

**Scalable Models and Policy Learning
for Online Marketplaces**

by
Madhav Kumar

B.Sc. (Honors), Physics, University of Delhi (2008)

M.Sc. Economics, IGIDR (2011)

S.M. Management Research, MIT (2020)

Submitted to the Department of Management
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Management
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

© Massachusetts Institute of Technology 2022. All rights reserved.

Author.....
Department of Management
April 29, 2022

Certified by.....
Sinan Aral
David Austin Professor of Management
Professor of Information Technology and Marketing
Thesis Supervisor

Certified by.....
Dean Eckles
Mitsubishi Career Development Professor
Associate Professor of Marketing
Thesis Supervisor

Accepted by.....
Catherine Tucker
Sloan Distinguished Professor of Management
Professor of Marketing
Chair of MIT Sloan PhD Program

Scalable Models and Policy Learning for Online Marketplaces

by

Madhav Kumar

Submitted to the Department of Management
on April 29, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Management

Abstract

This dissertation contains three essays on designing scalable models and policy learning methods for online marketplaces. The underlying theme across all chapters is the development of data-driven practical solutions that help improve business operations and customer experiences in e-commerce.

The first chapter offers a new perspective on creating promotional bundles in cross-category retail. A scalable approach is designed that efficiently leverages historical purchases and consideration sets to learn heuristics for complementarity and substitutability using machine learning-based embeddings. Subsequently, thousands of candidate bundles are created based on these heuristics and their effectiveness is tested using a field experiment. Offline policy learning is applied to the experimental data to optimize the retailer's bundle design policy. The optimized policy is robust across product categories, generalizes well to the retailer's entire assortment, and provides an expected improvement of 35% in revenue over the baseline policy.

The second chapter investigates the impact of algorithmic pricing on consumer behavior. The adoption of algorithmic pricing by an online retailer led to considerably higher price volatility. Analysis of detailed clickstream data, complemented with lab experiments, suggests that consumers become more price sensitive when exposed to frequently changing prices caused by algorithms. Furthermore, it shows that a key mechanism driving this behavior is price salience. This finding is economically consequential because even if implementing algorithmic pricing is profitable, it triggers unintended side effects that modify consumer behavior in ways that undermine those gains.

The third chapter augments choice models and recommendation systems with consumer consideration sets. Recommendations systems are commonly used in online marketplaces to suggest relevant items (products in case of e-commerce, content in case of social media, and music/movies in case of entertainment platforms) to users. In the case of online retail, these systems typically use histori-

cal purchases to learn consumer preferences and then predict what consumers are likely to buy next. The suggested method enhances the learning of consumer preferences by flexibly incorporating consumers' historical consideration sets along with purchases with a sequential deep learning model. The search augmented recommendation system better captures consumers' latent preferences, more accurately predicts future actions, and substantially outperforms strong baselines. Finally we show that these gains are distributed across the entire spectrum of consumers and not concentrated among a small subset of high usage consumers.

Thesis Supervisor: Sinan Aral

Title: David Austin Professor of Management

Professor of Information Technology and Marketing

Thesis Supervisor: Dean Eckles

Title: Mitsubishi Career Development Professor

Associate Professor of Marketing

Acknowledgments

I thank my dissertation committee members, Prof. Sinan Aral (co-chair), Prof. Dean Eckles (co-chair), and Prof. John Hauser for their support and guidance. I am grateful for the unconditional love and support of my family: Shanti, Ranbir, Renu, Dilip, Ritu, Keshav, and Esha. I also thank the maintainers of the Charles River Esplanade and the staff at the Erie Street Flour. I have spent countless hours in these places ruminating on the burden of choice.

This thesis is dedicated in its entirety to Lily.

Contents

1 Scalable Bundling via Dense Product Embeddings	11
1.1 Introduction	12
1.2 Related work	15
1.2.1 Economics, marketing, and operations research	15
1.2.2 Market-basket analysis and recommender systems	17
1.3 Data	19
1.3.1 Purchase baskets and consideration sets	20
1.4 Product embeddings	21
1.4.1 Training and validation	24
1.4.2 Visualizing embeddings	26
1.4.3 Consideration set embeddings	31
1.4.4 Product relationships	34
1.5 Bundle generation and experiment design	36
1.5.1 Candidate bundles	37
1.5.2 Experiment design	40
1.6 Optimized bundling policy	42
1.6.1 Learning the optimized policy	43
1.6.2 Comparing policies based on different relationship scores	45
1.6.3 Embeddings vs. co-purchase revisited	46
1.6.4 Limitations	48
1.7 Managerial insights	49
1.7.1 Consistency and variation across categories	50

1.7.2	Cross-category bundles	50
1.7.3	Relative prices	52
1.8	Discussion	53
A	Supplementary tables and figures	61
B	Embeddings model	72
B.1	Intuition	72
B.2	Formal model	74
C	Hierarchical model	77
2	Algorithmic Pricing and Consumer Sensitivity to Price Volatility	81
2.1	Introduction	82
2.2	Conceptual Framework	87
2.3	Data and Empirical Setting	90
2.4	Aggregate Purchase Behavior	92
2.4.1	Heterogeneity Across Product Categories	96
2.4.2	Before-and-After	98
2.5	Consumer-Level Exposure to Price Volatility	100
2.5.1	Instrumental Variables	105
2.5.2	Haphazard Visitation Timing	110
2.6	Lab Experiments	114
2.6.1	Experiment Design	115
2.6.2	Lab Experiment Results	116
2.6.3	Price Salience and Recall	117
2.7	Discussion	118
A	Algorithmic Pricing Periods	127
B	Statistical Tests of the Algorithmic Pricing Indicator	130
C	Definition of Algorithmic Pricing	131
D	Own-Price Elasticities	132
E	Types of Products	133
F	Before-and-After Events	135

G	Aggregate Robustness Checks	138
H	Consumer Level Robustness Checks	140
I	Exposure to Algorithmic Pricing Stock	142
J	Lab Experiment	143
	J.1 MBA Students	143
	J.2 MTurk Experiment	144
3	Search Augmented Choice Models and Recommendation Systems	147
3.1	Introduction	148
3.2	Relevant literature	150
3.3	Data	151
3.4	Consideration sets and discrete choice models	153
	3.4.1 Machine Learning models	158
3.5	Deep learning based recommender system	160
	3.5.1 Architecture	160
	3.5.2 Performance	162
3.6	Discussion	164
A	Supplementary tables and figures	168
B	Tuned hyper-parameters for DNN	170

Chapter 1

Scalable Bundling via Dense Product Embeddings

Abstract

Bundling, the practice of jointly selling two or more products at a discount, is a widely used strategy in industry and a well-examined concept in academia. Scholars have largely focused on theoretical studies in the context of monopolistic firms and assumed product relationships (e.g., complementarity in usage). There is, however, little empirical guidance on how to actually create bundles, especially at the scale of thousands of products. We use a machine-learning-driven approach for designing bundles in a large-scale, cross-category retail setting. We leverage historical purchases and consideration sets determined from clickstream data to generate dense representations (embeddings) of products. We put minimal structure on these embeddings and develop heuristics for complementarity and substitutability among products. Subsequently, we use the heuristics to create multiple bundles for each of 4,500 focal products and test their performance using a field experiment with a large retailer. We use the experimental data to optimize the bundle design policy with offline policy learning. Our optimized policy is robust across product categories, generalizes well to the retailer's entire assortment, and provides expected improvement of 35% (\sim \$5 per 100 visits) in revenue from bundles over a baseline policy using product co-purchase rates.

1.1 Introduction

Bundling is a widespread product and promotion strategy used in a variety of industries such as fast food (meal + drinks), telecommunications (voice + data plan), cable (TV + broadband), and insurance (car + home insurance). Given its pervasiveness, it has received considerable attention with over six decades of research analyzing conditions under which it is profitable, the benefits of different bundling strategies, and its welfare consequences. However, despite the vast literature, scholars have offered retailers little empirical guidance on how to create good promotional bundles. For example, consider a medium-sized online retailer with an inventory of 100,000 products across multiple categories. Which two products should the retailer use to form discount bundles? There are $\binom{10^5}{2} \approx 5$ billion combinations. Conditional on selecting a candidate product, there are 99,999 options to choose from to make a bundle. Is there a principled way that the managers can use to select products to form many bundles?

In this study, we offer a new perspective on the bundle design process which leverages historical consumer purchases and browsing sessions. We use them to generate latent dense vector representations of products in such a way that the proximity of two products in this latent space is indicative of “similarity” among those products. Importantly, we distinguish between the representation of product purchases and representation of consideration sets, where consideration sets include the products that were viewed together during a browsing session. We posit that products that are frequently bought together tend to be more complementary whereas products that are frequently viewed but *not* purchased together tend to be more substitutable. Then, depending on whether the products were more frequently co-purchased or co-viewed, the degree of similarity in the latent space suggests complementarity or substitutability respectively. We put minimal structure on this latent-space-based contextual similarity to generate product bundles. We learn consumers’ preferences over these suggested bundles using a field experiment with a large U.S.-based online retailer. We combine machine learning

methods with counterfactual off-policy evaluation to optimize the bundle design policy using the results of the field experiment. Our optimized policy improves revenue by 35% (\sim \$5 per 100 visits) over the benchmark policy.

Many of the earlier papers on bundling hinged on analytical models that relied on pre-specified product complementarity or substitutability (Adams and Yellen, 1976; Schmalensee, 1982, 1984; Venkatesh and Mahajan, 1993; Venkatesh and Kamakura, 2003). Furthermore, most studies work with the idea that a single firm is producing the goods, bundling them together, and then selling them at the discounted price (Derdenger and Kumar, 2013). However, a more realistic picture — and the one we consider in this study — is one of a downstream retailer bundling products from different firms. Our work focuses on the empirical side of bundling and enhances the existing literature in economics and marketing in three ways. First, instead of considering pre-defined relationships among products, we generate continuous metrics that are heuristics for the degree of complementarity and substitutability based on historical consumer purchases and consideration sets. Second, we test the effectiveness of our methodology by running a field experiment with a large online retailer in the US, providing empirical color to a largely theoretical literature. Third, we explore the idea of generating bundles from imperfect substitutes to tap into the variety-seeking behavior of consumers, which we call *variety* bundles.

More practically, our approach has several important features that add to the literature as well as the practice of creating bundles. For instance, a major strength of our approach is that we can learn relationships between two products which have very few, or even zero co-purchases, but may still be strongly related to each other. We do this in a scalable way which allows us to systematically explore a space that would otherwise be considered of limited value in designing bundles. This permits us to develop an effective bundle design strategy in a large-scale cross-category retail setting where co-purchases are sparse, a relatively unexplored context in bundling studies. In our results, we find that bundle purchases and revenue are highly correlated with purchase embeddings after netting out the effect of

historical co-purchases. Bundles in the top quintile of residualized complementarity scores are ~ 2 -3 times more likely to be purchased as compared to the bundles in the bottom quintile.

Further, we design our experiment to explore the potential bundle space efficiently. Rather than creating bundles completely at random, we use empirically-guided bundling strategies to create multiple bundles for the same product. This allows us to learn consumer preferences for a large and varied set of bundles while showing bundles that are “meaningful” and have a reasonable likelihood of purchase. We do this by using four simple but intuitive bundling strategies to generate candidate bundles. For a given focal product, we create - 1) a bundle with the strongest cross-category complement based on purchase embeddings (CC), 2) a bundle with the strongest cross-department complement based on purchase embeddings (DC), 3) a variety bundle with the closest imperfect substitute based on the consideration embeddings (VR), and 4) a bundle based on historical co-purchase frequency (CP). We use off-policy evaluation on the results of the field experiment to optimize the bundle design policy (Dudik et al., 2014; Zhou et al., 2018; Athey and Wager, 2021). The optimized policy increases the expected bundle purchases by $\sim 31\%$ and expected bundle revenue by $\sim 35\%$ over the benchmark policy.

Importantly, we also provide implementable strategies for managers. Identifying the best bundles for a retailer with an assortment of 100,000 products involves considering an action space with millions of potential bundles, a combinatorially challenging task. Our methodology allows us to filter this action space in a principled data-driven way using machine-learning-based heuristics, providing substantial efficiency gains while accounting for consumer preferences. For example, some of the bundles created by category managers include different volumes of the Harry Potter book series, branded sports team gear (hand towel and bath towel), and same-brand shampoo and conditioner. Our approach adds several types of bundles to this set: cross-category complements such as dryer sheets with scent boosters, fruit and vegetable snacks with protein supplements, mouthwash with

deodorants, and *variety* bundles such as rotini with penne pasta, and citrus soda with root beer. More broadly, we provide insights at the category level as well. For example, we find that fresh products make good bundles with pantry, snack foods, and dairy & eggs. Sports and nutrition products go well with candy, gum, & chocolate, and laundry + cleaning supplies work well too.

We provide references for related work in the next section. Sections 3 and 4 describe the data and embeddings. Section 5 provides the experiment design and the process we used to create candidate bundles for it. Section 6 describes the optimized bundling policy. Section 7 translates our results into actionable and implementable managerial insights. Section 8 concludes.

1.2 Related work

Our study draws inspiration from two distinct strands of literature: the bundling literature from economics, marketing and operations research, and the recommendation systems literature from computer science and marketing.

1.2.1 Economics, marketing, and operations research

Literature on bundling is vast and has evolved over the decades.¹ Early work focused on understanding bundling as an effective tool for price discrimination (Stigler, 1963; Adams and Yellen, 1976). Research interests further evolved to finding conditions under which a firm might choose to sell its products as independent components vs. pure bundles vs. mixed bundles (Schmalensee, 1982, 1984; Venkatesh and Mahajan, 1993; Venkatesh and Kamakura, 2003). With the availability of granular choice data, the interest shifted towards building complementarity-based structural models for bundle choice (Chung and Rao, 2003; Chao and Derdenger, 2013; Derdenger and Kumar, 2013; Prasad et al., 2015). For example, Chung and Rao (2003) build a multi-category choice model for bundles based on the at-

¹See Stremersch and Tellis (2002) for an introductory guide to bundling from a marketing perspective and Rao et al. (2018) for a chronological account of the bundling literature.

tributes of the products. They estimate their parameters by pre-defining a set of physical features and attributes for personal computers. Although their choice model does account for cross-category bundles and hence, heterogeneous components, they use narrowly defined categories with all products being complements in usage. More recently, [Derdenger and Kumar \(2013\)](#) empirically test some of the bundling theories described in the earlier bundling papers with hand-held video games. They investigate the options of pure bundling vs. mixed bundling along with the dynamic effects of bundling for durable complementary products and find that mixed bundling leads to higher revenues.

Much of the work on bundling is focused on a multi-product monopolist. [Bhargava \(2012\)](#) extends the ideas to study the impact of a merchant bundling products from different firms. He builds an analytical model to find conditions under which pure bundling and pure components are profitable. He further shows that bundling may not be profitable due to vertical and horizontal channel conflicts unless the firms can coordinate on prices. Among empirical works, [Yang and Lai \(2006\)](#) use association rules to create bundles of books based on shopping-cart data and browsing data. They find that these bundles sell more than the bundles based solely on order data or solely using browsing data. Although, [Yang and Lai \(2006\)](#)'s idea and our idea are similar in spirit, i.e., both use browsing and purchase data to generate bundles, our scopes are widely different - books vs. cross-category retail. Nevertheless, a key takeaway from their study is that they found incremental value in using browsing data in addition to purchase data in forming bundles, an idea that we leverage too. [Jiang et al. \(2011\)](#) also study bundling in the context of an online retailer selling books and use non-linear mixed integer programming to recommend the next best product given what it is currently in the basket. They do numerical studies to show that their method leads to more customers purchasing discounted bundles as well as improved profits for the retailer.

To summarize, previous research has carefully examined the efficacy of different bundling strategies in a variety of settings with a multitude of tools such as graphical analysis, analytical frameworks, probabilistic, and structural models,

survey-based empirical exercises, and modeling historical purchase data. Conclusions, though numerous, are contingent on the assumptions the researchers have made. Depending upon the context, researchers have found bundles of complementary, substitutes, and independent products to be profitable. With lessons from these papers as strong a foundation, our paper offers a new perspective to an old problem. We do not attempt to fill any “gap” in the literature but rather deliver a novel prescriptive methodology that is rooted in data, is empirically validated using a field experiment, and is practically implementable by retail managers.

1.2.2 Market-basket analysis and recommender systems

Work on recommendation systems is wide and varied spanning multiple academic genealogies. Our work is related to Grbovic and Cheng (2018); Barkan and Koenigstein (2016); Rudolph et al. (2016), who generate product embeddings for tasks such as recommendations and personalization of vacation stays, songs, and groceries respectively. The core data framework in these papers is similar to ours, in which there is unstructured data of a sequence of objects generated through repeated actions of an agent. Those actions could be rating different movies by a viewer (Rudolph et al., 2016), or listening to songs (Barkan and Koenigstein, 2016), or purchasing multiple products together (Rudolph et al., 2016, 2017; Ruiz et al., 2017).

Ruiz et al. (2017) also look at product baskets to build a model of consumer choice, eventually generating latent representations of products that can then be used to identify economic relationships among products such as complementarity and substitutability. We find their work insightful since our setting is quite similar — we also inspect product baskets to generate dense latent representations of products and then use them to learn product relationships. However, there are three important distinctions. First, they only consider products that were purchased together and use the embeddings from the purchase space to determine complements and substitutes. We, on the other hand, use clickstream data that

allows us to identify consideration sets before purchases and define a heuristic of substitutability through products that are viewed together but not purchased together. Second, our ultimate objective is different from theirs. They propose a novel model in the utility-choice framework; we are in effect taking the utility-choice framework as given and using our version of that framework to design retail product bundles. Third, though less important, is that our model training approaches are different. Their approach is based on variational inference (VI) while ours is based on a shallow neural network.

Our work is also closely related to the market-basket analysis literature from marketing. Especially relevant are [Jacobs et al. \(2016\)](#), [Gabel et al. \(2019\)](#), and [Chen et al. \(2020\)](#) who present scalable approaches to market-basket analysis. [Gabel et al. \(2019\)](#) provide a thorough introduction to product embeddings and their use in market mapping. [Chen et al. \(2020\)](#) study product competition using embeddings. We build upon their work and extend product embeddings to capture complementarity and substitutability by leveraging both purchase and consideration data. We also go further from a generalized introduction of the concept of product embeddings to a specific use-case of designing bundles and provide an empirical solution to the problem. [Jacobs et al. \(2016\)](#) propose LDA-X, an approach based on Latent Dirichlet Allocation to predict the customer’s next purchase. In spirit, our paper is also related to [Gabel and Timoshenko \(2020\)](#) who use the idea of product embeddings, combined with a custom neural network architecture, to predict consumer choice that can further help optimize coupon allocations. Our idea is along similar lines where we learn product embeddings to solve a downstream task.

Our paper adds to this literature in multiple ways. First, we combine both purchase and considerations set data to learn product embeddings that represent different dimensions of the consumer choice process, thereby extracting additional value from large-scale clickstream data. Second, we effectively combine these representations to learn consumer preferences for thousands of bundles using a field experiment. Third, we use off-policy learning on the results of the field experiment to design an optimized bundling policy that performs substantially better than the

benchmark policy (Zhou et al., 2018; Athey and Wager, 2021). To the best of our knowledge, these are all novel contributions to the literature on bundling.

1.3 Data

We use clickstream data from a large online retailer in the US in which we observe entire user sessions of views, clicks, and purchases. The retailer sells products across multiple categories such as grocery, household, health and beauty, pet, baby products, apparel, electronics, appliances, and office supplies. The data span all consumer activity on the retailer’s website from January 2018 to June 2018 during which we observe multiple users and multiple sessions of each user.²

For each consumer session, we observe all the products that the consumer viewed and/or purchased along with the number of units of each product bought and the price. For all products, we know the product category hierarchy. The product category hierarchy can be understood using a simple example. Consider *Chobani Nonfat Greek Yogurt, Strawberry*. Its hierarchy would be Grocery (*Department*) → Dairy & Eggs (*Aisle*) → Yogurt (*Category*), where *Department* represents the highest hierarchy, *Aisle* is a sub-level of *Department*, and *Category* is a sub-level of *Aisle* (and hence *Department*). Throughout the paper, we will refer to hierarchical categorical levels as *Department*, *Aisle*, and *Category*. In our data, we have products across 912 *Categories*. It is important to note that we do not use any product meta-data for training the model. The product category hierarchy is only used for qualitatively validating the model, a point we discuss later, and generating different bundles subject to constraints on category co-membership.

As is typical of e-commerce websites, the raw clickstream data include many views of very rarely purchased products. The retailer’s assortment consists of more than 500,000 products, of which most products have never been bought. We filter these rarely purchased products to retain the top 35,000 products by views,

²A session is defined as a visit to the retailer’s website by a user. A session continues until there is no activity by the user for 30 minutes on the website. If the user performs an action after 30 minutes of inactivity, it is considered to be a new session by the same user.

which include more than 90% of all purchases. After filtering, we cover about 947,000 sessions made by $\sim 534,000$ users, which generated $\sim 861,000$ purchase baskets (products purchased in the same transaction) and $\sim 589,000$ consideration sets consisting of products viewed together. In the raw data, the number of consideration sets is much larger than the number of purchase sessions since many browsing sessions don't have any purchase. For our purpose, we only consider sessions with purchases. Further, we include only those products in the consideration set whose detailed page the consumer visited. Hence, the number of consideration sets is smaller than the number of purchase sessions. Observation counts from our working sample are presented in Table A1 in Appendix A.

1.3.1 Purchase baskets and consideration sets

A typical user shopping session includes browsing a range of products, potentially across multiple categories, and then purchasing a subset of them. In this process, the user first forms a consideration set, i.e., a set of products from which the consumer intends to finally choose from. In effect, from our model's perspective, the user creates two product baskets during a shopping session — products viewed and products purchased, which form our units of analysis in this study. We consider the products viewed but not purchased as the consumer's consideration set and the products purchased as the purchase basket. For consideration sets, we include only those products whose description page the consumer visited. It is important for us to distinguish between these two sets of products since this separation allows us to learn different relationships between products, i.e., they could be potential complements or potential (imperfect) substitutes, as we explain later.

Figure 1-1(a) shows an illustrative purchase basket. In this case, the user bought coffee, milk, cookies (breakfast foods), along with chips and salsa (snacks), toothpaste, and dish pods (household products). Our model and associated heuristics have been designed to infer that coffee, milk, and cookies are potential complements. Not only that, we want to go one step further and infer that coffee and

milk are stronger complements than coffee and cookies. The signal for these relationships comes from thousands of purchase baskets where we are likely to find coffee and milk being purchased together more frequently than coffee and cookies. Similarly, we want to infer that chips and salsa are complements. The consumer in this case also purchased toothpaste and dish pods. Ex-ante we do not expect any complementarity between these items and the rest of the basket and this may just be idiosyncratic noise particular to this shopping session.³

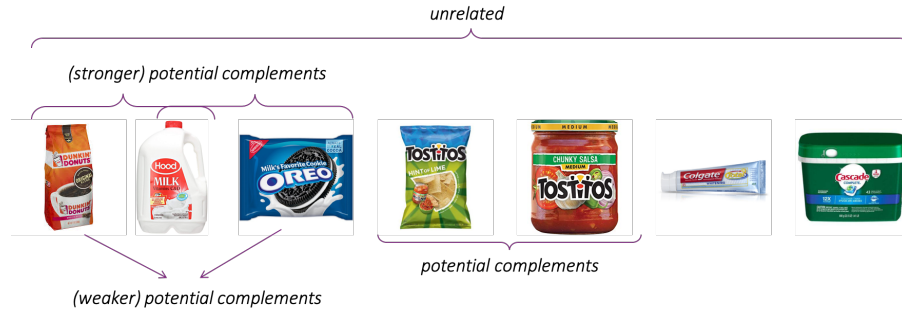
We also observe the corresponding consideration set for the same consumer, shown in Figure 1-1(b). The consideration set includes products that were viewed but *not* purchased together. We see that the consumer viewed different brands and flavors of coffee before purchasing one. Our model would infer them as potential substitutes. Further, the model would also pick out the different types of chips that the consumer searched. For inferring potential substitutes, we rely on the assumption that users search for multiple products before purchasing one, a pattern we do observe in the data.

Table 1.1 shows the summary statistics at a basket level. On average, a consumer searches 7 products for each one bought. The mean of products bought (or viewed) is higher than the median, indicating a long right tail of baskets with many products. Within each basket, the median number of departments is 1, alluding to the concept of a focused shopping trip, i.e., a particular shopping session for groceries, a different one for household supplies, a third one for apparel, and so on. Further, we see that the average purchase basket consists of products from 3 different categories.

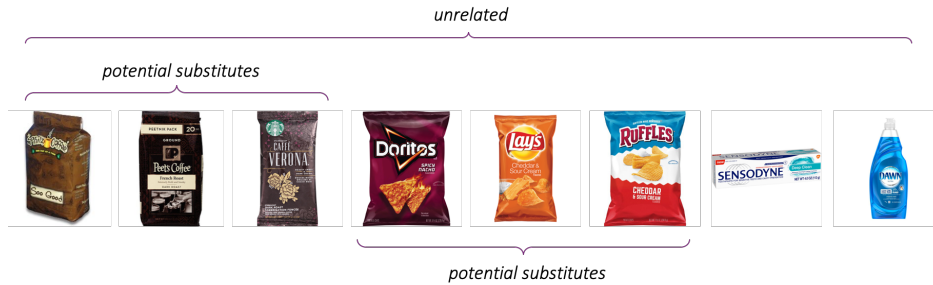
1.4 Product embeddings

Inferring product relationships from consumer choice has largely been the bastion of economics and marketing scholars studying micro-econometric discrete choice

³Note that the model does not make use of textual labels of the products. It ingests hashed product IDs and finds the relationships between these IDs without ever looking at the product name or category.



(a) Purchase basket



(b) Consideration set

Figure 1-1: Illustrative purchase baskets and consideration sets created during a user shopping session

models of consumer demand in which a consumer typically chooses one product out of an assortment of within-category options. Such models constrain the consumer's choice set to close (but potentially imperfect) substitutes, rendering cross-category comparison extremely difficult. Most of these models are also limited in the number of products and transactions they can handle, though the latter concern has been ameliorated with the rise in computational power. Furthermore, previous models only allow us to use features of products that are easily observable and quantifiable such as brand, price, and size. However, consumers make choices based on many factors such as the product description, packaging, and reviews, all of which are difficult to quantify and not intuitive to compare across categories of products.

Our approach loosens the grip of all these constraints by (1) considering multi-product choices in the same shopping session, (2) leveraging cross-category purchase baskets and consideration sets, (3) ensuring scalability in the number of

Table 1.1: Summary statistics per user session

		Purchased	Viewed
Products	Mean	3.6	21.3
	SD	4.2	34.1
	Median	2	9
	Max	202	1365
Category	Mean	3.0	4.0
	SD	2.9	5.3
	Median	2	2
	Max	69	122
Aisle	Mean	2.4	2.6
	SD	1.9	2.5
	Median	2	2
	Max	28	48
Department	Mean	1.6	1.7
	SD	0.8	1.0
	Median	1	1
	Max	10	14
Price	Mean	44.6	303.7
	SD	52.1	526.1
	Median	32.3	127.4
	Max	2,697	17,480

Note: A user session is defined a visit to the retailer’s website by a user. A session continues until there is no activity by the user for 30 minutes on the website. If the user performs an action after 30 minutes of inactivity, it is considered to be a new session by the same user.

products, transactions, and browsing sessions, and (4) imposing minimal structure on product characteristics. For example, in our setting of online retail with 35,000 products across hundreds of categories, inferring relationships between products through cross-price elasticity is not feasible. Co-purchases at the product level are too sparse to generate reliable estimates. Over 90% of the product pairs have never been purchased together. To analyze this sparse high-dimensional data efficiently, we adapt methods from the machine learning literature and customize them to suit our case. Our model condenses a large set of information about each product into dense continuous vector representations, which facilitate easy comparison of products across categories. Moreover, our method is also useful when considering categories of products with thin purchase histories, an area which is particularly difficult for structural choice models, allowing us to infer relationships even among products with few, or even no, purchases.

Our model belongs to the general class of vector space models where discrete tokens can be represented as continuous vectors in a latent space, such that tokens that are similar to each other are mapped to points that are closer in the space. Popular examples of vector space models include *tf-idf*, and the relatively newer,

word2vec (Mikolov et al., 2013a,b). Though both the examples above rely on the distributional hypothesis, models such as *tf-idf* are commonly referred to as count-based methods and are based on coarse statistics of co-occurrences of words in a text corpus, where models such as *word2vec* are based on prediction methods (Baroni et al., 2014). While a count-based model, such as an n-gram *tf-idf*, is simpler to understand, estimating the parameters of becomes increasingly complex as n increases ($\mathcal{O}(|\mathcal{V}|^n)$), where $|\mathcal{V}|$ is the size of the vocabulary. Count-based methods also cause problems when they face unforeseen n-grams and require smoothing to deal with them.

Neural probabilistic language models, like *word2vec*, deal with both these concerns by changing the objective function from modeling the likelihood of the corpus to predicting the probability of a word given its surrounding words. This not only condenses the representation of each word to a much lower dimension as compared to the size of the vocabulary but also removes the need for smoothing to generate probabilities estimates for new token sequences. It is important to note here that while neural models also rely heavily on co-occurrences, they go much beyond the simple notion of co-occurrence to help us learn about word pairs that may not have been frequently observed together in the past. In our context (as we will explain below), this implies that we can learn about product pairs that may have low co-purchases but are still be strongly related to each other. We describe our training and validation procedure below. In Appendix B, we provide an intuitive description and the formal equations for the model.

1.4.1 Training and validation

We train neural embeddings using a slightly modified version of *word2vec* on purchase baskets and consideration sets separately. We describe the training procedure on purchase baskets below (training on consideration sets similar). We start by only taking baskets that have more than one product. Since our model is designed to learn within-session relationships among products, we can only use pur-

chase baskets (or consideration sets) with 2 or more products. We then split the data into a development sample (Jan-2018 - May 2018) and a testing sample (June 2018). We further randomly split the development sample into a 90% training sample and a 10% validation sample. We use the validation sample to tune hyper-parameters.

We begin training by initializing the embeddings randomly with a $35,000 \times 100$ dimensional matrix. Here, 35,000 represents the size of our assortment and we are seeking a 100- d representation for each product. We split the training data into batches (as is standard in training neural networks) and subsequently split each basket into each combination of two-product pairs. We draw 20 times as many negative samples as positive samples using the marginal (uni-gram) distribution (Mikolov et al., 2013b). With the observed product pairs as positive labels and the generated negative samples as negative labels, we compute the cross-entropy loss and propagate it backward to update the embeddings. The model is trained with TensorFlow and we use the in-built Adam optimizer (Kingma and Ba, 2015) to update the weights.

The hyperparameters mentioned in the above paragraph are the optimal ones obtained by checking the performance of the model on 10% held-out validation data. To get these parameters, we first do a random search by running a smaller version of the model using product categories (level 3 hierarchy) to identify a narrower range of hyper-parameters over which the model performs well. This helps navigate the hyper-parameter space efficiently and iterate quickly over different configurations of the model (Bergstra and Bengio, 2012). We then run the complete model using the actual products on this narrower range to learn the optimal hyper-parameter values. This process is done separately for purchase baskets and consideration sets. While optimizing the model parameters, we learned that \mathcal{D} , the embedding dimension, has the biggest impact on model performance. A larger \mathcal{D} typically provides more accurate results since it can capture more complex relationships among products. However, this comes at the cost of a substantially larger training time. In our experiments, we found that $\mathcal{D} = 100$ provided the best

results on out-of-sample validation data. Reassuringly, other researchers have also found 100– d representation to work well for products on similar data sets (Ruiz et al., 2017).

We check the model’s fit on an out-of-time hold-out test set which consists of all shopping sessions in June-2018. We test the model on purchase baskets and consideration sets separately. We also compare the model’s performance to popular benchmarks used in recommendation systems and market-basket analysis. We train four models - 1) Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Jacobs et al., 2016), 2) Singular Value Decomposition (SVD)(Halko et al., 2010; Bell et al., 2008), 3) Non-negative Matrix Factorization (NNMF) (Cichocki and Phan, 2009), and 4) Item-based Collaborative Filtering (CF) (Linden et al., 2003). We also include a baseline model which uses only raw historical co-purchase/co-view rates. We compare each benchmark model’s score with our model using Hit Rate @ 10, which is a commonly used metric to evaluate recommendation systems. The results are shown in Table 1.2. We see that for both the cases - purchases and consideration sets, our product embeddings perform better.

Table 1.2: Model performance with Hit Rate @ 10 on held-out test data

	Purchases	Consideration sets
Product embeddings	0.039	0.023
LDA	0.035	0.018
SVD	0.033	0.021
Non-negative matrix factorization	0.034	0.016
Item-based collaborative filtering	0.029	0.020
Historical co-purchase/co-view rate	0.027	0.014

Note 1: The table shows performance for product embeddings and benchmark models on held-out test data as measured by hit-rate@10. The benchmark models were chosen based on literature review and include the common approaches used in market-basket analysis and recommender systems. Hit rate has been re-scaled by multiplying all values with 100.

1.4.2 Visualizing embeddings

It is instructive to visualize what the embeddings represent and understand why using them for designing bundles is a good idea. Below we visualize the purchase

and consideration set embeddings. Since our trained embeddings representations are $100-d$, we post-process them using t-SNE (van der Maaten and Hinton, 2008; Gabel et al., 2019) and project them to a $2-d$ map. To ensure proper convergence for t-SNE, we initialize the process by first running principal component analysis on the embeddings.

Purchase embeddings

Purchase embeddings are shown in Figure 1-2. For visual clarity, we highlight the department of the product and show products from the top five departments – groceries, health & beauty, household, baby, and pet supplies. A cursory glance reveals some confirmatory patterns. While we see a clear separation among products from different categories, there is also some overlap among departments. In this space proximity to the other products indicates a higher likelihood of the two products occurring in similar contexts, or in our case, similar types of baskets. Proximity among the products in this space suggests a higher degree of complementarity.⁴

As a more granular example, we zoom into the grocery department and look at snack foods, meats, dairy & eggs, and chocolates. Ex-ante we would expect snack foods to have a stronger positive relationship with candy & chocolates, and meats to have a stronger relationship with dairy & eggs. Figure 1-3 presents evidence for this hypothesis with snack foods being much closer to candy & chocolates than to either meat products or dairy & egg products. In fact, there is considerable overlap among snack foods with candy & chocolates, suggesting a high degree of complementarity between them.

In addition to testing relationships among products from different but pre-existing categories (typically created by the retailer), we can also generate new sub-categories of products and check how well they go with products from other categories. For instance, in Figure 1-4, we compare organic groceries with snack

⁴These embeddings are plotted in a latent space and hence the scale of this axis is immaterial; only proximity between the points is important.



Figure 1-2: Purchase embeddings for top-5 departments

foods. Although there is no pre-defined organic category of products, as a proof-of-concept, we do a simple string search of the word “organic” in the names of the products. We then visualize them along with snack foods to see what kind of organic products are related to snack foods. The upper highlighted portion of Figure 1-4 shows a high degree of complementarity among nuts, dried seeds such as watermelon seeds, trail mixes, jerky and dried meats, and seaweed snacks. On the other hand, the lower highlighted portion of the graph shown less of an overlap and mainly consists of cookies, chips & pretzels. We believe that having a flexible and scalable model such as this can provide crucial insights about market structure, brand competition, product positioning, user preferences, and personalized recommendations.

We dig deeper to the product level and inspect a few focal products. Consider, for instance, organic potatoes. In the purchase space, the products closest to organic potatoes include other organic fruits and vegetables such as organic celery, organic grape tomatoes, and organic green bell peppers. Similarly, products clos-

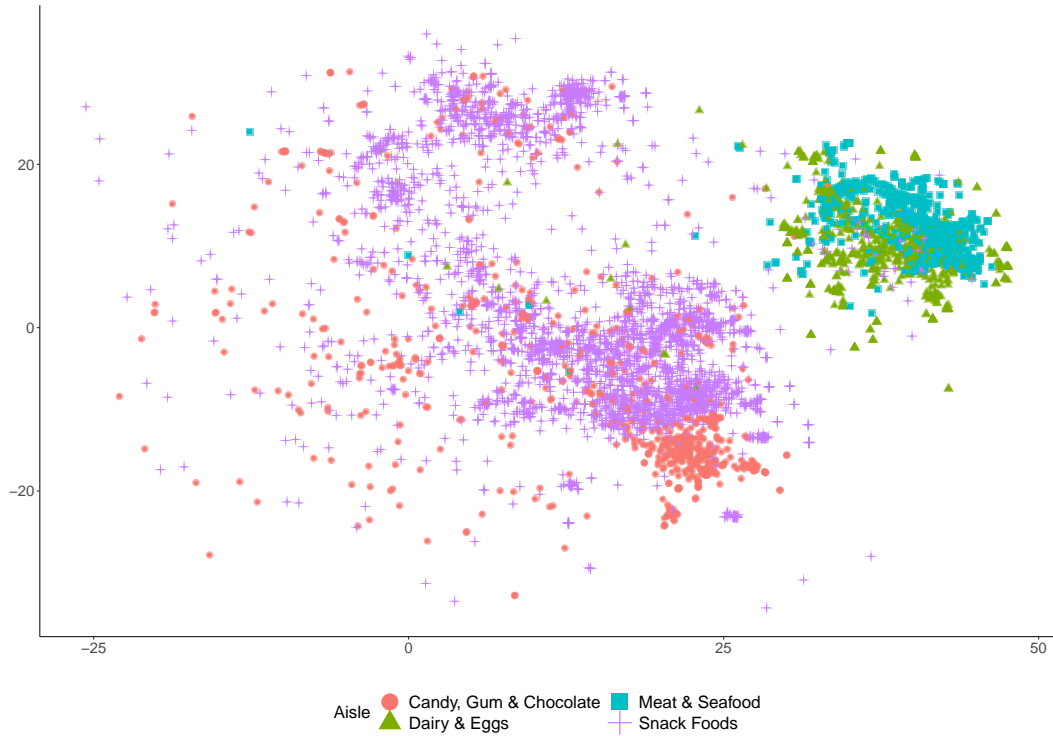


Figure 1-3: Purchase embeddings for within grocery aisles

est to dish-washing liquid include other household items, and in some cases can be narrowed to the space of cleaning products, such as paper towels, laundry detergents, and steel cleaners. As a third example, we look at a product from the health and beauty category — Neutrogena Oil-Free Acne Wash Redness Soothing Cream Facial Cleanser. Products that go along with this facial cleanser and include other hygiene and beauty products such as liners, rash cream, body wash, and deodorant. More details about the close complements of these focal products along with their complementarity score (described later) are presented in the Appendix A in Tables A2, A3, and A4.

