc is concordant pair loss. Random benchmark uses random prediction. Ens (ensemble) averages the predictions from all eight folds.

Figure 7.2: Distribution of Predicted vs Actual Slots



Note: Normalized distributions by row to adjust for different numbers of hotels appearing in search results

7.2.2 Platform Model Step 2: Sequential Logits

The first step model gives us the out-of-fold predicted deterministic component of relevance scores, $\hat{\psi}_{ijt}$, for each consumer-query-product, ijt.[43] The second step solved for the term that scales the deterministic portion of relevance scores, $\hat{\psi}_{ijt}$ to the scale of the random portion of relevance.

This second step estimates a slot-specific scale term, β_n^{slot}, on $\hat{\psi}_{ijt}$. For the first slot, this involves estimating a conditional logit with $\hat{\psi}_{ijt}$ as the right hand side variable, and an outcome of 1 if the target consumer-query-product is in the top slot. This regressions estimates the parameter β_n^{slot} from 7.24.

$$u^r_{ijtn}(\text{slot } n) = \beta_n^{slot}\hat{\psi}_{ijt} + \varepsilon_{ijt} \tag{7.24}$$

which give me the likelihood

$$P(j \text{ in slot } 1) = \frac{\exp(\beta_1^{slot}\hat{\psi}_{ijt})}{\sum_k \exp(\beta_1^{slot}\hat{\psi}_{ikt})} \tag{7.25}$$

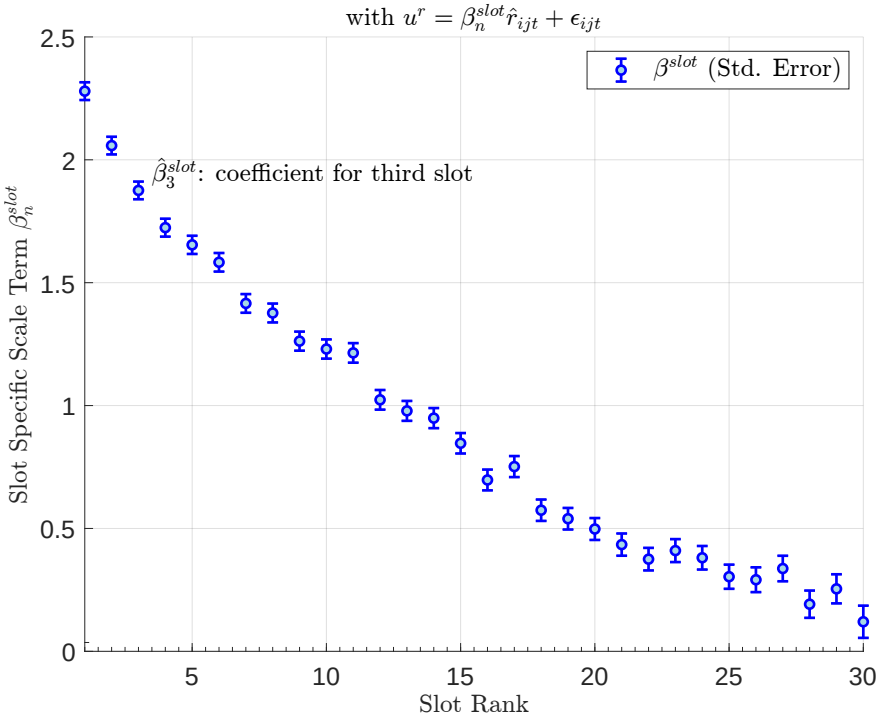
I then repeat this for target slot values of from 2 to 30. In each of these regressions, I use data for consumer-query-product with a slot at or below the target slot.

$$P(j \text{ in slot } n) = \frac{\exp(\beta_1^{slot}\hat{\psi}_{ijt})}{\sum_{k\notin \text{ slot 1 to } n-1} \exp(\beta_1^{slot}\hat{\psi}_{ikt})} \tag{7.26}$$

Figure 7.3 present the result from each of the conditional logit regressions. The parameters are higher for higher slots, meaning position on the page is more deterministic in $\hat{\psi}_{ijt}$ higher on the page.

[43] After step 1, I normalize the scores to be $N(0,1)$

Figure 7.3: Platform Model Sequential Logit Scale Parameters



In the supply side, I estimate own-price elasticities using the algorithms developed in step one, that predict $\hat{\psi}_{ijt}$, and the scale terms, β_n^{slot}, from step two.

7.3 Supply-Side Model: Hotel Pricing

The supply-side model captures hotel pricing behavior. In estimation, I use the seller's first-order conditions, the observed prices, and the expected quantities and elasticities from the demand and platform models to back out marginal costs. Since marginal costs depend on quantity, I use an IV approach with three-stage least squares to capture the relationship between marginal cost, quantity, and quantity squared.

Aggregation by Subperiod

The supply side (and counterfactual) requires some aggregation in terms of time of stay and time of search. In the hotel industry, prices change often. In part, these price changes serve as inter-temporal price discrimination. I define "subperiods" which are time of stay t and time of search t' pairs. In my primary specification, there are four subperiods per month, based on the combination of weekend vs weekday stays, and searches in advance of the stay or close to the date of the stay. This simplification

allows for some inter-temporal price changes but provides enough hotel-subperiod observations to calculate the own-price elasticities necessary for the supply side and counterfactuals.

Seller's Subperiod Expected Profits

Restating the hotel's pricing problem, we have

$$\underset{p_{jtt'}}{\operatorname{argmax}} \mathbb{E}\left[(1-\varphi)\, p_{jtt'} - c_{jtt'} q_{jtt'} \,|\, \Omega_{jtt'}\right] \tag{7.27}$$

Seller foc

$$\frac{mc_{jtt'}}{1-\varphi} = p_{jtt'} + \left(\frac{\partial q_{jtt'}}{\partial p_{jtt'}}\right)^{-1} q_{jtt'} \tag{7.28}$$

were $p_{jtt'}$ is the price for room–night j, staying period t, and searching period t' is denoted by $p_{jtt'}$. To aggregate to the subperiod, I use the median observed $p_{jtt'}$. $mc_{jtt'}$ represents the opportunity cost of having the marginal unit available to sell in the next (sub)period.[44] $c_{jtt'}$ denotes average variable opportunity cost. The seller's information set, $\Omega_{jtt'}$, marks that sellers are aware of their own costs, the elasticity of demand, and the features and availability of competing products.[45]

The remaining elements on the right hand side of the seller foc (equation 7.28) are the expected quantity $q_{jtt'}$, and the inverted $\frac{\partial q_{jtt'}}{\partial p_{jtt'}}$, which depends on the own-price elasticity, the expected quantity, and the median price. These two elements depend on the demand and platform models.

I use the results from the demand model to estimate supply side observation weights. I then combine the results from the demand and platform models to calculate expected the expected quantities and own-price elasticity needed for the supply side model. I detail these procedures in Appendix F.1.

7.3.1 Marginal Cost Recovery and Specification

I back out estimated marginal costs, $\hat{mc}_{jtt'}$, using the seller's first-order condition in Equation 7.28, the observed prices, and the expected quantities and elasticities derived from the demand and platform models.

[44] It can also include additional expected profits that are conditional on the purchase, such as room service, dining, or gambling.

[45] These are the standard assumptions about the seller information with some extensions to account for the e-commerce environment.

Economies of Scale and Capacity Constraints

The supply-side model captures three key features of the accommodation industry: at low occupancy, hotels have economies of scale, and at high occupancy, hotels face increasing costs and capacity constraints. I capture these features by allowing the marginal cost to depend on quantity and quantity squared.

$$\hat{mc}_{jtt'} = mc^{\text{base}}_{jtt'} + \gamma_1 q_{jtt'} + \gamma_2 q^2_{jtt'} \tag{7.29}$$

Where $\hat{mc}_{jtt'}$ is the marginal cost estimate recovered from the hotel's first-order condition (FOC). A negative coefficient for $q_{jtt'}$ captures economies of scale, while a positive coefficient for $q^2_{jtt'}$ captures increasing costs at high occupancy and serves as a soft-capacity constraint.[46]

I cannot estimate Equation 7.29 directly since quantity depends on prices, which are decided endogenously. To address the endogeneity concern, I use an instrumental variable (IV) approach with BLP-type instruments Berry et al. (1995). The instruments are demand shifters, including features and availability of competing products in the same market and interaction terms of a hotel's own star rating with the distribution of star ratings in the market. With these instruments, I estimate the supply-side model via three-stage least squares.

First stage: IV for $q_{jtt'}$

The first stage instruments for quantity

$$q_{jtt'} = \alpha_1 x_{jtt'} + \alpha_2 z_{jtt'} + \varepsilon_{jtt'} \tag{7.30}$$

where $x_{jtt'}$ includes product features, and market-subperiod specific effects. The instruments,$z_{jtt'}$ include product features and availability of other products in same market, and own-star rating interactions.

[46] Farronato and Fradkin (2022) use a similar approach to modeling soft-capacity constraints in the hotel industry by estimating fixed and variable costs at low capacity, and increasing costs at 85% occupancy (a hockey stick-type function).

Second Stage: IV for $q^2_{jtt'}$

In the second stage, I instrument for $q^2_{jtt'}$ using the squared predicted values from the first stage, $\hat{q}^{\text{step 1}}_{jtt'}$.[47]

$$q^2_{jtt'} = \alpha_3 \left(\hat{q}^{\text{step 1}}_{jtt'}\right)^2 + \varepsilon_{jtt'} \tag{7.31}$$

Third Stage

In the third stage, I include the predicted values from the first stage, $\hat{q}^{\text{step 1}}_{jtt'}$, and second stage, $\hat{q}^{\text{step 2}}_{jtt'}$. The parameters of interest are γ_1 and γ_2.

$$\hat{mc}_{jtt'} = \hat{\beta} x_{jtt'} + \gamma_1 \hat{q}^{\text{step 1}}_{jtt'} + \gamma_2 \left(\hat{q}^2_{jtt'}\right)^{\text{step 2}} + \nu_{jtt'} \tag{7.32}$$

7.3.2 Supply-Side Results

The supply-side results, presented in table 7.5, are consistent with intuition: costs are higher for higher star-rating (tier) hotels and reflect the expected relationship with quantity, with decreasing costs at low quantity, characteristic of economies of scale, and increasing costs at high quantities, characteristic of increasing costs near capacity constraints. An additional specification includes star-specific γ's.