We take this visual and tabular evidence as support for our claim that products that frequently co-appear in product baskets tend to have a higher degree of complementarity between them.

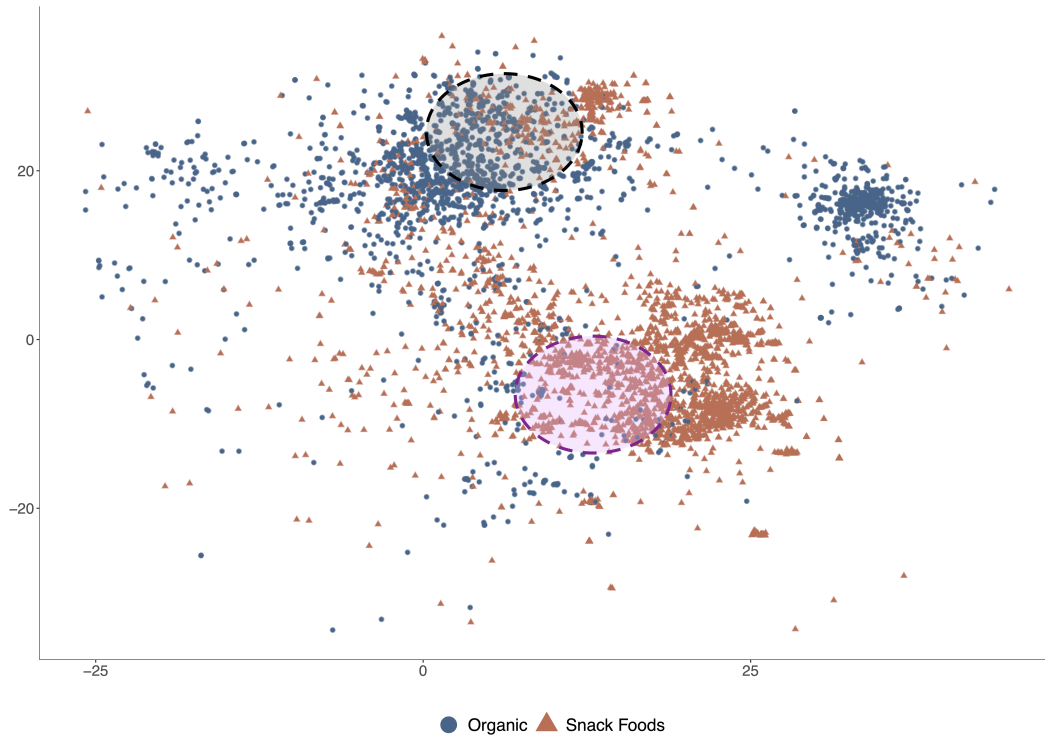


Figure 1-4: Purchase embeddings for organic groceries and snack foods

Embeddings vs. co-purchases

A natural contender for extracting signals of complementarity is using the co-purchase frequency directly. Hence, it is worth highlighting what we obtain from the embeddings that is otherwise not available through co-purchase counts. Figure 1-5 plots (log) historical co-purchase rate as observed in the data along with a heuristic for complementarity (described later) generated from the embeddings for a random sample of 5,000 product pairs. A quick glance reveals that although co-purchases are sparse (10%) as shown by points at the extreme left end of the graph, yet there is considerable variation in the scores of these products. This is because the model is able to smooth over the raw co-purchase counts over thousands of product pairs and learn relations even between products that have never been purchased together. For example, some of the product pairs with zero co-purchase history but high complementarity score include – 1) Colgate Total Whitening Toothbrush + Dove Sensitive Skin Body Wash Pump, 2) Breakstone’s Sour Cream +

Lemons, 3) Nutella & Go Pretzel + Clif Bar Energy Bar Variety Pack. Ex-post it is easy to see why these pairs should have a high complementarity score. However, identifying these pairs from a large assortment is challenging. Our approach is able to do this in a scalable way, which allows us to systematically explore a space that would otherwise be considered of limited value in designing bundles. Moreover, we highlight the department of the focal product to confirm that co-purchase is a limiting factor across the entire assortment and not just a few categories only.

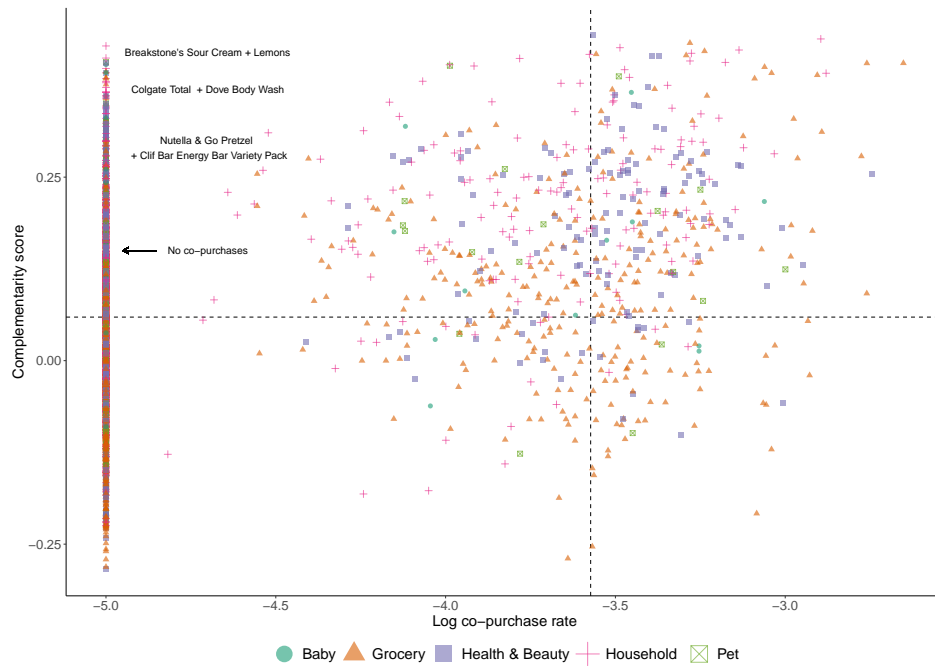


Figure 1-5: Historical co-purchase rate and complementarity score from purchase embeddings

Notes: Products with zero copurchase are shown at x = -5.0. Points are categorized with the department of the focal product.

1.4.3 Consideration set embeddings

Similar to purchase embeddings, we generate product embeddings using historical co-views of products, i.e., by observing how frequently do products co-occur in consideration sets formed during thousands of shopping trips. These embeddings also lie in a similar 100-dimensional space. We condense them to a 2-d space using t-SNE (van der Maaten and Hinton, 2008) and plot the top-5 departments in

Figure 1-6. The overall theme of the embeddings remains largely similar to that in purchase embeddings. However, there are two notable distinctions. First, the inter-department clusters are further separated away indicating that most views are confined to within-department products. This reinforces the evidence from Table 1.1, where we see that most search sessions are confined to one department. Second, there are well-defined sub-clusters within the department cluster, which self-classify into finer aisles and categories. For instance, the lowest olive colored cluster comprises only of “Breakfast Foods” (*Aisle*), primarily containing “Hot Cereals and Oats” (*Category*), with the occasional presence of “Granola & Muesli” (*Category*). On the other end of the plot, the topmost purple cluster consists of supplies for our furry friends. This cluster only contains meat-based meals (*Category*) for dogs (*Aisle*). These observations also lend merit to our hypothesis that co-views are good indicators of substitution across products, an insight we explore more below.



Figure 1-6: Consideration set embeddings for top-5 departments

At a more granular level, we look at aisles within the grocery department in

Figure 1-7. We see more refined sub-clusters as compared to purchase embeddings for the same grocery products. For example, the annotated cluster of purple points at the bottom of the graph is for popcorn and the annotated cluster of purple points in the center is for dried fruit and vegetable snacks.

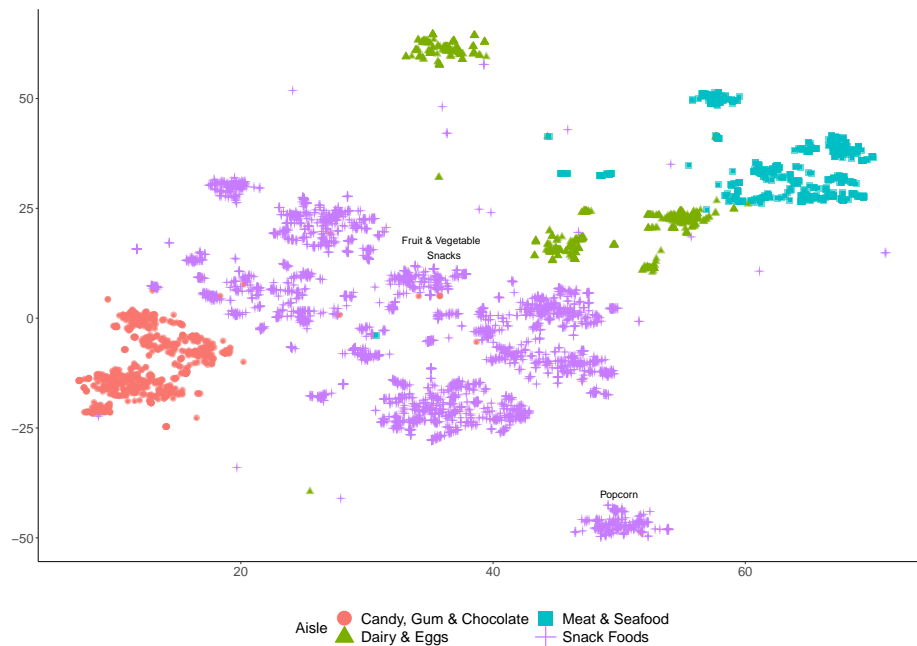


Figure 1-7: Consideration set embeddings for organic groceries and snack foods

Similar to the purchase space, we inspect the same three products and calculate their proximity to other products in the consideration space. For example, organic potatoes are now closer to other varieties of potatoes in the consideration space such as golden potatoes, red potatoes, and even sweet potatoes, indicating a higher degree of substitutability among them. This in contrast to the purchase space where organic potatoes were closer to other organic fruits and vegetables. Similarly, dish-washing liquid is now closer to other types and brands of dish-washing detergents such as liquids and soaps of different scents and sizes. Finally, the acne face wash shows considerable similarity with varieties of acne face washes. However, in this case, there is a strong brand effect with all potential substitutes being from the same brand - Neutrogena. It could be that users have strong preferences for brands when it comes to health and beauty products or that there is

a single dominant brand in the product line. More examples of products closer to each other in the consideration space are provided in the Appendix A in Tables A5, A6, and A7.

1.4.4 Product relationships

A critical ingredient in the recipe of our bundle generation process is the relationship between any two products in the retailer’s entire assortment. We want the relationship to be described by a metric that is continuous and category agnostic so that we can compare the strengths of the relationships that a particular product has with other products in the assortment as well as compare strengths of the relationships across product pairs. In other words, we’d like to be able to make both within-product as well as between-product comparisons. For example, we want to be able to say that coffee and cups are stronger complements than coffee and ketchup as well as that coffee and cups are stronger complements than tea and salt. This example seems obvious, however, it becomes increasingly hard to infer these relationships when there are thousands of products in the assortments and co-purchases among pairs of products are sparse as shown in Figure 1-5. With 35,000 products in the retailer’s assortment, there are close to 60 million product combinations with over 90% of the co-purchases being zero. Moreover, we want these relationships to be inferred from the data we observe and not be pre-imposed by the retailer. Analogously, we want to learn that coffee and tea are stronger substitutes than coffee and fruit juice.

With this objective in mind and based on the evidence described above, we use cosine similarity, a simple and intuitive metric, as a heuristic for product relationships. This heuristic is similar to the one used by Ruiz et al. (2017) and works well when trying to summarize distances in high-dimensional spaces.

Hence, we generate a heuristic for the degree of complementarity between

products i and j in the purchase space as:

$$c_{ij} = \frac{u_i^b \cdot u_j^b}{\|u_i^b\| \|u_j^b\|}, \quad (1.1)$$

where u_i^b and u_j^b are the embeddings of products i and j in the purchase space respectively, and $\|\cdot\|$ is the norm of the embedding vector.

Similarly, we generate a heuristic for the degree of substitutability between two products i and j in the consideration space,

$$s_{ij} = \frac{u_i^s \cdot u_j^s}{\|u_i^s\| \|u_j^s\|}, \quad (1.2)$$

where u_i^s and u_j^s are the embeddings of products i and j in the consideration space respectively, and $\|\cdot\|$ is the norm of the embedding vector.

To give an overview of what these product relationships look like, we present examples with a few focal products. Table 1.3 shows the top-5 complements and substitutes for products from three categories. In the first section, we consider hummus and we see that strong complements with it are baby carrots, greek yogurt, and mandarins. On the other hand, substitutes are other varieties of hummus. Similarly, for eyeliner, we find that complementary products include other skin-care and beauty products, whereas its substitutes are other varieties of eyeliner. Lastly, for household cleaning products such as dish soap, we find other types of cleaning products as strong complements and other varieties of dish soap as strong substitutes. These product relationship scores form the basis of our bundle design process and our field experiment, which we describe next.

A natural question ask is why we use two separate embeddings instead of combining them into a single set as done in Ruiz et al. (2017) and Gabel et al. (2019)? A qualitative answer here is interpretability, both for researchers and managers. The two embeddings capture different behaviors. By keeping the concept of complementarity and substitutability separately operable, we provide the option to explore the respective space in a guided way, especially in the case of designing

bundles. A more data-backed answer is that it works better. As we show later, using the two scores together delivers a better-optimized bundling policy than using either score individually (see Figure 1-8).

Table 1.3: Product relationships using the complementarity and substitutability heuristics

Product	Predicted complements in the purchase space	Predicted substitutes in the consideration space
Sabra Classic Hummus Cups	Cal-Organic Baby Carrot Snack Pack Chobani Fruit On The Bottom Low-Fat Greek Yogurt Sabra Hummus Singles Breakstone's 2% Milkfat Lowfat Cottage Cheese Halos Mandarins	Sabra Supremely Spicy Hummus Sabra Roasted Red Pepper Hummus Cups Sabra Greek Olive Hummus Sabra Classic Hummus Sabra Hummus Singles
L'Oreal Paris Infallible Eyeliner	Chloe Eau De Parfum Spray Olay Ultra Moisture Body Wash, Shea Butter John Frieda Frizz Ease Daily Nourishment Leave-In Smashbox NEW Photo Finish Foundation Primer Pore Olay Quench Ultra Moisture Lotion W/ Shea Butter,	L'Oreal Paris Brow Stylist Definer, Brunette L'Oreal Paris Brow Stylist Definer, Dark Brunette L'Oreal Paris Infallible Super Slim Eyeliner, Black Milani Eye Tech Extreme Liquid Liner, Blackest Black Revlon Colorstay Eyeliner, 203 Brown
Ecover Dish Soap, Pink Geranium	Forever New Fabric Care Wash Ecover Fabric Softener, Sunny Day Ecover Dish Soap, Lime Zest Full Circle Laid Back 2.0 Dish Brush Refill Giovanni Organic Sanitizing Towelettes Mixed	Ecover Dish Soap, Lime Zest Earth Friendly Products Ecos Dishmate Dish Soap, Lavender Ecover Zero Dish Soap Earth Friendly Products Ecos Dishmate Dish Soap Pear Earth Friendly Products Ecos Dishmate Dish Soap Almond

Note: For each focal product, the table shows the top-5 complements as determined by the embeddings in the purchase space and the top-5 substitutes as determined by the embeddings in the consideration space.

1.5 Bundle generation and experiment design

We follow a two-stage strategy to design bundles. In the first stage, we create a candidate set of bundles using the metrics of complementarity and substitutability described above and run a field experiment to gauge consumer responses to bundles with varying characteristics, especially scores derived from the product embeddings. The motivation here is to develop a principled exploratory strategy that is based on a more refined action space derived using historical purchases and consideration sets. In the second stage, we use the results of the field experiment to learn an optimized bundle design policy. We then scale the optimized policy to the entire assortment of products, identifying bundles with high success likelihood in terms of expected purchase rates and revenue.

We describe our strategy to create a candidate set of bundles for the field experiment below. In what follows, we consider bundles of two products — a “focal” product and an “add-on” product. The focal product is the main product on whose page the bundle offer is shown and the add-on is the product on which the

discount is applied. An illustrative example of how this is implemented on the retailer’s website is shown in Figure A1. For instance, consider ???. Here, a consumer visited the detailed page for ‘Doritos Tortilla Chips’ and was shown an option to purchase ‘Doritos Salsa’ along with the chips for a 10% discount on the price of the salsa. If the consumer would like to purchase the bundle, they can add it directly to the cart. This setup is consistent across all the promotional bundles we have in the experiment.

1.5.1 Candidate bundles

Our ultimate objective in this study is to identify the best promotional bundles from a large assortment. To this end, we need to learn consumer preferences over a wide set of bundles. A possible approach to learn these preferences is to create a large number of bundles using randomly drawn product pairs. However, this would be a poor approach – both in terms of power as well as in terms of creating a bad shopping experience for the consumers. At the other end, we can simply leverage the historical co-purchase rates among products and for each focal product pick the product with the largest co-purchase rate. However, as shown in Figure 1-5, co-purchases are sparse, and constraining ourselves to historical co-purchases would not allow us to explore the large assortment space. This would eventually not add much value to the consumer experience either.

A more principled way to navigate this space is to find candidate bundles with a “high” likelihood of purchase that are diverse enough to allow guided exploration. Our embeddings-based approach allows us to implement this strategy efficiently. We leverage the relationship scores to generate a varied set of bundles. Specifically, for each focal product, we create multiple bundles across different categories, casting a wide exploratory net for learning consumer preferences, while exploiting the strength of relationships between products to guide the learning.

It helps to fix notation and work with an example. In what follows, we seek to create a bundle b_{ij} for a focal i by choosing a product j from the assortment \mathcal{V}

of size $|\mathcal{V}|$, which in our case is 35,000. The bundle b_{ij} bundle will be shown on i 's page. c_{ij} and s_{ij} are the respective complementarity and substitutability scores between i and j . To make the exposition easier, we fix i for the examples below to be 'Domino Sugar' and call it \bar{i} . Our four candidate bundling strategies are:

1. **Co-purchase bundles (CP):** The first category of bundles is based on high observed co-purchase frequency. For each focal product, we select the product that it has been most frequently co-purchased in our training sample. These bundles are the natural contenders for a simple data-driven bundling strategy - products that have been purchased frequently together in the past will have a higher likelihood of being purchased together in the future as well, *ceteris paribus*. They also serve as a useful starting point of our bundle design strategy since we can map these bundles back to the underlying complementarity and substitutability scores, allowing us to learn more generalized patterns. However, these bundles are limited in scope since this strategy (a) does not generate bundles of products that have never been co-purchased before, (b) uses co-purchase information even when it is very noisy, e.g., bundling products if they have been co-purchased, say, 2 times in the past, (c) does not explore cross-category options since most of the bundles come from the same categories or aisles. For \bar{i} = 'Domino Sugar', the co-purchase bundle includes 'Sugar in the Raw Natural Cane Turbinado Sugar Packets, 100 Ct'.

$$b_{ij}^{CP} =_{j \in \mathcal{V}, j \neq \bar{i}} \{ \text{Co-purchase rate}_{\bar{i}j} \}, \bar{i} \in \mathcal{V} \quad (1.3)$$

2. **Cross-category complements (CC):** For a focal product i , we consider the strongest complement for i across a different category but within the same department based on c_{ij} as shown in Equation 1.4. The idea behind this strategy is to identify products that are most likely to be complements in usage and hence having the focal product under consideration would indicate a high likelihood of purchasing the add-on product as well. However, to add an element of exploration, we pair products across different cat-

egories. In case of a tie with the above co-purchase bundles, we use the second strongest complement. For ‘Domino Sugar’, the strongest cross-category complement is ‘International Delight Coffeehouse Inspirations Single Serve - Caramel Macchiato’ and hence we create a bundle with these two products.

$$b_{ij}^{CC} =_{j \in \mathcal{V}, j \neq \bar{i}} \{c_{ij} \mid \text{Category}(i) \neq \text{Category}(j)\}, \bar{i} \in V \quad (1.4)$$

3. **Cross-department complements (DC):** These bundles are similar in spirit to the cross-category complementary bundles mentioned above except that they specifically search over departments that are different from that of the focal product. Since most purchases within a trip come from the same department, as shown in Table 1.1, we tend to find stronger complements within the same department. Hence, the motivation in this arm is to explore cross-department bundles (e.g., household supplies and beauty products) of products that would otherwise not be considered. Equation 1.5 shows the formal criterion. For ‘Domino Sugar’, the strongest cross-department complement is ‘Solo Plastic Spoons, White, 500 Ct’.

$$b_{ij}^{DC} =_{j \in \mathcal{V}, j \neq \bar{i}} \{c_{ij} \mid \text{Dept.}(i) \neq \text{Dept.}(j)\}, \bar{i} \in \mathcal{V} \quad (1.5)$$

4. **Variety (VR):** Extant research has suggested the benefit of bundling (imperfect) substitutes to capture a larger portion of the consumer surplus and improve profitability (Lewbel, 1985; Venkatesh and Kamakura, 2003)⁵. We explore this idea empirically by creating bundles of products that are close to each other in the consideration space. The rationale here is that products that appear to be potential substitutes may in fact also be complements over time or complements within a household. If this is true, then bundling products that are imperfect substitutes could help exploit variety-seeking behav-

⁵There is the caveat that consumers may actually be forward looking and just buy the products ahead of time while they are being sold at a discount thereby having no impact on the overall sales of the retailer. We do not investigate inter-temporal substitution patterns here while noting that it is an interesting avenue to study further.

ior among consumers and generate incremental sales for the retailer. We use S_{ij} to find the strongest variety bundle as shown in Equation 1.6. For $\bar{i} = \text{'Domino Sugar'}$, the strongest variety bundles includes $\text{'Splenda No Calorie Sweetener 400 Count'}$.

$$b_{ij}^{VR} =_{j \in \mathcal{V}, j \neq \bar{i}} \{S_{ij}\}, \bar{i} \in \mathcal{V} \quad (1.6)$$

It is important to note that these strategies are not the "arms" of a randomized experiment which we plan to horse-race with each other. Rather they are a data-driven way to picking good yet varied candidate bundles to learn consumer preferences.

1.5.2 Experiment design

We use these bundling strategies to create four distinct bundles for 4,500 top-selling products after removing products with any retailer-specified restrictions for the field experiment.⁶ We select the top-selling products as they get maximum traffic on the retailer's website. We create $4,500 \times 4 = 18,000$ bundles across the different strategies mentioned above and institute them in the retailer's system. These different strategies for generating bundles should not be understood as different treatment arms that we aim to compare; rather they are different strategies for sampling *a priori* promising bundles, yielding data with which to optimize bundle selection.

Basic characteristics of the bundles created by the four strategies are shown in Table 1.4. We show the results for 11,296 bundles which were viewed at least once during the experiment, and hence are part of our subsequent analysis. All values in this table are calculated based on the pre-experiment data used for training. The number of bundles viewed is different across the four bundle types since there is

⁶These include products that cannot be part of sales or promotions due to special manufacturer-retailer contracts.

flux in the inventory and depending upon the location and time of the consumer, a bundle may or may not be available. We calculate the co-purchase rate for the product pairs. *Price-1*, *Rating-1*, *Purchase rate-1* correspond to the average price of the focal products, their average user-provided rating, and their historical individual purchase rates. Analogously, *Price-2*, *Rating-2*, and *Purchase rate-2* correspond to the same variables for the add-on product. The last four rows show the mean of binary variables which take the value 1 if both the product belong to the same department, aisle, category, and brand respectively.

Table 1.4: Mean pre-experiment product characteristics for candidate bundles in the experiment

	Cross category	Cross dept.	Variety	Co- purchase
	(CC)	(DC)	(VR)	(CP)
Bundles	3,092	2,418	3,287	2,499
Comp. score (c_{ij})	0.34	0.22	0.41	0.40
Sub. score (s_{ij})	0.39	0.20	0.74	0.58
Co-purchase rate	0.10	0.02	0.23	0.29
Price - 1	10.70	10.96	10.22	9.53
Price - 2	8.32	10.16	10.06	8.63
Purchase rate - 1	0.04	0.04	0.04	0.04
Purchase rate - 2	0.05	0.06	0.04	0.04
Product rating - 1	4.70	4.72	4.69	4.69
Product rating - 2	4.70	4.80	4.69	4.72
Same dept	0.99	0.00	1.00	0.88
Same aisle	0.55	0.00	0.99	0.71
Same category	0.02	0.00	0.98	0.53
Same brand	0.17	0.10	0.46	0.44

Note 1: Co-purchase rate has been multiplied by 100. Price-1, Purchase rate-1, and Product rating-1 show the average price, average historical purchase rate, and the average product rating of the focal product in each bundle type. Price-2, Purchase rate-2, and Product rating-2 are corresponding variables for the add-on product. Same department, Same aisle, Same category, and Same brand are binary variables that indicate if the two products are from the same department, aisle, same category, and brand respectively.

We ran the field experiment for ~ 3 months from July 2018 to September 2018. The experiment was run at a user-product level, such that if a user m searched for product i which has a bundle associated with it, then the user would be randomized into one of the four strategies, i.e., the user would be shown one of the four bundles associated with the focal product. Let's say that user m was randomized into the cross-category complement bundle strategy for product i , then every time m searched for i , she would be offered the opportunity to buy the cross-category

complement bundle (b_{ij}^{CC}) with the 10% discount. The user need not buy the bundle and can still purchase either the focal product directly or the add-on product without any discount. After searching for i , if m searched for product k , she would again be randomized into any of the four strategies. However, if she searched for i again, she would see the same cross-category complement bundle (b_{ij}^{CC}). To give a perspective of how the bundle offer is presented to the user, we show two illustrative examples in Figure A1.

1.6 Optimized bundling policy

An overview of the results from the field experiment is shown in Table 1.5. $\sim 180,000$ users viewed 11,296 bundles $\sim 227,000$ times during the experiment. Viewing a bundle is the same as visiting the focal product’s detailed page (as shown in Figure A1). We also capture clicks on the bundle component on the web page, the number of bundles added-to-cart (ATC), bundle purchases, and revenue. The third column shows the same metrics as a proportion of the total number of bundle views.⁷

Table 1.5: Key metrics from the field experiment

	Count	Count/View
Unique bundles viewed	11, 296	-
Users	180, 428	-
Total bundle views	227, 311	-
Bundles added-to-cart	1, 963	0.008
Bundle purchases	739	0.003
Bundle revenue	17, 136	0.07

Note 1: The third column is the second column divided by the total number of views and can be interpreted as the conversion rate conditional on exposure.

We further investigate the results for each bundling strategy. Table 1.6 shows variation in the views, cart-additions, purchases, and revenue from bundles across the three strategies which use product embeddings to pick candidate bundles, and

⁷These metrics are only from the transactions involving bundles. The users may have visited and bought other items from the retailer, including the focal or add-on products independently. Those transactions are not captured here.

Table 1.6: Experiment results split by candidate bundling strategy

	Cross category	Cross dept.	Variety	Co-purchase
Bundles viewed	3,092	2,418	3,287	2,499
Views	58,525	50,616	58,195	59,975
Users	55,104	47,730	54,840	56,526
Bundle ATC	468	227	707	841
Bundle purchases	173	83	343	432
Bundle revenue (\$)	2,795	1,062	6,080	7,199
ATC / View	0.008	0.004	0.012	0.014
Purchase / View	0.003	0.002	0.006	0.007
Revenue / View (\$)	0.048	0.021	0.104	0.120

Table 1.7: Results from optimized bundling policy on hold-out test data

	Optimized policy	Baseline policy	Optimized - Baseline
ATC / 100 Views	1.709	1.412	0.297 [0.127, 0.467]
Purchases / 100 Views	1.112	0.845	0.267 [0.059, 0.512]
Revenue / 100 Views (\$)	19.36	14.37	5.090 [0.217, 9.57]

Note 1: We learn the optimized policy using focal products for which all four bundles had at least one view during the experiment. This provides a more intuitive understanding of how the optimized policy is learned. We provide results for the policy trained on all bundles in the Appendix in Figure A2

Note 2: The numbers in Panel B are scaled by 100 and can be interpreted as ATC / 100 views or Revenue / 100 views. Confidence intervals are from 1000 bootstrap replications.

the fourth strategy which is based on historical co-purchase rates. The second half of Table 1.6 calculates the values of these metrics per view. We find good variation in bundle purchases across the four strategies, which provides us valuable training data to learn an optimized policy that picks the best bundle for each focal product. Table 1.7 presents the results from the optimized bundling design policy. We describe the details of how we learned the optimized policy next.

1.6.1 Learning the optimized policy

We assign multiple bundles to a focal product the field experiment. In essence, this means that each focal product has multiple treatments. We then ask which treatment is best for a *given* focal product. Note that this is different from just picking the strategy that performs the best on average across all products in the experiment. We want to pick the best bundling strategy for each focal product, given its

complementarity and substitutability scores, and other product characteristics.

Before learning the optimized policy, we run a comparative predictive modeling exercise to identify the best classifier suited to our case. We train five classifiers to predict bundle add-to-cart (ATC) as the binary label. We use an up-the-funnel metric such as ATC since it provides more power as compared to fitting the model directly on bundle purchases which is a highly imbalanced target variable. We choose XGBoost (Chen and Guestrin, 2016) after comparing to other popular benchmarks used for predictive modeling - logistic regression, hierarchical logistic regression, LASSO, and Random Forest. The results from this comparative exercise are available in Table A9.

A point to note is that not all candidate bundles were viewed by consumers during the experiment. Hence, we don't know what the outcome for the bundles would have been had they received views during the experiment. To circumvent this constraint, we learn the optimized policy using focal products for which all four bundles received at least one view during the experiment. This does not impact our results but makes the process easier to understand. We provide results for the optimized policy learned using all the bundles in the Appendix in Figure A2. They are qualitatively similar to results described below.

To learn the optimized policy, we first randomly split the experiment data into a training and testing sample. To ensure there is no leakage for the same product, we create the train-test split by randomizing focal products to either of the two samples. Hence, all bundles and observations for a particular focal product are either in the train sample or the test sample. We train an XGBoost model using cross-fitting on the training data with bundle add-to-cart as the binary label. We use the product relationship scores and other pre-treatment covariates as independent variables. This is our outcome model. We generate cross-fitted predictions and aggregate the predictions to bundle-level. Finally, for each focal product, we then select the bundle with the highest average predicted value.

For the selected bundles, we tabulate the average outcomes on the test data for the corresponding observed variables — bundle ATC, bundle purchases, and

bundle revenue. To compute the averages, we use the self-normalized IPW estimator, which is essentially a normalized version of the Horvitz-Thompson estimator. We run 1000 bootstrap replications of this process to estimate uncertainty and get the confidence intervals. We do a similar process for the baseline policy in which we select bundles based on the historical co-purchase rate only. We compare the performance of the policies on the test data. Table 1.7 shows the gains from the optimized policy over the benchmark policy. On average, we find that the optimized policy improves purchases by $\sim 31\%$ and revenue by $\sim 35\%$ over the benchmark policy (\$5 per 100 views). In the appendix, we replicate this process using all focal products. The results are shown in Figure A2 and are qualitatively similar.

1.6.2 Comparing policies based on different relationship scores

We highlight the benefits of using both the complementarity score and the substitutability score to optimize the bundling policy. In Figure 1-8 we show the results from an exercise where we optimize the bundling policy using both the scores (as described above), or select bundles using only the complementarity score, or the substitutability score. The policy using only complementarity scores is based on the process as described above, but can only select bundles from the “complementary” space. It uses the predictions from the XGBoost model to select the best bundle from one of the three complementary categories – cross-category complements, cross-department complements, co-purchase bundles. The policy using the substitutability score selects the variety bundle for each focal product. In theory, if there are more types of variety bundles, then this policy can select the best bundle using the predictions from the model as the optimized policy does.

The figure shows the results on held-out test data. The blue bars show the point estimates for revenue on test data and the red error bars are 95% confidence intervals for the difference between the optimized policy and the baseline policy (with the point estimate of the baseline policy added in to facilitate comparison). The baseline policy uses the historical co-purchase rates to select the bundles. We

find that the policy that uses both the scores performs considerably better than a policy that uses either scores individually or the baseline policy.

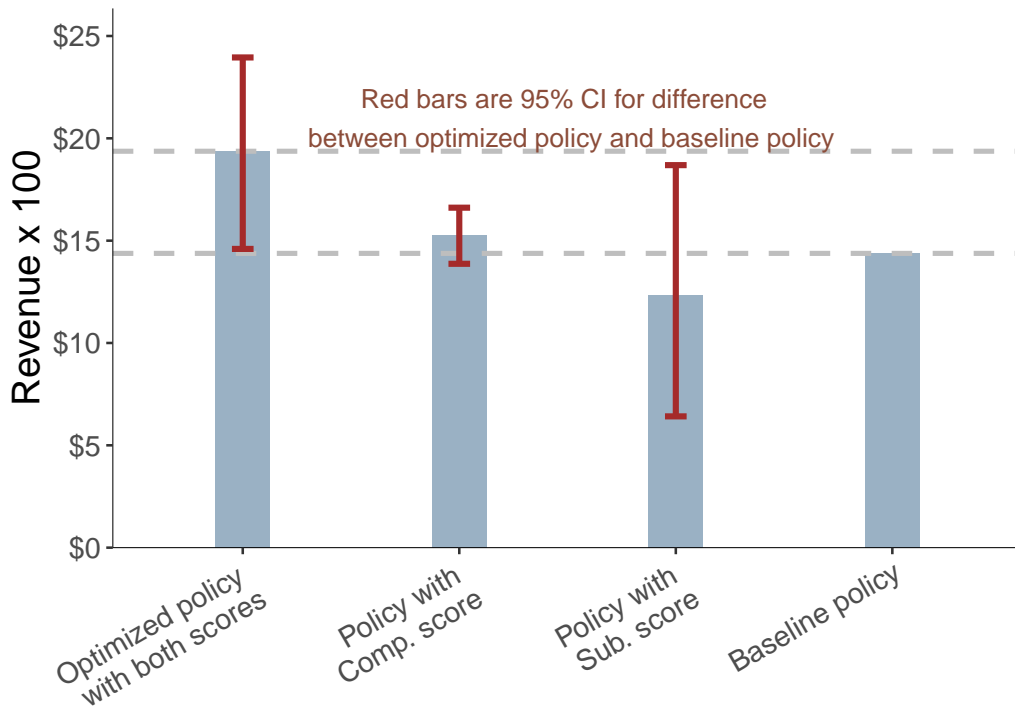


Figure 1-8: Revenue from the optimized bundling policy vs. policies that only use either the complementarity score (c_{ij}) or the substitutability score (s_{ij})

Note: The optimized policy is trained on 897 focal products for which all bundles received at least one view during the experiment. It uses both the scores to find the best bundle for a given focal product. The second bar only uses the complementarity score to find the best bundle and the third bar only uses the substitutability score, i.e., the variety bundle. The last bar is the benchmark policy based on co-purchase rate.

1.6.3 Embeddings vs. co-purchase revisited

A strength of our approach is that we can learn relationships among products that may have very few, or even zero co-purchases historically, but may still be strongly related to each other, as shown in Figure 1-5. Here, we revisit the issue and provide more concrete evidence for this using two approaches. In the first approach, we generate 1,000,000 potential bundles for 16,245 focal products. We then score these bundles using the optimized policy and the baseline policy described above. For each focal product, we select the best bundle as suggested by the respective

policies. For the optimized policy, the best bundle is selected using the score generated by the XGBoost model and for the baseline policy, it is based on the historical co-purchase rate. We record the predicted revenue from both the policies and the percentage of zero co-purchase bundles selected by the optimized policy. To estimate the uncertainty, we bootstrap this process 100 times.

The results are shown in Figure 1-9. To highlight the difference between the two policies we split the focal products into deciles based on their historical purchase rate and then graph the results within each decile. Two things are evident – 1) the optimized policy vastly outperforms the baseline policy in each group, and 2) the difference becomes larger as we select the right tail, i.e., the less popular products. We also record the proportion of bundles selected by the optimized policy in which the products had never been co-purchased before. We see that as we move towards the less-popular products, the optimized policy picks more and more bundles with no co-purchases. The baseline policy, which uses historical co-purchases only, is severely handicapped in generating additional value from the less popular products.