8 Personalized Recommendation Systems for Counterfactuals

Now, with a structural model of demand, platform product recommendations, and hotel pricing behavior, I turn to understanding the welfare effects of personalized recommendations. To do this, I first need to develop personalized recommendation systems. My model training approach is based on the winning entry in the Expedia Personalization competition. I use an ensemble of LambdaMARTs, which are learning-to-rank algorithms that use gradient-boosted decision trees, with NDCG as the loss function.

8.1 Training Four Recommendation Systems

As with estimating demand, a challenge in training recommendation systems is that slots influence consumer choices, but slots are highly correlated with product features. This is where the randomized

[47] Skipping this step, and including directly including $\hat{q}^{\text{step 1}}_{jtt'}$ in the regression on marginal cost would not produce the same results, and would be a "forbidden regression". For more info see chapter 9 of Wooldridge (2010)

Table 7.5: Supply Side IV Regression Analysis Results

	Pooled	Star Rating Specific			
Variable	**All Ratings**	**Two-Star**	**Three-Star**	**Four-Star**	**Five-Star**
Intercept	-0.307	0.527	1.022***	1.484***	3.685***
	(0.503)	(0.614)	(0.124)	(0.114)	(0.134)
$\hat{q}_j^{(1)}$	-0.198***	-0.091	-0.036	-0.091	-1.287***
	(0.037)	(0.072)	(0.094)	(0.092)	(0.149)
$\hat{q^2}_j^{(2)}$	0.033***	-0.016	0.033*	0.026**	0.199***
	(0.008)	(0.012)	(0.032)	(0.028)	(0.051)
Two/Three-Star	0.577***	–	–	–	–
	(0.088)				
Four-Star	1.010***	–	–	–	–
	(0.086)				
Five-Star	2.688***	–	–	–	–
	(0.111)				
Additional Controls					
Product Features	✓		✓		
Location Desirability	✓		✓		
Month–Weekend–Subgroup	✓		✓		
Observations	3492		3492		
Degrees of Freedom	3437		3429		
RMSE	0.783		0.761		
R^2	0.638		0.660		
Adjusted R^2	0.632		0.654		
First-stage F-statistic	103		103		

Note: The first column presents results from the pooled model, while the next four columns present results from the jointly estimated star-rating specific model. In the pooled model, product star-rating impacts the intercepts of marginal cost, but not the relationship with quantity. The estimates for each star-rating in the star-rating specific model are calculated by adding the base coefficients and interaction terms. For the Two-Star group, the estimates are the base coefficients. Both models include product feature controls, location desirability controls, and month–weekend–subgroup controls. The models share a first stage.

Instruments for $\hat{q}_j^{(1)}$: Mean product features of competing products, availability of other products, and own star rating interacted with the distribution of star ratings of other products.

data are incredibly useful. I train these recommendation systems using the subset of Expedia data where they randomized the product rankings.[48] I use the data from the five top markets, as there could be information spillovers from one market to another. For example, preferences may be similarly correlated across markets.

For the outcome variable, relevance score, I follow the approach from the original competition rules, where relevance scores are 5 for bookings, 1 for clicks, and 0 for impressions.

I train four increasingly personalized versions of the recommendation system. In counterfactuals,

[48] In fact, the winning entry from the Kaggle competition only used random data to train their models.

this helps understand not just what would happen with the most personalized recommendation system (possible with these data) but how welfare would change as we increase personalization from the least personalized to the most personalized. I adjust the level of personalization based on the variables available to the algorithms. The least personalized recommender only uses data on products. The next includes additional data on the consumer queries. This is information actively volunteered by consumers, such as length of stay and if they are traveling with children. The next recommendation system uses personal data based on the consumer's location, distance to the destination, booking window, and time of search. The most personalized recommendation system includes data on each consumer's past purchases, such as the average price, star rating of their previous purchases, and other tracked information.

Common Recommendations: Product features, price and availability on competing OTAs.

Query Adjusted: Added query features (e.g., length of stay, number of nights).

Personalize: Added consumer observables (e.g., booking window, consumer country).

Most Personalized: Included past transactions, tracked navigation data.

I use an ensemble approach for each recommendation system, in which I train multiple versions of LambdaMART and take the average of their predictions. Each of the four recommendation systems consists of 170 underlying LambdaMARTs. I constrain each of the 170x4 models to be monotonic in price.

8.2 Validating Recommendation Systems

I validate the models by estimating out-of-sample performance in predicting purchases and clicks. I should find that models with access to more personalized data have better out-of-sample performance. Table 8.1 presents the results. These models match that pattern with out-of-sample model performance increasing with personalization.

9 Counterfactual Simulations

This section presents the counterfactual simulations. I use the structural model to evaluate the welfare effects of deploying personalized recommendation systems.

Table 8.1: Comparison of Model Results

Measure	Model	NDCG Loss	Conc Loss	MAP	MRR
1	Random Benchmark	0.673	0.480	0.850	0.846
6	LambdaMART (Ensemble): Base Info	0.544	0.302	0.699	0.692
7	LambdaMART (Ensemble): with Query Info	0.540	0.301	0.695	0.686
8	LambdaMART (Ensemble): Personalized Basic	0.537	0.299	0.692	0.681
9	LambdaMART (Ensemble): Personalized Full	0.533	0.300	0.686	0.676

9.1 Counterfactual Setup

In the counterfactuals, I make a few necessary simplifications. In the hotel industry, prices change often. As a baseline counterfactual, I impose a sub-period uniform pricing constraint. In this setup, hotels can set four unique prices for room–nights in each month along two dimensions: weekends versus weekdays and searches in advance of the stay or close to the stay date. These subperiods match the supply side of the model. This simplification allows for some inter-temporal price changes but provides enough hotel–subperiod observations to calculate the own-price elasticities necessary for the supply side.

Next, I solve for the counterfactual equilibrium using the baseline, and the four recommendation systems from section 8. For each of these recommendation systems, I solve the equilibrium in three distinct phases: First, the platform updates the recommendation system. Second, sellers update prices. In the next phase of the project will include a third step where consumers update their beliefs about the recommendation system.[49]

Figure 9.1: Counterfactual Timing



My outcomes of interest are seller profits, quantity sold, platform revenue, and consumer surplus. I repeat the counterfactual analysis under different supply-side assumptions, fixed marginal cost, common economies of scale and soft capacity constraints, and star-level economics of scale and soft capacity constraints. There are 60 counterfactuals based on five recommendation systems, three supply-side assumptions, and updating versus not updating prices.

[49]This is a work in progress, these results will be available shortly.

9.2 Solving for Equilibrium

Here I briefly describe the process to solve for the new equilibrium. I first solve for the baseline counterfactual of subperiod uniform pricing, using the platform model to generate recommendations, and a contraction mapping of the firms first order conditions. The resulting prices serve as the baseline for the remaining counterfactual.

Without Price Updates. To solve for the new equilibrium without price updates, I take the structural model and replace the product recommendation system that provides the deterministic portion of relevance scores with the target recommendation system ψ. Then keeping prices fixed, I calculate quantities, gross booking revenue, and firm profits, and consumer surplus. The consumer surplus depends on the utility of the predicted purchases, and search costs of the predicted clicks.

With Price Updates. To solve for the new equilibrium with price updates follows the same process as without price updates, except I solve for new prices with a contraction mapping of the hotel's first order conditions.

Counterfactual Limitations. As noted above, one limitation of the counterfactuals comes from the need to aggregate to the subperiod level. There are a few other limitations, as this is a work in progress. Right now the counterfactual focus on four subperiods for the top market. There are over 700 unique hotels in the top market, but many appear rarely. I hold the prices of hotels that appear less than five times in a given subperiod fixed.

9.3 Counterfactual Results

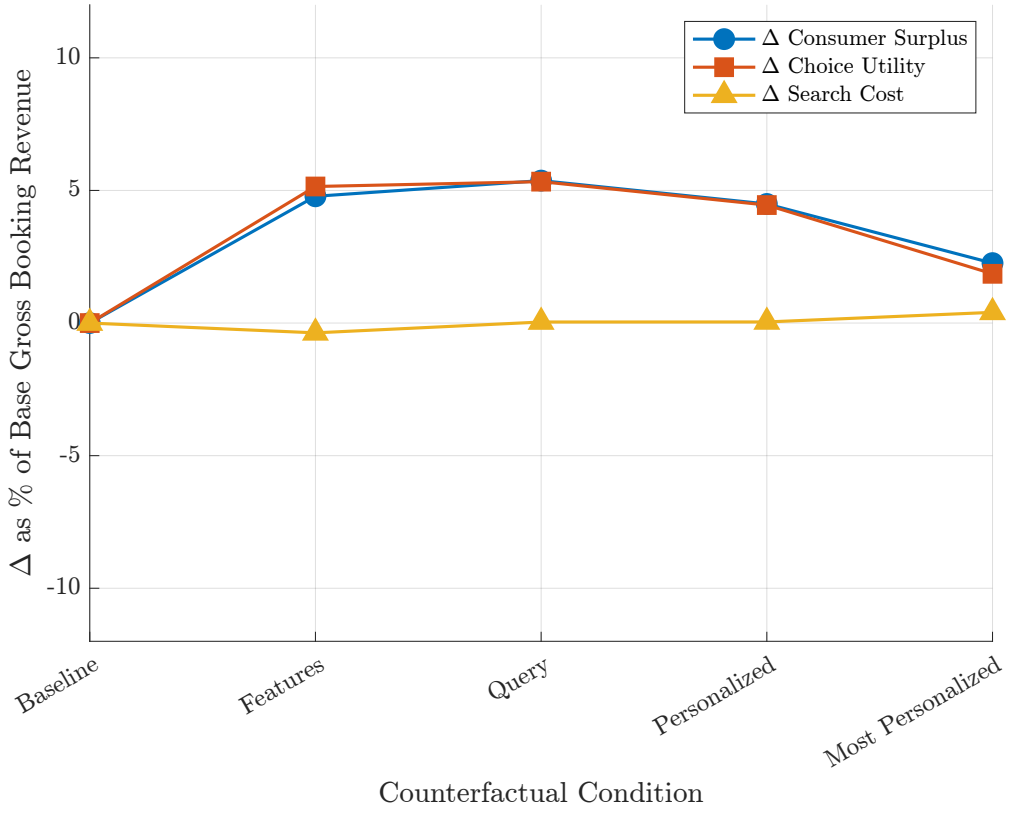
Here I present the counterfactual results for the primary specifications. In the primary specifications, the supply side includes star-rating specific marginal cost functions. The extended set of results are included in Appendix G.

9.3.1 Counterfactual without Price Updates: Welfare Gain

Holding prices fixed, I find a welfare gain from personalized recommendations. Figure 9.2 plots the consumer surplus results, including the utility of final choices and search cost of clicks. Table 9.1 includes the complete set of outcomes, quantity, gross booking revenue, and hotel profits. It also includes an approximate platform revenue, assuming they take a 10% commission. I present the

consumer welfare numbers relative to the baseline since, with discrete choice models, we can identify differences in consumer surplus but not the absolute level.

Figure 9.2: Counterfactual without price updates and with star-level economies of scale and soft capacity constraints: *welfare gain*



Note: Change in values represented as a percent of baseline gross booking revenue.

I do note an marginal decrease in consumer surplus going from the query level to personalized and most personalized. If we were to take the demand model as the truth, this would be due to potential over-fitting of the personalized recommendation systems. However, I suspect the more likely explanation is a limitation of the current demand specification. The personalized and most personalized models were trained using variables that the demand model does not include. As a next step, there are two options to address this concern: 1) A new demand model with more parameters to capture heterogeneous preferences. 2) Using a conditioning on individual tastes (COIT) post estimation procedure to make personalized welfare predictions (Revelt and Train, 2000).

The welfare gains come from consumers choosing higher utility products and lower incurred search costs. Gross booking revenue remains relatively unchanged, indicating that consumers choose higher utility products but do not, on average, substitute from the outside option to one of the

Table 9.1: Counterfactuals with No Price Updates, with Star-Level Economies of Scale and Soft-Capacity Constraints

		Recommendation System			
Outcomes	**Baseline**	**Features**	**Query**	**Personalized**	**Most Personalized**
Quantity	508.5	505.9	505.8	505.7	505.8
Gross Booking Revenue ($100s)	1,809.37	1,804.72	1,806.78	1,806.16	1,807.52
Hotel Profits ($100s)	984.96	985.14	985.18	984.75	984.99
Approx Platform Revenue ($100s)	180.94	180.47	180.68	180.62	180.75
Consumer Welfare					
Δ Consumer Surplus ($100s)	0	86.48	97.18	81.25	40.89
Δ Choice Utility ($100s)	0	93.17	96.47	80.50	33.59
Δ Search Cost ($100s)	0	-6.69	0.71	0.75	7.30

inside goods.

Back of the Envelope Welfare Change The gain in consumer surplus is 2.3% of total booking revenue. If we want an idea of the scale of the welfare effects of going from baseline to most personalized, we can scale the consumer surplus change by Expedia's gross booking revenue for the same year, 2013. This calculation would imply ∼$0.9 billion increase in consumer surplus. These results are consistent with previous literature that finds welfare gains from improving recommendation systems while holding prices fixed.

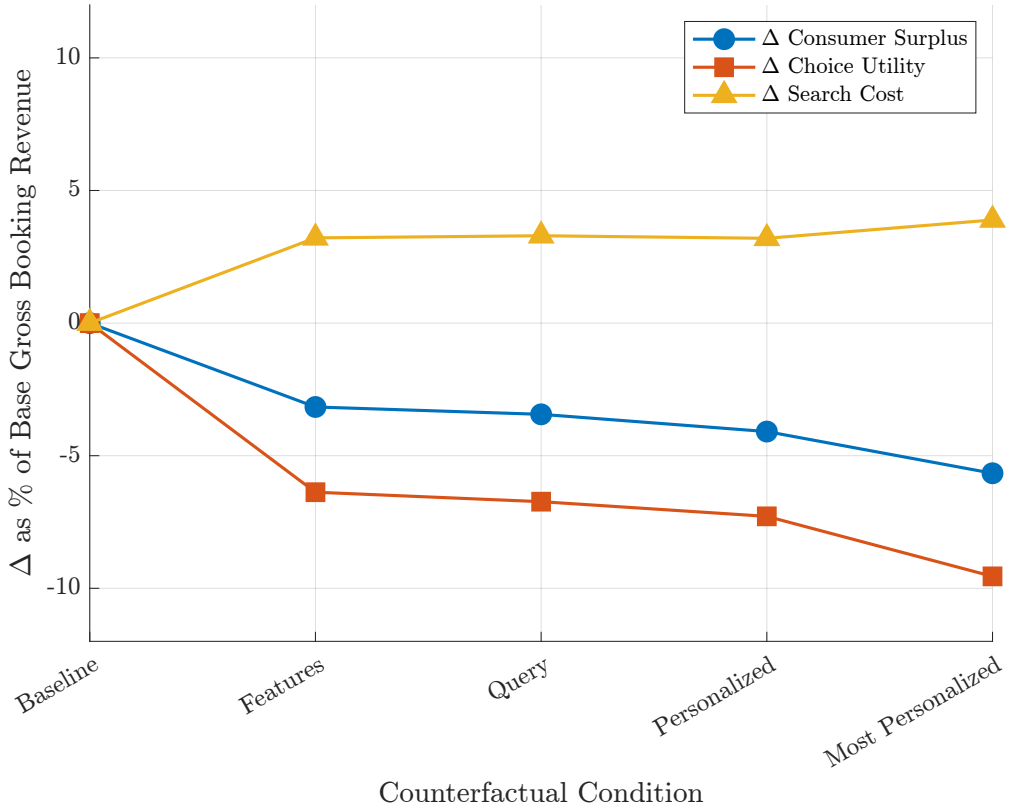
9.3.2 Counterfactual with Price Updates: Welfare Loss

Once sellers can update prices, I find a welfare loss from personalized recommendations. Figure 9.3 plots the consumer surplus results, including the utility of final choices and search cost of clicks. Table 9.2 includes the complete set of outcomes.

The results are consistent with a market becoming less competitive. Going from baseline to most personalized, sellers increase prices and see a 4.9% increase in profits. I also find a 4.5% decrease in quantity. Consumer surplus decreases by 5% of baseline gross booking revenue. Scaling these results up by the gross booking revenue of Expedia in 2013 would correspond to a ∼$2 billion decrease in consumer surplus. This would represent a total welfare loss, as the decrease in consumer surplus is nearly double the increase in hotel profits.

From these counterfactual simulations, I find that ignoring seller price adjustments causes considerable differences in the estimated impact of personalization. Without price adjustments, personalization would increase consumer surplus due to the efficiency gains from steering consumers

Figure 9.3: Counterfactuals with price updates and star-level economies of scale and soft capacity constraints: *welfare loss*



Note: Change in values represented as a percent of baseline gross booking revenue.

Table 9.2: Counterfactuals with Star-Level Economies of Scale and Soft-Capacity Constraints

		Recommendation System			
Outcomes	**Baseline**	**Features**	**Query**	**Personalized**	**Most Personalized**
Quantity	517.6	495.2	494.8	494.2	494.3
Gross Booking Revenue ($100s)	1,830.09	1,825.62	1,829.00	1,827.90	1,829.79
Hotel Profits ($100s)	974.02	1,020.00	1,021.20	1,021.32	1,022.03
Approx Platform Revenue ($100s)	183.01	182.56	182.90	182.79	182.98
Consumer Welfare					
Δ Consumer Surplus ($100s)	0	-27.37	-62.97	-66.19	-92.02
Δ Choice Utility ($100s)	0	-75.16	-124.19	-118.06	-158.50
Δ Search Cost ($100s)	0	47.79	61.22	51.88	66.48

to products that better match their tastes. However, the new algorithm and resulting reshaped demand would incentivize sellers to increase prices to profit from less price-sensitive demand. Once accounting for seller price adjustments I find that the price adjustments more than offset the efficiency

gains from personalizing. Indeed, once accounting for seller price adjustments, I find a total welfare loss from personalization.

10 Conclusion

In this paper, I explore the welfare effects of personalized recommendations in digital markets using data from Expedia Group. While this paper focuses on Expedia, it addresses a familiar dynamic between sellers and e-commerce platforms in the increasingly digital economy. The platform chooses its platform design, including the recommendation system, but third-party sellers, in this case hotels, set prices. Personalized recommendations can improve consumer welfare through the long-tail effect, where consumers find products that better match their tastes. However, third-party sellers, facing demand from better-matched consumers, may be incentivized to increase prices.

I develop a structural model of demand, platform product recommendations, and hotel pricing behavior to quantify the tradeoff between match quality and price competition. On the demand side, this paper proposes an optimal sequential search model where consumers have beliefs about the joint distribution of product features and recommendations, form consideration sets through clicks, and make a final purchase decision from their consideration set. For the product recommendation model, I use a "model of a model" machine learning approach to reverse engineer Expedia's default recommendation system. Combining the results from the demand and recommendation system models allows for the supply-side model where capacity-constrained hotels consider how changes in price impact position on the page in search results.

In addition to the structural model of demand, platform recommendations, and seller pricing behavior, I develop four increasingly personalized recommendation systems. I use an ensemble of LambdaMARTs, a popular machine-learning algorithm for ranking problems. I use Expedia data to train the recommendation systems, where Expedia randomized product rankings.

In my counterfactuals, I find that ignoring seller price adjustments would cause considerable differences in the estimated impact of personalization. Without price adjustments, personalized recommendations would increase consumer surplus by 2.3% of total booking revenue ($0.9 billion). However, once sellers update prices, personalization would lead to a welfare loss, with consumer surplus decreasing by 5% of booking revenue ($2 billion). This paper provides actionable insights relevant to researchers, platforms, and policymakers and highlights an overlooked concern in e-commerce platform research and regulation: Better recommendation systems may reduce competition

and harm consumer welfare. This finding is important to consider as e-commerce platforms' access to personal data grows, and technological improvements allow platforms to deploy increasingly sophisticated recommendation systems.