It is important to note that the gains reflected in Figure 1-9 heavily rely on the model. This is in contrast to Figure 1-8 where the model was just used to select the best bundles but the evaluation of the policy was done non-parametrically on held-out test data. Here, we don't observe yet observe the outcomes for the new bundles and hence extrapolate their success using the predictions from the model.

To provide more evidence for the benefits of our approach, we test its value for in-sample, i.e., the bundles used in the field experiment from a different perspective. We assess the value-added by the product relationship scores, specifically the complementarity score (c_{ij}) over and above the historical co-purchase rate. We do this by first residualizing the outcomes, bundle purchases and bundle revenue, using historical co-purchase rate and other pre-treatment covariates (except c_{ij} and s_{ij}). Second, we also residualize c_{ij} using the historical co-purchase rate and other pre-treatment covariates. We then regress the residuals from the first regression on the residuals from the second regression, which gives us the impact of comple-

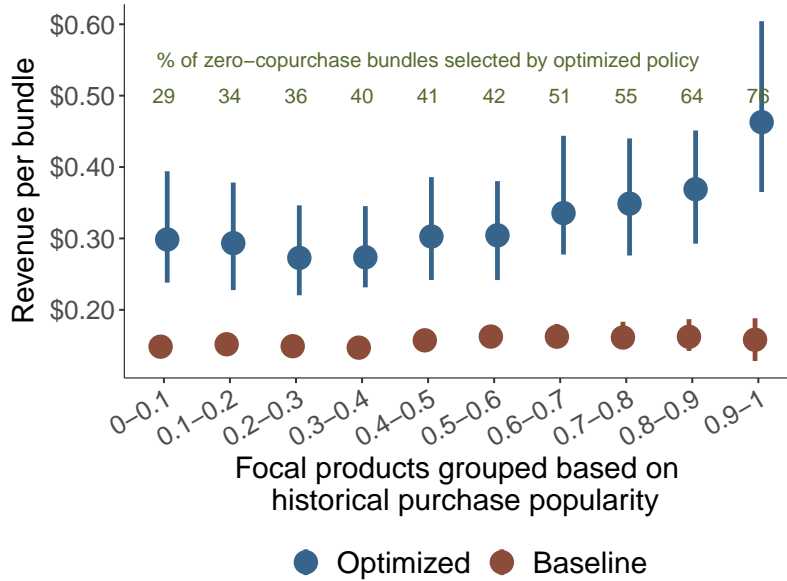


Figure 1-9: Revenue from optimized bundling policy and baseline policy on new bundles generated from the retailer's assortment

Note: We score 1M potential out-of-sample bundles for 16,000 focal products. The focal products are first ordered as per their historical purchase rate with the most popular products on the left. The products are then grouped into deciles and the policies are used to score all bundles for the products falling in each decile.

mentarity score on bundle success net of the effect of historical co-purchase. We repeat the same process of bundle revenue. In both cases, we find that the residualized complementarity score is highly predictive of residualized bundle success. The results are shown in the Appendix in Figure A3.

1.6.4 Limitations

In the literature, bundling involves two steps – 1) deciding which two products should be bundled together, and 2) setting their joint price. Both these problems are quite challenging to solve, especially at scale. In this study, we primarily focus on the first one and attempt to do it well. We choose prices based on institutional constraints and managerial suggestions. We use a reasonable discount of 10% which is commonly observed in online retail which allows us to focus on the problem of creating bundles with the best products. In this sense, our results can be considered a lower bound on the effect of scalable bundling. With more opti-

mally chosen prices, the retailer can expect to improve its profit and also serve its customers better. Modifying prices at our scale with thousands of products is not a feasible option. The trade-off is then to choose a small subset of products and determine their optimal bundle prices while sacrificing scale or developing a scalable approach with reasonable guardrails on price discounts. Both strategies have their merits and demerits. However, since much of the past literature has focused on the former, we chose to bring in a new perspective and focus on the latter.

1.7 Managerial insights

In this section, we build intuition about the results and focus on managerial insights. We use the results from the optimized policy as a guide to further uncover important insights. To motivate the analysis, we manually inspect the trees from the XGBoost model trained to learn the optimized policy. We present two sample trees in the Appendix in Figure A4. In many trees such as these, we find splits that interact the product relationship scores with product categories or prices of the two products with each other. We delve deeper into these findings below.

To make the managerial insights easily interpretable, we build a hierarchical logistic regression (Gelman and Hill, 2007) to predict bundle add-to-cart. We allow partial pooling across product aisles and the effects of the product relationship scores to vary by the aisle of the focal product. More details on the model along with the estimated coefficients are provided in Appendix C. We focus on three insights for managers in the bundle design process - 1) across-category robustness of the scores, 2) cross-category bundles, and 3) relative prices of bundle components. Before we proceed, we duly note that while these insights are correlational, we believe they provide valuable interpretable and implementable strategies for managers.

1.7.1 Consistency and variation across categories

We use the varying slopes in the hierarchical model to examine the robustness (homogeneity) of the predictive relationship between complementarity and substitutability scores and add-to-cart across different product aisles. Figure 1-10 shows the point estimate of the aisle-specific slopes (including the common slope parameter from Table C1) and the 95% confidence intervals. We plot the varying slopes for both the heuristics across different product aisles and find that the positive association of both the scores is fairly robust across all aisles, alluding to the ability of our approach to generalize across the retailer’s entire assortment.



Figure 1-10: Aisle-varying slopes for product relationship scores from hierarchical logistic regression

1.7.2 Cross-category bundles

A feature of our approach is the ability to form cross-category bundles at scale. This is important for a large retailer which sells products across hundreds of categories. We can use the hierarchical model to learn bundle success likelihood across

multiple product category combinations allowing us to generalize our findings outside of the bundles in the experiment. To this end, we first randomly create 20,000 out-of-sample bundles from the retailer's assortment across all product categories. We score these bundles by generating predictions using the model. We then aggregate the predictions to category-combination levels using the categories of both the products and inspect the patterns we see.

A condensed view of the result is shown in Figure 1-11, which plots the average predicted probability, expressed in percentage, for each category combination. Larger darker circles imply a higher average likelihood of bundle add-to-cart and the color bar below shows the percent likelihood of success. The product-category combinations are sorted using spectral clustering. Note that this matrix is asymmetric, the probability of success for bundles with focal product A + add-on product B is different from the probability of success of a bundle with focal product B + add-on product A. To make the visual example easier to read, we average these probabilities and make the matrix symmetric.

A few interesting patterns are visible and we highlight certain cells for discussion using (*). For example, the two clusters at the extreme ends of the graph — the top left, and the bottom right, show aisles of products that would be good contenders for cross-category bundles. Fresh produce, dairy and eggs, meat and seafood, snacks, pantry, and soups and side dishes make good bundles with each other. Similarly, sports nutrition products, breakfast foods, and candy make good bundles with each other. Among other combinations, cleaning products go well with laundry, skin care, and interestingly, candy. Candy and chocolates also make a good combination with pantry goods. We also see product combinations in the mid-left of the graph that show cases where cross-category bundling may not be effective.

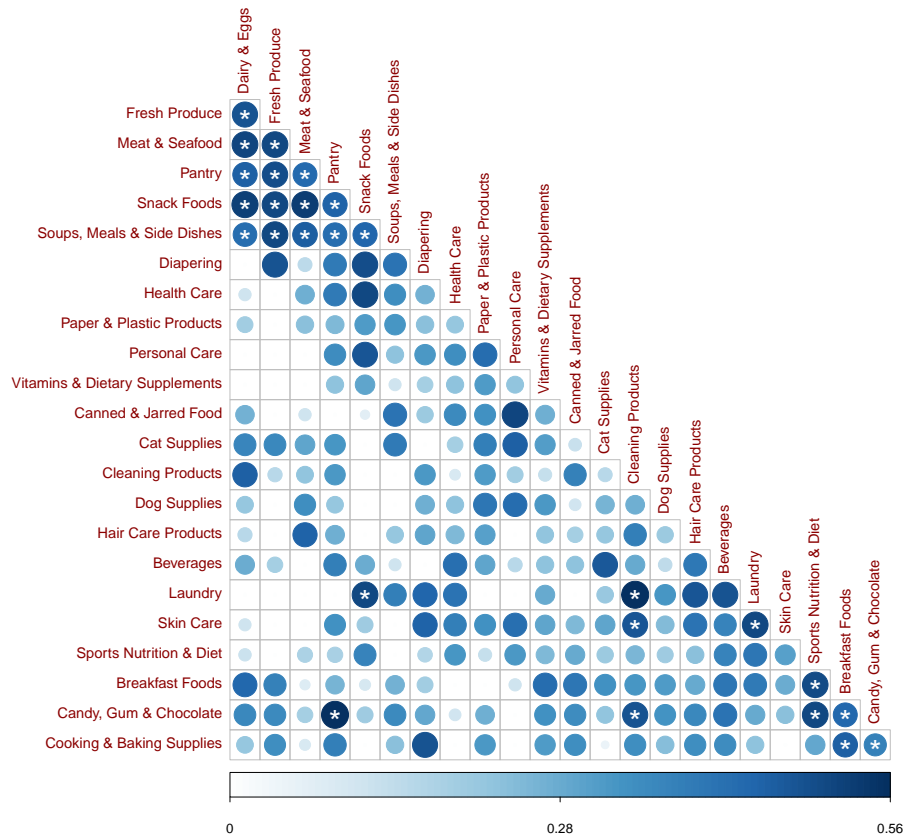


Figure 1-11: Predicted probabilities aggregated to product category combinations.

Note: The color bar shows % likelihood of success. (*) Cells highlighted for discussion.

1.7.3 Relative prices

Real-world retail bundles, as in our case, are typically asymmetric, i.e., there is a “focal” product that the consumer is seeking to buy and an add-on product is included with a discount to sell both products to the consumer. To incentivize the consumer to buy both products, either the second one is sold at a discount, as we do, or the bundle has a single joint price that is lower than the sum of prices of the two components. While we cannot directly identify the optimal price/discount for the bundle due to the reasons mentioned above, we highlight general-purpose guidelines on which products to choose to make a bundle, given their prices.

We first test whether there is an asymmetric impact of the prices of the two

products, i.e., whether consumers are more sensitive towards the price of the focal product vs. the add-on product or vice versa. We do a Wald's test for the equality of the two price coefficients and reject the null that the coefficients are equal. Consumers are more sensitive towards the price of the add-on product as compare to the focal product. The results are shown in Table C2.

Next, we include a binary variable in the model that takes the value 1 if the price of the focal products is greater than or equal to the price of add-on product. The coefficient for the indicator is shown in Model (2) in table C1. We find a strong positive effect of this asymmetric price relation on bundle success.

We provide visual evidence of this effect in Figure 1-12 in which we first create 6 buckets for prices for both the focal and add-on products and then aggregate posterior predictions from the hierarchical model in these groups. Most of the mass of the predictions is in the upper left corner, where most of the bundles also lie. We can clearly see the asymmetry in the effect of relative prices. Consumers prefer bundles where the add-on product is cheaper than the focal product. This effect can partly be explained by consumer's shopping intentions. Bundles are shown on the focal product's page which the consumer has self-selected to view with the intention of purchasing the focal product. The add-on is then an additional purchase and getting a discount on a cheaper item induces the consumer to purchase the bundle. A few examples of successful bundles where the focal product's price is strictly greater than the add-on product's price include – '1) Method 4X Laundry Detergent, Beach Sage + Method Fabric Softener, Beach Sage, 2) Blue Diamond Almonds, Bold Wasabi & Soy Sauce + Hapi Snacks Hot Wasabi Coated Green Peas, and 3) Dole California Whole Pitted Dates + Prince Of Peace 100% Natural Ginger Candy.

1.8 Discussion

We propose a novel machine learning-based bundle design methodology for a large assortment of retail products from multiple categories. Our methodology is

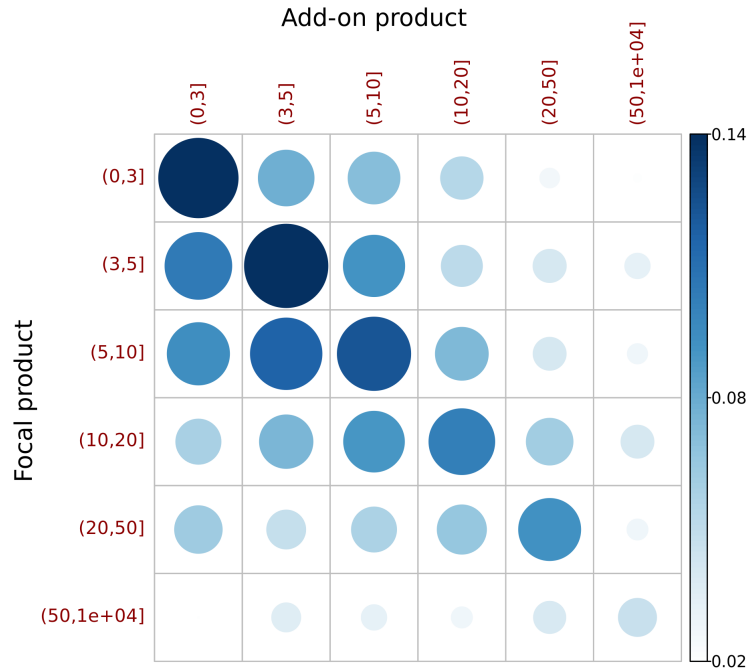


Figure 1-12: Predicted bundle add-to-cart probabilities across prices groups of focal and add-on products

based on the historical purchases and considerations sets generated by consumers while shopping online. We create two continuous dense representations of products (embeddings) in the purchase space and in the consideration space using purchase baskets and consideration sets respectively. We put minimal structure on these embeddings and create heuristics of complementarity and substitutability between products. In essence, we exploit the notion that products that are “close” in the purchase space are potential complements, and products that are “close” in the consideration space are potential substitutes.

We use these heuristics to create multiple bundles for each of 4,500 focal products and learn consumer preferences over these bundles using a field experiment run in collaboration with a large online retailer in the US. We especially create bundles across product categories and using imperfect substitutes to explore the potential bundle space in a principled way. We apply offline policy evaluation to the results of the field experiment to learn an optimized bundling policy. The optimized policy increase expected revenue by $\sim 34\%$ ($\sim \$5$ per 100 views) over the

benchmark policy.

To the best of our knowledge, ours is the first study to leverage historical purchase and search patterns to generate discount bundles at this scale. Our setting of cross-category online retail is also relatively unexplored in marketing and economics bundling studies. Moreover, previous studies have primarily been theoretical or lab-based and have typically pre-assumed relationships among products to derive their insights. On the other hand, combining a machine learning model with an online field experiment, we provide empirical evidence and generate generalizable insights from a large number of bundles across multiple product categories. For example, we find that beverages, snacks, and laundry products are good contenders for cross-category bundles with most categories. Meat and seafood go quite well with canned food and fresh produce. On the other hand, health care and baby supplies are not good candidates for cross-category bundles.

Our study has some constraints as well. We duly note that we trade-off “structure” for scale and this has its pros and cons. With our method, we are able to work with a much larger set of products and explore a combinatorially complex space efficiently. As a result, we don’t focus on the micro-foundations of the model or attempt to tie the model to theory. For instance, we do not look at cross-price elasticities to identify complements or substitutes but rather define them in a way that suits our purpose.

Additionally, although we include price by controlling for it while designing the optimized bundling policy, we do not explicitly include it in the experiment. We believe it would be insightful to randomize the discount in the experiment and investigate the impact on the results. For example, we hypothesize that the retailer would need to provide a smaller discount for complementary bundles and a relatively larger one for variety bundles. However, experimenting with prices at this scale is not trivial. Not only because of institutional/bureaucratic constraints but the sheer number of guardrails one would need to ensure the stability of the system could lead to a very challenging implementation.

Finally, we believe that our work is just the first step in a new direction for bun-

dle design. A valuable next step would be to further extend this method to design personalized bundles. Personalized bundling strategies, in addition to providing more value to customers, could also help the retailer segment the market better.

Bibliography

- Adams, W. and Yellen, J. L. (1976). Commodity bundling and the burden of monopoly. *The Quarterly Journal of Economics*, 90(3):475–498.
- Athey, S. and Wager, S. (2021). Policy learning with observational data. *Econometrica*, 89(1):133–161.
- Barkan, O. and Koenigstein, N. (2016). ITEM2VEC: neural item embedding for collaborative filtering. In *26th IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2016, Vietri sul Mare, Salerno, Italy, September 13-16, 2016*, pages 1–6.
- Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.
- Bell, R. M., Koren, Y., and Chris, V. (2008). The bellkor 2008 solution to the netflix prize.
- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305.
- Bhargava, H. K. (2012). Retailer-Driven Product Bundling in a Distribution Channel. *Marketing Science*, 31(6):1014–1021.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.
- Chao, Y. and Derdenger, T. (2013). Mixed bundling in two-sided markets in the presence of installed base effects. *Management Science*, 59(8):1904–1926.
- Chen, F., Liu, X., Proserpio, D., Troncoso, I., and Xiong, F. (2020). Studying product competition using representation learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’20*, page 1261–1268, New York, NY, USA. Association for Computing Machinery.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16*, pages 785–794, New York, NY, USA. ACM.
- Chung, J. and Rao, V. R. (2003). A general choice model for bundles with multiple-category products: Application to market segmentation and optimal pricing for bundles. *Journal of Marketing Research*, 40(2):115–130.

- Cichocki, A. and Phan, A.-H. (2009). Fast local algorithms for large scale non-negative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E92.A(3):708–721.
- Derdenger, T. and Kumar, V. (2013). The dynamic effects of bundling as a product strategy. *Marketing Science*, 32(6):827–859.
- Dudik, M., Erhan, D., Langford, J., and Li, L. (2014). Doubly Robust Policy Evaluation and Optimization. *Statistical Science*, 29(4):485 – 511.
- Gabel, S., Guhl, D., and Klapper, D. (2019). P2v-map: Mapping market structures for large retail assortments. *Journal of Marketing Research*, 56(4):557–580.
- Gabel, S. and Timoshenko, A. (2020). Product choice with large assortments: A scalable deep-learning model.
- Gelman, A. and Hill, J. (2007). *Data analysis using regression and multi-level/hierarchical models*, volume Analytical methods for social research. Cambridge University Press, New York.
- Grbovic, M. and Cheng, H. (2018). Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, page 311–320, New York, NY, USA. Association for Computing Machinery.
- Halko, N., Martinsson, P.-G., and Tropp, J. A. (2010). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.
- Jacobs, B. J., Donkers, B., and Fok, D. (2016). Model-based purchase predictions for large assortments. *Marketing Science*, 35(3):389–404.
- Jiang, Y., Shang, J., Kemerer, C. F., and Liu, Y. (2011). Optimizing E-tailer Profits and Customer Savings: Pricing Multistage Customized Online Bundles. *Marketing Science*, 30(4):737–752.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Lewbel, A. (1985). Bundling of substitutes or complements. *International Journal of Industrial Organization*, 3:101–107.
- Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, pages 3111–3119, USA. Curran Associates Inc.
- Prasad, A., Venkatesh, R., and Mahajan, V. (2015). Product bundling or reserved product pricing? price discrimination with myopic and strategic consumers. *International Journal of Research in Marketing*, 32(1):1–8.
- Rao, V. R., Russell, G. J., Bhargava, H., Cooke, A., Derdenger, T., Kim, H., Kumar, N., Levin, I., Ma, Y., Mehta, N., Pracejus, J., and Venkatesh, R. (2018). Emerging Trends in Product Bundling: Investigating Consumer Choice and Firm Behavior. *Customer Needs and Solutions*, 5(1):107–120.
- Rudolph, M., Ruiz, F., Athey, S., and Blei, D. (2017). Structured embedding models for grouped data. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 251–261. Curran Associates, Inc.
- Rudolph, M., Ruiz, F. J. R., Mandt, S., and Blei, D. M. (2016). Exponential family embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, pages 478–486, USA. Curran Associates Inc.
- Ruiz, F. J. R., Athey, S., and Blei, D. M. (2017). SHOPPER: A Probabilistic Model of Consumer Choice with Substitutes and Complements. Papers 1711.03560, arXiv.org.
- Schmalensee, R. (1982). Commodity bundling by single-product monopolies. *The Journal of Law & Economics*, 25(1):67–71.
- Schmalensee, R. (1984). Gaussian demand and commodity bundling. *The Journal of Business*, 57(1):S211–30.
- Stigler, G. J. (1963). United states vs. loew's inc.: A note on block booking. *Supreme Court Review*, pages 152–157.
- Stremersch, S. and Tellis, G. J. (2002). Strategic Bundling of Products and Prices: A New Synthesis for Marketing. *Journal of Marketing*, 66(1):55–72.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.
- Venkatesh, R. and Kamakura, W. (2003). Optimal bundling and pricing under a monopoly: Contrasting complements and substitutes from independently valued products. *The Journal of Business*, 76(2):211–231.
- Venkatesh, R. and Mahajan, V. (1993). A probabilistic approach to pricing a bundle of products or services. *Journal of Marketing Research*, 30(4):494–508.

Yang, T. C. and Lai, H. (2006). Comparison of product bundling strategies on different online shopping behaviors. *Electronic Commerce Research and Applications*, 5(4):295–304.

Zhou, Z., Athey, S., and Wager, S. (2018). Offline multi-action policy learning: Generalization and optimization. *arXiv*.

Appendix

A Supplementary tables and figures

Table A1: Observation counts from the working sample

	Count
Total users	534,284
Total sessions	947,955
Total purchase baskets	861,963
Total consideration sets	589,552
Unique products	35,000

Note 1: The table shows the size of the working sample after filtering out purchases and searches involving right tail products. We retain the top-35,000 products that include more than 90% of the purchases in our sample period.

Note 2: Purchase baskets include products purchased and consideration sets include products viewed but *not* purchased. The number of consideration sets are less than the number of purchase baskets because we define a product viewed only if the user opens the description page of the product. The user can, however, purchase without opening the product description page by directly adding the product to the cart while browsing.

Examples of products close to the focal product in the purchase space

Table A2: Products close to “Organic Russet Potatoes, 5 Lb (10-12 Ct)” in the purchase space

Product	Comp. score
Organic Celery Hearts, 16 Oz	0.75
Organic Grape Tomatoes, 1 Pint	0.74
Organic Green Bell Peppers, 2 Ct	0.74
Organic Carrots, 2 Lb	0.73
Organic Cauliflower, 1 Ct	0.73
Organic Garlic, 8 Oz	0.72
The Farmers Hen Large Organic Eggs, 1 Dozen	0.70
Organic Bananas, Minimum 5 Ct	0.69
Organic Broccoli Crowns, 2 Ct	0.69
Organic Romaine Hearts, 3 Ct	0.69

Note 1: The complementarity score (Comp. score) is a measure of proximity in the purchase space and is indicative of complementarity. A higher score implies stronger complementarity. The score is normalized such that the maximum possible value is 1.

Table A3: Products close to “Joy Ultra Dishwashing Liquid, Lemon Scent, 12.6 oz” in the purchase space

Product	Comp. score
Bounty Paper Towels, White, 12 Super Rolls	0.50
Tide PODS Plus Downy HE Turbo Laundry Detergent Pacs	0.48
P&G 45Oz Cmp Gel Detergent	0.48
The Art of Shaving Shave Cream, Sandalwood	0.47
Bounty Towel, Bounty Essentials	0.46
Bounty Paper Towels, Select-A-Size, 6 Triple Rolls	0.46
Lillian Dinnerware Pebbled Plastic Plate	0.45
Saratoga Spring Water	0.45
CLR Stainless Steel Cleaner	0.45
Pepcid Complete Dual Action Acid Reducer and Antacid Chewcap	0.45

Note 1: The complementarity score (Comp. score) is a measure of proximity in the purchase space and is indicative of complementarity. A higher score implies stronger complementarity. The score is normalized such that the maximum possible value is 1.

Table A4: Products close to “Neutrogena Oil-Free Acne Wash Redness Soothing Cream Facial Cleanser, 6 Fl. Oz” in the purchase space

Product	Comp. score
U By Kotex Barely There Daily Liners	0.53
Aveeno Active Naturals Daily Moisturizing Body Yogurt Body Wash	0.52
Palmer’s Cocoa Butter Formula Bottom Butter	0.52
Neutrogena Oil-Free Acne Wash Redness Soothing Facial Cleanser	0.51
Neutrogena Oil-Free Acne Face Wash Pink Grapefruit Foaming Scrub	0.49
Maybelline New York Fit Me Matte & Poreless Foundation, Natural Beige	0.48
Secret Invisible Solid Anti-Perspirant Deodorant 2 Ct	0.47
Motrin IB, Ibuprofen, Aches and Pain Relief	0.47
Nature’s Bounty Hair, Skin & Nails Gummies Strawberry	0.47
Equate Ibuprofen Pain Reliever/Fever Reducer 200 mg Tablets	0.46

Note 1: The complementarity score (Comp. score) is a measure of proximity in the purchase space and is indicative of complementarity. A higher score implies stronger complementarity. The score is normalized such that the maximum possible value is 1.

Examples of products close to the focal product in the consideration space

Table A5: Products close to “Organic Russet Potatoes, 5 Lb (10-12 Ct)” in the consideration space

Product	Sub. score
Green Giant Organic Golden Potatoes, 3 Lb	0.95
Green Giant Organic Red Potatoes, 3 Lb	0.95
Organic Russet Potatoes, 3 Lb	0.95
Green Giant Klondike Gourmet Petite Purple-Purple Fleshed Potatoes, 24 Oz	0.93
Green Giant Klondike Fingerling Potatoes, 24 Oz	0.93
Green Giant Golden Potatoes, 5 Lb	0.92
Organic Sweet Potatoes, 3 Lb	0.92
Green Giant Klondike Petite Red-White Fleshed Potatoes, 24 Oz	0.92
The Little Potato Garlic Herb Potato Microwave Kit, 16 Oz	0.92
The Little Potato Company Garlic Herb Oven Griller Kit, 16 Oz	0.91

Note 1: The substitution score (Sub. score) is a measure of proximity in the consideration space and is indicative of substitutability. A higher score implies stronger substitutability. The score is normalized such that the maximum possible value is 1.

Table A6: Products close to “Joy Ultra Dishwashing Liquid, Lemon Scent, 12.6 oz” in the consideration space

Product	Sub. score
Joy Dishwashing Liquid, Lemon, 5gal Pail	0.83
Joy Dishwashing Liquid 38 oz Bottle	0.79
Joy Dishwashing Liquid Lemon Scent 12.6 oz Bottle	0.71
Palmolive Ultra Anti-Bacterial Dish Soap, Orange, 56 Oz	0.70
Ajax Triple Action Dish Soap, Orange, 12.6 Oz	0.69
Palmolive Ultra Dish Soap, Orange, 25 Fl Oz	0.69
Biokleen Natural Dish Liquid, Citrus, 32 Oz, 12 Ct	0.69
Palmolive OXY Plus Power Degreaser Dish Soap, 10 Oz	0.69
Ajax Super Degreaser Dish Soap, Lemon, 52 Oz	0.69
Ajax Dish Soap, Tropical Lime Twist, 52 Oz	0.68

Note 1: The substitution score (Sub. score) is a measure of proximity in the consideration space and is indicative of substitutability. A higher score implies stronger substitutability. The score is normalized such that the maximum possible value is 1.

Table A7: Products similar to “Neutrogena Oil-Free Acne Wash Redness Soothing Cream Facial Cleanser, 6 Fl. Oz” in the consideration space

Product	Sub. score
Neutrogena Oil-Free Acne Face Wash With Salicylic Acid, 6 Oz.	0.84
Neutrogena Oil-Free Acne Face Wash Daily Scrub With Salicylic Acid, 4.2 Fl. Oz.	0.84
Neutrogena Oil-Free Acne Face Wash Pink Grapefruit Foaming Scrub	0.83
Neutrogena Naturals Purifying Pore Scrub, 4 Fl. Oz.	0.82
Neutrogena Rapid Clear Stubborn Acne Cleanser, 5 Oz	0.82
Neutrogena All-In-1 Acne Control Daily Scrub, Acne Treatment 4.2 Fl. Oz.	0.82
Neutrogena Oil-Free Acne Wash Pink Grapefruit Cream Cleanser, 6 Oz	0.82
Neutrogena Oil-Free Acne Face Wash With Salicylic Acid, 9.1 Oz.	0.81
Neutrogena Men Oil-Free Invigorating Foaming Face Wash, 5.1 Fl. Oz	0.80
Neutrogena Oil-Free Acne Face Wash Pink Grapefruit Foaming Scrub, Salicylic Acid Acne Treatment, 6.7 Fl. Oz.	0.80

Note 1: The substitution score (Sub. score) is a measure of proximity in the consideration space and is indicative of substitutability. A higher score implies stronger substitutability. The score is normalized such that the maximum possible value is 1.

Table A8: Popular bundles from each strategy

Bundle Type	Category-1	Product-1	Category-2	Product-2	Views	ATC Count
CC	Scent Boosters	Downy Unsopables In-Wash Premium Scent Booster with Fabric Conditioner	Dryer Sheets	Downy Infusions Botanical Mist Fabric Softener Dryer Sheets	141	11
CC	All-Purpose Cleaners	Method Antibac All Purpose Cleaner, Wildflower	Bathroom Cleaners	Method Antibac Bathroom Cleaner Spray, Spearmint	75	5
CC	All-Purpose Cleaners	Mrs. Meyer's Vinegar Gel Cleaner, Lemon Verbena	Bathroom Cleaners	Mrs. Meyer's Clean Day Tub and Tile Cleaner Spray, Lemon	212	5
CC	Miracle and Accessories	Custom Variety Pack Single-Serve In-Kitchen	Team and Creamers	International Delight Collection Inspiraions Single-Serve - Caramel Macchiato	245	5
CC	All-Purpose Cleaners	Method All-Purpose Cleaner, Original	Hand Soaps	Method All-Purpose Cleaner, Original	86	5
DC	Hand Soaps	Method All-Purpose Cleaner, Original	Hand Soaps	Method All-Purpose Cleaner, Original	67	7
DC	Fruit and Vegetable Snacks	Mrs. Meyer's Clean Day Liquid Hand Soap, 16 Oz.	Dish Soap	Mrs. Meyer's Clean Day Liquid Dish Soap, Honey Suckle, 16 Fl Oz	23	6
DC	Laundry Detergent	Nature's All Purpose Powerfully Clean Laundry Detergent, Clean Burst, 140 Loads	Protein and Meal Replacement	RxBAR Protein Bar, Chocolate Sea Salt, 1.8 Oz.	14	6
DC	Laundry Detergent	Arm and Hammer Powerfully Clean Laundry Detergent, Clean Burst, 140 Loads	Coffee	Maxwell House Medium Roast Ground Coffee, Original, 42.5 Oz	425	5
VR	Glass Cleaners	Method Glass + Surface Cleaner Spray, 28 Oz	Hand Soap	Method Foaming Hand Soap, Waterfall, 10 Oz	66	4
VR	Jerky and Dried Meats	Duke's Hot and Spicy Smoky Smoked Sausages, 5 Oz	Jerky and Dried Meats	Duke's Original Smoky Smoked Sausage Jerky 5 Ounce	30	15
VR	Disposable Tableware	Disie Everyday Paper Bowls, 10 Oz, 60 Count	Disposable Tableware	Disie Ultra Paper Plates 94 Ct	39	11
VR	Fabric Softener	Mrs. Meyer's Clean Day Fabric Softener, Lemon Verbena, 32 Loads	Fabric Softener	Mrs. Meyer's Clean Day Fabric Softener, Basil, 32 Loads	16	10
VR	Pasta and Noodles	Barilla Gluten Free Rotini, 12 Oz	Pasta and Noodles	Barilla Gluten Free Penne, 12 Oz	53	9
VR	Folding Tables	Lifetime 5' Essential Fold-in-Half Table	Folding Tables	Cosco 6' Centerfold Table	266	9
CP	Laundry Detergent	Method 4X Laundry Detergent, Beach Sage, 66 Loads	Water	San Pellegrino Sparkling Natural Mineral Water, 16.9 Fl Oz	162	14
CP	Chocolate	Mars Chocolate Favorites Mini Bars Variety Mix Bag	Fabric Softener	Method Fabric Softener, Beach Sage, 45 Loads	53	13
CP	Protein and Granola Bars	Nature Valley Protein Chewy Bar	Candy	Skittles/Lifesavers/Starburst Candy Variety Pack, 22.7 oz	133	10
CP	Laundry Detergent	Arm and Hammer Powerfully Clean Laundry Detergent	Protein and Granola Bars	Nature Valley Crunchy Granola Bar, Variety Pack	64	9
CP			Stain Removers	OxClean Versatile Stain Remover	422	9

Table A9: AUC for predicting bundle add-to-cart on hold-out test data

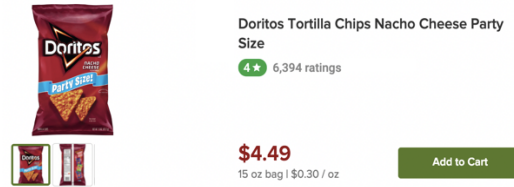
Model	AUC
Baseline	0.6635
Logistic	0.7023
Hierarchical Logistic	0.6967
LASSO	0.7009
Random Forest	0.7073
XGBoost	0.7276

Note 1: The baseline model excludes the product relationship heuristics – c_{ij} & s_{ij}

Note 2: Hierarchical Logistic regression is a mixed effects model with varying slopes that allows the effects of c_{ij} & s_{ij} to vary by product category. We estimate it using Restricted Maximum Likelihood.

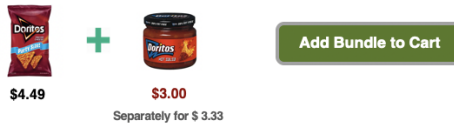
Illustrative examples of bundles from the field experiment

Grocery → Party → Snack → Chips → Tortilla Chips



Doritos Tortilla Chips Nacho Cheese Party Size
4★ 6,394 ratings
\$4.49
15 oz bag | \$0.30 / oz
Add to Cart

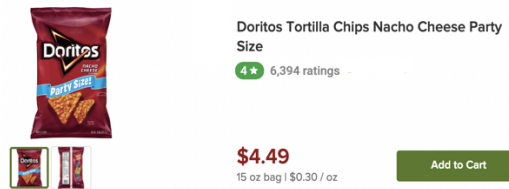
Buy products together and save 10% on the second product



\$4.49 + **\$3.00**
Separately for \$ 3.33
Add Bundle to Cart

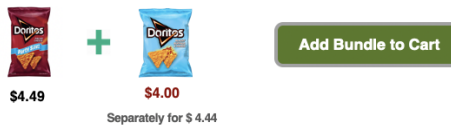
(a) Complementary bundle example

Grocery → Party → Snack → Chips → Tortilla Chips



Doritos Tortilla Chips Nacho Cheese Party Size
4★ 6,394 ratings
\$4.49
15 oz bag | \$0.30 / oz
Add to Cart

Buy products together and save 10% on the second product



\$4.49 + **\$4.00**
Separately for \$ 4.44
Add Bundle to Cart

(b) Variety bundle example

Figure A1: Illustrative examples from the field experiment

Policies optimized using the outcome model directly

In Figure 1-8, we use 897 focal products for which all four bundles types had at least one view during the experiment. Here, we repeat the analysis using all the focal products. We learn the optimized bundling policy in the same way as before. For focal products that did not have all four bundles viewed, the optimized policy selects from the remaining set. The results are shown in Figure A2. The policy which uses both scores preforms the best here as well. Moreover, the results for this policy are qualitatively similar to the policy optimized in Figure 1-8.

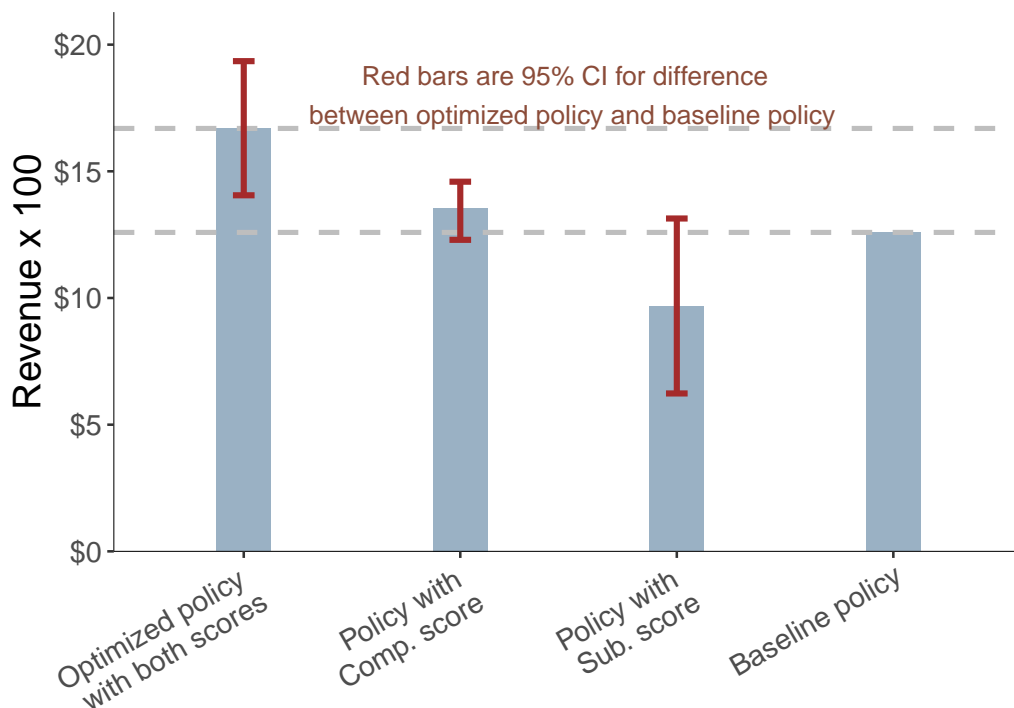
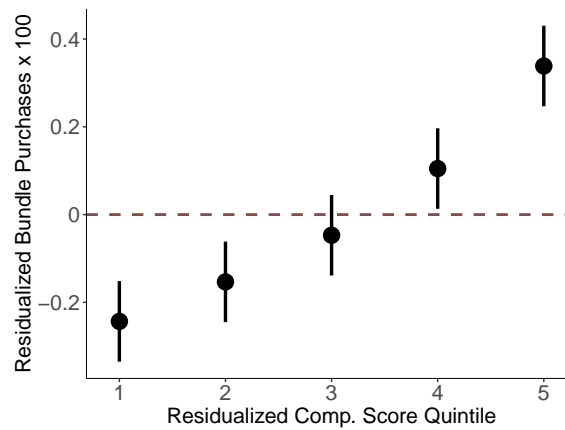


Figure A2: Revenue from the bundling policy optimized using all focal products

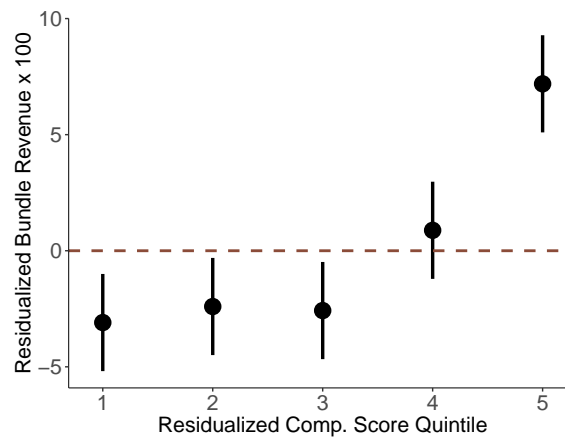
Note: The optimized policy uses both the scores to find the best bundle for a focal product. The second bar only uses the complementarity score to find the best bundles and the third bar only uses the substitutability score and selects the variety bundle. The last bar is the benchmark policy based on co-purchase rate.

Effect of complementarity score net of co-purchase

Figure A3 shows the results from the exercise described in where we regress residualized bundle success (purchases and revenue) on residualized complementarity score (c_{ij}). We residualize both variables by regressing them on historical co-purchase rate. The left panel of the figure shows the effect on residualized bundle purchases by residualized complementarity score quintiles. The right panel shows the impact on bundle revenue. In both cases, we see a large increase in bundle success as the complementarity score increases.



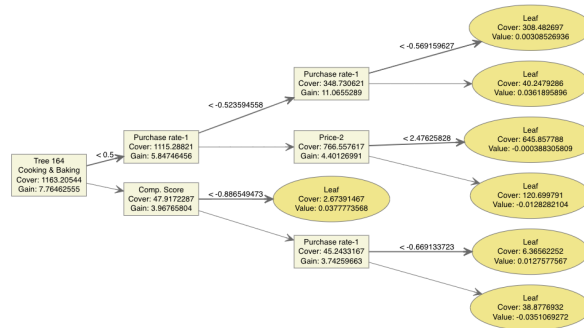
(a) Effect on bundle purchases



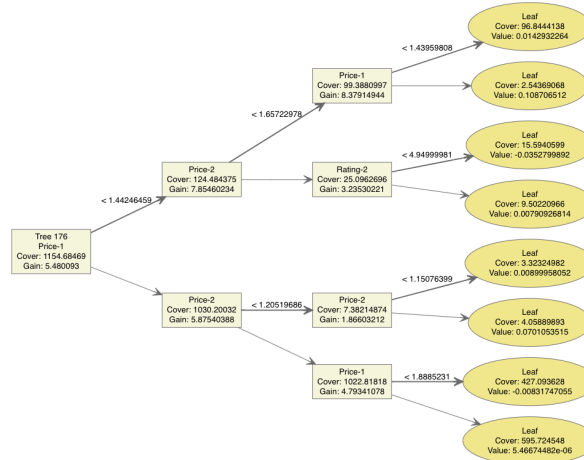
(b) Effect on bundle revenue

Figure A3: Effect of complementarity score on bundle success net of historical co-purchase rate

Sample trees from XGBoost model used to optimize the bundling policy



(a) Sample tree showing the interaction between the complementarity score and focal product aisle



(b) Sample tree showing the interaction between prices of the two products

Figure A4: Sample trees from the XGBoost model trained to learn the optimized bundle design policy

B Embeddings model

We provide intuition behind the product embeddings model below and formalize it subsequently. We explain the model for purchase baskets. The case for consideration sets is analogous; only the input products change.

B.1 Intuition

Our model is a slightly modified version of a widely used shallow learning technique from the machine learning literature used to analyze discrete, sparse data (Mikolov et al., 2013a,b). It has been fairly popularized in recent years due to its application in analyzing text. In the natural language processing (NLP) field, the intuition behind this method is simple — words that occur frequently in the same context are likely to have a semantic, and syntactic, relationship with each other. For instance, consider the following sentences:

Esha has **milk**, **cereal**, and **coffee** for breakfast

The tragedy is that she pours her **milk** before the **cereal**

She also has **coffee** with **milk** in the evening

She prefers **coffee** with a little bit of **sugar**

In these sentences, milk and cereal appear together frequently (relatively speaking) and that milk and coffee also appear together frequently. Our understanding of language, plus banal observation of the world, tells us that milk and cereal are “related” and that milk and coffee are also “related”. Essentially, these are the kinds of associations that we attempt to capture with our model, albeit with some refinements.

Translating the language from text documents to retail products, we exploit the notion of product baskets, i.e., we take products purchased together by consumers, and think of them as text sentences. The underlying idea is that products that appear frequently together in multiple baskets have a relationship that is beyond

mere random co-occurrence. To make this idea clear, consider Esha's consumption basket as shown below. It reproduces the sentences from above with everything but the products consumed removed. For the sake of exposition, we also add variants of the products consumed. The baskets then look like:

b_1 : low fat milk, crunchy cereal, dark roast coffee

b_2 : low fat milk, dark roast coffee

b_3 : dark roast coffee, raw sugar

These baskets are perfectly valid sentences for our algorithm to process with each product being a word and each sentence being a combination of these products. We can then build a model similar to the one used in NLP to learn relationships among products, with two important caveats: (1) the order of the products in our basket does not matter, and (2) our model needs to consider only two products at a time since we are building bundles with only two-component products. We thus transform each basket to a two-product combination with all possible permutations. This gives us the following baskets:

b_{11} : low fat milk, crunchy cereal

b_{12} : crunchy cereal, dark roast coffee

b_{13} : low fat milk, dark roast coffee

b_{21} : low fat milk, dark roast coffee

b_{31} : dark roast coffee, raw sugar

This transformation effectively converts our unstructured data to a simple classification problem where all the instances above form positive cases. To operationalize this model, we need two more inputs: (1) negative cases for the model to distinguish between products purchased together and products not purchased together, and (2) an optimization algorithm to learn the parameters. One can simply sample negative cases by considering pairs of products that do not occur in the

same baskets, but are present in the inventory (Mikolov et al., 2013b). This process is called negative sampling in the NLP literature. We can think of more complex negatively sampled procedures where we generate negative samples taking into account the product category hierarchy. For example, for generating negative samples for milk + coffee, one should use milk + tea instead of milk + batteries. We tried this approach while training our model. It ended up adding substantial complexity to the training procedure without any gains. Hence, we used the method recommended in the literature to generate negative samples from a unigram distribution (Mikolov et al., 2013b).

Returning to our example, we now have both positively labeled samples and negatively labeled samples. Hence, we can run our favorite classification algorithm to train the parameters. Of course this is an overly-simplified stylized example. We present a more formal treatment of the underlying process and the model in the next sub-section.

To complete the picture, along with products purchased together, we also consider products from users' consideration sets, which include products that were viewed together in the same browsing session. We similarly break them into pairs of two products to form positive cases, and likewise generate negative cases.

Lastly, with recent advances in machine learning methods and computational infrastructure there are now multiple ways to train these models (e.g., word2vec⁸, glove⁹). We use Tensorflow¹⁰, which provides gpu support so that we can easily scale the model to a large volume of data. We describe the model formally below in the context of purchase baskets. The reasoning can be easily extended to consideration sets.

B.2 Formal model

Consider a retailer with an assortment \mathcal{V} of size n . Suppose our representative consumer, Esha, purchases 5 products, forming the product basket $b_1: \{w_1, w_2, w_3, w_4, w_5\}$.

⁸<https://radimrehurek.com/gensim/models/word2vec.html>

⁹<https://nlp.stanford.edu/projects/glove/>

¹⁰<https://www.tensorflow.org/>

Our objective is then to predict the products $\{w_2, w_3, w_4, w_5\}$ given the product w_1 . Unlike natural language models, we do not consider the order of the products, but use the entire remaining basket to be the *context* for product w_1 . Let \mathcal{C} be the set of context products, such that, $\mathcal{C}(w)$ represents the set of products in the context for product w . With the basket above, given the product w_1 and its context $\mathcal{C}(w_1) = \{w_2, w_3, w_4, w_5\}$, we want to maximize the log-likelihood of the basket,

$$\mathcal{L}_{b_1} = \sum_{w \in b_1} \log P(\mathcal{C}(w)|w). \quad (1.7)$$

Here we introduce the concept of embeddings, the dense continuous representations we are trying to estimate. Suppose that each product in the assortment is represented by two d -dimensional real-valued vectors, v and u . The matrices \mathbb{U} ($|\mathcal{V}| \times d$) and \mathbb{V} ($d \times |\mathcal{V}|$) are the embedding matrices, where u_i and v'_i give two representations for product w_i . \mathbb{V} is the input matrix and \mathbb{U} is the output matrix. The process of predicting $\mathcal{C}(w_1)$, given w_1 boils down to estimating the probability $P(\mathcal{C}(w)|w)$ mentioned in 1.7. Considering one element w_2 from $\mathcal{C}(w_1)$, we can write this probability using the logit model,

$$P(w_2|w_1) = P(u_2|v'_1) = \frac{\exp(u_2 \cdot v'_1)}{\sum_{k=1}^{|\mathcal{V}|} \exp(u_2 \cdot v'_k)}, \quad (1.8)$$

where u_2 is the second row from the output embedding matrix \mathbb{U} and v'_1 is the first column from the input embedding matrix \mathbb{V} .

Generalizing expression 1.8 to account for all products in the context, we can write the conditional probability term in the objective function shown in 1.7 as:

$$P(\mathcal{C}(w_1)|w_1) = \prod_{w_j \in \mathcal{C}(w_1)} \frac{\exp(u_{w_j} \cdot v'_{w_1})}{\sum_{k=1}^{|\mathcal{V}|} \exp(u_{w_k} \cdot v'_{w_1})} \quad (1.9)$$

A point to note is the calculation of the denominator in the above expression. Typically, $|\mathcal{V}|$ is quite large and hence for computational efficiency we employ neg-

ative sampling as described in (Mikolov et al., 2013b) to approximate the denominator. With negative sampling, we only select a sample of the negative examples to update at each iteration. We use a unigram distribution to sample negative examples such that more frequently occurring products across baskets are selected more likely to be chosen. Assuming we select, N_s negative examples, we can write the approximate probability expression as

$$P(\mathcal{C}(w_i)|w_i) = \prod_{w_j \in \mathcal{C}(w)} \frac{\exp(u_{w_j} \cdot v'_{w_i})}{\sum_{k=1}^{N_s} \exp(u_{w_k} \cdot v'_{w_i})}. \quad (1.10)$$

Plugging this value in the log-likelihood function to estimate the probability of each product in the context $\mathcal{C}(w_i)$ for given a target product w_i , we get

$$\mathcal{L}_{b_1} = \sum_{w \in b_1} \left[\sum_{w_j \in \mathcal{C}(w_i)} \left(\log \sigma(u_{w_j} \cdot v_{w_i}) + \sum_{k \neq j, k=1}^{N_s} \log \sigma(-u_{w_k} \cdot v_{w_i}) \right) \right], \quad (1.11)$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function.

We estimate the parameters \mathbb{U} and \mathbb{V} by maximizing the likelihood of all baskets in the data set. The log-likelihood for the entire data set is given in Equation 1.12, where \mathcal{B} is the set of all product baskets observed in the data,

$$\mathcal{L}_{\mathcal{B}} = \sum_{b \in \mathcal{B}} \sum_{w \in b} \left[\sum_{w_j \in \mathcal{C}(w_i)} \left(\log \sigma(u_{w_j} \cdot v_{w_i}) + \sum_{k \neq j, k=1}^{N_s} \log \sigma(-u_{w_k} \cdot v_{w_i}) \right) \right]. \quad (1.12)$$

In practice, we use Adam (Kingma and Ba, 2015) to update the embedding vectors while minimizing the negative log-likelihood. Optimal hyper-parameters of the training algorithm including the dimensions of the embedding matrices are found using a hold-out validation set. In our model, we use $\mathcal{D} = 100$ and $N_s = 20$.

C Hierarchical model

To build intuition on how the product relationship scores influence bundle success and also to check their robustness across categories, we build a hierarchical logistic regression with varying slopes. We allow the effects of c_{ij} and s_{ij} to vary by the focal product aisle and the intercepts to vary by both – the focal product aisle and the add-on product aisle. We estimate the model shown in Equation 1.13.

$$Pr(Y_{ij} = 1 | \text{View}_i) = \text{logit}^{-1} \left(\alpha_{k[i]j} + \alpha_{k[j]i} + \beta_{k[i]j}^c c_{ij} + \beta_{k[i]j}^s s_{ij} + \gamma^T W_{ij} \right) \quad (1.13)$$

$$\alpha_{k[j]} \sim \mathcal{N}(0, \sigma^2)$$

$$\begin{pmatrix} \alpha_k[i] \\ \beta_k^c[i] \\ \beta_k^s[i] \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \mu_\alpha \\ \mu_{\beta^c} \\ \mu_{\beta^s} \end{pmatrix}, \Sigma \right)$$

where i indexes the focal product, j is the add-on product, $k[i]$ is the aisle of the focal product and $k[j]$ is the aisle of the add-on product. Together, ij make one bundle and we estimate the probability of the bundle being added-to-cart, conditional on the user viewing the focal product i . The intercept α_k is allowed to vary by both aisles. The slopes β^c and β^s , i.e., the coefficients for the complementarity heuristic c_{ij} and the substitutability heuristic s_{ij} vary by the aisle of the focal product. Pre-treatment variables and other product meta-data are captured by the vector W with their parameters γ held fixed. The varying parameters are estimated jointly with each parameter having a separate mean and variance. Their variances and covariances are given by the 3×3 matrix Σ .

Setting up the model in a hierarchical fashion helps us account for unobserved variation in product aisles that is not captured in a pooled regression model. Further, since we model the product aisles separately, we can use the model to generate predictions for aisles that were not part of our training data, and hence generalize our findings to a larger domain. In addition to statistical benefit, the hierar-

chical modeling also provides a systematic way to analyze the robustness of our model across aisles, as we show in Figure 1-10.

Estimation results from Model 1.13 are shown in the first column of Table C1. We find the both the scores are significant and positive, even after controlling for basic product features including the historical co-purchase rate. In the second column, we test whether asymmetry in prices has an effect on bundle success. We include a binary indicator that takes value 1 if $Price - 1 \geq Price - 2$. We find that bundles where the add-on product has the same or lesser price are more successful. The last coefficient in column 2 shows the result. We also test this using ANOVA. The results are shown in Table C2.

Table C1: Mixed effects hierarchical logistic regression to predict bundle add-to-cart

	<i>Dependent variable:</i>	
	Bundle add-to-cart	
	(1)	(2)
Comp. score	0.377*** (0.046)	0.373*** (0.046)
Sub. score	0.276*** (0.048)	0.265*** (0.048)
Hist. co-purchase rate	0.068*** (0.010)	0.066*** (0.010)
Price-1	-0.064 (0.047)	-0.154*** (0.056)
Price-2	-0.350*** (0.051)	-0.251*** (0.060)
Same brand	0.342*** (0.062)	0.330*** (0.062)
Diff. category Same aisle	0.212*** (0.059)	0.209*** (0.059)
Hist. purchase rate-1	-0.014 (0.027)	-0.011 (0.027)
Hist. purchase rate-2	0.027 (0.027)	0.026 (0.027)
Rating-1	0.078 (0.070)	0.078 (0.070)
Rating-2	-0.068 (0.063)	-0.067 (0.064)
Price-1 \geq Price-2		0.245*** (0.077)
Observations	227,311	227,311
Log Likelihood	-10,706	-10,701
Bayesian Inf. Crit.	21,647	21,649

Note 1: *p<0.1; **p<0.05; ***p<0.01

Note 2: Intercepts vary by focal and add-on product aisles. c_{ij} and s_{ij} vary by focal product aisle.

Table C2: Wald's test for equality of two price coefficients in Model 1

	χ^2	Chi Df	$\text{Pr}(>\chi^2)$
Price-1 = Price-2 in Model 1	10.44	1	0.001

Chapter 2

Algorithmic Pricing and Consumer Sensitivity to Price Volatility

Abstract

Algorithmic pricing can be broadly defined as a formula to set prices by a computer. It is typically associated with a lower cost of changing prices and a greater frequency of price changes. While commonly observed in ride-sharing, lodging, and airline tickets, there has been recent evidence of its implementation in pharmaceutical drugs, gasoline, and online retail. However, little is known about how consumers respond to encountering frequently changing prices for goods. Here we use detailed clickstream data from an online retailer that varied pricing methods to examine how exposure to the frequently-changing prices feature of algorithmic pricing affects purchase behavior. Aggregate analysis at the product-week level, before-and-after event studies around adoption time, and granular user-level models, all show a consistent pattern — exposure to price volatility increases price sensitivity. This is economically consequential because, even if implementing algorithmic pricing is profitable, it triggers unintended side effects that modify consumer behavior in ways that may undermine those gains. We complement these empirical findings with laboratory experiments and provide evidence for a key underlying mechanism—price salience.

2.1 Introduction

\$6.19 at 10:30 pm on Sunday, \$6.39 at 3:28 am on Monday, \$5.99 at 3:42 am, \$2.99 at 4:28 am, \$4.26 at 4:44 am, \$3.99 at 8:40 am, and \$4.47 at 12:21 pm. One may be forgiven for assuming these are prices for a stock listed on the stock exchange. These are, in fact, seven distinct prices of a single regular carbonated cola drink over the course of only two days in an online grocery retailer in the United States. How do consumers react when they see prices changing frequently?

Algorithmic pricing has been expanding across industries and channels. What perhaps used to be a specialized feature of airline tickets (McAfee and Te Velde, 2006) has now been documented in ride-sharing platforms (Chen, 2016; Cohen et al., 2016), gasoline markets (Assad et al., 2020), allergy drugs in online retailers (Brown and MacKay, 2019), and Amazon’s durable goods marketplace (Chen et al., 2016).

However, as with artificial intelligence or other forms of automation technologies (Brynjolfsson and McAfee, 2014; Ford, 2015; Agrawal et al., 2018), algorithmic pricing is an intangible concept that is not easy to dissect. The most salient feature identified in the literature is the striking price volatility, as measured by the number of price changes, over time (Chen et al., 2016; Calvano et al., 2019; Brown and MacKay, 2019; Assad et al., 2020). Research has shown that sellers that adopt algorithmic pricing are found to update prices several times per day. For example, Amazon is known to change product prices ~ 2.5 million times a day or, equivalently, the price for a product listed on Amazon changes every 10 minutes on average (Business Insider, 2018). Comparable examples from other industries include *Smart Pricing* by Airbnb (Airbnb, 2017) and *Surge Pricing* by Uber (Dholakia, 2015). In Uber’s case, prices change as frequently as every three to five minutes (Washington Post, 2015).

We obtain clickstream data from an online retailer that contains abundant examples similar to the one in the introductory paragraph. Figure 2.1 shows visually compelling evidence. We can distinguish between periods of stable prices initially

versus those of extremely volatile prices later on.

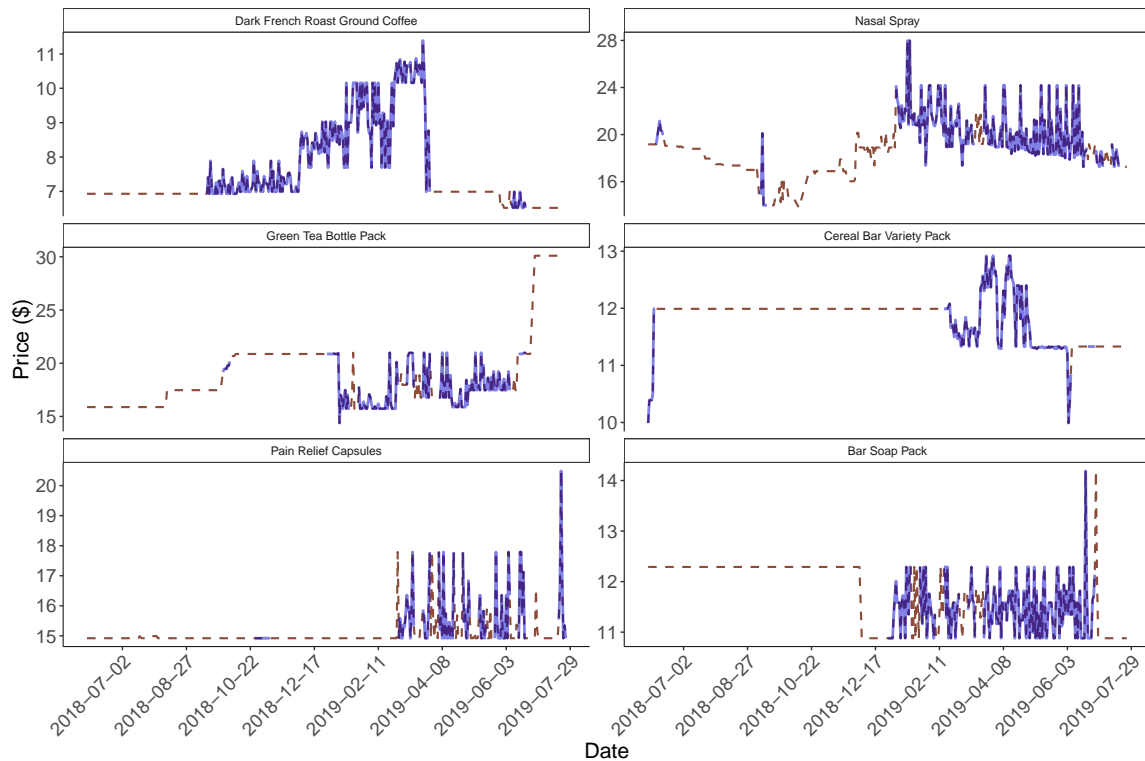


Figure 2.1: Examples of Price Variation: Daily Price with Periods of High Volatility Highlighted in Blue

A primary focus of the literature has been on studying whether, and how, the adoption of algorithmic pricing can alter competition incentives across rival firms (Miklós-Thal and Tucker, 2019; Calvano et al., 2019; Brown and MacKay, 2019; Hansen et al., 2021; Asker et al., 2021). However, despite its increasing prevalence, we know little about how *consumers* react when they are exposed to the strikingly high price volatility of machine algorithms. We contribute to this discussion by studying whether and how consumers' price sensitivity reacts to heightened price volatility, as measured by frequency of price changes and exposure to multiple unique prices, caused by pricing algorithms. While we are mindful that price sensitivity covers just one dimension of the greater realm of consumer behavior, price sensitivity has been a fundamental question in the economics and marketing literature. To illustrate, early papers on advertising were in fact absorbed about the connection between advertising and price sensitivity (Dorfman and Steiner, 1954;

Becker and Murphy, 1993). And to date, this question remains contested (Sethuraman et al., 2011).

Important exceptions in the area of consumer behavior are the studies of Haws and Bearden (2006) and Weisstein et al. (2013), which show that unusual price differences may evoke feelings of unfairness and thereby reduce willingness-to-pay.¹ These findings are obtained in the context of laboratory experiments, and therefore highlight the need to understand more “realistic shopping environments and under conditions of higher involvement” (Haws and Bearden, 2006). Although not in the context of algorithmic pricing, prior studies have shown that deep price promotions can trigger customer antagonism or incentivize promotion-seeking behaviors (Mela et al., 1997; Anderson and Simester, 2004; Hendel and Nevo, 2004; Rotemberg, 2005; Anderson and Simester, 2010; Elberg et al., 2019). Reflecting upon this collection of evidence, it is reasonable to assume that consumer behavior will not be indifferent to algorithmic pricing; but it is not immediately clear in which direction.

We begin with a conceptual discussion of how algorithmic pricing affects consumers’ price sensitivity in Section 2.2. The conceptual framework discusses two conflicting behavioral components. On the one hand, algorithmic pricing heightens price salience. We connect to Chetty et al. (2009); Bordalo et al. (2013, 2020), who study consumer choice in the context of boundedly rational consumers and salience effects. Relatedly, Finkelstein (2009); Busse et al. (2013); Hastings and Shapiro (2013); Busse et al. (2015); Aparicio and Rigobon (2020); Blake et al. (2021) find empirical support to the role of salience in offline and online markets. On the other hand, algorithmic pricing obfuscates the price anchor, namely “jams” the signal of good or bad deals. A number of conceptual and behavioral articles have examined limited price recall and constrained attention across attributes, such as Monroe (1973); Dickson and Sawyer (1990a); Lichtenstein et al. (1993); Thomas et al. (2010); Caplin and Dean (2015); Jung et al. (2019). The novelty of our work lies

¹Some studies use the term “dynamic pricing”. Throughout this work, we maintain the algorithmic pricing or machine pricing terminology.

in conceptualizing the ambiguous effect of algorithmic pricing through the lens of consumer behavior; and more precisely, the connection of price sensitivity to the role of price salience and price anchors—two critical behavioral findings that are often treated separately.

With these ideas in mind, we proceed to study algorithmic pricing in a field setting. We collaborate with an online retailer in the United States that implemented algorithmic pricing. Overall, the data covers a subset of 2,044 products and more than 670,000 distinct consumers for 15 months. Critically, the clickstream dataset covers both search and purchases, allowing tracking patterns of visitation and purchases across and within users. This is important because, intuitively, we can exploit the fact that distinct consumers browsed and purchased the same product, but they had different exposure to prices (and price volatility).

Three core empirical strategies are used to estimate the effect of price volatility on price sensitivity. First, we build intuition by estimating aggregate models at the weekly-UPC level, allowing us to obtain comparable estimates to frequently used scanner data. We find own-price elasticities that are qualitatively similar to those in prior work (Anderson and Simester, 2008; Hitsch et al., 2019; DellaVigna and Gentzkow, 2019; Semenova et al., 2017). Second, we consider a before-and-after event study around the time of adoption of algorithmic pricing in each product. Similar to Assad et al. (2020), adoption dates can be recovered by observing unusual spikes in price volatility. Importantly, this experimentation varied across products and across categories over time, and is presumably exogenous to a customer’s decision to visit the platform. Section 2.4 presents these results. Finally, in Section 2.5, we build upon these motivating signs to estimate a more stringent model of exposure to price volatility at the user level. We use two identification strategies - an instrumental variables approach and randomization inference to pin down the causal effect of frequently changing prices, and hence exposure to multiple unique prices *for the same product*, on purchase behavior. We find a consistent pattern throughout: price volatility makes demand more price sensitive.

We are mindful that, despite the granular clickstream dataset, it is not possi-

ble to exert complete control over a large-scale and long time-span field setting. Therefore, we conduct laboratory experiments to test the key effect of algorithmic pricing in a controlled environment. We implement a between-subject design in which participants are randomly assigned to two treatment conditions: stable pricing and algorithmic pricing. Participants in each cell are asked to simulate online purchases over a set of periods, and the price fluctuates from period to period. Importantly, the price series are calibrated with the real data, i.e. the stable prices and algorithmic prices mimic the prices of the online platform. Once again, and most importantly, participants are more price sensitive when exposed to higher price volatility. The experiment is implemented in two different subject pools—Amazon Mechanical Turk and MBA students. Section 2.6 presents the details.

Returning to the conceptual framework, we provide evidence that price *saliency* is a key behavioral mechanism through which sensitivity to price volatility operates. We motivate a behavioral model with field data, obtained from a technology company, supporting the intuition that prices capture additional “bits” of attention ((Jacoby, 1984)). Using eye-tracking technology installed in digital screens (placed in physical stores), we find that showing prices captures more attention, compared to signage without prices. We interpret this evidence, albeit secondary, as a valuable step in the direction of price saliency: if prices were to actually *change* on the screens, the attention (saliency) would very likely be much greater. We then formally test for saliency effects in the lab experiment. More precisely, following a vast tradition in the literature (Alba and Chattopadhyay (1986); Kissler et al. (2007); Finkelstein (2009); Kroft et al. (2013); Gaspelin et al. (2015)), a series of recall questions in the lab indicate that price volatility exacerbates price saliency. While we emphasize the role of saliency, we make no claim that it precludes other processes to operate as well—an interesting question for future research.

Taken together, these findings shed light on the non-obvious side effects of algorithmic pricing. Consumers are not indifferent to price volatility: it modifies the shopping behavior and increases price sensitivity. Intuitively, a more price-sensitive demand “eats” some of the benefits that presumably could have been ex-

tracted had price elasticity remained unaffected by the extreme price fluctuation. While beyond the scope of our work, in theory, this side effect could be utilized as “input” to perfect the technology. Said differently, it speaks to the possibility of *personalizing* algorithmic pricing to mitigate behavioral reactions. For example, suppose that the machine determines that the optimal price is \$3.17. Moreover, suppose that a given consumer has recently visited that product twice, and on those occasions, the price was \$3.09 and \$3.99. Perhaps it is better, for this consumer, to coarsen the price to the already-seen \$3.09, rather than to show \$3.17, a new price for that consumer. In statistical parlance, one can think of this as adding a penalty or regularization term on the number of price changes the algorithm is allowed to make. The new price may be a “better” price but the trade-off needs to be judged after netting out the negative impact of increased price sensitivity.

2.2 Conceptual Framework

In a standard friction-less shopping process, a representative consumer decides whether to buy a single product based on two attributes, namely the quality and the price (Lilien et al. (1995)). The decision is often summarized as $v = q - p$, and the outside option $v = 0$ implies no purchase. Algorithmic pricing alters the role of the price attribute, in ways that connect to various mechanisms studied in the literature.

Saliency: Increasing the frequency of price changes makes the price attribute more salient, relative to the attributes that remain static (brand, package, features, etc.). The shift in relative saliency can be thought of as changing the decision weights between a product’s price and value (Bertini and Wathieu, 2008; Aparicio and Rigobon, 2020; Bordalo et al., 2020; Blake et al., 2021). Saliency shifts irrespective of the sign of the price change, although the effect may be exacerbated when prices increase. In fact, Rotemberg (2005) and Anderson and Simester (2010) have shown that consumers develop antagonism when they realize that they have paid a higher price. That price variation might attract attention to prices can be indi-

rectly related to evidence of how salient, visual attributes are over-weighted in the decision (Krider et al. (2001); Folkes and Matta (2004)).

Signal to Noise: Consumers retrieve (or form) an anchor or reference price and compare it with the current price. Abundant research has explored how the price anchor is formed and the extent to which it can be manipulated by various forms of price presentation strategies (Kalyanaram and Winer (1995); Anderson and Simester (2003); Amaldoss and He (2018)). Typically, the price anchor is formed and updated upon exposure to past prices of the same product, advertised prices, or reference products in the category (Vanhuele and Drèze (2002); Jindal and Aribarg (2021); André et al. (2021)). Echoing prior studies showing limited price recall (Monroe (1973); Dickson and Sawyer (1990a); Lichtenstein et al. (1993)), algorithmic pricing exposes consumers to a complicated price path, often iterating between many distinct prices. This unstable path makes the price anchor noisier. Algorithmic pricing obfuscates the price anchor and, thereby, reduces sensitivity to notions of good or bad deals. Returning to the example in the Introduction: it is not obvious what the typical price for the carbonated cola should be.

This conceptual discussion captures two critical conflicting effects of algorithmic pricing. On the one hand, it increases price sensitivity by making the price a more salient element in the decision. The salience occurs as a result of shifting the relative variation between product attributes. Note that, interestingly, price sensitivity may be exacerbated even for consumers for whom the willingness-to-pay is greater than the actual price, i.e. an unnecessary side effect given that those consumers would have purchased this product regardless. On the other hand, it decreases price sensitivity by obfuscating the anchor price. Constant iterations between prices make the anchor price noisier (it “jams” the signal), thereby mitigating the reaction to a better or worse deal. We return to testable implications of this model in the context of lab experiments in Section 2.6.

To motivate the discussion that follows, we provide empirical evidence of price salience using novel experiment data from brick-and-mortar retailers. Digital screens are often placed in physical stores (e.g., supermarkets, gas stations, fashion stores)

to advertise selected products of the assortment. We collaborate with a European marketing analytics company which manages the content of these campaigns with its partner retailers. Throughout a period of approximately two months, the company placed regular advertisements on those digital screens; in some cases with prices and in some other cases without prices. Importantly, the screen is equipped with an eye-tracking technology that records consumer-level eye views and time spent viewing.

While in practice it is infeasible to capture a price *change* (i.e., the price is constant in the digital screens), the eye-tracking sensor allows testing whether prices increase attention. Our empirical strategy resembles prior work in which salience of an attribute entails attention to that attribute (Duncan, 1984; Folkes and Matta, 2004), and time is used as a measure of attention (Townsend and Kahn, 2014; Cian et al., 2015).

We test whether showing prices in the screens captures additional signage attention (controlling for the number of eye-views). In total, the data includes 3,570,646 distinct eye-views and 42 digital campaigns throughout two months. The average per-person view time of a screen, conditional on viewing, is approximately 7 seconds. Let v_{it} be the total number of views to screen i on day t and let t_t be the total time spent viewing screen i on day t . Time is measured in milliseconds. The measure of interest is $\tau \equiv \frac{t_{it}}{v_{it}}$, i.e. the time spent per eye view. We then estimate the following model:

$$\tau_{it} = \beta_0 + \beta_1 PriceDisplayed_i + \delta_t + \gamma_s + \epsilon_{it} \quad (2.1)$$

where $PriceDisplayed_i$ is an indicator variable that takes value 1 when the screen i contains a price (and 0 otherwise); and δ_t and γ_s denote day- and store- fixed effects, respectively.

Table 2.1 shows the results. When the digital screens display prices, time spent viewing the screen increases by 227 milliseconds ($p < 0.01$). This evidence supports the idea that price is a product feature prone to be salient and thereby to

Table 2.1: Eye-Tracking and Price Saliience

Attention Time	
Price Displayed	226.68*** (33.03)
<i>Fixed-effects</i>	
Store	✓
Day	✓
<i>Fit statistics</i>	
Obs.	139,978
R^2	0.15

capture additional cognitive attention. Furthermore, it complements the implications of algorithmic pricing. That is, because prices tend to capture attention, high-frequency price variation would presumably capture even more “bits” of attention and thereby heighten the role of price saliience. Further research, perhaps in a laboratory setting with the availability of fMRI technology (like Karmarkar et al. (2015)), is needed to better examine the behavioral decision-making process. For our purpose, we find this evidence motivating to more keenly study how heightened price volatility, caused by algorithmic pricing, makes price more salient and hence may influence consumer price sensitivity.

2.3 Data and Empirical Setting

We use data from an online retailer in the United States to examine the scope and implications of algorithmic pricing. Throughout the relevant time period, the retailer tried out algorithmic pricing for thousands of products across a wide range of categories, departments, and price levels. This empirical setting is particularly well-suited to studying behavior in response to algorithmic pricing for two reasons. First, the data includes clickstream records at the user level, which allows us to observe the entire sequence of the click activity (e.g., image impressions, search queries, product views, add-to-carts, orders placed). Moreover, the online groceries context involves repeated purchases across users and products, as well as a

relatively large assortment breadth.

Table 2.2 reports summary statistics on the data. We focus on a subset of products that experienced algorithmic pricing and, additionally, a minimum threshold of purchase records. Overall, the data covers 2,044 distinct products across groceries, household supplies, baby products, health and beauty, and pet supplies. The data covers 15 months, 673,677 distinct consumers, and over 2.6 million units sold.

Table 2.2: Data Description

Summary Statistics		
(1)	Distinct consumers	673,677
(2)	Categories	Household Supplies, Baby, Health & Beauty, Grocery, Pet Supplies
(3)	Distinct products	2,044
(4)	Time period	15 months
(5)	Units sold	2, 659, 906
(6)	Algorithmic periods	32%

There is no single definition of algorithmic pricing. In this work, we focus on one dimension of algorithmic pricing—a greater frequency of price changes. This echoes a growing literature that identifies price volatility as one of its most distinctive features (Chen et al. (2016); Brown and MacKay (2019); Assad et al. (2020)). We operationalize this concept in the field data as follows. For each product and week pair, we compute the sum of absolute price changes and the number of unique prices; if the first measure in any given week is greater than its respective median values across all weeks, *and* if the second one is greater than three, then we classify that week as an algorithmic pricing period.² Return to Figure 2.1 for some visual examples using this definition. The pattern of results remains quantitatively similar under alternative thresholds and definitions, e.g. based on the standard deviation of prices, number of price changes, or absolute size of price changes.