Policies that mitigate these pricing effects could be available to regulators and platforms. In the next step of this project, I plan to consider a policy alternative that tunes the recommendation systems to account for price competition. Operationally, this would involve increasing or decreasing the price sensitivity of the personalized recommendations to achieve a policy goal, for example, maximizing total surplus or equating equilibrium prices with those that would prevail without personalization. These price-tuned recommendation systems can potentially improve platform revenue, hotel profits, and consumer surplus.

References

Abaluck, J., G. Compiani, and F. Zhang (2020, March). A Method to Estimate Discrete Choice Models that is Robust to Consumer Search. Technical Report w26849, National Bureau of Economic Research, Cambridge, MA.

Agrawal, K., S. Athey, A. Kanodia, and E. Palikot (2022, December). Personalized Recommendations in EdTech: Evidence from a Randomized Controlled Trial. arXiv:2208.13940 [econ, q-fin].

Akerlof, G. A. (1970, August). The Market for "Lemons": Quality Uncertainty and the Market Mechanism. *The Quarterly Journal of Economics 84*(3), 488.

Andrews, I., M. Gentzkow, and J. M. Shapiro (2020, October). Transparency in Structural Research. *Journal of Business & Economic Statistics 38*(4), 711–722.

Athey, S. and G. Ellison (2011). Position Auctions with Consumer Search. *The Quarterly Journal of Economics 126*(3), 1213–1270. Publisher: Oxford University Press.

Berry, S., J. Levinsohn, and A. Pakes (1995, July). Automobile Prices in Market Equilibrium. *Econometrica 63*(4), 841.

Berry, S. T. (1994). Estimating Discrete-Choice Models of Product Differentiation. *The RAND Journal of Economics 25*(2), 242.

Betancourt, J., A. Hortaçsu, A. Oery, and K. Williams (2022, August). Dynamic Price Competition: Theory and Evidence from Airline Markets. Technical Report w30347, National Bureau of Economic Research, Cambridge, MA.

Blake, T., S. Moshary, K. Sweeney, and S. Tadelis (2021, July). Price Salience and Product Choice. *Marketing Science 40*(4), 619–636.

Brown, Z. Y. and J. Jeon (2022). Endogenous Information and Simplifying Insurance Choice.

Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An Overview. *Learning 11*.

Cardell, N. S. (1997, April). Variance Components Structures for the Extreme-Value and Logistic Distributions with Application to Models of Heterogeneity. *Econometric Theory 13*(2), 185–213.

Chapelle, O. and Y. Chang (2011, January). Yahoo! Learning to Rank Challenge Overview. In *Proceedings of the Learning to Rank Challenge*, pp. 1–24. PMLR. ISSN: 1938-7228.

Chen, Y. and S. Yao (2017, December). Sequential Search with Refinement: Model and Application with Click-Stream Data. *Management Science 63*(12), 4345–4365.

Choi, M., A. Y. Dai, and K. Kim (2018). Consumer Search and Price Competition. *Econometrica 86*(4), 1257–1281.

Compiani, G., G. Lewis, S. Peng, and W. Wang (2021). Online Search and Product Rankings: A Double Index Approach. *SSRN Electronic Journal*.

De los Santos, B. and S. Koulayev (2016, July). Optimizing Click-Through in Online Rankings with Endogenous Search Refinement.

Diamond, P. A. (1971, June). A model of price adjustment. *Journal of Economic Theory 3*(2), 156–168.

Dinerstein, M., L. Einav, J. Levin, and N. Sundaresan (2018, July). Consumer Price Search and Platform Design in Internet Commerce. *American Economic Review 108*(7), 1820–1859.

Donnelly, R., A. Kanodia, and I. Morozov (2020). Welfare Effects of Personalized Rankings. *SSRN Electronic Journal*.

Ellison, G. and S. F. Ellison (2009). Search, Obfuscation, and Price Elasticities on the Internet. *Econometrica 77*(2), 427–452.

Farronato, C. and A. Fradkin (2022, June). The Welfare Effects of Peer Entry: The Case of Airbnb and the Accommodation Industry. *American Economic Review 112*(6), 1782–1817.

Farronato, C., A. Fradkin, and A. MacKay (2023, January). Self-Preferencing at Amazon: Evidence from Search Rankings. Technical Report w30894, National Bureau of Economic Research, Cambridge, MA.

Galichon, A. (2022, April). ON THE REPRESENTATION OF THE NESTED LOGIT MODEL. *Econometric Theory 38*(2), 370–380.

Gardete, P. M. and M. Hunter (2018). Avoiding Lemons in Search of Peaches: Designing Information Provision. Research Papers, Stanford University, Graduate School of Business.

Goldfarb, A. and C. Tucker (2019, March). Digital Economics. *Journal of Economic Literature 57*(1), 3–43.

Greminger, R. P. (2022). Heterogeneous Position Effects and the Power of Rankings. Publisher: arXiv Version Number: 2.

Honka, E. (2014, December). Quantifying search and switching costs in the US auto insurance industry. *The RAND Journal of Economics 45*(4), 847–884.

Honka, E. and P. Chintagunta (2017, January). Simultaneous or Sequential? Search Strategies in the U.S. Auto Insurance Industry. *Marketing Science 36*(1), 21–42.

Hortaçsu, A., O. Natan, H. Parsley, T. Schwieg, and K. Williams (2021, December). Demand Estimation with Infrequent Purchases and Small Market Sizes. Technical Report w29530, National Bureau of Economic Research, Cambridge, MA.

Hu, H. and J. Pang (2021). Model Extraction and Defenses on Generative Adversarial Networks. Publisher: arXiv Version Number: 1.

Kim, J. B., P. Albuquerque, and B. J. Bronnenberg (2010). Online Demand Under Limited Consumer Search. *Marketing Science 29*(6), 1001–1023. Publisher: INFORMS.

Koulayev, S. (2014). Search for differentiated products: identification and estimation. *The RAND Journal of Economics 45*(3), 553–575. Publisher: Wiley.

Lam, H. T. (2021). Platform Search Design and Market Power.

Lee, K. H. and L. Musolff (2021). Entry Into Two-Sided Markets Shaped By Platform-Guided Search.

McFadden, D. (1989). A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration. *Econometrica 57*(5), 995–1026. Publisher: [Wiley, Econometric Society].

Moehring, A. (2023). Personalized Rankings and User Engagement: An Empirical Evaluation of the Reddit News Feed.

Morozov, I. (2023, March). Measuring Benefits from New Products in Markets with Information Frictions. *Management Science*, mnsc.2023.4729.

Morozov, I., S. Seiler, X. Dong, and L. Hou (2021, September). Estimation of Preference Heterogeneity in Markets with Costly Search. *Marketing Science 40*(5), 871–899.

Oh, S. J., M. Augustin, B. Schiele, and M. Fritz (2018, February). Towards Reverse-Engineering Black-Box Neural Networks. arXiv:1711.01768 [cs, stat].

Orekondy, T., S. J. Oh, Y. Zhang, B. Schiele, and M. Fritz (2020, September). Gradient-Leaks: Understanding and Controlling Deanonymization in Federated Learning. arXiv:1805.05838 [cs, stat].

Papernot, N., P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami (2017, March). Practical Black-Box Attacks against Machine Learning. arXiv:1602.02697 [cs].

Platt, J. (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif. 10*.

Reimers, I. and J. Waldfogel (2023, October). A Framework for Detection, Measurement, and Welfare Analysis of Platform Bias. Technical Report w31766, National Bureau of Economic Research, Cambridge, MA.

Revelt, D. and K. Train (2000). Customer-Specific Taste Parameters and Mixed Logit.

Stigler, G. J. (1961, June). The Economics of Information. *Journal of Political Economy 69*(3), 213–225.

Teng, X. (2022). Self-Preferencing, Quality Provision, and Welfare in Mobile Application Markets. *SSRN Electronic Journal*.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Tramer, F., F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart (2016). Stealing Machine Learning Models via Prediction APIs. *25th USENIX Security Symposium (USENIX Security 16)*.

Ursu, R., S. Seiler, and E. Honka (2023, January). The Sequential Search Model: A Framework for Empirical Research.

Ursu, R. M. (2018, August). The Power of Rankings: Quantifying the Effect of Rankings on Online Consumer Search and Purchase Decisions. *Marketing Science 37*(4), 530–552.

Weitzman, M. L. (1979, May). Optimal Search for the Best Alternative. *Econometrica 47*(3), 641.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data* (2nd ed ed.). Cambridge, Mass: MIT Press. OCLC: ocn627701062.

Appendix

A Data

A.1 Additional Expedia Data Images

These images come from the competition website and the data summary from the IDCM 2013 presentation of the competition data.