²An important digression is helpful. Conceptually, it is possible that the output of the algorithm is to set a flat price, e.g., collusion between two rival firms (Miklós-Thal and Tucker (2019); Calvano et al. (2019)). This definition essentially implies that there the price changed frequently and substantially within a week. At least in our field setting, institutional knowledge strongly indicates that periods in which the price fluctuates intensively are driven by the implementation or experimentation of price algorithms (e.g., price matching, a grid of mark-up rules, inventory triggers).

Robustness results are presented in Appendix G.

Summary statistics split by algorithmic pricing and stable pricing periods are shown in Appendix A. Furthermore, a variance decomposition test, shown in Appendix B, indicates that the product-week indicator of algorithmic pricing significantly explains a large portion of the price variation.

2.4 Aggregate Purchase Behavior

We proceed with a series of models with data aggregated at the product-week level, which allows us to more directly contrast our demand estimates with those using scanner data (typically at the same aggregation level). Later in Section 2.5 we consider consumer-level exposure to price volatility. Before we proceed, it would help to fix notation. Users are indexed with $i = \{1, \dots, I\}$, products with $j = \{1, \dots, J\}$, product categories with $c = \{1, \dots, C\}$, and time with $t = \{1, \dots, T\}$. In much of our analysis cases, t is year-week unless specified otherwise. Y is the number of units purchased and P is price.

We initially consider a reduced form demand model similar to Hitsch et al. (2019). We aggregate purchases at the weekly level, compute the unit-weighted price, and estimate the following baseline fixed-effects model:

$$\log(Y_{jt}) = \beta_0 + \beta_1 \log(P_{jt}) + \mu_j + \tau_t + \epsilon_{jt} \quad (2.2)$$

where Y_{jt} is the number of units sold for product j in week t and P_{jt} is the quantity-weighted price for product j at time t . μ & τ are product and time fixed effects respectively.

To understand the potential impact of algorithmic pricing on consumer behavior, we augment Equation 2.2 by including indicators for weeks during which the price for a product was highly volatile, as per the definition in the previous section. We simply call these weeks "algorithmic pricing weeks". Further, we interact these indicators with (log) price to test whether these high price-volatility periods

influence consumer price sensitivity. The updated model is:

$$\log(Y_{jt}) = \beta_0 + \beta_1 \log(P_{jt}) + \beta_2 A_{jt} + \beta_3 \log(P_{jt}) \times A_{jt} + \mu_j + \tau_t + \epsilon_{jt} \quad (2.3)$$

where A_{jt} is a binary indicator that equals 1 if product j is under an algorithmic pricing week during year-week t .

The results are shown in Table 2.3. Our focus is on β_3 ; the interaction between price and the algorithmic pricing indicator. A negative β_3 indicates that demand is more price sensitive when exposed to high-frequency price variation. Consider the baseline own-price elasticity of -1.51 in column (2). In periods of algorithmic pricing, the price sensitivity increases 0.087, which represents a sizable 5.8% in relative terms.

If algorithmic pricing makes the demand more price-sensitive, one might imagine that a greater intensity of price volatility exacerbates price sensitivity. Indeed, we find that as we require a higher number of distinct prices in a given week to be classified as an algorithmic pricing week, the estimand of interest becomes more price-sensitive. For example, the effect increases from approximately -0.07 to -0.10 when the threshold of distinct prices increases from two to five, as shown in Appendix C.

To provide visual intuition for what these results imply, we draw the demand curves for three product categories – dog supplies, chips, and fresh produce. We estimate the demand separately during algorithmic and stable pricing weeks after residualizing the quantity and price. The results are shown in Figure 2.2. For all three categories, we find the demand curve becomes flatter, i.e., the demand becomes more price sensitive. Furthermore, the rotation in the demand curve varies across the categories indicating potential heterogeneity. For example, the demand for dog supplies and chips changes much more than the demand for fresh produce. We explore this heterogeneity later in the section.

Next, in column (3) of Table 2.3, we remove the holiday period (mid-November to mid-January), a time when retailers typically run multiple promotions, and re-

estimate model 2.3. Here again, we find that even outside the holiday period, the effect is strong. We believe that this evidence is suggestive of important changes in consumer behavior as a result of the firm’s pricing policy. We explore this hypothesis more in later sections and use consumer-level data to causally estimate the impact. In the remainder of this section, we provide more evidence for the aggregate result using multiple specifications.

First, as in Anderson and Simester (2008), we estimate a quasi-Poisson demand model as follows:

$$\mathbb{P}(Y_{jt} = y) = \frac{e^{-\lambda_{jt}} \lambda_{jt}^q}{q!}, \quad q = 0, 1, 2, \dots \quad (2.4)$$

$$\log(\lambda_{jt}) = \beta_0 + \beta_1 \log(P_{jt}) + \beta_2 A_{jt} + \beta_3 \log(P_{jt}) \times A_{jt} + \mu_j + \tau_t + \epsilon_{jt} \quad (2.5)$$

The results are presented in column (4) in Table 2.3. Overall, we find very close estimates to those of the baseline model, reported in column (2).

Table 2.3: Aggregate Elasticity Estimates with Multiple Specifications

Dependent Variables:	Log units		Units	Log units		
	Gaussian	Gaussian	Poisson	Ortho ML	Mixed effects	
Model:	(1)	(2)	(3)	(4)	(5)	(6)
<i>Variables</i>						
Elasticity	-1.525*** (0.100)	-1.508*** (0.101)	-1.545*** (0.101)	-1.558*** (0.135)	-1.052*** (0.109)	-1.424*** (0.108)
Algo		0.260*** (0.036)	0.263*** (0.041)	0.232*** (0.040)	0.032*** (0.011)	0.285*** (0.034)
Elasticity × Algo		-0.087*** (0.015)	-0.088*** (0.017)	-0.081*** (0.018)	-0.150** (0.062)	-0.103*** (0.017)
<i>Fixed-effects</i>						
Product	Yes	Yes	Yes	Yes	Yes	Yes
Year week	Yes	Yes	Yes	Yes	Yes	Yes
<i>Fit statistics</i>						
Observations	122,309	122,309	103,954	122,309	59,157	122,309
R ²	0.558	0.559	0.558	-	0.399	-
Log-Likelihood	-112,205	-111,987	-95,450	-591,338	-46,113	-116,096

Two-way (Product & Year week) standard-errors in parentheses

Signif. Codes: ***: 0.01, **: 0.05, *: 0.1

Additionally, we consider recent methods in machine learning. We imple-

ment Semenova et al. (2017)'s approach of estimating own-price elasticities using orthogonal-machine learning. The methodology allows us to include a high-dimensional set of features as controls, including lagged values for price and purchases. The model is estimated in two-stages. In the first stage, we residualize the outcome (purchases) and the independent variable of interest (price) using lagged values of purchases, price, indicator for algorithmic pricing week, indicators for product, product category, and time. In the second stage, we regress the residualized outcome on the residualized price, the indicator for algorithmic pricing week, and the interaction of the two. To ensure unbiased estimates, regressions in the two stages are estimated on different sub-samples. More formally, the model is defined as follows:

First stage, estimated on sample S :

$$\log(Y_{jt}) = \beta_{0y} + g_y(\beta_y A_{jt}, \delta_y \sum_{t-1}^{t-4} \log(Y_{jt}), \gamma_y \sum_{t-1}^{t-4} \log(P_{jt}), \mu_j, \tau_t) + \epsilon'_{jt} \quad (2.6)$$

$$\log(P_{jt}) = \beta_{0p} + g_p(\beta_p A_{jt}, \delta_p \sum_{t-1}^{t-4} \log(Y_{jt}), \gamma_p \sum_{t-1}^{t-4} \log(P_{jt}), \mu_j, \tau_t) + \epsilon''_{jt} \quad (2.7)$$

Second stage, estimated on sample S' ($S \cap S' = \phi$):

$$\log(\tilde{Y}_{jt}) = \beta_0 + \beta_1 \log(\tilde{P}_{jt}) + \beta_2 A_{jt} + \beta_3 \log(\tilde{P}_{jt}) \times A_{jt} + \epsilon_{jt} \quad (2.8)$$

where \tilde{Y}_{jt} are the residuals from Model 2.6 and \tilde{P}_{jt} are the residuals from Model 2.7. $g_y()$ & $g_p()$ are functions that control for lagged features, product, category, and time effects. In our case, we use penalized l_1 regressions. Column (5) in Table 2.3 then shows the result from Equation 2.8. We again see that the results lead to the same conclusion as our baseline model, with the coefficient on the interaction term being negative.

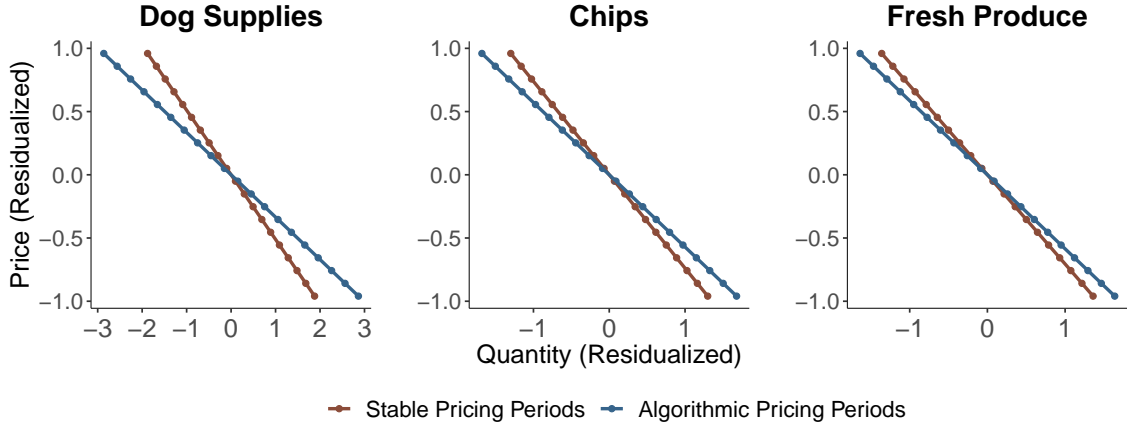


Figure 2.2: Price-Sensitive Demand Rotation

Notes: The graphs show demand curves across three categories after taking out the effects of product and time from log quantity and log price. The red demand curve is estimated separately by only considering periods of stable pricing. Analogously, the blue demand curve is estimated separately for periods of algorithmic pricing.

2.4.1 Heterogeneity Across Product Categories

To unpack the heterogeneity across product categories, we estimate a hierarchical mixed-effects model (Gelman and Hill, 2006). Previous literature has used similar random effects models to study price elasticities and online consumer behavior (e.g. Hoch et al., 1995). Mixed-effects models allow for partial pooling of information across products and categories. In our case, we use them to efficiently estimate product-level elasticities and conduct sub-group analysis. We use a nested hierarchical approach where we allow the intercept and slopes to vary by category and by each product within that category; we also allow the intercept to vary over time. The model is estimated using Restricted Maximum Likelihood. Varying the price elasticities with categories and products allows pooling of information across levels and regularizes coefficients in a data-driven way. We estimate the following model:

$$\log(Y_{jt}) = \beta_{0cj} + \beta_{1cj} \log(P_{jt}) + \beta_{2cj} A_{jt} + \beta_{3cj} \log(P_{jt}) \times A_{jt} + \epsilon_{jt} \quad (2.9)$$

where β_{0cj} is the intercept that is allowed to vary by category, product, and year-week, and all three slope coefficients $\beta_{1cj}, \beta_{2cj}, \beta_{3cj}$ are allowed to flexibly vary by

category and by product within a category.

Results from the mixed-effects model are shown in column (6) in Table 2.3. Once again, the results are qualitatively similar to the previous models. Reassuringly, the estimated own-price elasticities are qualitatively similar to recent studies using grocery data (Hitsch et al., 2019; Semenova et al., 2017). See the product-level distribution of own-price elasticities in Appendix D. In Figure D.5, we show the distribution of elasticities during algorithmic pricing and stable pricing periods, as depicted. We see that during periods of algorithmic pricing, there is a significant shift in greater price sensitivity across most products.

As a motivation to understand the value of these flexible models, multilevel analysis of variance (ANOVA) supports the inclusion of varying intercept and slope parameters. The results in Appendix B show that varying slopes explain a significant portion of the purchase variation.

The mixed-effects models allow us to take a step further in decomposing the results across products categories. Figure 2.3 shows the percentage change in price elasticity during algorithmic pricing weeks split by product category. For simplicity, we visualize 20 categories, 10 with the smallest change in elasticity and 10 with the largest change in elasticity. The red dashed line is the global average across all categories. Overall, and interestingly, we observe some but not fundamental heterogeneity across products. The biggest change is seen in stockable snacks, cleaning products, and pet supplies. On the other hand, health and beauty products such as skin care, hair care, and digestion & nausea see the smallest changes. In Appendix E we test for heterogeneity using different sub-groups such as expensive and cheap products, popular and unpopular products, and perishable and non-perishable products.

Our set of analyses serves two purposes – 1) it facilitates comparison with previous research using scanner data and helps establish common ground with existing literature, 2) it provides motivation to explore implications of algorithmic pricing at a more granular level. The results from Table 2.3, while significant and robust to specifications, do not allow causal identification of the impact of algo-

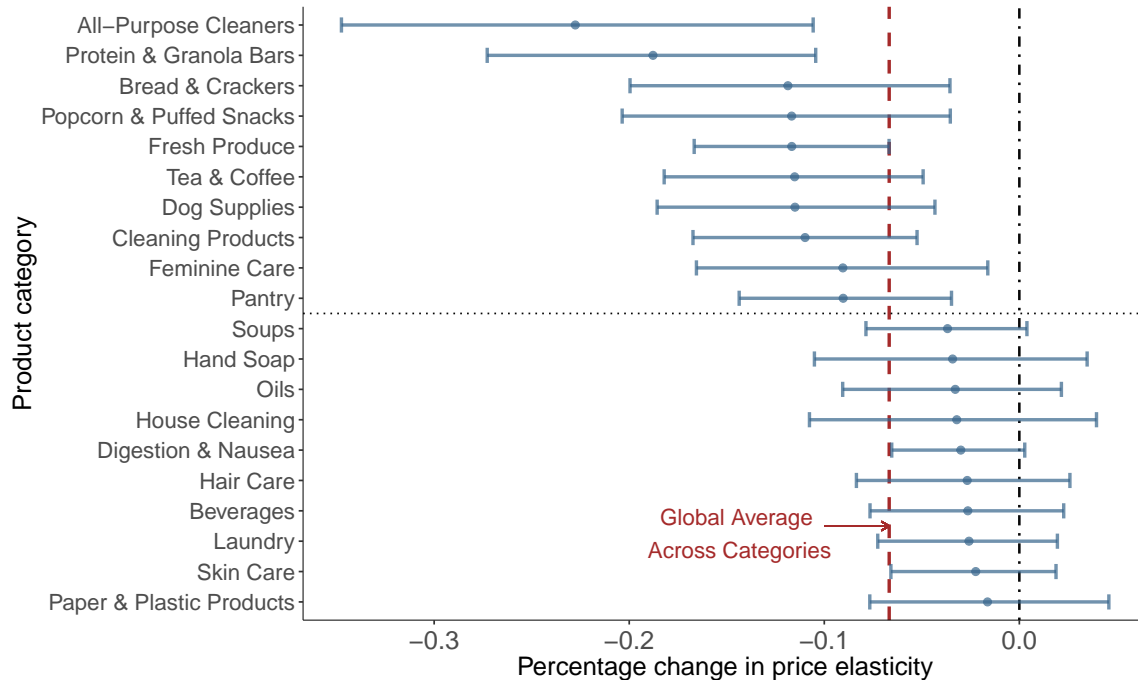


Figure 2.3: Top-10 and Bottom-10 Product Categories Based on Percentage Change in Price Elasticity During Algorithmic Pricing Weeks

algorithmic pricing on consumer behavior. Other than the potential impact of algorithmic pricing, current estimates could also be a reflection of either common demand shocks or compositional changes in demand, or a mix of the three. We conduct a more granular analysis with consumer visit-level data in the coming sections to identify this effect.

2.4.2 Before-and-After

We build upon the aggregate analysis with an event-study approach exploiting the first time algorithmic pricing was adopted in each product. We identify the date of the first algorithmic pricing week as per Section 2.3 and conduct an event study before-and-after the adoption. Importantly, we observe significant variation across products, i.e. different products had their first experimentation of algorithmic pricing in different dates. The same is true across product categories. (Appendix F presents the distribution of the timestamps of the first events.)

We report two main specifications: one in which we pool observations at the

product-week level, and a second in which we utilize observations at the user \times product \times week level. Appendix F reports robustness specifications. First, we estimate the following fixed-effects model:

$$\log(Y_{jt}) = \beta_0 + \beta_1 \log(P_{jt}) + \beta_2 Post_{jt} + \beta_3 \log(P_{jt}) \times Post_{jt} + \mu_j + \tau_t + \epsilon_{jt} \quad (2.10)$$

where $Post_{jt}$ is a binary indicator that takes the value 1 if $t' < t < t' + 8$ weeks for product j and t' is the first week when product j is under algorithmic pricing. To estimate the models, we use a window ± 8 weeks around the first week when a product adopted algorithmic pricing. However, to ensure robustness of our results, we consider three window cut-offs, i.e. adoption after 12 weeks of our sample start date, adoption after 20 weeks, and adoption after 28 weeks.

The results for products who adopted 20 weeks or after are shown in the first column of Table 2.4. Our main coefficient of interest is the one on the interaction between log price and the post-period indicator. We find that once the product switches to an algorithmic pricing regime, sensitivity to price changes increases.

The above result provides suggestive evidence towards increased price sensitivity. However, these results could partly be driven by compositional changes in the underlying user population. The granular scope of the data allows us to exclude this explanation by examining the same set of users who visit the product before and after the adoption of algorithmic pricing. We do this by using a different specification in which we estimate price sensitivity before-and-after the algorithmic pricing adoption at the user level. We use the same cut-offs as the aggregate model; however, we only consider users who browsed the product in both periods (i.e., before adoption and after adoption). More formally, we estimate the following fixed-effects model:

$$\log(Y_{ijt}) = \beta_0 + \beta_1 \log(P_{ijt}) + \beta_2 Post_{jt} + \beta_3 \log(P_{ijt}) \times Post_{jt} + \delta_{ij} + \epsilon_{ijt} \quad (2.11)$$

where Y_{ijt} is the number of units of product j purchased by user i at time t and P_{ijt} is the average price for product j seen by consumer i during time-period t .

Note that here there are only two time-period observations per user-product pair, i.e. one before the product adopts algorithmic pricing and the second after the product adopts algorithmic pricing. Importantly, this allows to control for user-product fixed effects (δ_{ij}). We are interested in β_3 , the coefficient on the interaction of average price and the indicator for the post-period. Similar to Equation 2.10, $Post_{jt}$ takes the value 1 if $t' < t < t' + 8$ weeks, where t' is the first week when product j is under algorithmic pricing. As in the aggregate case, to ensure the robustness of our results, we consider three window cut-offs for adoption – 12 weeks, 20 weeks, and 28 weeks.

The results are shown in the second column of Table 2.4. Since we control for user-product fixed effects, i.e., we estimate the coefficient using variation only within the same user-product pair. We see that after products adopt algorithmic pricing, consumers become more price sensitive. We posit that this effect is primarily driven by repeated exposure to different prices for the same product, which makes price more salient, making consumers put more weight on it during the purchase decisions. We explore this hypothesis in subsequent sections.

2.5 Consumer-Level Exposure to Price Volatility

The field data from the online retailer covers detailed browsing and shopping clicks at the consumer level. This granular data allows to estimate the effect of algorithmic pricing by exploiting consumer-level exposure to price variation while controlling for user-, product-, and time- fixed effects. Intuitively, consumers that had exposure to more distinct prices and a higher frequency of price changes, had a larger exposure to algorithmic pricing when shopping online groceries.

We introduce a model at the user level that bears resemblance to prior studies investigating the “stock” of advertising (Erdem et al. (2008); Shapiro et al. (2021)). In our case, we translate the model to one where, instead of advertising, we keep track of the stock of price volatility exposure. Said differently, the model accounts for spillovers of price volatility across products when accumulating the exposure

Table 2.4: Aggregate and User-Level Price Sensitivity Before and After Adoption of Algorithmic Pricing

Dependent Variables: Model:	Log units (1)	Units (2)
<i>Variables</i>		
Log price	-1.03** (0.389)	-1.12*** (0.222)
Post period	0.238*** (0.084)	0.208*** (0.058)
Log price × Post period	-0.101*** (0.035)	-0.072*** (0.025)
<i>Fixed-effects</i>		
Product	Yes	
Year week	Yes	Yes
User-product		Yes
<i>Fit statistics</i>		
Observations	7,652	42,864
R ²	0.687	0.465
Log-Likelihood	-5,003	-57,195

Model (1): Two-way (Product & Year week) standard-errors

Model (2): Three-way (User, Product & Year week) standard-errors

*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Notes: The first column is OLS regression of total weekly sales on average weekly price. Post period is a binary variable that takes the value 1 after a product adopts algorithmic pricing. Second column emulates the spirit of the first regression at the user-level. Each observation is at the user-product-period level, i.e., there is one observation for a user-product pair before algorithmic pricing adoption for that product and one observation after it. The dependent variable is the number of units purchased and price is average across all exposure for that user-product pair.

over time. Conceptually thinking of algorithmic pricing through the lens of advertising and price sensitivity (Dorfman and Steiner (1954); Becker and Murphy (1993)) is helpful because it illustrates that algorithmic pricing cannot be reduced to a simple A/B test. Instead, its core effect must consider the accumulation of price volatility across products and over time.

With these ideas in mind, we define the following user-level model:

$$Y_{ijt} = f(P_{ijt}, A_{it}, X_{ijt}; \epsilon_{ijt}) \quad (2.12)$$

where Y_{ijt} is the number of units of product j purchased by user i at time t . A_{it} is the cumulative effect of algorithmic pricing that user i has accumulated till time t . It is the total number of unique prices that the consumer has seen over the past L days across all products that the consumer browsed. In our regressions,

we use $L \in \{7, 15, 30, 60\}$ days to account for different intensities of exposure to algorithmic pricing. We refer to A_{it} as the algorithmic pricing stock or, succinctly, the algo-pricing stock. X_{ijt} are user history variables that account for the user's search and purchase intensity. We are interested in understanding how exposure to A_{ijt} modifies user behavior.

We use the following econometric specification for the model:

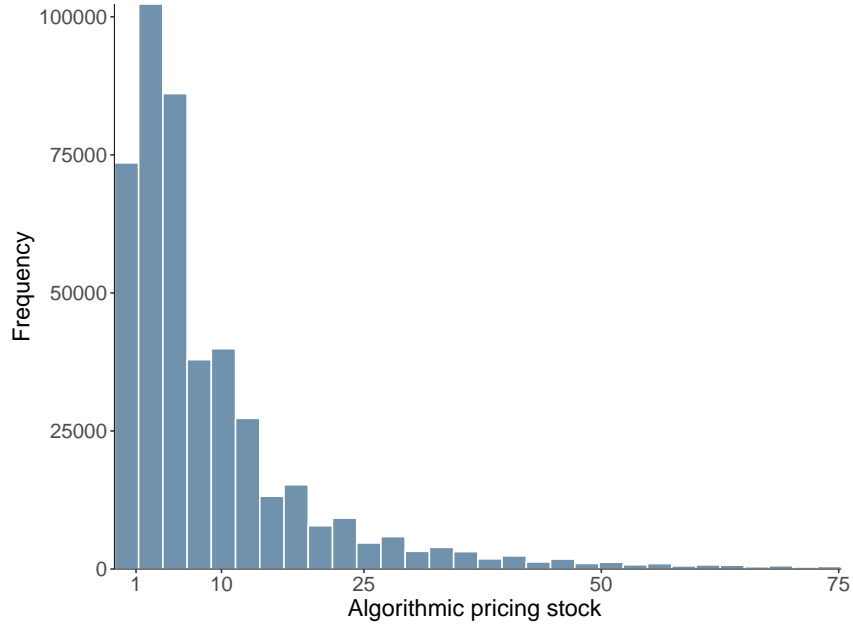
$$\begin{aligned} \mathbb{P}(Y_{ijt} = y) &= \frac{e^{-\lambda_{ijt}} \lambda_{ijt}^q}{q!}, \quad q = 0, 1, 2, \dots \\ \log(\lambda_{ijt}) &= \beta_0 + \beta_1 \log(P_{ijt}) + \beta_2 A_{it} + \beta_3 \log(P_{ijt}) \times A_{it} + \\ &\quad \delta X_{ijt} + \Gamma_i + \mu_j + \tau_t + \epsilon_{ijt} \end{aligned} \tag{2.13}$$

Here again, we are interested in the coefficient β_3 , which is the interaction between price and the algorithmic pricing stock. A negative value for β_3 indicates that consumers become more price sensitive after exposure to algorithmic pricing. In the regression, we measure the stock using the total number of unique prices that a consumer has been exposed to for the products that she visited *more than once*. As an example, say the consumer visited the product page for *Nutella* thrice in the past 15 days and saw two different prices. In addition, she visited the page for *Diet Coke* once and hence just saw one price for it. Her algorithmic pricing stock A_{it} is two. The price for *Diet Coke* is not counted since she only visited the product once. X_{ijt} are controls that account for the browsing intensity of the consumer. They include the total number of visits that the consumer made in the past L days, the total number of products browsed per visit, and the total number of purchases made. Γ_i are user-level fixed effects, μ_j are product fixed effects and τ_t are week fixed effects.

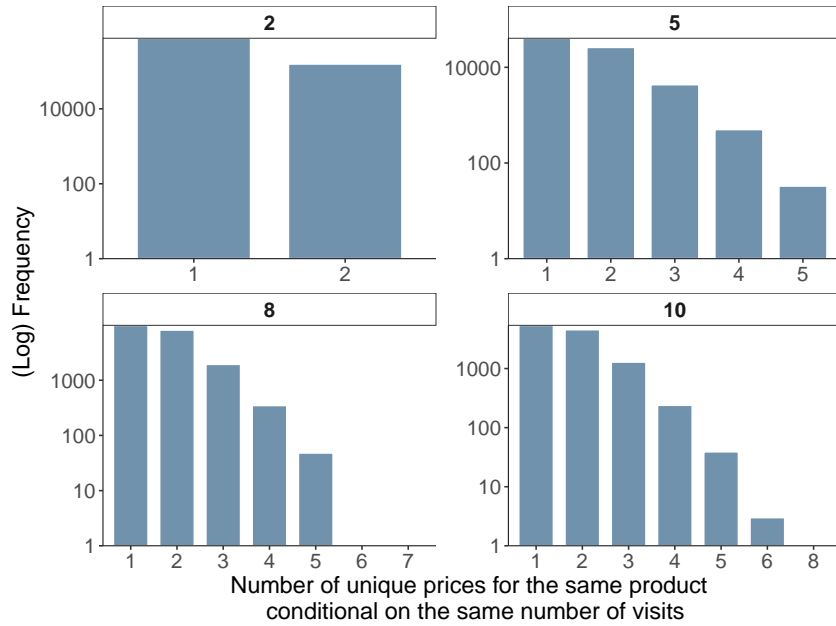
The motivation behind this model is to investigate whether consumers who are exposed to more unique prices for the same product, i.e. they have a larger A_{it} , tend to become more price sensitive. Arguably, one may worry that since A_{it} is not randomly assigned but rather dictated by the user's search process, the

effect of exposure to algorithmic pricing on purchase behavior is not identified. For example, a given consumer who tends to search more may intrinsically be more price-sensitive and hence may repeatedly visit the retailer's website to fetch a good deal. As a consequence of their repeated visits, they naturally get exposed to different prices for the same product. Hence, the effect we estimate is just an artifact of browsing intensity and not necessarily a change in behavior.

Fortunately, the granular nature of clickstream data allows to finely control for time-varying browsing and purchase intensity of consumers. Estimation of the model then critically depends upon conditional variation in A_{it} . More specifically, we need variation in A_{it} conditional on the number of visits, i.e., we need users who visited the same product the same number of times but were exposed to a different number of prices. Figure 2.4 presents this variation. Panel (a) shows the marginal distribution of algorithmic pricing stock aggregated at the user-date level. Panel (b) shows the conditional distribution of algorithmic pricing stock aggregated at the user-product-date level. Each facet in Panel (b) conditions on the number of visits made by users for the same product. For example, the top-right facet of Panel (b) shows that users who visited a product five times in the past 30 days could have been exposed to anywhere between one and five unique prices. This variation allows us to control for the browsing intensity of users. To control for unobserved time-invariant user and product heterogeneity, we use user- and product- fixed effects.



(a) Marginal Distribution of Algo-Pricing Stock



(b) Distribution of Algo-Pricing Stock Conditional on the Number of Visits. Each panel is a different number of visits for the same product.

Figure 2.4: Marginal and Conditional Distributions of Algorithmic Pricing Stock over a 30-day Period

Finally, to causally pin down the effect of algo-pricing stock A_{it} , we use two

identification strategies – one based on instrumental variables and the second is based on randomization inference. Both strategies crucially depend upon the variation in the timing of user’s visits to the website. Specifically, we assume that the exact time a user visits the retailer’s website is as-good-as-random. Then, if she would have visited a few hours earlier or later, then she may have seen a different price for the products she visits. Consequently, this changes her exposure to algorithmic pricing, i.e., it changes the number of unique prices she ends up seeing. We explain both approaches in the sections below.

2.5.1 Instrumental Variables

To capture observed time-varying heterogeneity we include detailed user and user-product level controls such as the number of products searched, the number of total purchases made in the past, and the number of purchases for this particular product made in the past. Further, to causally pin down the effect of algo-pricing stock A_{it} , we exploit variation in the timing of user visits and calculate the number of unique prices the user *could have seen*, had she come at a different time, but *did not see*. This gives us an instrument for A_{it} . The intuition behind the instrument is that the purchase decision of a consumer naturally depends on the prices seen, but does not depend upon the prices not seen. However, prices for a particular product are correlated across time. Hence, the prices the consumer *did not see* cannot influence the outcome directly, except through their correlation with the prices she did see.

To make things concrete, consider two users A and B who both visited the retailer’s website thrice in the past 15 days, albeit on different days or at different times on the same day. For simplicity, assume that both saw the same product three times. During the past 15 days, the price for this product was fluctuating independently of these two users’ visits (because of competitor effects, inventory state, and/or aggregate demand). Because of the difference in timing of their visits, user A was exposed to only one unique price for the product whereas user B

was exposed to three unique prices. The purchase decision that both the users are to make today depends upon the current price as well as the history of prices observed. However, it does not depend upon the prices not observed, except through their correlation with the observed prices. This makes the prices for the same product during the same time period that the consumer could have seen but did not see a valid instrument.

It is worth pointing out that this is a non-causal instrument. We don't observe exogenous shocks to prices at the system level. Rather we carefully isolate independent variation for each user based on their historical visit times. While not directly comparable, this instrument is similar in spirit to the one used by [Assad et al. \(2020\)](#) and [Ellison and Ellison \(2009\)](#). [Assad et al. \(2020\)](#) first identify station-level adoption of algorithmic pricing software in the gasoline markets using changes in high-frequency markers of prices³ and then identify brand-level adoption using the proportion of the brand's stations who have adopted. Their instrument is non-causal as well and works on the assumption that brand-level adoption decisions are independent of local station-level shocks. [Ellison and Ellison \(2009\)](#) use prices of products from one category with prices from another category to estimate price elasticities for PC-RAM modules.

For estimation, we use a two-stage control function approach as described in [Petrin and Train \(2010\)](#) where we instrument for the algo-pricing stock (A_{it}) and its interaction with price ($\log(P_{ijt}) \times A_{it}$). In the first stage, we run two regressions – 1) A_{ij} on the two instruments, exogenous controls (X_{ijt}) and the fixed effects from Equation 2.13, and 2) $\log(P_{ijt}) \times A_{it}$ on both instruments, exogenous controls, and fixed effects. In the second stage, we run the regression from Equation 2.13 by including the residuals from the two first-stage regressions. This control function is equivalent to running a TSLS procedure for instrumental variables when the outcome is linear. In our case, since our model for user purchases is a Poisson regression, we use the control function approach instead. Finally, to take into account the uncertainty from the first-stage estimation, we use clustered bootstrap to

³We follow a similar idea in our before-and-after analysis in Section 2.4.2.

Table 2.5: User-Level Price Sensitivity Estimates using Two-Stage Control Functions

Dependent Variable: Model:	Baseline		Units Control Function	
	(1)	(2)	(3)	(4)
<i>Variables</i>				
Log price	-1.13*** (0.086)	-1.11*** (0.079)	-1.03*** (0.087)	-1.01*** (0.081)
Log price x Algo pricing stock	-0.045*** (0.008)	-0.052*** (0.015)	-0.206*** (0.021)	-0.211*** (0.021)
Algo pricing stock	0.284*** (0.018)	0.268*** (0.033)	0.566*** (0.043)	0.669*** (0.059)
Log # of prior purchases - total		-0.268*** (0.006)		-0.281*** (0.010)
Log # of total visits		-0.247*** (0.009)		-0.273*** (0.017)
Log # of product visits		1.86*** (0.030)		1.85*** (0.032)
First stage residual - 1			-0.330*** (0.045)	-0.486*** (0.066)
First stage residual - 2			0.197*** (0.023)	0.199*** (0.020)
<i>Fixed-effects</i>				
User, Product, and Year-Week				
<i>Fit statistics</i>				
Observations	7,891,405	7,891,405	7,891,405	7,891,405
Pseudo R ²	0.206	0.245	0.206	0.245
Log-Likelihood	-3,084,927.3	-2,934,026.1	-3,083,960.9	-2,933,163.3
<i>Two-way (User & Product) standard-errors in parentheses</i>				
<i>Signif. Codes: ***: 0.01, **: 0.05, *: 0.1</i>				

Notes: The table shows baseline estimates (columns 1 and 2) and two-stage control function estimates (columns 3 and 4) for the consumer-level model in Equation 2.13. Algorithmic pricing stock and time-varying history variables are calculated over a 30-day period. The dependent variable in all models is the number of units of a particular product, purchased by a user during a single visit. All models control for user, product, and year-week fixed effects. Standard errors for columns 3 & 4 are estimated using clustered bootstrap to account for first-stage estimation error.

estimate standard errors.

The results are shown in Table 2.5. The first two columns show the baseline model where we don't use the control functions. After controlling for the user's browsing and purchase intensity, plus the user, product, and year-week fixed effects, we find that consumers who were exposed to more unique prices do become more price sensitive. In columns 3 and 4, we use a control function approach which accounts for potential endogeneity in algorithmic pricing stock. We find that, after correcting for endogeneity, the effect of algorithmic pricing stock almost doubles in absolute value. These results provide direct evidence of consumers becoming more sensitive due to heightened volatility in prices caused by algorithmic pricing.

Table 2.6: First stage results for user level control function regression

Dependent Variables: Model:	Algo pricing stock (1)	Log price x Algo pricing stock (2)
<i>Variables</i>		
# of prices not seen	-0.127*** (0.004)	-0.900*** (0.018)
Log price x # of prices not seen	-0.012*** (0.001)	0.237*** (0.006)
Log price	0.079*** (0.012)	-1.06*** (0.046)
Log # of prior purchases - total	0.144*** (0.003)	0.290*** (0.006)
Log # of total visits	0.590*** (0.006)	1.31*** (0.015)
Log # of product visits	0.219*** (0.005)	0.441*** (0.019)
<i>Fixed-effects</i>		
User, Product, and Year-Week		
<i>Fit statistics</i>		
Observations	8,918,548	8,918,548
R ²	0.783	0.733
Log-Likelihood	-4,015,435.5	-11,739,843.7
F-test	152.1	115.5

Two-way (User & Product) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

ing. In all models, the algorithmic pricing stock and user history variables are calculated over a 30-day period. In the appendix, we provide robustness checks where user history is calculated over 7, 15, and 60-day periods. Furthermore, we also test for the robustness of functional form in Equation 2.13 by running vanilla OLS and correcting for endogeneity using two-stage least squares. Across all specifications, we unanimously find that exposure to more unique prices for the same product makes consumers more price sensitive.

Heterogeneity by Consumer Type

We investigate how the change in sensitivity varies by consumer type. We use two measures of consumer heterogeneity – historical purchases and tenure. For the first one, we calculate the number of total purchases made by a user in the past 60 days and split the consumer base at the median. Similarly, we calculate the tenure of the user on the retailer’s platform from the date of the user’s first visit and split at the median. We then re-estimate Model 2.13 separately for each sub-group using

the two-stage control function approach described above.

The results are shown in Table 2.7. Columns (1) and (2) show the estimates for high-value and low-value customers, as defined by historical purchases. Overall, we find the effect of exposure to algorithmic pricing on price sensitivity to be strong and negative, i.e., both “high-value” and “low-value” consumers become more price sensitive. However, as a proportion of their baseline elasticity, high-value consumers, on average, experience twice as large of a change in price elasticity as compared to low-value consumers. This comparatively larger change also holds for consumers with a longer tenure with the retailer.