Figure A.1: Kaggle Competition



Figure A.2: Expedia search can be preceded by Google or Bing search

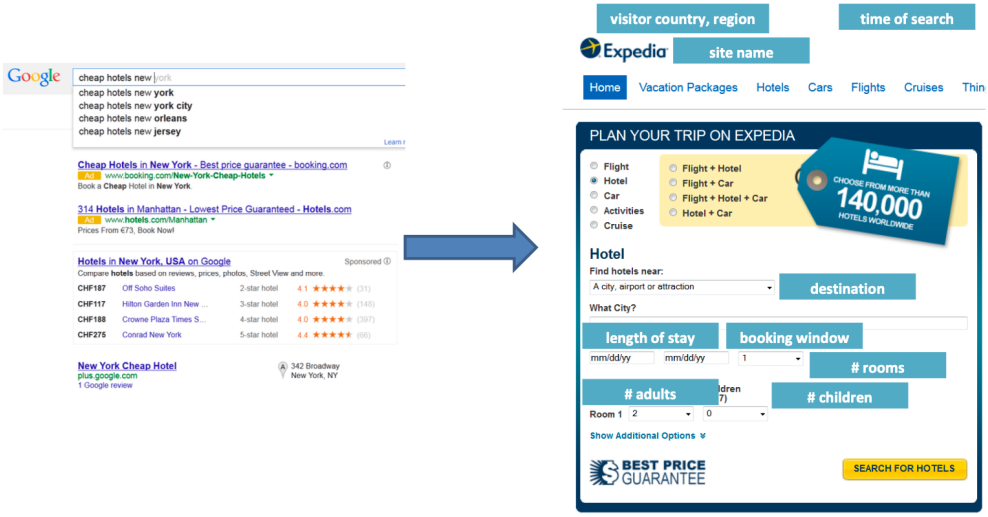


Figure A.3: Purchase Information

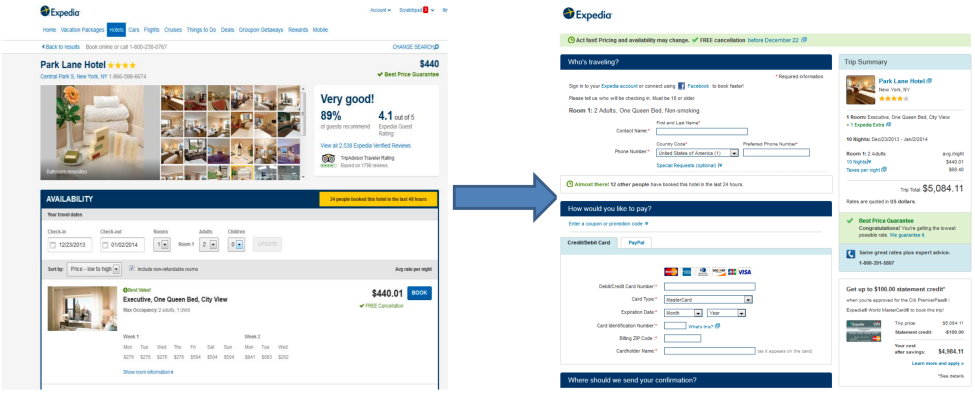
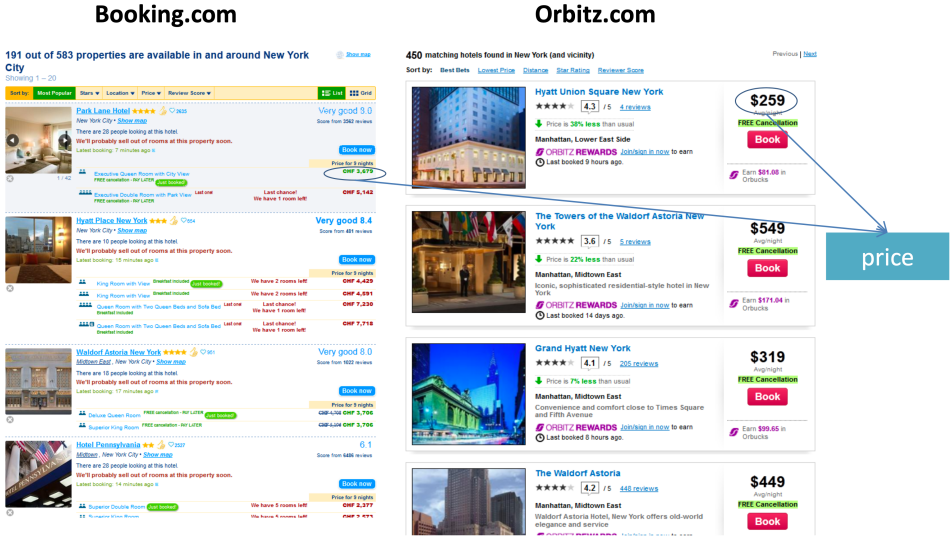


Figure A.4: Sometimes Expedia tracks prices and availability on other OTAs



A.2 Data Processing Details

This section highlights some of the key data processing steps. The Expedia data were released for a data science competition and are well-suited for training recommendation systems. There are, however, a few limitations that present difficulties in conducting the type of demand estimation and counterfactual analysis in this paper. This section highlights those challenges and the approaches that I use to address them. More details on the data processing approach are included in the appendix.

A.2.1 Market Definitions via K-means Clustering

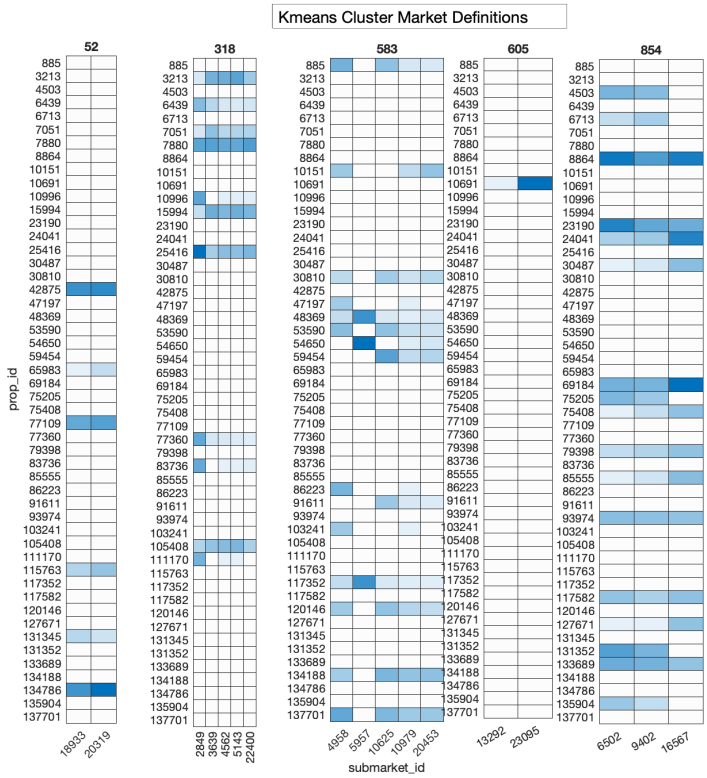
In this section, I describe how I define markets from the deidentified data using a data-driven approach. The data include ID variables for properties, countries, and search terms but lack specific keys. For example, while an identifier might indicate "search term 52," there is no direct link to a specific term such as "Manhattan, NY." Multiple search terms could correspond to the same underlying market. As this paper evaluates a supply-side problem of hotel pricing behavior, it is crucial to ensure that I do not mistakenly exclude observations from a significant portion of the market. I define markets using K-means clustering, an unsupervised machine-learning technique. This clustering procedure matches search terms based on the similarity of their results. In plain terms, this procedure aims to match search terms that produce a common set of hotels in the displayed results. Further details on the k-means clustering procedure are included in the appendix.

Figure A.5 displays cluster definitions for five markets. Each column stands for a search term, while the rows denote hotels. The shade of a hotel search term cell reflects the frequency with which that hotel appears in searches, weighted by position.

A.2.2 Final Transaction Price Prediction

A limitation of this dataset is that it records final transaction prices only when there is a purchase. When a result for a hotel is clicked but no purchase is made, the consumer may still discern the final transaction price, but this price is omitted from the data. Transaction prices are important for two reasons. First, they influence consumer search and purchase decisions. A consumer might, through a click, learn the final price, which informs their next search or purchase decisions. Second, for an accurate measure of consumer welfare, the final prices are essential, as these represent the actual expenditures by consumers.

Figure A.5: Market Definitions by Search Term Clusters

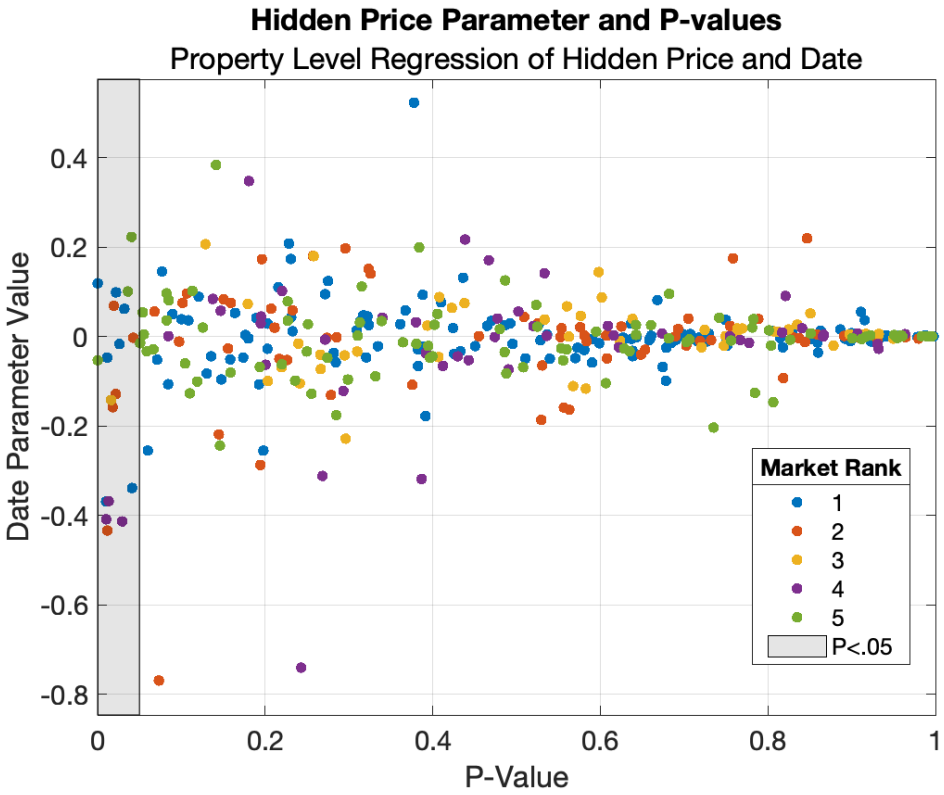


Notes: visualization includes subset of search terms matched to top 5 markets. The prop_id refers to hotels. The figure includes a subset of hotels that appears in on of the top markets. The k–means clustering procedure has one dimension for each hotel that appears in the data in at least five searches.

To address this missing data issue, I impute the percent difference between the headline price and the final per-night transaction prices using the median hotel stay length. I use hotel-length of stay median, to impute the percent difference between headline prices and final transaction prices. I use the hotel median for hotels with a limited number of transactions, while hotels with fewer than 3 observations are assigned the market-length of stay median. One potential concern with this methodology is that hotels could have modified their concealed pricing strategy during the study. To evaluate this concern I run a regression for each hotel in the top-5 markets with hidden price as the left hand side variable, and date as the right hand side variable. If many firms changed hidden pricing strategy during my period of study, I would expect many significant coefficients on date. Figure A.6, plots the results of this robustness check.

In Figure A.6, each point represents the property-specific coefficient on date from each regressions. The parameter value is plotted vertically, and the p-value is plotted horizontally. The shaded regions

Figure A.6: Date parameter and P-value of hotel-level regressions on hidden price



covers marks the estimates that are significant on the 5% level. f many firms changed hidden pricing strategy during my period of study I would expect to see bunching in the statistically significant region of the figure. Instead, roughly 5% of points are significant, which is in line with what one would expect to see just by chance.

A.2.3 Click Order Prediction

For each consumer-query, I observe which items each consumer clicked and which they purchased, but I do not know the order of clicks. For the naturally ordered data in the top five markets (by booking revenue), 92.5% of the consumer-sessions have only one click. I use a linear prediction model of clicks, where the click likelihood depends on the visible price, price interacted with promotions, the non-date visible product features, and a spline of slots. I then impute the order based on predicted click likelihood.

An alternative approach is to assume that the clicks happen in order of slot. However, this could overestimate position effects since this would force parameter estimates based on data where higher-ranked products are always clicked before lower-ranked ones. Another approach is to adjust

the likelihood function. Alternative methodologies might assume that clicks occur in the sequence of the product slot.

A.3 Sample Selection

These data were initially intended for a data science competition aimed at developing recommendation systems. In that context, consumer-sessions that include a purchase are more valuable than consumer-sessions that have only clicks, and consumer-sessions with clicks tend to be more valuable than consumer-sessions without them. It is also important to note that e-commerce platforms often consider conversion rates proprietary information.

Three issues arise from the competition's data sampling method: 1) selection on clicks, 2) oversampling of transactions, and 3) ambiguity in the sample size. The following subsections detail each of these concerns and my approach to addressing them.

A.3.1 Selection on Clicks

The data include only consumer-sessions for which the consumer clicked at least one product. In demand estimation, I address this issue by using conditional likelihoods, conditioning each target consumer's joint likelihoods of observed clicks and purchases on the likelihood of clicking at least one product. On the supply side, I address this issue by using my parameter estimates from demand to reweight observations. The estimation section of this paper offers a more detailed discussion of these adjustments.

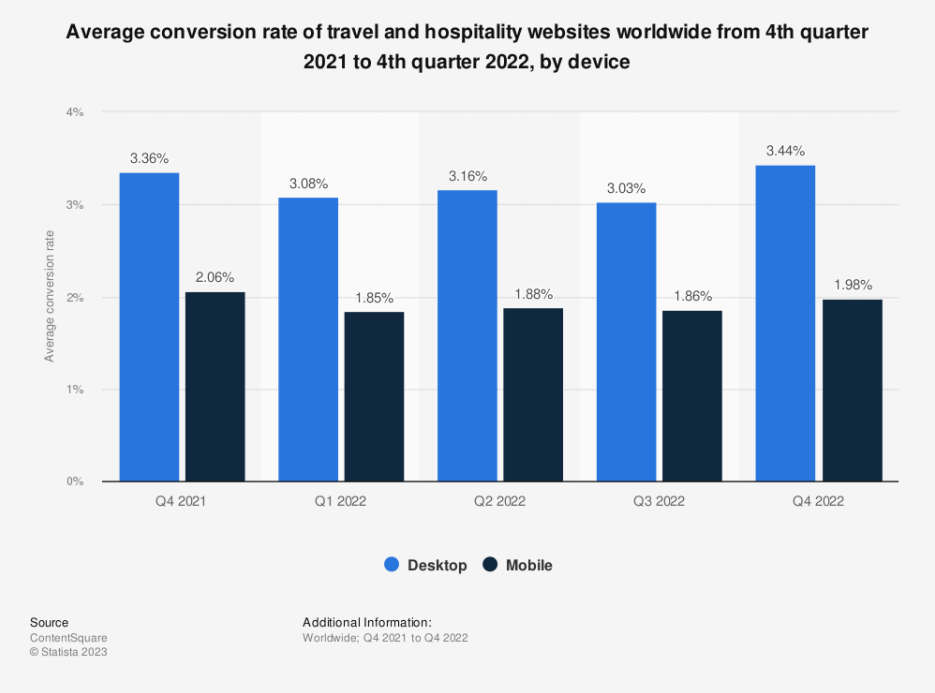
A.3.2 Oversampling of Transactions

In the original competition data, observations are sampled randomly but at different rates depending on whether the observation includes a purchase or only clicks. Consequently, this means oversampling of observations in which the consumer made a purchase relative to those in which the consumer clicked but did not purchase anything (chose the outside option).

It is important to address this issue as I need to estimate the population distribution of preferences and capture substitution patterns in my counterfactual simulations.

Again, the true underlying conversion rates are confidential, so here, the task is to choose sensible parameters. To select the structural weights, I look to other sources for reasonable values. First, I need the percentage of searches with at least one click. I chose 30%—which is within the range of numbers reported in De los Santos and Koulayev (2016), which finds that 33% of searches

Figure A.7: Hospitality Average Conversion Rates



resulted in a click for an OTA in Chicago, and Ursu (2018), which finds a 23% click rate for an OTA in Manhattan. Next, I need a conversion rate. I choose 5%, meaning that 5% of searches end in a transaction.[50] These numbers, together, imply a conversion rate conditional on at least one click of 16.67%. I then apply inverse sampling weights to the log-likelihoods during demand estimation to align with this conversion rate.

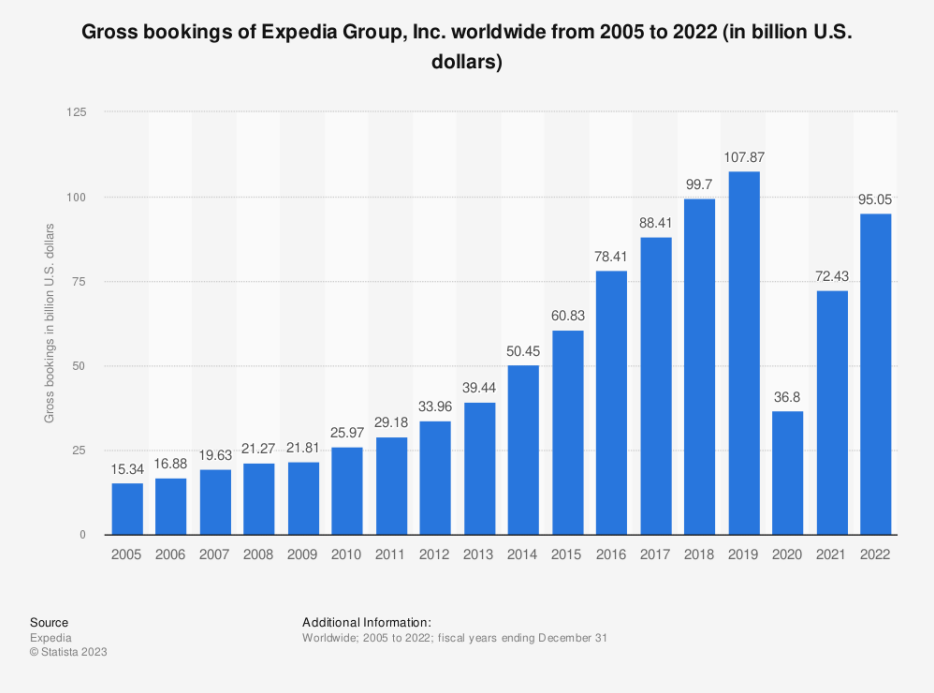
While these weights might not capture the full picture, the proprietary nature of exact conversion rates necessitates imposing a reasonable structural assumption instead of attempting to recover exact conversion rates. An important next step of this project is to conduct a sensitivity analysis in line with Andrews et al.'s (2020) guidelines on transparency in structural modeling. In the exercise, I will assess the robustness of the primary findings by rerunning the main analysis under different assumed conversion rates.

A.3.3 Sample Rates

The last issue relates to sample size. The data's scope does not cover all Expedia searches and transactions. However, the granularity of the dataset, which includes gross booking revenue combined

[50] 2021-2022 statistic on travel and hospitality websites range from 1.8-3.5%. Ursu (2018) notes a 1% rate in supplementary data.

Figure A.8: Reported gross bookings of Expedia Group, 2005-2022



source: https://www.statista.com/statistics/269386/gross-bookings-of-expedia/

with Expedia's publicly disclosed revenue figures, permits broader extrapolation.

B Demand Notes

This section provides additional detail on the features of the demand model.

I develop a novel optimal sequential search demand model, based on Weitzman (1979), and estimation strategy. To measure consumer welfare and have realistic substitution patterns in counterfactuals, my demand model accounts for important aspects of platform design, including four innovations over standard search models. The model can be applied more broadly to other online or offline settings where consumers engage in costly search and where the researcher observes search and purchase decisions. I develop a novel optimal sequential search demand model, based on Weitzman (1979), and estimation strategy. To measure consumer welfare and have realistic substitution patterns in counterfactuals, my demand model accounts for important aspects of platform design, including four innovations over standard search models. The model can be applied more broadly to other online or offline settings where consumers engage in costly search and where the researcher observes search and purchase decisions.

First, to account for feature emphasis, the demand model allows for both visible and hidden product features. This means that consumers know some of the product features prior to searching and learn about other product features after searching. In the context of Expedia, consumers know the product features that appear on the landing page and can learn the remaining product features by clicking through to the product-specific page. The standard search model assumption is that consumers have full information on product features and search only over an independent and identically distributed match quality term. This part of the model complements recent work by Compiani et al. (2021).

The second innovation relaxes the standard assumption that consumers learn the match term from search. Some recent work has introduced a visible and hidden term (Ursu et al., 2023; Morozov et al., 2021; Morozov, 2023). This paper builds on that by introducing a data-driven approach based on the variance structure of the match quality term used in nested logit established by Cardell (1997) and recent advances by Galichon (2022). With this data-driven approach, the model subsumes both the full-information demand model and the traditional search model.