Table 2.7: User Level Sub-Group Analysis using Two-Stage Control Functions

Dependent Variable:	Units			
	Purchases >= Median	Purchases < Median	Tenure >= Median	Tenure < Median
Model:	(1)	(2)	(3)	(4)
<i>Variables</i>				
Log price	-0.802*** (0.072)	-1.27*** (0.099)	-0.835*** (0.076)	-1.17*** (0.091)
Log price x Algo pricing stock	-0.275*** (0.023)	-0.256*** (0.026)	-0.266*** (0.023)	-0.170*** (0.031)
Algo pricing stock	0.664*** (0.072)	1.74*** (0.094)	0.649*** (0.076)	1.48*** (0.095)
Log # of prior purchases - total	-0.243*** (0.012)	-	-0.206*** (0.012)	-1.29*** (0.028)
Log # of total visits	-0.249*** (0.025)	0.256*** (0.034)	-0.275*** (0.024)	0.124*** (0.037)
Log # of product visits	1.65*** (0.034)	3.19*** (0.036)	1.61*** (0.032)	2.84*** (0.044)
First stage residual - 1	-0.459*** (0.083)	-0.794*** (0.087)	-0.430*** (0.087)	-1.01*** (0.090)
First stage residual - 2	0.233*** (0.024)	0.186*** (0.023)	0.215*** (0.025)	0.169*** (0.026)
<i>Fixed-effects</i>				
User, Product, and Year-Week				
<i>Fit statistics</i>				
Observations	3,846,987	3,483,403	3,833,666	3,357,457
Pseudo R ²	0.252	0.298	0.252	0.295
Log-Likelihood	-1,583,530.4	-1,174,606.6	-1,540,480.2	-1,206,230.1

Two-way (User & Product) standard-errors in parentheses

Signif. Codes: ***: 0.01, **: 0.05, *: 0.1

2.5.2 Haphazard Visitation Timing

Our second identification strategy is based on the assumption that the exact time users visit is as-good-as random. Consequently, the actual price they end up seeing depends on the time they visit. Because prices change frequently, had they come a few hours earlier or later, they may have seen a different price. We build on this thought experiment and posit a randomization scheme that allows us identification and inference — using Fisherian randomization inference — of the effect of algorithmic pricing stock on consumer behavior.

An Ideal Experiment of Exposure to Algorithmic Pricing

It is instructive to ponder what an ideal experiment for exposure to algorithmic pricing, i.e. exposure to high volatility in prices would look like. One may presume that an A/B test at the user or product level could help us achieve a clear identification of the impact of price volatility on consumer purchase behavior. However, notice that even if such a test were possible, the real exposure cannot be captured in a single event, rather it accumulates over time and this stock would be heavily driven by the user’s browsing intensity. That is, a simple A/B test can be understood as an encouragement design (Holland, 1988) that indirectly induces variation in exposure to varied prices.

We try to emulate an “ideal” experiment with observational data using randomization inference. The idea is similar to how Donnelly et al. (2019) use surrounding weekly in-store price changes to estimate price elasticities for groceries. However, in our case, there are complex dependencies in the data since consumers visit multiple times to purchase multiple products. Furthermore, price changes occur at many different times. Independently of any particular user’s visit, prices for products are changing due to different factors such as inventory or competitive pressures. Hence, the actual price a consumer sees conditional on visit depends upon her time of visit. If she were to visit a few hours earlier or later, then she may end up seeing a different price for the same product. We use the idea that the

actual visit time of a particular user is as-good-as-random to generate counterfactual distributions of price exposure and algorithmic pricing stock. Subsequently, we use these counterfactual exposures to causally pin down the effect of exposure to multiple prices on consumer behavior.

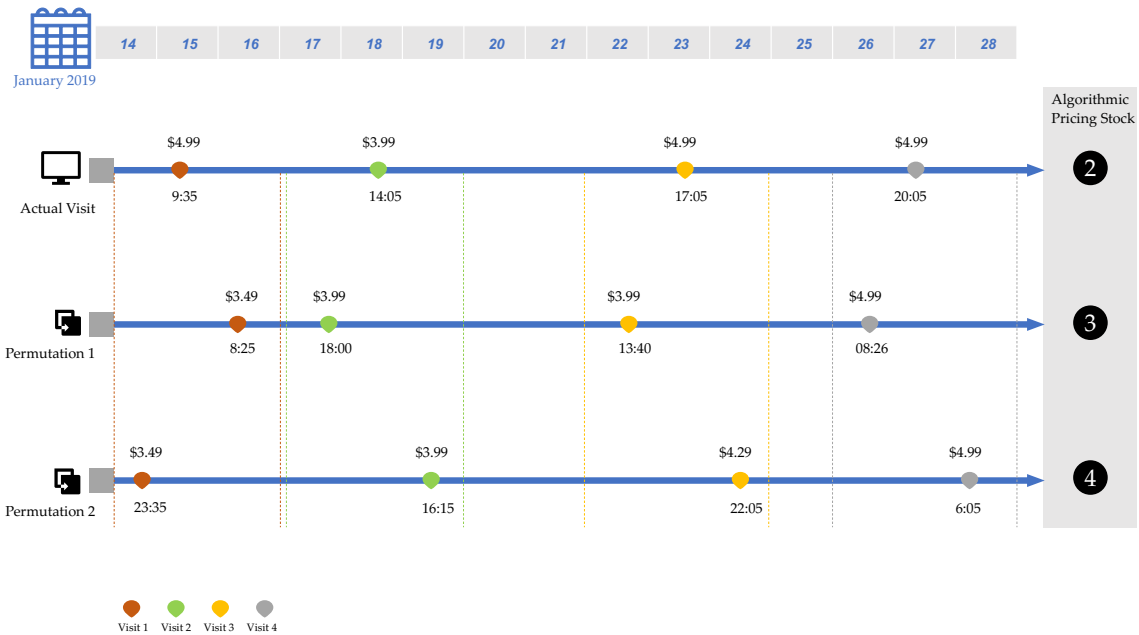


Figure 2.5: Permuted user visits by re-drawing user visit times within a ± 48 hour window of actual visit time.

Figure 2.5 helps build intuition behind this procedure. For simplicity, consider a single user who visits a single product four times during a two-week period. The dates and times the user visits are shown in the first row of the figure. Across these four visits, the user sees two distinct prices, and hence the total algorithmic pricing stock is 2. Consider the first visit of the user on Jan-15 at 9:15 AM. Suppose that instead of 9:15 AM on Jan-15, she visited the product at 8:25 AM on Jan-16. Then, all else equal, she would have been exposed to three prices, and her algorithmic pricing stock would be 3.

We generalize this idea and shuffle all visits for a user within ± 48 hours of the original visit, keeping the total number of visits and the products visited each time fixed. Since the prices are fluctuating independently of this user's visit and not personalized, she could get exposed to a different price in each permutation.

Consequently, each permutation of a user’s visit to the retailer’s website creates a counterfactual price exposure and algorithmic pricing stock. The collection of all permutations for a user gives a vector of counterfactual price exposures and algo-pricing stocks. We use these counterfactual exposures and algo-pricing stocks for identification and inference. To give a sense of how the permuted assignments look like, Figure 2.6 shows the probability distribution of observed algo-pricing stock and permuted algo-pricing stock averaged across permutations. Overall we find that randomizing user’s visit time does expose them to different prices for the same product. The distribution of observed algo-pricing stock exposure is shown in Figure I.10 in the Appendix.

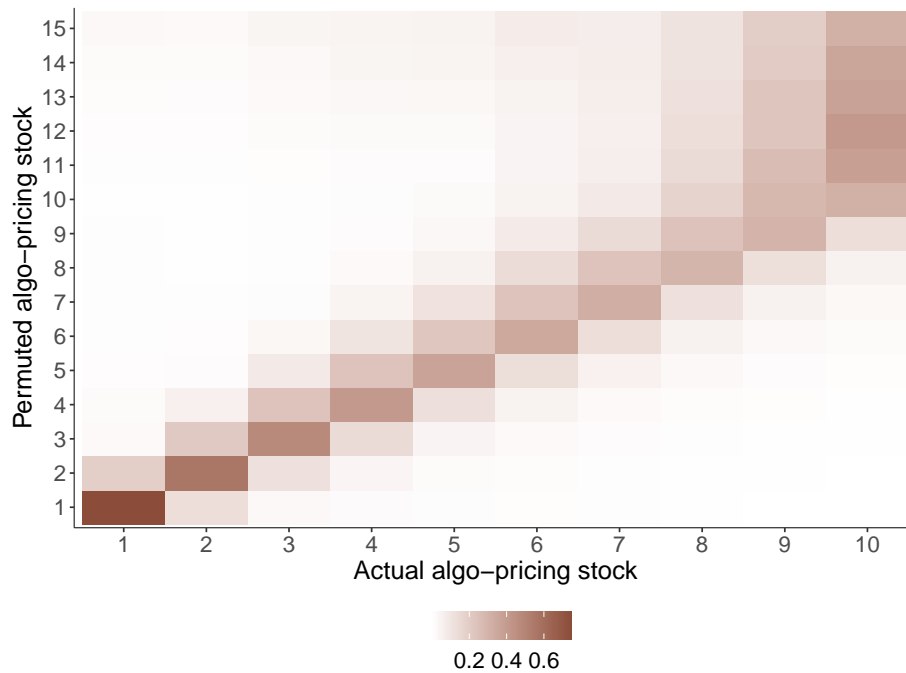


Figure 2.6: Conditional Distribution of Permuted Algo-Pricing Stock Given Observed Algo-Pricing Stock

It is important to test the validity of the assumption that consumer visit times are random. We present analysis similar to Donnelly et al. (2019) in which we compare the coefficients obtained by running the model on the actual data with the coefficients obtained by running the model on the permuted data. Since our data has more complex dependencies at the user and product level, we estimate the full

fixed effects model specified in Equation 2.14 for each counterfactual exposure. For each permutation, we calculate the algo-pricing stock and user history variables over a 30-day period.

$$\begin{aligned}
\mathbb{P}(Y_{ijt} = y) &= \frac{e^{-\lambda_{ijt}} \lambda_{ijt}^q}{q!}, \quad q = 0, 1, 2, \dots \\
\log(\lambda_{ijt}) &= \beta_0 + \beta_1 \log(P_{ijt}) + \beta_2 A_{it} + \beta_3 \log(P_{ijt}) \times A_{it} + \\
&\quad + \overline{\log(P_{ijt})} + \overline{A_{it}} + \overline{\log(P_{ijt}) \times A_{it}} \\
&\quad + \Gamma_i + \mu_j + \tau_t + \epsilon_{ijt}
\end{aligned} \tag{2.14}$$

where $\overline{\log(P_{ijt})}$ is the average log price for a user, product, visit combination across all permutations, $\overline{A_{it}}$ is the average algo-pricing stock across all permutations, and $\overline{\log(P_{ijt}) \times A_{it}}$ is the average value of the interaction between log price and algo-pricing stock across all permutations. If the treatments have no effect, then on average across permutations, we would expect their effect to be centered at zero. The results from this test are shown in Figure 2.7 where we plot the distribution of z -stats from each permuted regression. We do indeed find that the treatments affect the outcome. The red line in each panel is the observed z -stat from the actual regression run using the observed exposures.

Estimation Results

Finally, for inference, we estimate Equation 2.14 using the observed values of log price, algorithmic pricing stock, and their interaction while still controlling for the mean value of the independent variables across permutations. The results are shown in Table 2.8. As with the permutations, we use a 30-day window to calculate the observed algorithmic pricing stock. We estimate the model for a random sample of 30,000 consumers. As with the instrumental variables approach, we find that exposure to multiple prices for the same product does make the consumers substantially more price sensitive.

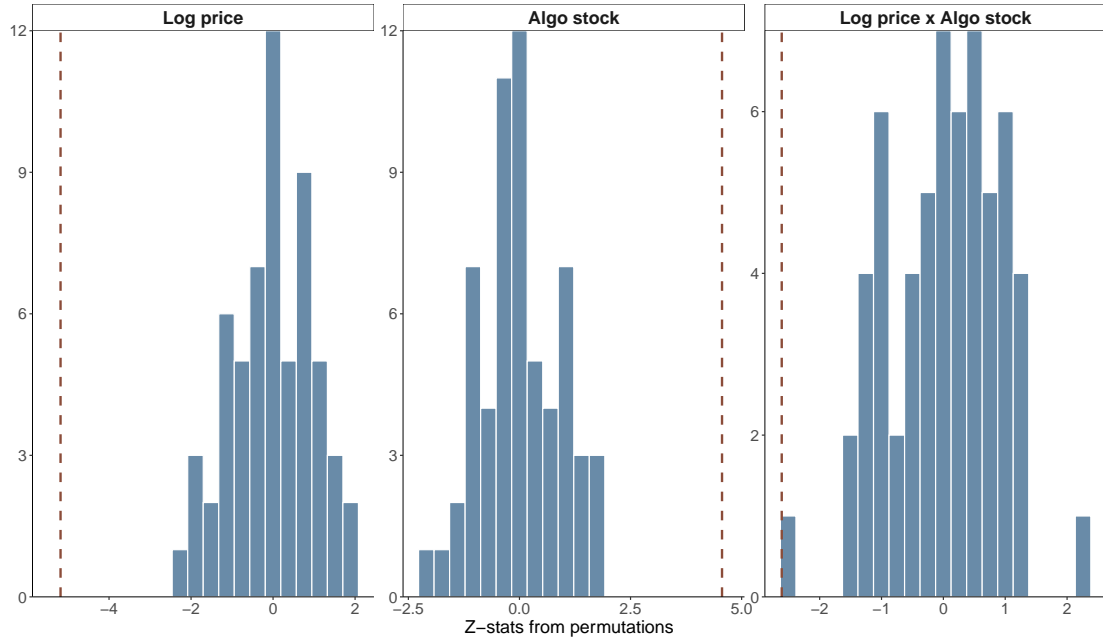


Figure 2.7: Placebo Tests using Counterfactual Price Exposure and Algo-Pricing Stock

Note: The blue histogram shows the distribution of z -scores obtained by running Model 2.14 on permuted data. The dashed red line is the z -score from the regression using observed data. All permutations and models are estimated on a random sample of 30,000 customers and the algo-pricing stock is calculated over a 30-day period.

2.6 Lab Experiments

The analysis in Section 2.5 shows that exposure to more number of unique prices increases price sensitivity. We are mindful that in a field setting it is not possible to exert full control over the unobserved reasons consumers decide to visit the online grocery platform. To address this limitation, we conduct a laboratory experiment to test the effect of price volatility in a controlled environment. The experimental design is simple: we ask participants to simulate purchase decisions, i.e. participants must click how many units they intend to buy each period. Participants are randomly assigned to two treatment conditions: stable prices and algorithmic prices. To the best of the authors' knowledge, there are no studies in the literature that have explored algorithmic pricing and price sensitivity in a laboratory experiment.

Table 2.8: User-level Price Sensitivity Estimates using Randomization Inference

Dependent Variable:	Units purchased
<i>Variables</i>	
Log price	-1.06*** (0.205)
Log price \times Algo stock	-0.110*** (0.042)
Algo stock	0.425*** (0.093)
Mean price across perm.	0.034 (0.203)
Mean algo stock across perm.	-0.126 (0.081)
Mean price \times Algo stock across perm.	-0.003 (0.039)
<i>Fixed-effects</i>	
User, Product, and Year-Week	
<i>Fit statistics</i>	
Observations	505,425
Pseudo R ²	0.236
Log-Likelihood	-192,254.8
<i>Two-way (User & Product) standard-errors in parentheses</i>	
<i>Signif. Codes: ***: 0.01, **: 0.05, *: 0.1</i>	

Notes: The table shows the estimates from Model 2.14 estimated using the observed values of price, algo-pricing stock, and their interaction. The algo-pricing stock is computed over a 30-day period and the model is estimated for a random sample of 30,000 users.

2.6.1 Experiment Design

The online shopping simulation involves a single product (e.g., a Nutella or Nestle’s Cocoa), it lasts 12 periods, and the price might fluctuate from period to period. Participants decide how many units to buy (from 0 to 5) each period. They receive a budget at the beginning which is automatically adjusted based on the units that they have bought so far. Answers are sequential, i.e. participants answer period 1, then period 2, etc. Responses are incentivized by offering a bonus payout that depends on the total units bought and total savings, i.e. users that buy more (less) units when the price is low (high) receive a larger payout. Finally, many series of 12 prices are simulated based on the *real* data according to two pricing regimes: stable pricing and algorithmic pricing. In particular, the price sequences across conditions have a very similar average price (but different volatility and frequency of price changes).

For example, four periods under stable pricing might be (\$5.98, \$5.98, \$5.76,

\$5.76); while the same periods under algorithmic pricing might be (\$6.01, \$5.88, \$5.63, \$5.91). Importantly, the price variation closely resembles the online grocer’s strategy of stable pricing and algorithmic pricing, respectively. In periods of algorithmic pricing prices fluctuate frequently and often in tiny amounts, and in periods of stable pricing, prices are fairly stationary with small infrequent jumps.

We run two lab experiments – one on Amazon Mechanical Turk⁴ and the other with MBA students from a large European university. Among the MBA students, the study covers 139 distinct users (self-reported average age 29.8 and 68% male), 52% randomly assigned to stable pricing, and 48% randomly assigned to algorithmic pricing. Additional methodological details and robustness specifications are reported in Appendix J. Reassuringly, we find similar results from the lab experiment on MTurk (Appendix J).

2.6.2 Lab Experiment Results

We estimate the following model:

$$\log(Y_{it}) = \beta_0 + \beta_1 \log(P_{it}) + \beta_2 Algo_i + \beta_3 \log(P_{it}) \times Algo_i + \epsilon_{it} \quad (2.15)$$

where Y_{it} and $P_{i,t}$ denote the quantity and price, respectively; $Algo_i$ is an indicator variable that takes value 1 if user i was assigned to the online shopping simulation with algorithmic prices (and 0 otherwise). As with the models before, we are interested in β_3 , the coefficient on the interaction.

The results from the lab experiment are presented in Table 2.9. Consistent with the findings in Sections 2.4 and 2.5, participants exhibit a more price sensitive demand when exposed to prices with high volatility.

⁴MTurk has become a standard platform to conduct lab experiments in pricing (e.g., Wadhwa and Zhang (2015))

Table 2.9: Price Elasticity – Lab Experiment with MBA Students

Dependent Variables: Model:	Log units		Units
	Gaussian (1)	Gaussian (2)	Poisson (3)
<i>Variables</i>			
(Intercept)	1.70*** (0.101)	1.23*** (0.135)	1.38*** (0.162)
Elasticity	-0.651*** (0.059)	-0.371*** (0.076)	-0.722*** (0.091)
Algo		0.863*** (0.195)	1.61*** (0.266)
Algo x Elasticity		-0.516*** (0.114)	-0.962*** (0.160)
<i>Fit statistics</i>			
R ²	0.03901	0.04511	
Log-Likelihood	-1,479.5	-1,474.2	-2,632.7

One-way (User) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

2.6.3 Price Salience and Recall

We now revisit the conceptual framework discussed in Section 2.2 to conceptualize potential mechanisms underlying the effects. A critical role in that framework is price salience: variation in prices shifts attention to the price attribute thereby making consumers more price sensitive.

We make progress showing that salience indeed is a relevant mechanism. After the 12-period simulated shopping trip, we show participants in the lab experiment a different product (Oreo) for 5 seconds. Along with the standard product packaging, the image includes the product’s price. We then ask participants in both groups to recall the price and size of the Oreo. We hypothesize that, if algorithmic pricing makes prices more salient, then a larger proportion of users in that treatment condition will be able to better recall the price of Oreo’s. We operationalize salience as recall, consistent with the tradition in behavioral sciences, economics, and marketing. See, for example, Alba and Chattopadhyay (1986); Kissler et al. (2007); Finkelstein (2009); Kroft et al. (2013); Gaspelin et al. (2015).

We test whether the proportion of correct responses is higher in the algorithmic pricing condition. Thus, we first classify a participant with a correct response if their answer was within \$0.05 of the correct price. More formally, consider a

participant i who answers X_i ; we define recall R_i as follows:

$$R_i = \begin{cases} 1, & \text{if } |X_i - P^*| \leq 0.05 \\ 0, & \text{otherwise} \end{cases} \quad (2.16)$$

where P^* is the correct prices of Oreo's. Therefore, the proportion of correct respondents in each treatment condition is:

$$p_{algo} = \frac{1}{n_{algo}} \sum_i R_i$$

$$p_{stable} = \frac{1}{n_{stable}} \sum_i R_i$$

We then test the null whether $p_{algo} \leq p_{stable}$ using a two-sample proportions test. The results are shown in Figure 2.8. We find that participants in the algorithmic pricing condition are more likely to correctly recall the price of Oreo ($p - value < 0.03$). The proportions test is robust to using different cut-offs, e.g. $\{\$0.02, \$0.03, \$0.04\}$. Furthermore, we find no difference between the two conditions when asked to recall the size of the Oreo's package. Taken together, this evidence supports that a process through which high volatility increases price sensitivity is price salience. We emphasize that further research should examine the existence of additional mechanisms.

2.7 Discussion

Industry practitioners often express the concern of "lagging behind" in the race of adopting state-of-the-art pricing technology. Improvements in this technology typically include some form of machine-based tool to set or update prices. In this paper, we show that a stylized distinctive feature of algorithmic pricing, namely

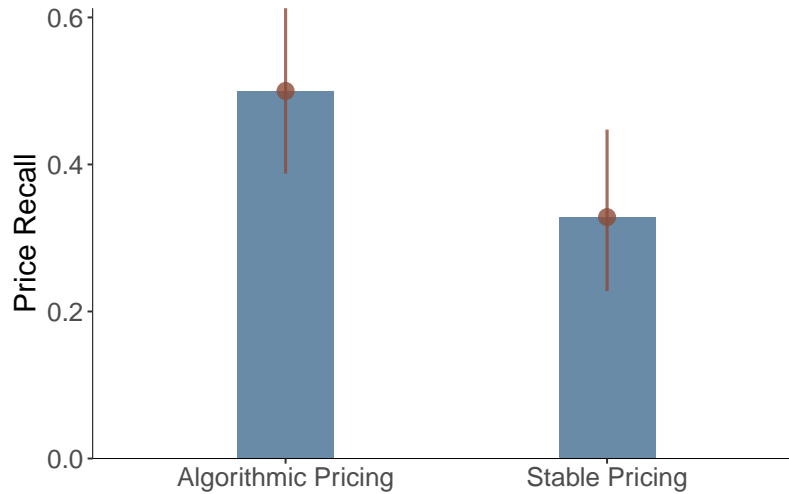


Figure 2.8: Testing Price Salience using Recall

Notes: Blue bars indicate the proportion of respondents who were able to recall the price of a second product (Oreo’s) within \$0.05 of the correct price. Red confidence bands are the 95% interval for the point estimate.

price volatility, modifies consumer behavior. We show, in the field and in the lab, that price volatility makes demand more price sensitive. Once again, the effect is identified within-consumers: greater exposure to algorithmic pricing makes a given consumer more price-sensitive. Our findings also indicate that a key mechanism through which this happens is salience in the price attribute.

This set of results encourages scholars to further connect technical innovations and consumer behavior. In light of the role of price salience, consumers are not indifferent to how online retailers change prices. Therefore, methodological improvements in the back-end (e.g., speed of optimization, high-dimensional inputs, price matching) are not sufficient in isolation; their connection to front-end user experiences is extremely relevant. Even if some form of machine-based pricing tool is profitable, it may trigger or shift salience to prices. And it is not obvious that all retailers benefit from price salience.

Thinking more broadly, which businesses want their customers to become more price-sensitive? The answer is probably very few. Perhaps price aggregator platforms or everyday low prices (EDLP) retailers might stand to benefit, but in general, businesses would like to avoid this side effect of algorithmic pricing. Said

differently, while a retailer would not want to shut down algorithmic pricing, it would like to avoid the negative effect on price sensitivity. Our work suggests that price algorithms could be improved by accounting for consumer-level sensitivity to price volatility—a dimension often overlooked.

Finally, there are promising paths to further expand the analysis presented here. While we focus on understanding one process mechanism, namely salience, we are aware that it does not exclude other processes. Future work could explore the moderating role of price knowledge (Dickson and Sawyer, 1990b), price fairness (Xia et al., 2004; Anderson and Simester, 2008; Allender et al., 2021), fairness to machine algorithms (Lee, 2018), limited memory (Chen et al., 2010), or the formation of price cues.

It is also interesting to differentiate short-term reactions from long-term implications. Algorithmic pricing technology is fairly new and even specialized AI vendors are continually experimenting and updating their models. Studying the long-term impact of this new-age pricing technology on consumer behavior and market structure will help inform both business strategies and regulatory policies. Another important dimension, which is beyond the scope of the current paper, to consider is competition. Firms are not using pricing algorithms in isolation and a key input to these algorithms is competitor price. Characterizing the equilibrium effects between consumers and firms when multiple players in the market adopt algorithmic pricing is a promising, albeit challenging, avenue to pursue.

Bibliography

- Agrawal, A., Gans, J., and Goldfarb, A. (2018). *Prediction machines: the simple economics of artificial intelligence*. Harvard Business Press.
- Airbnb (2017). What's smart about smart pricing? <https://blog.airbnb.com/smart-pricing/>.
- Alba, J. W. and Chattopadhyay, A. (1986). Salience effects in brand recall. *Journal of Marketing Research*, 23(4):363–369.
- Allender, W. J., Liaukonyte, J., Nasser, S., and Richards, T. J. (2021). Price fairness and strategic obfuscation. *Marketing Science*, 40(1):122–146.
- Amaldoss, W. and He, C. (2018). Reference-dependent utility, product variety, and price competition. *Management Science*, 64(9):4302–4316.
- Anderson, E. T. and Simester, D. I. (2003). Effects of 9 price endings on retail sales: Evidence from field experiments. *Quantitative Marketing and Economics*, 1(1):93–110.
- Anderson, E. T. and Simester, D. I. (2004). Long-run effects of promotion depth on new versus established customers: Three field studies. *Marketing Science*, 23(1):4–20.
- Anderson, E. T. and Simester, D. I. (2008). Research note—does demand fall when customers perceive that prices are unfair? the case of premium pricing for large sizes. *Marketing Science*, 27(3):492–500.
- Anderson, E. T. and Simester, D. I. (2010). Price stickiness and customer antagonism. *The Quarterly Journal of Economics*, 125(2):729–765.
- André, Q., Reinholtz, N., and De Langhe, B. (2021). Can consumers learn price dispersion? evidence for dispersion spillover across categories. *Journal of Consumer Research*.
- Aparicio, D. and Rigobon, R. (2020). Quantum prices. *NBER Working Paper No. 26646*.
- Asker, J., Fershtman, C., and Pakes, A. (2021). Artificial intelligence and pricing: The impact of algorithm design. *NBER Working Paper No. w28535*.
- Assad, S., Clark, R., Ershov, D., and Xu, L. (2020). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *CESifo Working Paper 8521*.
- Becker, G. S. and Murphy, K. M. (1993). A simple theory of advertising as a good or bad. *The Quarterly Journal of Economics*, 108(4):941–964.

- Bertini, M. and Wathieu, L. (2008). Research note—attention arousal through price partitioning. *Marketing Science*, 27(2):236–246.
- Blake, T., Moshary, S., Sweeney, K., and Tadelis, S. (2021). Price salience and product choice. *Marketing Science*, forthcoming.
- Bordalo, P., Gennaioli, N., and Shleifer, A. (2013). Salience and consumer choice. *Journal of Political Economy*, 121(5):803–843.
- Bordalo, P., Gennaioli, N., and Shleifer, A. (2020). Memory, attention, and choice. *The Quarterly Journal of Economics*, 135(3):1399–1442.
- Brown, Z. and MacKay, A. (2019). Competition in pricing algorithms. *Available at SSRN 3485024*.
- Brynjolfsson, E. and McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company.
- Business Insider (2018). Amazon changes prices on its products about every 10 minutes — here’s how and why they do it. <https://www.businessinsider.com/amazon-price-changes-2018-8>.
- Busse, M. R., Lacetera, N., Pope, D. G., Silva-Risso, J., and Sydnor, J. R. (2013). Estimating the effect of salience in wholesale and retail car markets. *American Economic Review*, 103(3):575–79.
- Busse, M. R., Pope, D. G., Pope, J. C., and Silva-Risso, J. (2015). The psychological effect of weather on car purchases. *The Quarterly Journal of Economics*, 130(1):371–414.
- Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2019). Artificial intelligence, algorithmic pricing and collusion. *Available at SSRN 3304991*.
- Caplin, A. and Dean, M. (2015). Revealed preference, rational inattention, and costly information acquisition. *American Economic Review*, 105(7):2183–2203.
- Chen, L., Mislove, A., and Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1339–1349.
- Chen, M. K. (2016). Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 455–455.
- Chen, Y., Iyer, G., and Pazgal, A. (2010). Limited memory, categorization, and competition. *Marketing Science*, 29(4):650–670.
- Chetty, R., Looney, A., and Kroft, K. (2009). Salience and taxation: Theory and evidence. *American Economic Review*, 99(4):1145–77.

- Cian, L., Krishna, A., and Elder, R. S. (2015). A sign of things to come: behavioral change through dynamic iconography. *Journal of Consumer Research*, 41(6):1426–1446.
- Cohen, P., Hahn, R., Hall, J., Levitt, S., and Metcalfe, R. (2016). Using big data to estimate consumer surplus: The case of uber. *NBER*.
- DellaVigna, S. and Gentzkow, M. (2019). Uniform pricing in us retail chains. *The Quarterly Journal of Economics*, 134(4):2011–2084.
- Dholakia, U. M. (2015). Everyone hates uber’s surge pricing – here’s how to fix it. *Harvard Business Review*.
- Dickson, P. R. and Sawyer, A. G. (1990a). The price knowledge and search of supermarket shoppers. *The Journal of Marketing*, pages 42–53.
- Dickson, P. R. and Sawyer, A. G. (1990b). The price knowledge and search of supermarket shoppers. *Journal of Marketing*, 54(3):42–53.
- Donnelly, R., Ruiz, F. R., Blei, D., and Athey, S. (2019). Counterfactual inference for consumer choice across many product categories.
- Dorfman, R. and Steiner, P. O. (1954). Optimal advertising and optimal quality. *The American Economic Review*, 44(5):826–836.
- Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology: General*, 113(4):501.
- Elberg, A., Gardete, P. M., Macera, R., and Noton, C. (2019). Dynamic effects of price promotions: Field evidence, consumer search, and supply-side implications. *Quantitative Marketing and Economics*, 17(1):1–58.
- Ellison, G. and Ellison, S. F. (2009). Search, obfuscation, and price elasticities on the internet. *Econometrica*, 77(2):427–452.
- Erdem, T., Keane, M. P., and Sun, B. (2008). The impact of advertising on consumer price sensitivity in experience goods markets. *Quantitative Marketing and Economics*, 6(2):139–176.
- Finkelstein, A. (2009). E-ztax: Tax salience and tax rates. *The Quarterly Journal of Economics*, 124(3):969–1010.
- Fisher, M., Gallino, S., and Li, J. (2018). Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. *Management science*, 64(6):2496–2514.
- Folkes, V. and Matta, S. (2004). The effect of package shape on consumers’ judgments of product volume: attention as a mental contaminant. *Journal of Consumer Research*, 31(2):390–401.

- Ford, M. (2015). *Rise of the Robots: Technology and the Threat of a Jobless Future*. Basic Books.
- Gaspelin, N., Leonard, C. J., and Luck, S. J. (2015). Direct evidence for active suppression of salient-but-irrelevant sensory inputs. *Psychological science*, 26(11):1740–1750.
- Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multi-level/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.
- Hansen, K. T., Misra, K., and Pai, M. M. (2021). Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*.
- Hastings, J. S. and Shapiro, J. M. (2013). Fungibility and consumer choice: Evidence from commodity price shocks. *The Quarterly Journal of Economics*, 128(4):1449–1498.
- Haws, K. L. and Bearden, W. O. (2006). Dynamic pricing and consumer fairness perceptions. *Journal of Consumer Research*, 33(3):304–311.
- Hendel, I. and Nevo, A. (2004). Intertemporal substitution and storable products. *Journal of the European Economic Association*, 2(2-3):536–547.
- Hitsch, G. J., Hortacsu, A., and Lin, X. (2019). Prices and promotions in us retail markets: Evidence from big data. *NBER*.
- Hoch, S. J., Kim, B.-D., Montgomery, A. L., and Rossi, P. E. (1995). Determinants of store-level price elasticity. *Journal of marketing Research*, 32(1):17–29.
- Holland, P. W. (1988). Causal inference, path analysis, and recursive structural equations models. *Sociological Methodology*, pages 449–484.
- Jacoby, J. (1984). Perspectives on information overload. *Journal of Consumer Research*, 10(4):432–435.
- Jindal, P. and Aribarg, A. (2021). The importance of price beliefs in consumer search. *Journal of Marketing Research*, forthcoming.
- Jung, J., Kim, J.-H., Matejka, F., Sims, C. A., et al. (2019). Discrete actions in information-constrained decision problems. *The Review of Economic Studies*, 86(6):2643–2667.
- Kalyanaram, G. and Winer, R. S. (1995). Empirical generalizations from reference price research. *Marketing Science*, 14(3_supplement):G161–G169.
- Karmarkar, U. R., Shiv, B., and Knutson, B. (2015). Cost conscious? the neural and behavioral impact of price primacy on decision making. *Journal of Marketing Research*, 52(4):467–481.

- Kissler, J., Herbert, C., Peyk, P., and Junghofer, M. (2007). Buzzwords: early cortical responses to emotional words during reading. *Psychological Science*, 18(6):475–480.
- Krider, R. E., Raghurir, P., and Krishna, A. (2001). Pizzas: π or square? psychophysical biases in area comparisons. *Marketing Science*, 20(4):405–425.
- Kroft, K., Lange, F., and Notowidigdo, M. J. (2013). Duration dependence and labor market conditions: Evidence from a field experiment. *The Quarterly Journal of Economics*, 128(3):1123–1167.
- Lee, M. K. (2018). Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1):2053951718756684.
- Lichtenstein, D. R., Ridgway, N. M., and Netemeyer, R. G. (1993). Price perceptions and consumer shopping behavior: a field study. *Journal of Marketing Research*, pages 234–245.
- Lilien, G. L., Kotler, P., and Moorthy, K. S. (1995). *Marketing models*. Prentice Hall.
- McAfee, R. P. and Te Velde, V. (2006). Dynamic pricing in the airline industry. *Handbook on economics and information systems*, 1:527–67.
- Mela, C. F., Gupta, S., and Lehmann, D. R. (1997). The long-term impact of promotion and advertising on consumer brand choice. *Journal of Marketing research*, 34(2):248–261.
- Miklós-Thal, J. and Tucker, C. (2019). Collusion by algorithm: Does better demand prediction facilitate coordination between sellers? *Management Science*, 65(4):1552–1561.
- Monroe, K. B. (1973). Buyers' subjective perceptions of price. *Journal of Marketing Research*, pages 70–80.
- Petrin, A. and Train, K. (2010). A control function approach to endogeneity in consumer choice models. *Journal of Marketing Research*, 47(1):3–13.
- Rotemberg, J. J. (2005). Customer anger at price increases, changes in the frequency of price adjustment and monetary policy. *Journal of Monetary Economics*, 52(4):829–852.
- Semenova, V., Goldman, M., Chernozhukov, V., and Taddy, M. (2017). Estimation and inference about heterogeneous treatment effects in high-dimensional dynamic panels. *arXiv e-prints*, pages arXiv–1712.
- Sethuraman, R., Tellis, G. J., and Briesch, R. A. (2011). How well does advertising work? generalizations from meta-analysis of brand advertising elasticities. *Journal of Marketing Research*, 48(3):457–471.

- Shapiro, B. T., Hitsch, G. J., and Tuchman, A. E. (2021). Tv advertising effectiveness and profitability: Generalizable results from 288 brands. *Working Paper*.
- Thomas, M., Simon, D. H., and Kadiyali, V. (2010). The price precision effect: Evidence from laboratory and market data. *Marketing Science*, 29(1):175–190.
- Townsend, C. and Kahn, B. E. (2014). The “visual preference heuristic”: The influence of visual versus verbal depiction on assortment processing, perceived variety, and choice overload. *Journal of Consumer Research*, 40(5):993–1015.
- Vanhuele, M. and Drèze, X. (2002). Measuring the price knowledge shoppers bring to the store. *Journal of Marketing*, 66(4):72–85.
- Wadhwa, M. and Zhang, K. (2015). This number just feels right: The impact of roundedness of price numbers on product evaluations. *Journal of Consumer Research*, 41(5):1172–1185.
- Washington Post (2015). How uber surge pricing really works. <https://www.washingtonpost.com/news/wonk/wp/2015/04/17/how-uber-surge-pricing-really-works/>.
- Weisstein, F. L., Monroe, K. B., and Kukar-Kinney, M. (2013). Effects of price framing on consumers’ perceptions of online dynamic pricing practices. *Journal of the Academy of Marketing Science*, 41(5):501–514.
- Xia, L., Monroe, K. B., and Cox, J. L. (2004). The price is unfair! a conceptual framework of price fairness perceptions. *Journal of Marketing*, 68(4):1–15.

Appendix

A Algorithmic Pricing Periods

Table A.1 shows additional descriptive statistics for periods (weeks) identified with and without algorithmic pricing. Overall, the statistics indicate that the measure of algorithmic pricing does capture periods in which the price of a product experienced intense variation.