The third innovation relates to the mechanisms underlying position effects. The standard approach in the empirical literature is to impose a structural assumption that position impacts demand only through search cost. I allow product ranking to influence demand both through search cost and through rational expectations. In terms of rational expectations, prior to a search, consumers form beliefs on the hidden product features conditional on position on the page. As a robustness check, I test this structural assumption against one where position impacts only search costs. To allow for these sophisticated beliefs, in estimation, I include a value function approximation in an inner loop. This setup introduces sufficient flexibility to test competing structural assumptions about consumer beliefs. For example, this also could allow for consumers' having higher-order beliefs about how the relationship between position on the page impacts the variance of hidden product features.

The fourth innovation takes advantage of a useful source of variation in the data. In my case, consumers arrive at the platform by searching for stays of different lengths. I focus on stays of one to four nights. Presumably, a consumer searching for a one-night stay and a consumer searching for a four-night stay face similar search costs. However, their returns to search are quite different. Since the consumer with the longer stay would consume more of the product and pay for each night, her returns to search are higher. This variation helps me separately identify consumer preferences from search costs, a common challenge in the search literature. This is consistent with other areas

of the search literature in which consumers engage in more search when the returns to search are higher (Brown and Jeon, 2022). The best of my knowledge, this is the first paper to take advantage of variation in quantity in this way.

C Model

C.1 Structure of Match Quality Term

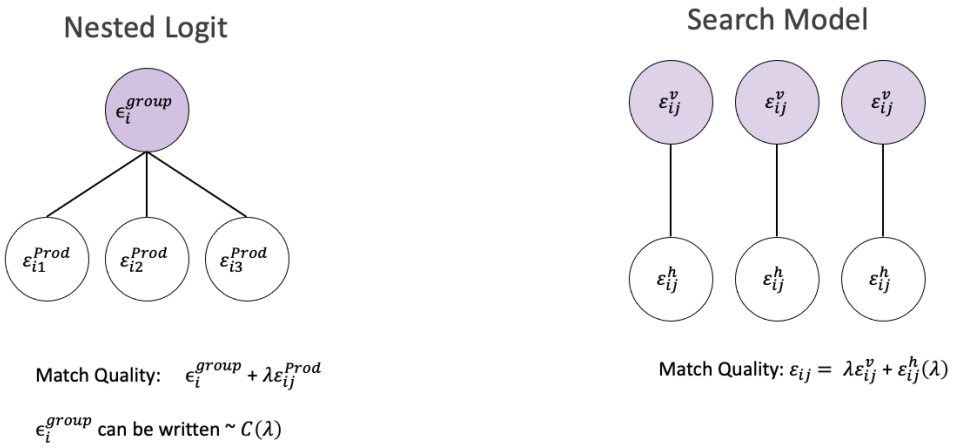
This section describes the details of the match quality term. In the demand model, I treat product features as either visible or hidden, with visible features appearing on the landing page. The match quality term follows a similar structure, with visible and hidden components, but with an added parameter λ that determines how much of the match quality term is known before search and how much is learned from search with along with the hidden product features. We can express the sum of the terms as:

$$\epsilon_{ijt} = \lambda\varepsilon^{v}_{ijt} + \varepsilon^{h}_{ijt}(\lambda) \tag{C.1}$$

where ϵ_{ijt} is consumer i's match quality for product j at time t and follows an i.i.d. type-1 extreme value distribution. ε^{v}_{ijt} is the match quality known before search and follows an i.i.d. type-1 extreme value distribution. It is multiplied by $\lambda \in (0, 1)$. ε^{h}_{ijt} follows a Cardell(λ) distribution , whose characteristic function depends on λ, and ε_{ijt} follows an i.i.d. type-1 extreme value distribution. As λ approaches 1, more of the match quality is visible, and the Cardell(λ) distribution collapses to zero. As λ approaches 0, more of the match quality is hidden, and the Cardell(λ) distribution approaches the type-1 extreme value distribution. The split error structure is similar to the nested logit, which includes nest-level and the item-level variance components of the error term. In my setup, both match quality terms are at the consumer–product–time level and are split based on the information available to the consumer. Identification of λ comes, in part, from the correlation between click decisions and product features.

Including the visible and hidden match quality term offers two advantages over reverting to the common assumption that consumers learn the entire error term from search. First, it overcomes one of the weaknesses of optimal sequential search models, where the search order is deterministic in product features. Second, the data-driven approach relaxes the assumption that the entire error is

Figure C.1: Split Error Term Structures: Nested Logit vs Search Model



learned from search, allowing more model flexibility.[51] Recent papers, for example, Morozov et al. (2021), Morozov (2023), and Ursu et al. (2023) use normal distributions (double-probit) instead of extreme value (EV) and normalize one of the match quality terms. In contrast, I use the properties of the Cardell distribution Cardell (1997) and extreme distributions to create an EV-1 combined match quality term, which provides additional benefits. Since I normalize ϵ_{ijt} to be EV-1, the other utility parameters are scaled to this combined ϵ_{ijt} term instead of the hidden or visible component of match quality. This approach allows a straightforward interpretation of the utility and search cost parameters consistent with that under more conventional demand models. My setup also provides for within-simulation analytic expressions of choice and click likelihoods, which are necessary for the supply-side estimation and counterfactual simulations.[52]

C.1.1 Approximating the Cardell Distribution

Having discussed the split match quality term, I now turn to the challenges of taking draws from the Cardell distribution. Cardell (1997) proves the existence of the distribution. The nested logit, for example, used in Berry (1994), implicitly depends on the Cardell distribution to create an analytic expression of choice probabilities and market shares and and identify a parameter λ from the diversion ratio. In contrast, my estimation strategy requires taking draws from the Cardell distribution, which is not straightforward since the distribution does not have a closed-form probability density function (PDF) or cumulative distribution function (CDF) and there is no ready to use software package to

[51] In fact, this nests both the full-information demand ($\lambda = 1$) and sequential search ($\lambda = 0$) models.

[52] By within-simulation expression, I mean that I take simulated draws of all random parameters except ε_{ijt}^{v} and then use the property that ε_{ijt}^{v} is also EV-1 to construct choice and click likelihoods.

take these draws.

A recent advance by Galichon (2022) proves that the Cardell distribution with parameter λ is related to the positive stable distribution with parameter λ, specifically if $Z \sim P_{\text{stable}}(\lambda)$, where $P_{\text{stable}}(\lambda)$ is the stable distribution. With this property, the problem of drawing from the Cardell distribution reduces to sampling from a positive stable distribution and applying the correct transformation. In practice, drawing from the stable distribution is slow since each point is solved numerically. To avoid this slowdown, I create a fine grid of points in terms of λ and v, the probability value, numerically solved for the point in the inverse CDF of the Cardell distribution. Then, I create an interpolation object $f(\lambda, v)$.

D Estimation

E Demand Estimation and Results

E.0.1 Demand Estimation Procedure

Algorithm 1 summarizes the estimation procedure.

Algorithm 1 Demand Estimation Procedure

procedure PRE-ESTIMATION
    Select Hyperparameters
    Generate Splines
    Sample Beliefs Data
    Take Scrambled Halton Draws $D = 400$
    Estimate Expected Hidden Features: $E[x^h|\Omega]$
    Estimate Expected Price: $E[p_{ijt}|\Omega]$
end procedure
procedure MAXIMUM SIMULATED LIKELIHOOD
    Initialize Parameters θ
    while Not Converged **do**
        Update Random Coefficients Using Draws and Cholesky Factor
        Approximate Hidden-Match Quality $\varepsilon_{ijt}^{h[s]}(\lambda) = \text{ICDF}(\lambda, d_{ijt}^{[s]})$
        Calculate Utilities and Search Costs
        Calculate Expected Hidden Utility $E[\delta_{ijt}^{h[s]}|\Omega_{it}] = -e^{\rho_i^{[s]}} E[p_{ijt}|\Omega_{it}] + \beta_{ijt}^{h[s]} E[x_{ijt}^{h[s]}|\Omega_{it}]$
        procedure RESERVATION UTILITY VALUE FUNCTION APPROXIMATION
            Initialize Grid of State Variables
            Solve for ζ at Each Point on Grid
            Fit Spline Interpolation Object to ζ and Grid of State Variables
            Predict Each $\zeta_{ijt}^{[s]}$ Using Interpolation Object
        end procedure
        procedure LOGIT SMOOTHING
            Scale Per-Night Utility and Reservation Utility by Length of Stay
            Apply Logit-Smoothing Parameter ω
        end procedure
        procedure LIKELIHOODS
            Calculate Click, Purchase, and Joint Likelihoods
            Calculate Any-Click Likelihood
            Calculate Conditional Likelihood
            Calculate Log-Likelihood
            Apply Weights
        end procedure
        Calculate Weighted Log Likelihood
        Calculate Gradient $\nabla_\theta \mathcal{L}(\theta)$ (Finite-Differences)
        Update Parameters $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}(\theta)$
        Check Convergence Criteria
    end while
    return θ
end procedure

E.1 Search Cost and Slot Results

Figure E.1 plots the distribution of search cost (in utils) by position on the page.

Figure E.1: Search Cost Distribution

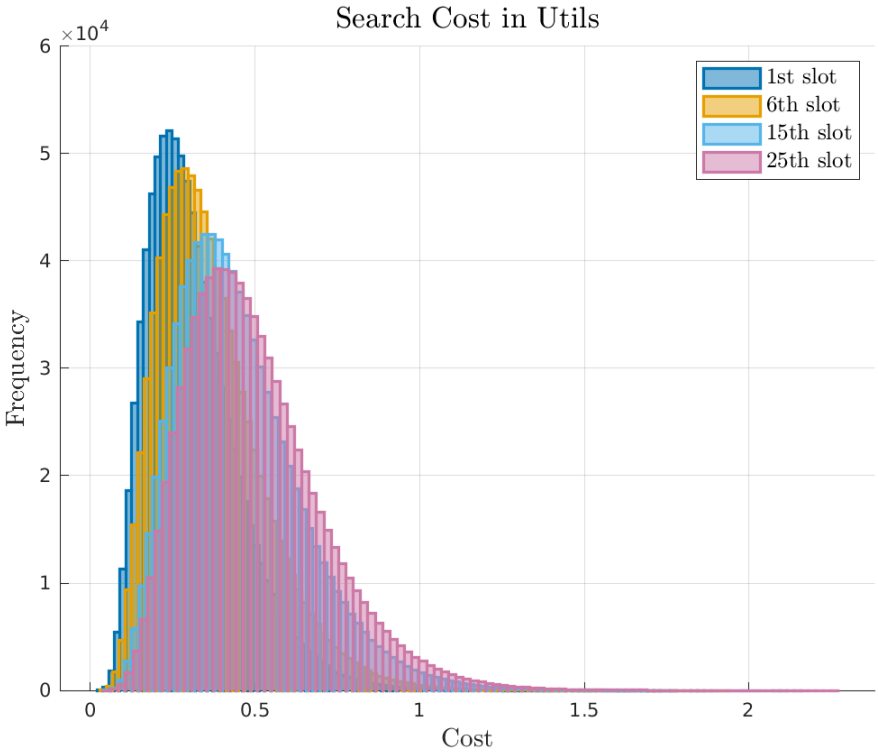


Table E.1 shows the implied median search cost, per-night utility, and reservation utility by slot and number of nights. Here, we see the dynamic where reservation utility increases with the length of stay.