Table A.1: Summary Statistics During Algorithmic and Non-Algorithmic Weeks

	Stable pricing	Algo. pricing
Observations	81,337	40,972
Std. price	0.08	0.33
Mean price	9.14	9.23
Min price	9.05	8.82
Max price	9.27	9.75
Weekly price changes	0.7717	3.536

Figure A.1 visualizes, in the form of a heat map, the percent of products across categories that experience algorithmic pricing over time. Reassuringly, there is variation in exposure to algorithmic pricing across products within and across categories, as well as throughout the time-series of the data.

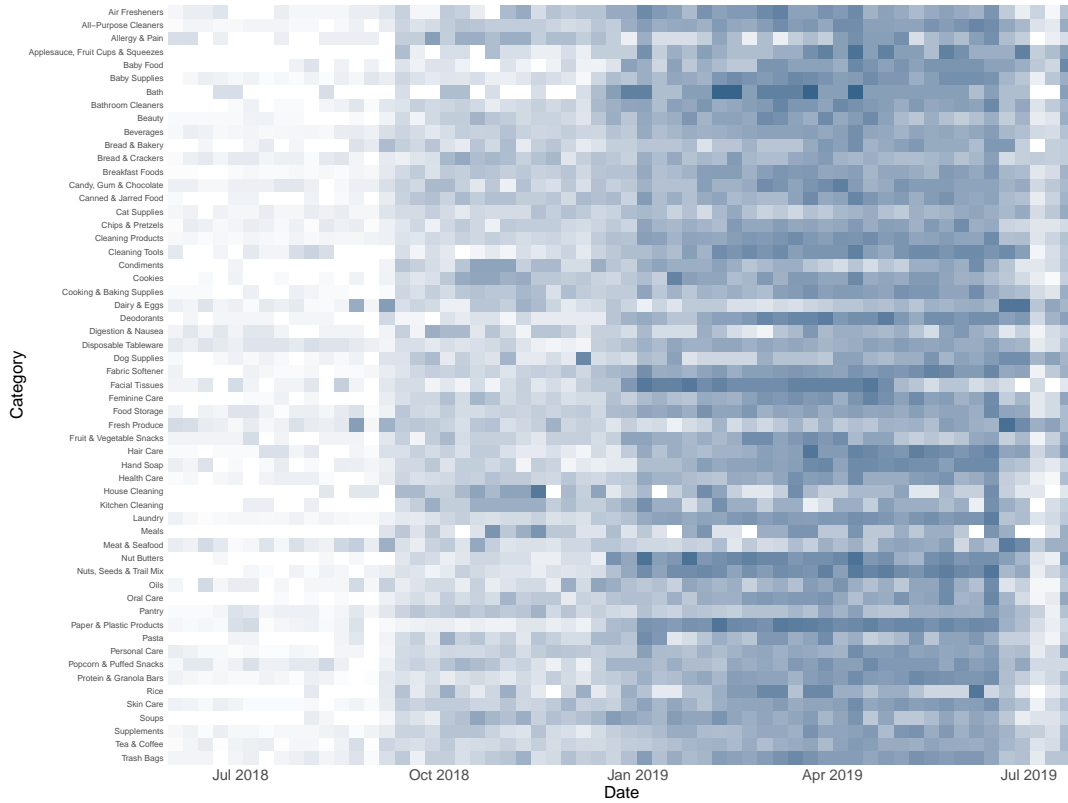
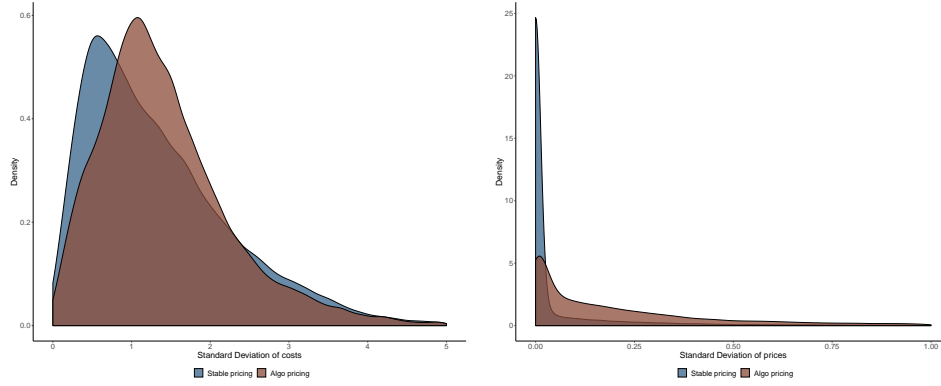


Figure A.1: Share of Products per Category under Algorithmic Pricing

Notes: Each cell is a category-week combination

If prices were dynamically updated according to a mark-up rule (e.g., $p_{it} = m * c_{it}$), changes in price (p_{it}) might be triggered by changes in cost (c_{it}). However, and interestingly, periods of algorithmic pricing do not appear to be driven by high-frequency changes in costs. Panels (a) and (b) of Figure A.2 show the distribution of the cost and price changes in algorithmic pricing weeks and in non algorithmic pricing weeks, respectively. Summary statistics are reported in Table A.2. Overall the evidence indicates, similar to Fisher et al. (2018), that algorithmic or dynamic pricing is not primarily driven by cost shifters.



(a) Variation in cost

(b) Variation in prices

Figure A.2: Costs and Price Changes during Algorithmic and Non-Algorithmic Pricing Periods

Table A.2: Variation in Costs and Prices during Algorithmic and Non-Algorithmic Pricing Periods

	Cost Std.	Price Std.
Stable pricing	1.419	0.078
Algorithmic pricing	1.476	0.379

B Statistical Tests of the Algorithmic Pricing Indicator

Table B.3 shows the results of a χ^2 test for the significance of the algorithmic pricing indicator, as defined in Section 2.3. We find that both the algorithmic pricing indicator and its interaction with price capture a statistically significant portion of the purchase variation.

Model	Log Lik.	χ^2	p.value
Only price	-112,205.29	-	-
Price + algorithmic pricing indicator	-112,074.71	256.95	< 0.001
Price + algorithmic pricing indicator + interaction	-112,004.51	138.06	< 0.001

Table B.3: χ^2 Test for Algorithmic Pricing

Additionally, Table B.4 shows the results of the ANOVA test for varying intercepts by product and by category. The results support the existence of individual differences in price elasticities across products and categories.

Model	Log Lik.	χ^2	p.value
Varying intercept by product	-118514.4	-	-
Varying intercept and slope by product	-114814.5	7399.65	< 0.001
Varying intercept and slope by category and product (nested)	-114565.3	498.44	< 0.001

Table B.4: ANOVA Test for Mixed Effects

C Definition of Algorithmic Pricing

Does more intensity in algorithmic pricing magnify the price sensitivity? Figure C.3 shows that that it does. More precisely, a more stringent definition of algorithmic pricing (i.e., a product-week pair is required to exhibit a more intense price variation to be classified as an algorithmic pricing week) increases the interaction with the price elasticity. The intensity is measured by the minimum number of unique prices in a given week.

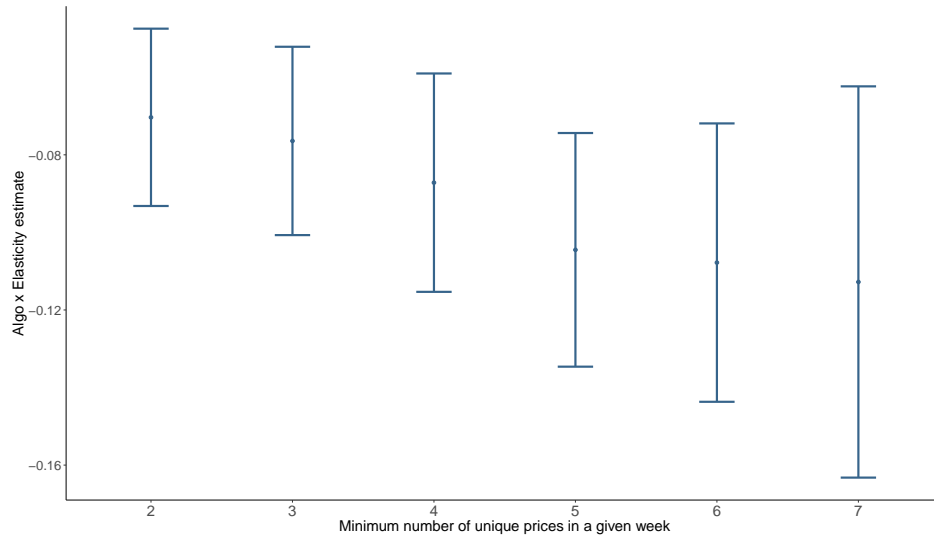


Figure C.3: Change in Price Elasticity with Increasing Variation in Prices

D Own-Price Elasticities

Figure D.4 shows the distribution of the own-price elasticities computed at the product-level. Price elasticities are restricted to those significant at the 10%. Panel (a) depicts the distribution using a separate linear regression for each product, and Panel (b) depicts the distribution using a multilevel model allowing the elasticity to vary by product.

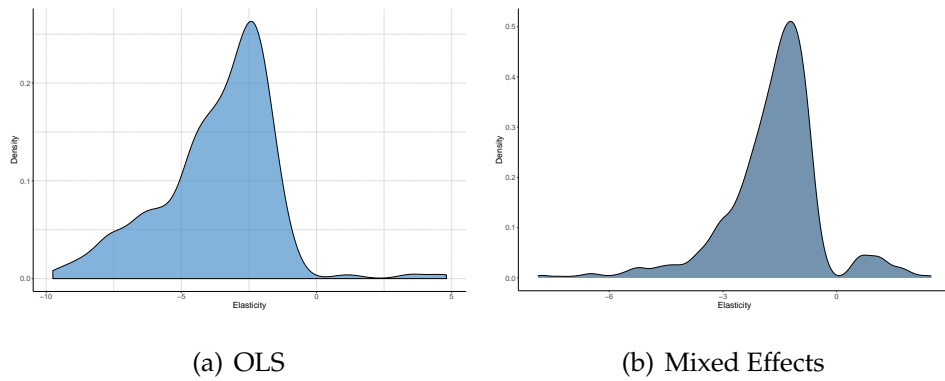


Figure D.4: Distribution of Own-Price Elasticities

Figure D.5 shows the distribution of the product-level own-price elasticities during algorithmic and stable pricing weeks.

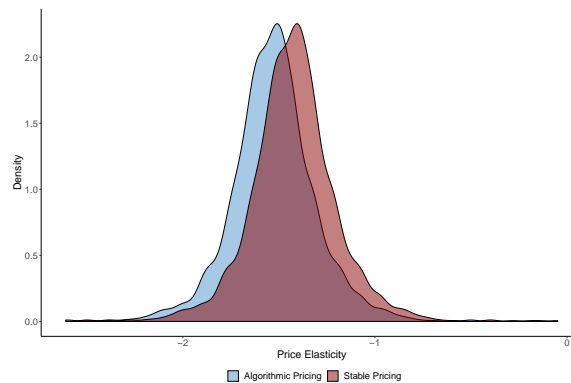


Figure D.5: Product-level distribution of price elasticities during stable and algorithmic pricing periods estimated using mixed-effects model

E Types of Products

We examine whether the effect of algorithmic pricing on price sensitivity varies across types of products. We consider three classifications of products: cheap and expensive, high-revenue and low-revenue, perishable or non-perishable. The results are shown in Figures E.6, E.7, and E.8, respectively. In all graphs, the dashed red line in the center is global average across categories.

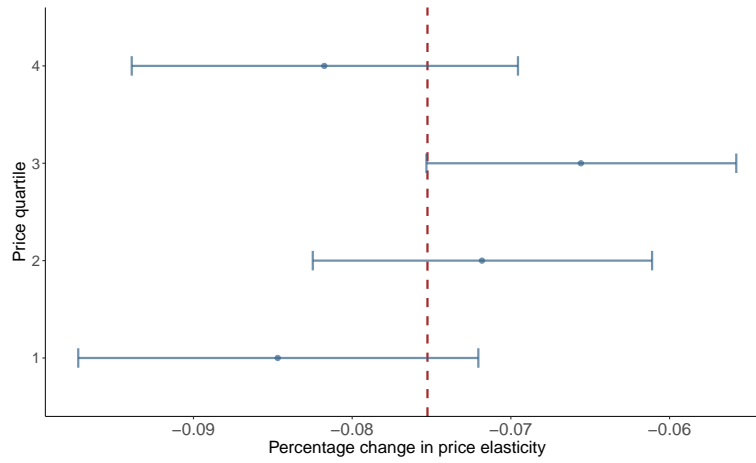


Figure E.6: Estimated price elasticity across products split by price quartile.

We use first 16 weeks of our sample to calculate the average price for each product and categorize them into quartiles based on average price. Elasticities are estimated using mixed-effects model similar to Equation 2.9 on the remaining sample.

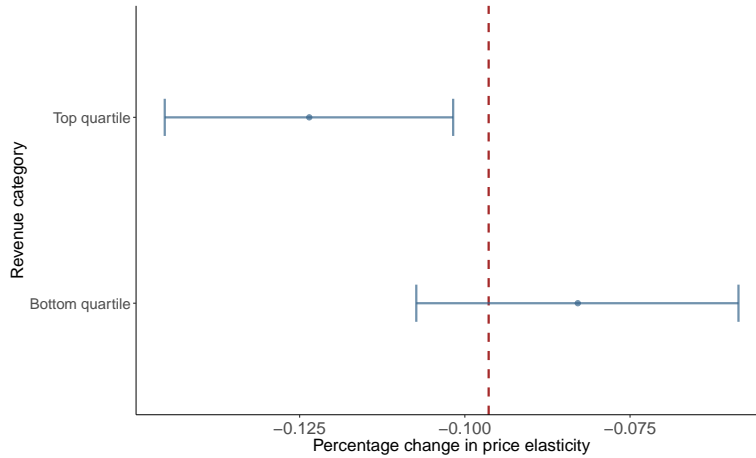


Figure E.7: Estimated price elasticity across less popular and more popular products.

We use first 16 weeks of our sample to calculate the total revenue for each product and categorize them into quartiles based on total revenue. Elasticities are estimated using mixed-effects model similar to Equation 2.9 on the remaining sample. We chose the top and bottom quartile for clarity. The middle two quartiles were centered at the global average.

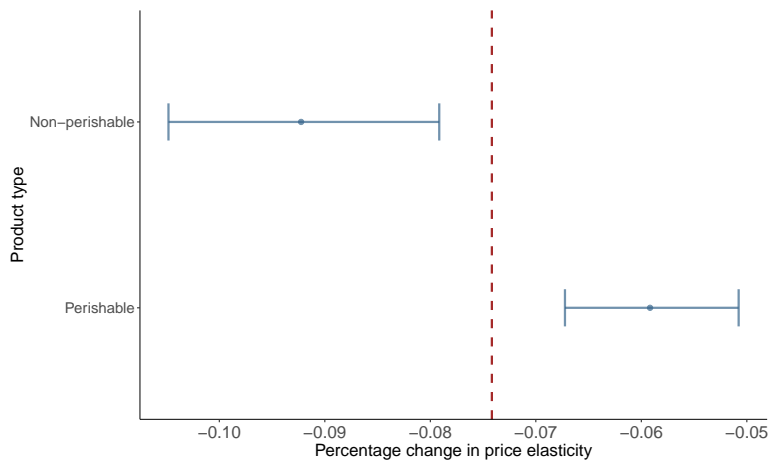


Figure E.8: Estimated price elasticity across perishable and non-perishable products.

Perishable products include categories such as dairy & eggs, meat & seafood, and fresh produce. Elasticities are estimated using mixed-effects model similar to Equation 2.9 on the remaining sample.

F Before-and-After Events

We estimate the effect of algorithmic pricing through a before-and-after event study, exploiting variation in the timestamp different products had their first algorithmic pricing week. Figure F.9 shows that there is considerable variation in the timing of adopting algorithmic pricing across products. See also Figure A.1 for heatmap split by category.

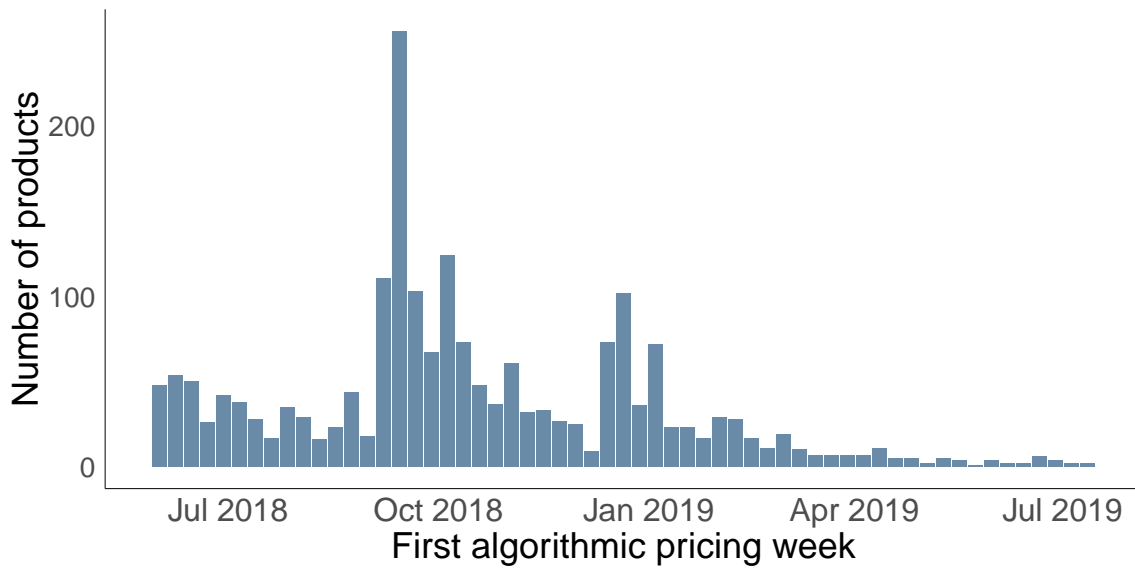


Figure F.9: Histogram of the First Event of Algorithmic Pricing for All 2,044 Products

Table F.5: Robustness checks for aggregate elasticity estimates before and after algorithmic pricing periods

Dependent Variable:	Log units	
	12 weeks	28 weeks
Model:	(1)	(2)
<i>Variables</i>		
Elasticity	-1.25*** (0.258)	-1.03** (0.389)
Post period	0.154*** (0.057)	0.238*** (0.084)
Elasticity \times Post period	-0.065*** (0.024)	-0.101*** (0.035)
<i>Fixed-effects</i>		
Product	Yes	Yes
Year week	Yes	Yes
<i>Fit statistics</i>		
Observations	19,607	7,652
R ²	0.644	0.687
Log-Likelihood	-14,557	-5,003
<i>Two-way (Product & Year week) standard-errors</i>		
<i>Signif. Codes: ***: 0.01, **: 0.05, *: 0.1</i>		

Table F.6: Robustness tests for user-level price sensitivity before and after algorithmic pricing periods

Dependent Variable:	Units				
	12 weeks	28 weeks	12 weeks	20 weeks	28 weeks
Model:	(1)	(2)	(3)	(4)	(5)
<i>Variables</i>					
Log price	-1.13*** (0.181)	-0.953*** (0.287)	-0.950*** (0.140)	-0.964*** (0.172)	-0.877*** (0.233)
Post period	0.187*** (0.052)	0.245*** (0.073)	0.184*** (0.052)	0.199*** (0.061)	0.240*** (0.073)
Log price × Post period	-0.049** (0.024)	-0.084*** (0.032)	-0.049** (0.024)	-0.070** (0.027)	-0.083** (0.032)
<i>Fixed-effects</i>					
User			Yes	Yes	Yes
Product			Yes	Yes	Yes
User-product	Yes	Yes			
Year week	Yes	Yes	Yes	Yes	Yes
<i>Fit statistics</i>					
Observations	60,404	27,264	99,008	65,894	38,784
Pseudo R ²	0.450	0.496	0.429	0.448	0.479
Log-Likelihood	-79,801	-36,816	-114,626	-78,589	-48,176

Three-way (User & Product & Year week) standard-errors in parentheses

Signif. Codes: ***: 0.01, **: 0.05, *: 0.1

G Aggregate Robustness Checks

We test whether the effect of algorithmic pricing on price sensitivity is robust to the definition of price sensitivity. In Table G.7 we repeat the models of Table 2.3 using the following definition of algorithmic pricing week:

A product is under algorithmic pricing in a given week if:

1. The total absolute change in price in that week is greater than the median total absolute change across all weeks; **AND**,
2. The number of changes in price in that week is greater than the median number of changes in prices across all weeks

Note that in both the Tables, 2.3 and G.7, the idea is to capture high frequency price variation in prices; what differs is how we quantify those changes. Table G.7's results show that our results are robust to a different definition of algorithmic pricing.

Table G.7: Elasticity estimates with multiple specifications and different algo-week indicator

Dependent Variables:	Log units	Units	Log units	
	Gaussian	Poisson	Gaussian	Ortho ML
Model:	(1)	(2)	(3)	(4)
<i>Variables</i>				
Elasticity	-1.518*** (0.100)	-1.581*** (0.135)	-1.371*** (0.128)	-2.533*** (0.044)
Algo	0.1051*** (0.0328)	0.094*** (0.034)		0.029** (0.014)
Elasticity × Algo	-0.031** (0.014)	-0.032** (0.015)		-0.273*** (0.063)
Post period			0.202*** (0.056)	
Elasticity × Post period			-0.062** (0.024)	
<i>Fixed-effects</i>				
Product	Yes	Yes	Yes	
Year week	Yes	Yes	Yes	
<i>Fit statistics</i>				
Observations	122,309	122,309	122,309	90,316
R ²	0.558		0.567	0.326
Log-Likelihood	-112,172.42	-592,395.11	-82,303.68	-49,620.81

Product & Year week standard-errors in parentheses

*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

H Consumer Level Robustness Checks

Table H.8: Robustness checks for user level elasticity estimates using OLS and TSLS

Dependent Variable:	Log units	
	(1)	(2)
Model:	OLS	TSLS
<i>Variables</i>		
Log price	-0.093*** (0.007)	-0.088*** (0.008)
Log price x Algo pricing stock	-0.001** (0.0006)	-0.003*** (0.001)
Algo pricing stock	0.006*** (0.001)	0.036*** (0.011)
# of sku per visit	-0.227*** (0.006)	-0.131*** (0.042)
# of total visits	-0.018*** (0.001)	-0.037*** (0.009)
# of prior purchases	-0.025*** (0.0007)	-0.026*** (0.0008)
<i>Fixed-effects</i>		
User	Yes	Yes
Product	Yes	Yes
Year week	Yes	Yes
<i>Fit statistics</i>		
Observations	4,621,411	4,621,411
Log-Likelihood	-292,530.2	-295,076.2
<i>Two-way (User & Product) standard-errors in parentheses</i>		
<i>Signif. Codes: ***: 0.01, **: 0.05, *: 0.1</i>		

Notes: The table shows OLS and two-stage least square estimates for the consumer-level model in Equation 2.13. Algorithmic pricing stock and time-varying history variables are calculated over a 30-day period. The dependent variable is the log number of units of a particular product, purchased by a user during a single visit.

Table H.9: Robustness checks for user level elasticity estimates using different number of days for historical data

Dependent Variable:	Units					
Model:	7 days		15 days		60 days	
	(1)	(2)	(3)	(4)	(5)	(6)
<i>Variables</i>						
Log price	-1.05*** (0.083)	-0.837*** (0.091)	-1.01*** (0.081)	-0.840*** (0.087)	-0.948*** (0.079)	-0.854*** (0.083)
Log price x Algo pricing stock	-0.074*** (0.008)	-0.203*** (0.018)	-0.079*** (0.007)	-0.170*** (0.013)	-0.063*** (0.005)	-0.104*** (0.010)
# of sku per visit	-1.74*** (0.071)	-0.472 (0.315)	-1.92*** (0.064)	-0.422 (0.319)	-2.78*** (0.074)	-1.39*** (0.363)
Log # of total visits	-0.150*** (0.015)	-0.434*** (0.071)	-0.142*** (0.013)	-0.455*** (0.066)	0.013 (0.014)	-0.285*** (0.076)
Log # of prior purchases - total	-0.265*** (0.010)	-0.279*** (0.010)	-0.248*** (0.008)	-0.268*** (0.009)	-0.366*** (0.010)	-0.386*** (0.011)
Algo pricing stock	0.269*** (0.022)	0.925*** (0.098)	0.241*** (0.017)	0.853*** (0.088)	0.182*** (0.014)	0.648*** (0.095)
First stage residual - 1		-0.711*** (0.097)		-0.667*** (0.087)		-0.517*** (0.094)
First stage residual - 2		0.153*** (0.020)		0.115*** (0.015)		0.063*** (0.011)
<i>Fixed-effects</i>						
User	Yes	Yes	Yes	Yes	Yes	Yes
Product	Yes	Yes	Yes	Yes	Yes	Yes
Year week	Yes	Yes	Yes	Yes	Yes	Yes
<i>Fit statistics</i>						
Observations	2,672,388	2,672,388	3,680,937	3,680,937	5,206,155	5,206,155
Pseudo R ²	0.235	0.235	0.231	0.231	0.228	0.228
Log-Likelihood	-1,029,052.5	-1,028,759.8	-1,443,524.4	-1,443,161.3	-2,086,407.0	-2,086,154.9

Two-way (User & Product) standard-errors in parentheses

*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Notes: The table shows two-stage control function estimates for the consumer-level model in Equation 2.13. Algorithmic pricing stock and time-varying history variables are calculated over 7-day, 15-day, and 60-day periods. The dependent variable in all models is the number of units of a particular product, purchased by a user during a single visit. All models control for user, product, and year-week fixed effects. Standard errors for columns 3 & 4 are estimated using clustered bootstrap to account for first-stage estimation error.

I Exposure to Algorithmic Pricing Stock

Figure I.10 shows the distribution of user level algorithmic pricing stock over a 30-day period.

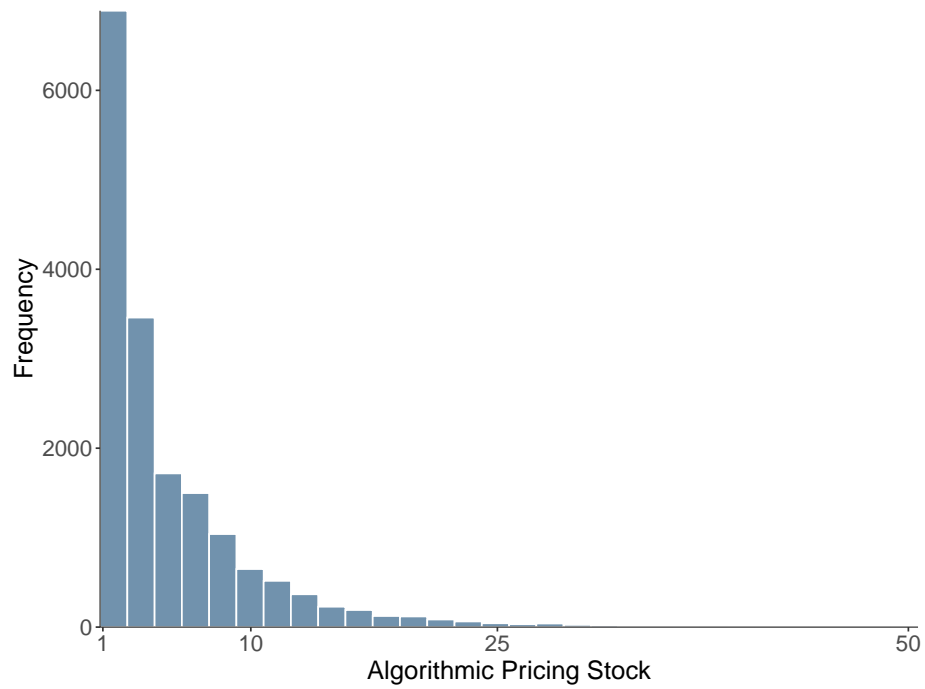


Figure I.10: Observed 30-day Algo-Pricing Stock Across Users

J Lab Experiment

We conduct two lab experiments – one on Amazon Mechanical Turk and the other with MBA students from a large European university. Participants are randomly assigned to two pricing regimes: stable pricing or algorithmic pricing. The set-up and task are similar in both studies, only the prices are re-drawn randomly. As described in Section 2.6, the algorithmic pricing condition is characterized by high frequency price changes (calibrated using real data from the online retailer). Figure J.11 shows one pair of price series used in the experiment (we used 4 pairs in total).

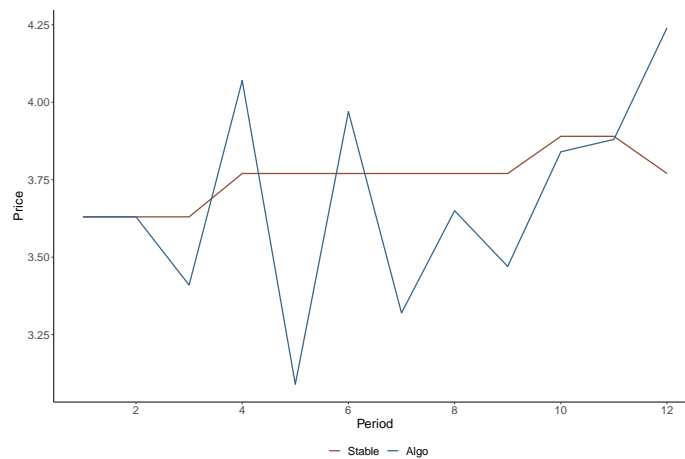


Figure J.11: Sampled price series for lab experiment

Note: A user was randomly assigned to either of these pricing conditions and then their purchase behavior was recorded.

J.1 MBA Students

Table J.10 shows the summary statistics across the two conditions. Important to note is that the average price is similar but the standard deviation of prices is much higher in the algorithmic pricing condition.

Table J.13 in the main text shows the results from the main model. Here, we do a robustness check by including user fixed effects in the estimation. In the lab experiment, since a given participant is only allocated to one of the conditions and hence the main effect of the pricing regime is not identified. Hence we run separate regressions for the stable pricing and algorithmic pricing users, while accounting

Table J.10: Summary Statistics from Lab Experiment with MBA Students

	Algo	Stable	<i>p</i>
Obs.	864	804	
Users	72	67	
Mean price	4.42	4.49	0.66
SD price	0.33	0.11	
Units purchased	14.9	14.2	0.08
	(3.66)	(4.04)	
Spend	64.7	62.6	0.20
	(9.52)	(9.54)	

The third column shows the p-value from a t-test testing the difference in means across the two conditions.

for user-level fixed effects. The results are presented in Table J.11 and, once again, show that price sensitivity is higher in the algorithmic pricing regime.

Table J.11: Robustness check for lab experiment with MBA students using fixed effects estimation

Dependent Variable:	Log units		
	Combined	Algo	Stable
Model:	(1)	(2)	(3)
<i>Variables</i>			
Elasticity	-3.23*** (0.259)	-3.68*** (0.269)	-1.25 (0.759)
<i>Fixed-effects</i>			
User	Yes	Yes	Yes
<i>Fit statistics</i>			
R ²	0.16332	0.25041	0.07662
Log-Likelihood	-1,364.0	-681.39	-673.43

*One-way (User) standard-errors in parentheses
Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

J.2 MTurk Experiment

Table J.12 shows the summary statistics across the two conditions from the MTurk experiment. Again the average prices are quite similar, only the standard deviation of price varies.

We repeat the exercises of estimating elasticity using two specifications for the MTurk lab experiment as well. The results are in Tables J.13 and J.14. Again, in both specifications we find that consumers in the algorithmic pricing condition

Table J.12: Summary Statistics from MTurk Lab Experiment

	Algo	Stable	<i>p</i>
Obs.	504	564	
Users	42	47	
Price	6.17	6.40	0.18
SD Price	2.83	1.21	
Units purchased	11.2 (3.66)	10.2 (4.04)	0.25
Spend	60.1 (8.1)	56.2 (9.71)	0.041

The third column shows the p-value from a t-test testing the difference in means across the two conditions. Standard errors in parentheses.

exhibit greater price sensitivity.

Table J.13: Price Elasticity – Lab Experiment

Dependent Variables:	Log units		Units
	Gaussian	Gaussian	Poisson
Model:	(1)	(2)	(3)
<i>Variables</i>			
(Intercept)	1.22*** (0.053)	1.12*** (0.085)	1.72*** (0.186)
Log price	-0.385*** (0.025)	-0.340*** (0.038)	-1.00*** (0.091)
Algo		0.207** (0.102)	0.500** (0.228)
Log price x Algo		-0.099** (0.047)	-0.244** (0.118)
<i>Fit statistics</i>			
R ²	0.064	0.066	-
Log-Likelihood	-815.50	-814.75	-1,384.3

*One-way (User) standard-errors in parentheses
Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Table J.14: Robustness check for lab experiment results using fixed effects estimation

Dependent Variable:	Log units		
	Combined (1)	Algo (2)	Stable (3)
<i>Variables</i>			
Elasticity	-1.51*** (0.292)	-1.54*** (0.338)	-1.38** (0.680)
<i>Fixed-effects</i>			
User	Yes	Yes	Yes
<i>Fit statistics</i>			
R ²	0.35	0.37	0.33
Log-Likelihood	-620.59	-295.86	-324.62

One-way (User) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Chapter 3

Search Augmented Choice Models and Recommendation Systems

Abstract

Choice models and recommendation systems are commonly used in online marketplaces to suggest relevant items (products in case of e-commerce, content in case of social media, and music/movies in case of entertainment platforms) to users. These systems include large-scale machine learning models that are trained on historical interaction data. For example, in case of online retail, recommender systems use historical purchases to learn consumer preferences and recommend products that consumers would like to buy in future. Similarly, econometric choice models of demand are typically designed to live in the purchase space, i.e., they are based on the set of products historically purchased. We augment these systems by including information from users' historical consideration sets. We first show how one can improve logit-type demand models using data from consideration sets. Subsequently, we enhance recommendation systems by flexibly incorporating granular consumer search data along with purchase data using a sequential deep learning-based approach. The search augmented recommendation system better captures consumers' latent preferences, more accurately predicts future actions, and substantially outperforms strong baselines. Finally we show that these gains are distributed across the entire spectrum of consumers and not concentrated among a small subset of high usage consumers.

3.1 Introduction

Choice models and recommendations systems are ubiquitous tools for learning consumer preferences and predicting future consumption. These models are typically estimated using historical transaction/consumption data where one observes previous choices made by the consumer. Common examples include logit choice models using scanner data (Guadagni and Little, 2008), demand models estimated using hierarchical bayes (Allenby and Rossi, 1998), and cross-category purchase models using modern machine learning methods (Donnelly et al., 2021). These models have proven immensely useful in deepening our understanding of consumer behaviour, decoding the underlying decision process, and subsequently generating prescriptive guidelines on how to better serve consumers.

A limitation of these models is that they all live in the purchase space, i.e., they are estimated and evaluated on products that consumers have historically purchased. However, extant research has shown that consideration sets are an important factor that eventually influence choice (Hauser and Wernerfelt, 1990). Given the nature of the data that these choice models work with (most commonly scanner data), observing the consideration set is quite difficult. Often to get to the consideration set scanner data need to be enhanced using consumer surveys which are neither scalable nor economical. To get around the limitation of not observing the consideration set, typical choice models make the strong assumption that all the products in the assortment (within the same category) that the user did not eventually choose, were part of the user's consideration set. For instance, if a user is shopping for coffee and purchases Starbucks, the choice modeler assumes that all other brands that were available at the time, were part of the consumer's consideration set, irrespective of whether the consumer actually considered them or not.

Aside from choice models, many online marketplaces use recommender systems to learn consumer preferences and predict future demand. These models are heavily used in industry across a range of platforms such as e-commerce, on-

line travel, media streaming, and online news. Most recommendation systems are based on collaborative filtering in which consumer preferences are learned through similarity within users and within items. The similarity is estimated by generating a low-rank approximation of large sparse interaction matrix. In e-commerce, this matrix includes the set of products that the user has purchased in the past. Here again, most of these models are trained on the purchase space and do not account for the set of products users considered but did not purchase.

In this study, we extend the scope of choice models and recommendation systems by including information from users' consideration sets, i.e., the products that consumer consider but do not eventually buy. We partner with an online retailer in the US and use granular clickstream in which we observe all consumer activity. Specifically, we observe the products that consumers visit by navigating to their product detailed page and the products that consumers buy. We classify the former as a user's consideration set and the latter as their purchase basket.

With this granular data, we augment choice models and recommendation systems by including information for all the products that users visit but do not buy. This allows us to better learn consumer preferences and more accurately predict future purchases. Using multiple econometric and machine learning models, we show the information from consideration sets is valuable and robust to different model choices. Subsequently, we design a scalable sequential deep learning framework that flexibly accounts for information from historical consideration sets. We allow for cross-category and cross-product pooling of information. We learn consumer preference weights for products by optimizing for a personalized pairwise ranking objective, the Bayesian Personalized Ranking (BPR) loss. Here again we find that information from consideration sets help drive substantial gains in model performance. Moreover, these gains are not concentrated among a small set of heavy users but rather distributed across the entire spectrum.

3.2 Relevant literature

This work connects to the literature on choice models, consideration sets, recommendation systems, and applications of deep learning in marketing.

Literature on choice models is deep and wide. Starting with [Guadagni and Little \(2008\)](#), who build a logit demand model of brand choice using scanner data, research has evolved both in substantive scope and applied methods ([Winer, 1986](#); [Tellis, 1988](#); [Kamakura and Russell, 1989](#); [Mela et al., 1997](#)). [Allenby and Rossi \(1998\)](#) introduce a hierarchical Bayesian model to estimate consumer preferences while accounting for consumer heterogeneity. More recently, machine learning has gained popularity in estimating scalable choice models. For instance, [Dew et al. \(2020\)](#) use Gaussian Processes, a Bayesian non-parametric method, to learn dynamics in consumer preferences over time. [Ruiz et al. \(2017\)](#) and [Donnelly et al. \(2021\)](#) use methods from probabilistic matrix factorization to learn consumer preferences for products across multiple categories. We build on this work by incorporating information consideration sets in demand models.

Our work is also closely related to the research on consideration sets. Part of our underlying motivation dwells from the idea that consideration sets are a “real” phenomena that strongly influence eventual consumption. [Hauser and Wernerfelt \(1990\)](#) make the distinction between consideration sets and consumption clear based on the net utility of evaluating another brand. [Roberts and Lattin \(1991\)](#) estimate a model for consideration set composition on ready-to-eat cereals and compare their two-stage model of consideration and choice against the popular one-stage choice model. They find that the two-stage model more accurately predicts eventual consumption. [Hauser et al. \(2010\)](#) investigate how consumers select products that they will include into their consideration set using cognitively simple decision rules. [Dzyabura and Hauser \(2011\)](#) develop a method for active machine learning that adaptively selects questions to maximize information about consumers’ decision rules. We approach the consideration set literature from an empirical standpoint and ask how we can effectively use consideration sets at scale

to more accurately learn consumer preferences?

Recommendation systems are a well-studied topic in both marketing and computer science. Typically, the objective in these papers is to design recommendation systems so that the platform can better learn user preferences. For example, Ansari et al. (2018) build a supervised topic model for recommending movies, Jacobs et al. (2016) use LDA to model purchase baskets in retail, and Lu et al. (2016) build a computer vision-based garment recommender for in-store recommendations. Dzyabura and Hauser (2019) look at the problem differently and suggest designs for recommendation systems when consumers themselves don't really know their preferences. They suggest a system that encourages consumers to search products with diverse attribute levels.

Finally, deep learning is being increasingly used for solving many core marketing problems. For example, Dhillon and Aral (2021) develop a custom deep learning architecture to model dynamic user consumption patterns for online news. Chen et al. (2020) and Gabel and Timoshenko (2020) use the idea of product embeddings to learn consumer preferences for offline grocery retail. More broadly, Timoshenko and Hauser (2019) and Chakraborty et al. (2022) use deep learning on online reviews to understand customer needs and score attribute sentiments respectively.

On the recommender systems and deep learning front, we contribute to the literature two novel ideas – 1) a flexible way of incorporating consideration sets, and 2) a new architecture that accounts for sequential shopping patterns and scales well to large assortments.

3.3 Data

We use clickstream data from a large online retailer in the U.S. Our data span 9 months of activity from October 2018 to June 2019 and include everything the consumers do on the retailer's website. The retailer's assortment includes general purpose products such as groceries, pet supplies, household supplies, baby prod-

ucts, health & beauty products etc.

Importantly, we observe all the products that consumers view and those that consumers buy along with the price, category, and activity timestamp. This allows us to observe each consumer’s consideration set and her purchase basket. From the clickstream logs, we sample data for 30,000 consumers and the top 2,000 most popular products by revenue. The observation counts from our working sample are given in Table 3.1. The most popular categories by revenue are shown in the Appendix in Figure A.1.

Table 3.1: Observation counts from working sample

# Obs	1, 292, 297
# Users	30, 000
# Products	2, 044
# Categories	177
Departments	Grocery, Household, Pet Baby, Health & Beauty

An important element in our analysis is the classification of products based on consumer actions. Specifically, we classify the assortment for each user-session into three parts: the products that the user buys, the products that the user views but does not buy, and the products that the user does not view. Subsequently, we label these three disjoint sets as the user’s purchase basket, consideration set, and not considered products. Table 3.2 shows the summary statistics for purchase baskets and consideration sets at the session level.

Table 3.2: Consideration sets and purchase baskets

	Consideration Sets	Purchase Baskets
# Products	8.44 (10.07)	2.57 (2.02)
# Categories	3.47 (3.54)	2.31 (1.65)
Price	10.79 (8.63)	10.58 (9.11)

Note 1: Mean values for each variable along with standard deviations in parentheses.

A point worth highlighting here is the decision rule to classify a product as being viewed or not. For instance, if a consumer searches for coffee and the prod-

uct listing page presents 20 products on the first results page, then which of these are taken as products viewed. Here, we use the most stringent definition of products viewed and take only those products to be part of the consideration set whose detailed page the consumer clicked on. An illustrative consumer journey with purchase baskets and consideration sets over time is shown in Figure 3.1. The example highlights the first interaction the user had on the retailer’s website. It also highlights sessions when the user only viewed certain products but did not purchase anything. Our granular clickstream records these interactions and our model is designed to leverage this data efficiently.



Figure 3.1: Illustrative consumer journey showing purchase baskets and consideration sets at multiple time points

3.4 Consideration sets and discrete choice models

In the first set of analysis, we augment typical discrete choice models with data from consumers’ consideration sets. Typical discrete choice models estimate user preferences for products within a particular category based on historical purchases. To estimate the model, they assume that all the products not purchased were part

of the consideration set. Mechanically, for the regression, products purchased have an outcome value 1 and all the products (within the same category) have outcome value 0. One can then estimate the parameters using maximum likelihood.

We start with the question: what if we observe the consideration set? Many on-line retailers, like the one we partner with, track the entire consumer journey, logging products that consumers view but do not buy. This allows us to approximate the set of products the user considered before making her decision. Observing the consideration set, or a noisy approximation of it, can then, in theory, help in better estimating user preferences and demand. We explore the idea of leveraging consideration sets with two strategies – 1) improving the measurement of the outcome (Y) variable, and 2) enhancing the feature set (X).

For the first case, we estimate a discrete-choice demand model for the coffee category in two ways – a) including everything in the assortment that was not purchased as 0, 2) including only those products that were viewed but not purchased, i.e., the consideration set, as 0. The model specification is shown in Equation 3.1. The model is estimated at a user-session level, and Y_{ijt} takes the value 1 only if the user i purchased product j at time t . The parameter estimates and in-sample fit statistics are shown in Table A.1.

$$Y_{ijt} = \beta_0 + \beta_1 \log(P_{jt}) + \gamma_i + \mu_j + \tau_t + \epsilon_{ijt} \quad (3.1)$$

We first evaluate the model using the U^2 measure as prescribed in Hauser (1978). U^2 essentially measures amount of reduction in uncertainty due to a model. As prescribed in Hauser (1978), we use the null model as all the product choices being equally likely, and compare it with the model in which product choices are equally likely within the consideration set of each user. That is, for the null model, we have $p_{ij} = \frac{1}{J}$, and for the model being evaluated, we have $p_{ij} = \frac{1}{J_i} \forall J_i \in C_i$, where J_i is the total number of products and C_i is the consideration set for user i . The results are shown in the second column of Table 3.4. The results show a U^2 of

0.4, which implies that only using the consideration set explains about 40% of the total uncertainty in final choice.

Furthermore, we investigate the out-of-sample fit for these models on future purchases. It is important to note that both models have the same specification and the same number of cases where $Y_{ijt} = 1$; the only difference is in the cases where $Y_{ijt} = 0$. Consequently, we evaluate the models using metrics that focus on getting the positive outcomes right. We use two metrics for this: *F1-score*, which is the harmonic mean of precision (the positive predictive value) and recall (the true positive rate), and *AUC*, which measures the probability that a randomly sampled positive cases is ranked higher than a randomly sampled negative case. The results are shown in Table 3.4. We find that on both measures, the choice model that incorporates users' consideration set performs substantially better.

Table 3.3: Performance of discrete-choice model in the coffee category

	Uncertainty	F1-score	AUC
Typical choice model	2.30	0.11	0.62
Choice model with consideration set	0.94	0.21	0.71
Gain/Uncertainty explained	$U^2 = 0.40$	(+90.0%)	(+14.5%)

Note 1: Both choice models are estimated on data using the coffee category only. Typical choice model is a logistic regression where the dependent variable is 1 if the user purchased a product during a shopping session and all the other coffee products are labelled 0. The choice model with consideration set is also a logistic regression where the dependent variable is 1 if the user purchases the product in a given session. The products from the user's consideration set are taken as 0. The first evaluation criterion is in-sample U^2 from Hauser (1978). The latter two are out-of-sample.

We explore the second strategy of incorporating consideration sets in choice models by enhancing the feature set in a standard machine learning framework. Specifically, we construct features related to the user's visitation patterns over time that indicate how frequently a product was considered but not purchased. We first explore the predictive value if these features in a regression framework. Specifically, we estimate the following model:

$$\log(Y_{ijt}) = \beta_0 + \beta_1 \log(P_{jt}) + \beta_2 \log\left(\sum_T Y_{ijt-1}\right) + \beta_3 \log\left(\sum_T S_{ijt-1}\right) + \gamma_i + \mu_j + \tau_t + \epsilon_{ijt} \quad (3.2)$$

where Y_{ijt} is the number of units of product j purchased by user i at time t . P_{jt} is the price of product j at time t . $\sum_T Y_{ijt-1}$ is the number of units of j bought by i in the past T days. We vary T to be 7, 15, 30, and 60 days. $\sum_T S_{ijt-1}$ is the number of times i visited the product detail page of j in the past T days but did not purchase it, i.e., it is the number of times j was in i 's consideration set. γ_i, μ_j, τ_t are user, product, and time fixed effects. The model is estimated at a user-session level.

The goal of this exercise is to understand how much of an effect S , the consideration set, has after controlling for previous purchases, and user and product fixed effects. Accounting for fixed effects here is important since we don't want to pick up the effect of users' latent tendency to search more, and hence having a high S value. Analogously, we want to remove the tendency of certain products to get searched more. If this effect is significant and substantial, then it is important that online marketplaces effectively leverage this information to better target and serve their customers.

We first show the effect of S on future Y using a binscatter plot presented in Figure 3.2. We partial out user and product fixed effects and then regress partialled-out Y on partialled-out S . We see that, even after controlling for user and product fixed effects, consideration set significantly predicts future purchases.

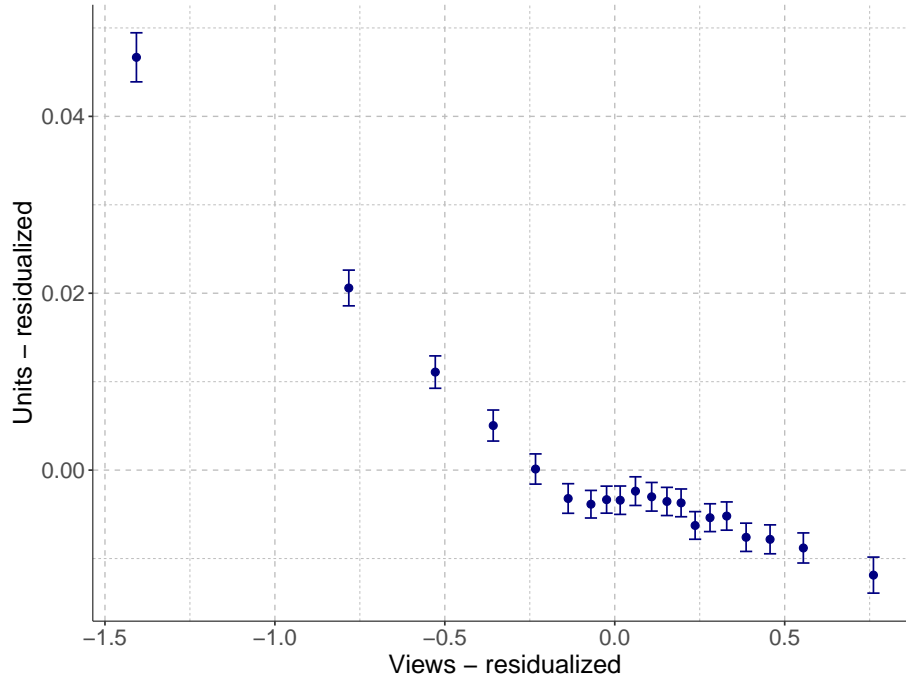


Figure 3.2: Binscatter plot of (residualized) future purchases on (residualized) past consideration sets

We extend this analysis by estimating the full model shown in Equation 3.2, which controls for historical purchases. Accounting for historical purchases is common in choice models estimated using scanner data. Our goal is to investigate whether consideration sets can provide information over and above historical purchases. The results are shown in Figure 3.3. The points and bars show the estimate of β_3 and 95% confidence intervals from Model 3.2. Each point is from a separate regression where S is calculated over different time periods - 7 days, 15 days, 30, days, and 60 days. The graph shows that even after controlling for historical purchases, historical consideration sets are predictive of future purchases.

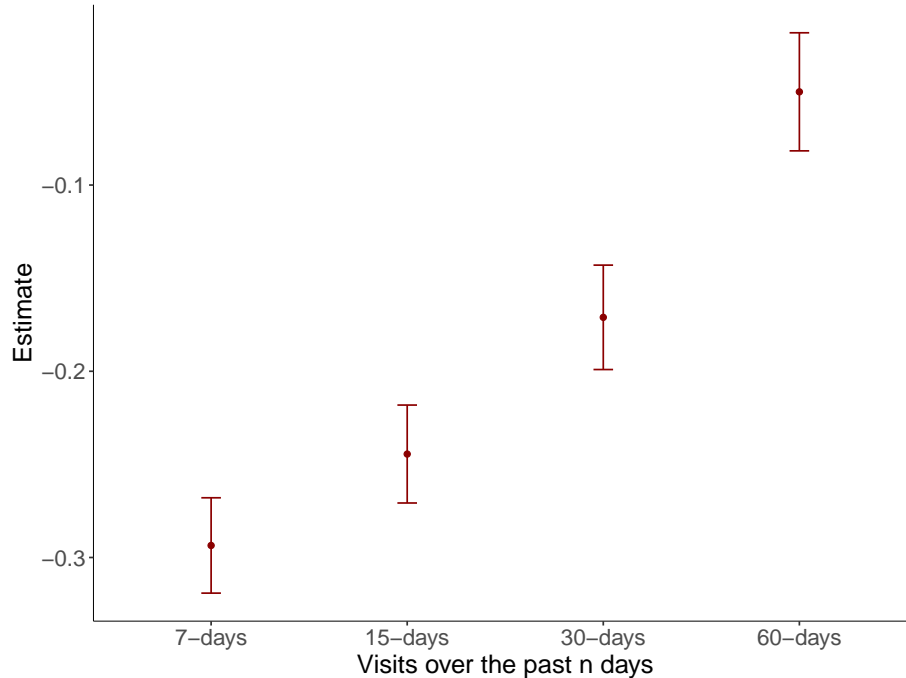


Figure 3.3: Estimates and 95% confidence intervals for the number of past visits. The outcome variable is the number of units of a product purchased during a user-session. The standard errors are clustered by user and product.

3.4.1 Machine Learning models

In our second set of analysis, we use two classes of machine learning models – 1) penalized regression and 2) boosted trees to further investigate the predictive impact of consideration sets on future purchases. For both classes of models, we predict the set of products purchased at time t for each user with historical data (purchases and consideration sets) up to time $t - 1$. In addition, we also include variables such as price, product category, hour of day, and day of week.

To evaluate the model, we keep the last visit for each user aside as held-out test data. For hyper-parameter tuning we randomly split the training data into 90% data for fitting and 10% data for tuning. Penalized regression is fit using Lasso and boosted trees using XGBoost (Chen and Guestrin, 2016). We evaluate the models using F1-score and AUC on the test data. The results are shown in Table 3.4. We see that for both the model classes, including the consideration set substantially improves the performance. It is important to note that these gains come even after

including the historical purchases in the model.

Table 3.4: Performance of machine learning models in predicting future purchases

	F1-score	AUC
Previous purchase (Binary indicator)	0.12	0.5
Lasso (Only purchases)	0.29	0.601
Lasso (Purchases + consideration)	0.32	0.645
Boosted Trees (Only purchases)	0.32	0.581
Boosted Trees (Purchases + consideration)	0.36	0.671
	(+12.5%)	(+15.4%)

Note 1: Out of sample performance of machine learning models in predicting future purchases. The number in parenthesis are the percentage increase in performance metrics of models with purchase and consideration history over the models that only include purchase history.

We go a step further and plot the most important features in the XGBoost models. Feature importance plots, like the one shown in Figure 3.4 show the relative importance of individual features in predicting the outcome, after controlling for all other variable in the model. Figure 3.4 shows that, even after accounting for price and historical purchases, the variable that is most influential in predicting future purchases is the number of times the product was part of the consideration set.

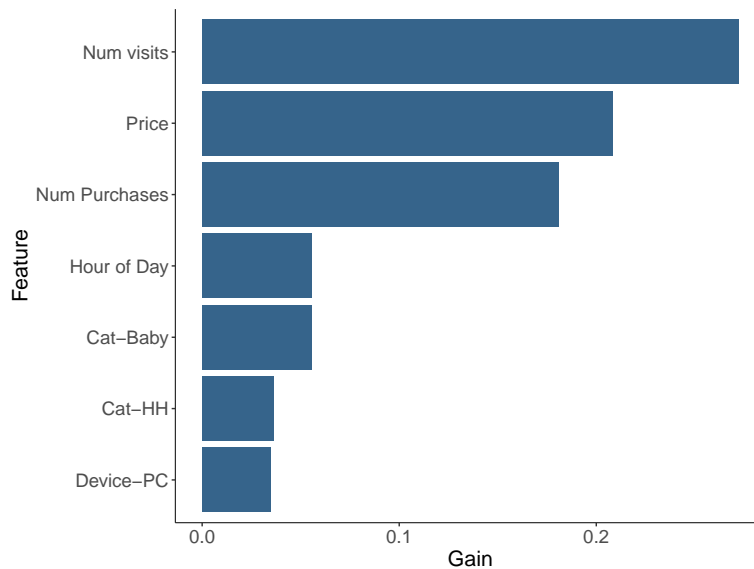


Figure 3.4: XGBoost feature importance for predicting future purchases

3.5 Deep learning based recommender system

We now develop a scalable sequential deep learning framework that flexibly accounts for historical purchases and consideration sets. The analysis in the previous section, while insightful, was constrained in a number of ways. For example, it only used historical purchases and considerations of the same product to predict future purchases, completely ignoring cross-product or cross-category relations. So, for instance, if a user purchases *coffee* in the previous session, the models don't take that information into account while predicting the likelihood of purchasing *creamers* today.

We expand the scope the analysis from the previous section by developing a sequential deep learning based recommender system that flexibly accounts for historical cross-product purchases and considerations, easily accommodates other high-dimensional sparse features such as product category and user zip code, and scales well to large data sizes. Furthermore, as recommendation systems are typically used to generate user-specific rankings of products, we directly optimize for ranking using Bayesian Personalized Ranking.

Bayesian Personalized Ranking is a pairwise personalized ranking loss commonly used in recommendation systems with implicit feedback, i.e., in cases where positive examples are observed (clicks or purchases) but negative examples are not. Negative examples need to be first generated, through sampling or some other pre-defined procedure, and then fed to the model to optimize the loss.

3.5.1 Architecture

Our model architecture is shown in Figure 3.5. For a given user i , the model combines user features one-hot encoded user-id, with pooled historical purchases and historical consideration sets. Historical purchases included one-hot encoded product ids that were purchased in the previous session. The product ids are passed through a 64-dim embedding layer. Since the number of purchases vary by session, we max-pool the embeddings to get a 64-dim representation of the previ-

ous visit's purchase basket. Similarly, all the products that were considered but not purchased are passed through a separate embedding layer of 64-dim and then max-pooled to get a 64-dim representation of the entire consideration set. Both representations are then concatenated and passed through a fully-connected network (FCN) head. Each layer in the FCN has a \tanh activation function. A dropout layer with 0.3 probability is added for regularization.

The user representation and the historical user state (combination of historical purchase basket and consideration set) are multiplied to get an updated user-state representation. This representation is then used to optimize for the Bayesian Personalized Ranking (BPR) loss using items that were actually purchased as positive cases and randomly sampled items as negative cases. All sets of embeddings are then jointly trained using backpropagation with an Adam optimizer and BPR loss.

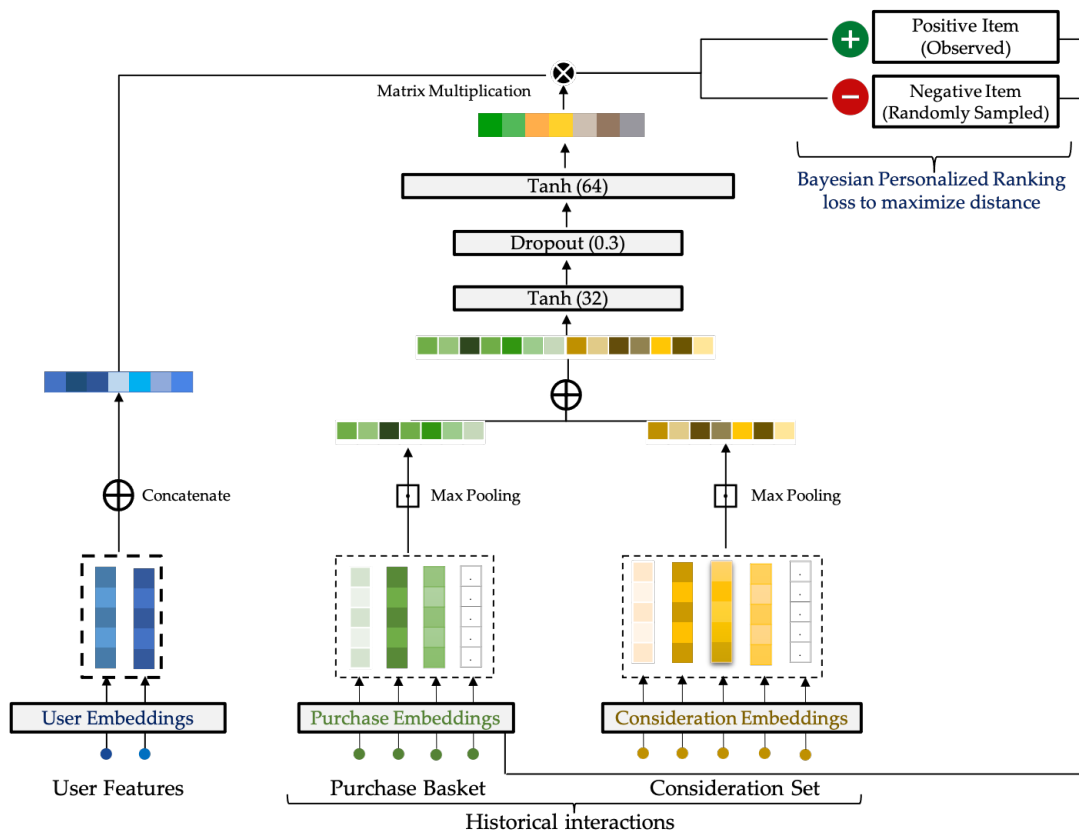


Figure 3.5: Sequential deep learning architecture to flexibly predict future purchases

3.5.2 Performance

We evaluate the model on data from the last visit of each user, which is not used to fit the model. We look at three metrics typically used for evaluating recommender systems, Hit Rate, Mean Recall, and Mean Average Precision. All the metrics are calculated by generating a rank ordered list of top-10 items the model thinks the user is likely to purchase in the last session.

In addition to our sequential deep learning model, we also test a few baselines starting with random predictions, most popular predictions, matrix factorization (MF), and matrix factorization optimized for BPR loss. For both sets of matrix factorization based recommender systems, we estimate the model with and without consideration set and highlight the gains from including consideration sets. Across all the different models, we see that including consideration set data substantially improves model performance on all three metrics.

Table 3.5: Performance of deep learning based recommender system

	Hit-Rate	MAP	Mean Recall
Random Baseline	0.48	0.12	0.12
Popular Baseline	7.39	2.28	2.27
MF	30.32	15.74	15.59
(Only purchases)			
MF	36.98	19.48	19.32
(Purchases + consideration)	(+21.9%)	(+23.7%)	(+23.9%)
MF: BPR Loss	36.11	19.36	19.23
(Only purchases)			
MF: BPR Loss	41.19	23.60	23.46
(Purchases + consideration)	(+14.0%)	(+21.9%)	(+21.9%)
Sequential DNN: BPR Loss	37.73	21.85	21.74
(Only purchases)			
Sequential DNN: BPR Loss	47.67	27.66	27.48
(Purchases + consideration)	(+27.0%)	(+26.5%)	(+26.4%)

Note 1: Metrics are calculated using top-10 recommendations generated by the model for the last visit of each user in the sample.

We further investigate the source of these gains across users and products. In

Figure 3.6, we plot the performance (Recall @ 10) of the sequential model with and without consideration sets across quintiles of users. The quintiles are created based on users' purchase history in the training data, i.e., users in the 5th quintile include the top 20% users in terms of revenue on the retailer's platform. Ex-ante, one might worry that using the consideration sets generates benefits for only those users for whom we have a lot of history, which tend to be the ones in the 5th quintile. However, Figure 3.6 shows that this is not the case. In fact, the gains are distributed across the entire spectrum of consumers.

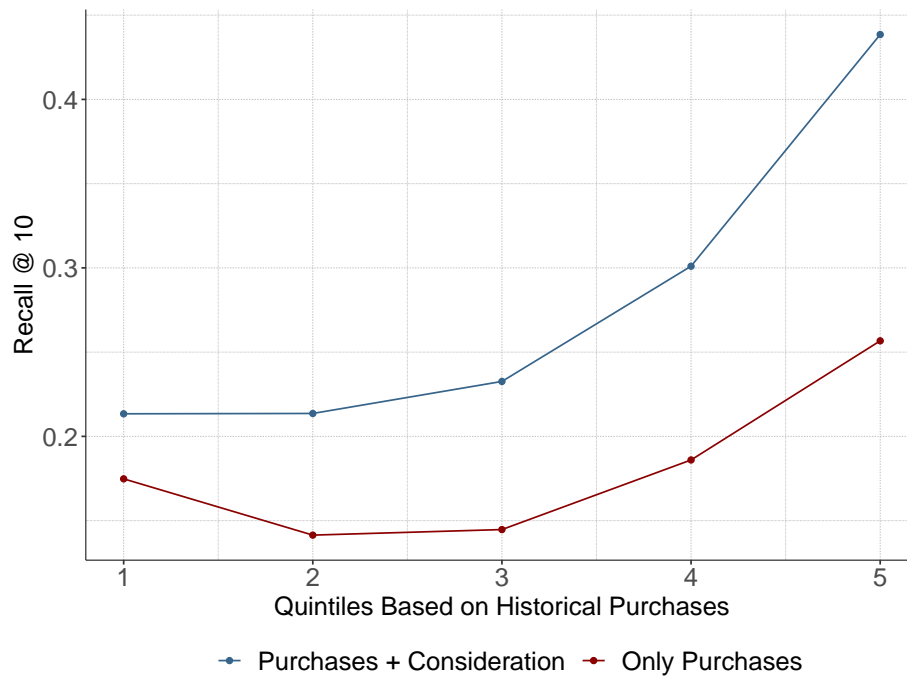


Figure 3.6: XGBoost feature importance for predicting future purchases

We also split the gains based on product to check if the model performs better on certain types of products. Figure 3.7 shows the top-10 and worst-10 categories based on model performance on the test data. While not as clearly discernible as the user case, we find that food based products are generally well-predicted by the model as compared to household and beauty products. One reason behind this difference in gains is the relative frequency with which products within these categories are purchased. Generally, food related products are purchased at more regular intervals as compared to household products, and the model is able to pick

that up easily.

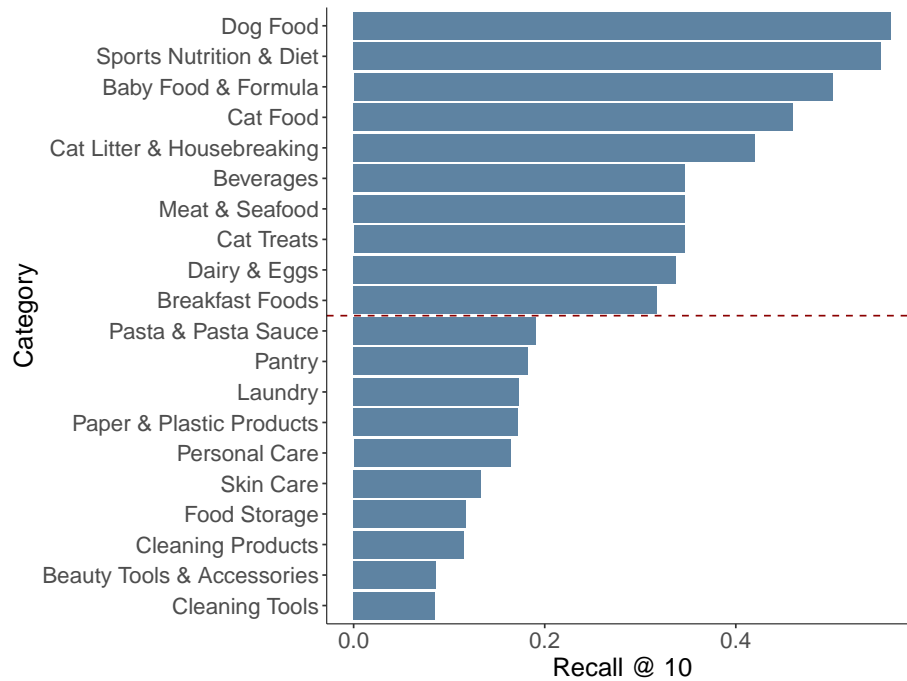


Figure 3.7: XGBoost feature importance for predicting future purchases

3.6 Discussion

We investigate the benefit of using historical consideration set data to predict future purchases. We use multiple computational approaches such as discrete-choice logit models, quasi-Poisson demand models, penalized regression, boosted trees, and a custom sequential deep learning model to evaluate the gains provided by consideration sets. Overall, we find robust and substantial gains in using data from historical consideration sets to predict future purchases. For example, including consideration sets in a simple penalized regression model improves out-of-sample performance by 10% and more flexibly including them in a sequential deep learning model improves performance by 27%. Moreover, these gains are distributed across the entire spectrum of consumers as defined by quintiles based on historical revenue.

Consideration sets, more broadly, are an important phenomena that are crucial

to more deeply understand consumer behavior. We find this to be a promising area of research, especially given the increasing share of online marketplaces in consumer spending. The rise of online marketplaces such as e-retailers, online travel portals, and content discovery platforms gives researchers and firms an opportunity to observe the entire journey of consumer decision processes at scale. Tying this fine grained view of consumer search, consideration, and choice, can unlock many insights into how consumer preferences form and evolve.

Bibliography

- Allenby, G. M. and Rossi, P. E. (1998). Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89(1):57–78.
- Ansari, A., Li, Y., and Zhang, J. Z. (2018). Probabilistic topic model for hybrid recommender systems: A stochastic variational bayesian approach. *Marketing Science*, 37(6):987–1008.
- Chakraborty, I., Kim, M., and Sudhir, K. (2022). Attribute sentiment scoring with online text reviews: Accounting for language structure and missing attributes. *Journal of Marketing Research*, 0(0):00222437211052500.
- Chen, F., Liu, X., Proserpio, D., Troncoso, I., and Xiong, F. (2020). Studying product competition using representation learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, page 1261–1268, New York, NY, USA. Association for Computing Machinery.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 785–794, New York, NY, USA. ACM.
- Dew, R., Ansari, A., and Li, Y. (2020). Modeling dynamic heterogeneity using gaussian processes. *Journal of Marketing Research*, 57(1):55–77.
- Dhillon, P. S. and Aral, S. (2021). Modeling dynamic user interests: A neural matrix factorization approach. *Marketing Science*, 40(6):1059–1080.
- Donnelly, R., Ruiz, F. J., Blei, D., and Athey, S. (2021). Counterfactual inference for consumer choice across many product categories. *Quantitative Marketing and Economics (QME)*, 19(3):369–407.
- Dzyabura, D. and Hauser, J. R. (2011). Active machine learning for consideration heuristics. *Marketing Science*, 30(5):801–819.
- Dzyabura, D. and Hauser, J. R. (2019). Recommending products when consumers learn their preference weights. *Marketing Science*, 38(3):417–441.
- Gabel, S. and Timoshenko, A. (2020). Product choice with large assortments: A scalable deep-learning model.
- Guadagni, P. M. and Little, J. D. C. (2008). A logit model of brand choice calibrated on scanner data. *Marketing Science*, 27(1):29–48.
- Hauser, J. R. (1978). Testing the accuracy, usefulness, and significance of probabilistic choice models: An information-theoretic approach. *Operations Research*, 26(3):406–421.

- Hauser, J. R., Toubia, O., Evgeniou, T., Befurt, R., and Dzyabura, D. (2010). Disjunctions of conjunctions, cognitive simplicity, and consideration sets. *Journal of Marketing Research*, 47(3):485–496.
- Hauser, J. R. and Wernerfelt, B. (1990). An Evaluation Cost Model of Consideration Sets. *Journal of Consumer Research*, 16(4):393–408.
- Jacobs, B. J., Donkers, B., and Fok, D. (2016). Model-based purchase predictions for large assortments. *Marketing Science*, 35(3):389–404.
- Kamakura, W. A. and Russell, G. J. (1989). A probabilistic choice model for market segmentation and elasticity structure. *Journal of Marketing Research*, 26(4):379–390.
- Lu, S., Xiao, L., and Ding, M. (2016). A video-based automated recommender (var) system for garments. *Marketing Science*, 35(3):484–510.
- Mela, C. F., Gupta, S., and Lehmann, D. R. (1997). The long-term impact of promotion and advertising on consumer brand choice. *Journal of Marketing Research*, 34(2):248–261.
- Roberts, J. H. and Lattin, J. M. (1991). Development and testing of a model of consideration set composition. *Journal of Marketing Research*, 28(4):429–440.
- Ruiz, F. J. R., Athey, S., and Blei, D. M. (2017). SHOPPER: A Probabilistic Model of Consumer Choice with Substitutes and Complements. Papers 1711.03560, arXiv.org.
- Tellis, G. J. (1988). Advertising exposure, loyalty, and brand purchase: A two-stage model of choice. *Journal of Marketing Research*, 25(2):134–144.
- Timoshenko, A. and Hauser, J. R. (2019). Identifying customer needs from user-generated content. *Marketing Science*, 38(1):1–20.
- Winer, R. S. (1986). A Reference Price Model of Brand Choice for Frequently Purchased Products. *Journal of Consumer Research*, 13(2):250–256.

Appendix

A Supplementary tables and figures

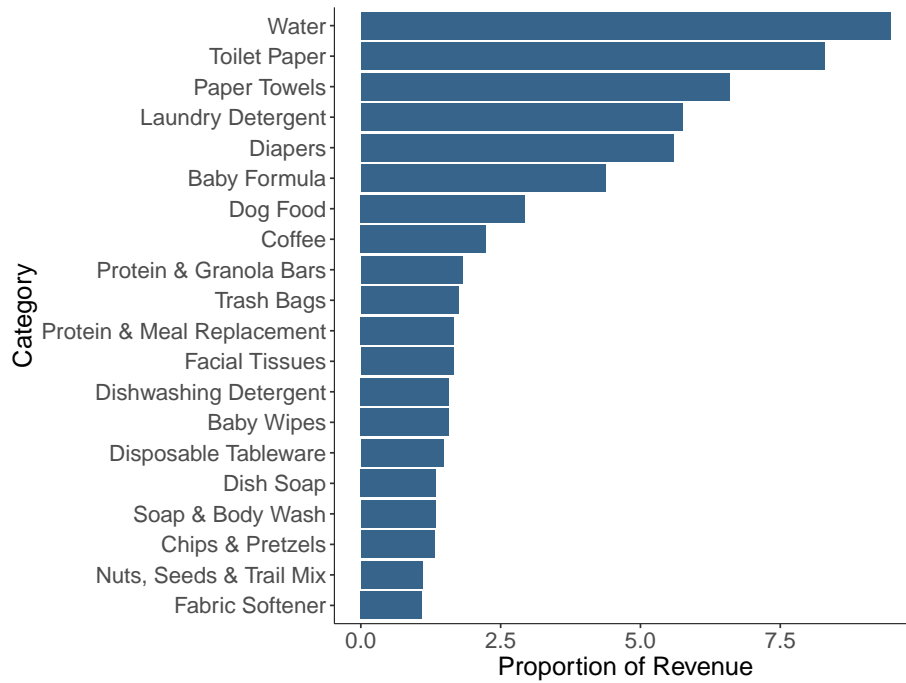


Figure A.1: Top-20 most popular categories by revenue

Table A.1: Discrete choice model with and without consideration sets

Dependent Variable: Model:	Purchase indicator	
	Typical	With consideration set
<i>Variables</i>		
Log price	-2.33*** (0.130)	-3.25*** (0.297)
<i>Fixed-effects</i>		
<i>User & Product</i>		
<i>Fit statistics</i>		
Observations	56,192	10,982
Squared Correlation	0.026	0.223
Pseudo R ²	0.054	0.206
BIC	50,130.4	26,768.8

Clustered (User & Product) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

B Tuned hyper-parameters for DNN

Hyper-parameters were tuned using a random held-out 10% validation sample.

- **Embedding dimensions:** 64
- **Optimizer and learning rate:** Adam, 0.001
- **Number of epochs:** 40
- **Batch size:** 1,024