Table E.1: Demand Results: Median Search Cost and Per-Night Utilities

Slot	**Search Cost ($)**	**Utility ($)**	**Reservation Utility ($)**			
			1 night	2 nights	3 nights	4 nights
1	99	97	563	871	995	1141
6	143	57	341	665	774	943
15	163	29	260	577	710	866
25	181	24	202	516	669	808

F Supply Side Estimation and Results

F.1 Expected Quantity

While the demand estimation relies on logit-smoothing to estimate the utility and search cost parameters, once I have the parameter estimates, the properties of the split error structure allow me to estimate the choice, click, and selection likelihoods without smoothing. To calculate the expected quantity at the observed prices I sum choice weighted probabilities using both the demand and platform models.

Choice probability

Choi et al. (2018) show that under optimal sequential search, consumers choose the product with the highest $\min(u_{ijt}, r_{ijt})$. This is where the split-error distribution described in Section C.1 becomes useful. Recall the visible component of match quality $\lambda\varepsilon^{v}_{ijt}$, where λ is a scale term estimated in the demand model and ε^{v}_{ijt} follows a type-1 extreme value distribution. These two properties let me construct an analytic expression for simulation level choice probabilities (without needing to use logit smoothing). To start, we need a term for the min of reservation utility and utility that excludes the visible error term (from now on called the *adjusted value*):

$$\mu^{[s]}_{ijt} = \frac{\min(u^{[s]}_{ijt}, r^{[s]}_{ijt}) - \lambda\varepsilon^{v[s]}_{ijt}}{\lambda} \tag{F.1}$$

$u^{[s]}_{ijt}$, $r^{[s]}_{ijt}$ share the same visible error term $\varepsilon^{v[s]}_{ijt}$, so I subtract to $\varepsilon^{v[s]}_{ijt}$ get the adjusted value. I also divide by λ to simplify the expression for choice probabilities. I make the same adjustments to the outside.

From here, we can construct the choice likelihoods using the demand model and observed rankings:

$$Pr(i \text{ choose } j|\theta) = \frac{1}{D}\sum_{s=1}^{D} \frac{\exp\left(\mu^{[s]}_{ijt}\right)}{\exp\left(\mu^{[s]}_{i0t}\right) + \sum_{k\in J_{it}} \exp\left(\mu^{[s]}_{ijt}\right)} \tag{F.2}$$

where θ are the parameter estimates and J_{it} is the set of all the hotels on the first page.

Observation Weights

I can similarly use the effective values, $\mu^{[s]}_{ijt}$ to calculate the observation weights. This also requires $\varrho^{[s]}_{ijt}$, which is the scaled reservation utility excluding the visible match quality term. The first step

is to calculate the selection probability,$P_{it}^{\text{selection}}$, of each consumer-query. $P_{it}^{\text{selection}}$ depends on the sampling weights (w^{in},w^{out}), the likelihood of clicking at least one product and making a purchase, and the likelihood of clicking at least one thing and choosing the outside option (no purchase).

$$P_{it}^{\text{selection}} = \frac{1}{D}\sum_{s=1}^{D}\Bigg[\overbrace{\frac{1}{w^{\text{in}}}\left(\frac{\sum_{k\in J_{it}}\exp\left(\mu_{ijt}^{[s]}\right)}{\exp\left(\mu_{i0t}^{[s]}\right)+\sum_{k\in J_{it}}\exp\left(\mu_{ijt}^{[s]}\right)}\right)}^{\text{sampling likelihood from click and purchase}} \tag{F.3}$$

$$+\quad\overbrace{\frac{1}{w^{\text{out}}}\left(\frac{\sum_{k\in J_{it}}\exp\left(\varrho_{ijt}^{[s]}\right)}{\exp\left(\mu_{i0t}^{[s]}\right)+\sum_{k\in J_{it}}\exp\left(\varrho_{ijt}^{[s]}\right)}-\frac{\sum_{k\in J_{it}}\exp\left(\mu_{ijt}^{[s]}\right)}{\exp\left(\mu_{i0t}^{[s]}\right)+\sum_{k\in J_{it}}\exp\left(\mu_{ijt}^{[s]}\right)}\right)}^{\text{sampling likelihood from click w/o purchase}}\Bigg]$$

The observation weights are the inverse sampling likelihood from the demand model

$$\hat{w}_{it}^{obs} = \left(P_{it}^{\text{selection}}\right)^{-1} \tag{F.4}$$

Now we can express the expected quantity as

$$E[q_{jtt'}|\hat{\theta}_{\text{est}}] = \sum_{i\in I_{tt'}}\left[\hat{w}_{it}^{obs}\left(\frac{1}{D}\sum_{s=1}^{D}x_{it}^{nights}P_{it}^{[s]}\left(i \text{ chooses } j|\hat{\theta}^{[s]}\right)\right)\right] \tag{F.5}$$

where $E[q_{jtt'}|\hat{\theta}_{\text{est}}]$ is the expected quantity of product j, for stays in period t, from searches happening in period t'. $I_{tt'}$ denotes the set of consumers searching for stays in period t, from searches happening in period t'. Since the choice probabilities are based on the estimated population distribution of utility parameters $\hat{\theta}$, I multiply by the inverse of the selection likelihood, $\hat{w}_{it}^{obs}$. Note that when estimating elasticity and in counterfactual simulations, $\hat{w}_{it}^{obs}$ remains fixed, but $P_{it}^{[s]}\left(i \text{ chooses } j|\hat{\theta}^{[s]}\right)$ will change due to different prices and product rankings.

F.1.1 Own Price Elasticity

I calculate the own price elasticity via finite differences. The elasticity requires both the platform and demand model results. The elasticity requires the following steps outlined in algorithm 2

Algorithm 2 Own-Price Elasticity Procedure

1: Identify the target product j^* and subperiod
2: Increase j^*'s final and headline prices by $\epsilon\%$ ▷ Small + perturbation
3: Calculate ψ_{ij^*t} using the platform model ▷ Deterministic relevance score
4: Re-assign slots using ψ, β^{slot}, and $\varepsilon_{ijt}^{\text{rec}[s]}$
5: Update expectations of hidden features based on new slots
6: Recompute utilities, search costs, and reservations with new prices and slots
7: Derive new effective values $\mu_{ijt}^{[s]}$
8: Compute expected quantity, $q_{jtt'}^{+}$ from Eq. F.5 ▷ Subperiod demand
9: Repeat steps 1-6, instead decreasing price by $\epsilon\%$ ▷ Small - perturbation for $q_{jtt'}^{-}$
10: Calculate arc elasticity for j^* using $q_{jtt'}^{+}, q_{jtt'}^{-}, \epsilon, p_{jtt'}$
11: **Repeat** for each product-subperiod ▷ Iterate over j

G Extended Counterfactual Results

G.1 Sellers Do Not Update Prices

Table G.1: Counterfactuals with No Price Updates, Fixed Marginal Cost

		Recommendation System			
Outcomes	**Baseline**	**Features**	**Query**	**Personalized**	**Most Personalized**
Quantity	508.5	505.9	505.8	505.7	505.8
Gross Booking Revenue ($100s)	1,809.37	1,804.72	1,806.78	1,806.16	1,807.52
Hotel Profits ($100s)	1,019.69	1,019.26	1,019.43	1,018.98	1,019.31
Approx Platform Revenue ($100s)	180.94	180.47	180.68	180.62	180.75
Consumer Welfare					
Δ Consumer Surplus ($100s)	0	86.48	97.18	81.25	40.89
Δ Choice Utility ($100s)	0	93.17	96.47	80.50	33.59
Δ Search Cost ($100s)	0	-6.69	0.71	0.75	7.30

G.2 Sellers Update Prices

Table G.2: Counterfactuals with Fixed Marginal Costs

		Recommendation System			
Outcomes	**Baseline**	**Features**	**Query**	**Personalized**	**Most Personalized**
Quantity	518.6	496.8	496.2	496.2	495.4
Gross Booking Revenue ($100s)	1,832.93	1,830.82	1,832.84	1,832.78	1,834.03
Hotel Profits ($100s)	1,004.22	1,049.46	1,050.32	1,050.31	1,051.70
Approx Platform Revenue ($100s)	183.29	183.08	183.28	183.28	183.40
Consumer Welfare					
Δ Consumer Surplus ($100s)	0	-37.11	-54.95	-61.11	-87.01
Δ Choice Utility ($100s)	0	-91.53	-116.13	-116.68	-159.24
Δ Search Cost ($100s)	0	54.42	61.18	55.56	72.23

Table G.3: Counterfactuals with Common Economies of Scale and Soft-Capacity Constraints

		Recommendation System			
Outcomes	**Baseline**	**Features**	**Query**	**Personalized**	**Most Personalized**
Quantity	519.0	496.2	496.1	495.7	495.3
Gross Booking Revenue ($100s)	1,832.98	1,830.09	1,830.90	1,831.48	1,833.68
Hotel Profits ($100s)	969.88	1,015.12	1,014.21	1,015.42	1,016.96
Approx Platform Revenue ($100s)	183.30	183.01	183.09	183.15	183.37
Consumer Welfare					
Δ Consumer Surplus ($100s)	0	-57.98	-63.06	-74.96	-103.79
Δ Choice Utility ($100s)	0	-116.88	-123.42	-133.56	-175.05
Δ Search Cost ($100s)	0	58.90	60.37	58.60	71.27