Personalized Retail Promotions Through a Directed Acyclic Graph–Based Representation of Customer Preferences

Srikanth Jagabathula, Dmitry Mitrofanov, Gustavo Vulcano

1. Introduction

The availability of individual-level transaction data allows retailers to implement personalized operational decisions. Although such decisions have been around for several years now in online platforms, where e-tailers can profile their shoppers toward personalized pricing according to geographical location, past purchases, or device used for access (desktop versus mobile), recent technological developments have allowed to extend similar practices to bricks-and-mortar settings where most of the sales still take place (Pounder 2015).

For instance, in 2013, about 45% of Safeway’s sales had already come from specialized offers that customers could get through desktop or mobile applications (Kharif 2013). Lowes Foods, a grocery store chain with around 100 stores in North Carolina, South Carolina, and Virginia, launched personalized promotions in May 2016 in which every registered store guest receives the most relevant deals, based on her purchasing history, through the mobile phone (Denman 2016).

To implement customized promotions, retailers now have a range of technological options, including the use of electronic price tags to show different prices to different customers, the use of beacon-based technology to send promotion offers to targeted customers, and the use of computer vision to provide customized deals depending on the items the customer has added to the shopping cart (e.g., see Apricart’s application (Ladd 2019)). These technologies have already been adopted in practice. For instance, B&Q retail chain (Pounder 2015) has used electronic price tags. And Macy’s, Marsh supermarkets, GameStop, and mall developers, such as Simon Property Group and Macerich, have already tested the beacon-based technology. In fact, Simon Property Group installed about 4,800 beacons over 192 malls to target customers using the Simon app, and the top-100 retailers saw around $4 billion of sales from the beacon-based technology in 2015.

Personalized promotions offer several benefits to the retailers, becoming an effective tool for individual-
level price discrimination. They reduce competition by making the price paid by customers opaque to other retailers as sticker price is no longer the same as the price paid by a customer. They also induce stronger customer relationship and drive up sales. According to an Accenture survey (Vujanic and Goldstein 2015), more than 60% of customers want to participate in customized promotions and explore real-time deals. Along the same lines, a more recent study conducted on 1,250 global shoppers (Brinks 2018) reveals that 65% of customers appreciate personalized prices. It appears that consumers appreciate services accompanied with personalization more than they dislike sharing the personal information about their purchasing habits (Farhnam 2013).

Motivated by the significance of personalized promotions, we provide a full methodological roadmap to run them in both online and offline retail settings. The required input data consist of a history of sales transactions for a category of substitutable products (e.g., coffee in a grocery store, women’s shirts in a fashion apparel retailer) tagged by individual customer IDs. With each transaction, the data also supply the set of products available for purchase (product availability) and the subset of which are offered on promotion (product promotions). Using these data, the retailer must first infer customer-level preferences for the items within the category of analysis, which allows him to predict each customer’s purchases in response to the retailer’s promotion decisions. This inference problem faces three main challenges: (i) data sparsity, because only a few observations per customer may be readily available for a particular category; (ii) variation in the availability of products (e.g., due to stockouts); and (iii) presence of promotions that may alter ex ante customer choices. The first challenge is the most significant one for any personalized prediction. The latter two challenges complicate preference inference because it is hard to tell if a customer switched her purchase because of a change in preferences or because of a stock-out or promotion. Once customer-level preferences are estimated, the retailer must decide the optimal subset of products to put on promotion (if any) for each individual customer visit to the store or the website, with the objective of maximizing the revenue from the visit.

Our focus in this paper is on the immediate to short-term effect of brand-switching of the promotion decisions. Therefore, we focus on a retailer who wants to maximize the immediate revenue from a customer visit. Promotions also have medium- to long-term effects, such as stockpiling, consumption stimulation (leading to general increase in consumption levels of the product), new customer attraction, and customer retention (e.g., store loyalty, category loyalty, or brand loyalty). Our focus on brand-switching effects allows us to develop methods that can scale to practical-sized problems with thousands of customers and millions of transactions. In the conclusions section, we briefly comment on how our decision system can be extended to account for some of the medium- to long-term impacts of promotions. We also argue that our proposal here may still be helpful in mitigating the stockpiling effect.

### 1.1. Summary of Results

The building block of our proposal is a nonparametric, choice-based demand model where each customer is characterized by a directed acyclic graph (DAG) representing a partial order among products in a category. In the DAG, each product is represented by two nodes: a full price version and a discounted counterpart. A directed edge from node $a$ to node $b$ indicates that the customer prefers the product corresponding to node $a$ over the product corresponding to node $b$. The DAG captures the fact that customer preferences are acyclic. Unlike a full preference list, a DAG specifies pairwise preferences for only a subset of pairs of products; therefore, it represents a partial order. When visiting the store, the customer samples a full preference list consistent with her DAG according to a prespecified distribution and chooses the available product that ranks the highest.

Inferring customer preferences from transaction data consists of two key elements. The first element is the construction of the DAGs. Starting from an empty graph (i.e., a collection of isolated nodes representing products), and using historical data as a source of revealed preferences for each individual, we start adding edges from the purchased product to the other products (i.e., nodes) that were on offer, distinguishing between full price and discounted versions of the products. This process can lead to a graph with cycles, reflective of the fact that a number of “incorrect edges” could have been added along the way. In order to associate each customer with a partial order, we run a decycling procedure with the objective of dropping spurious edges.

The second key element is to fit a choice model that specifies the distribution with which the customer samples full preference lists consistent with her DAG. We fit a multinomial logit (MNL) model and a multiclass version of it. The estimation requires computing the likelihood of the constructed DAGs, which is a computationally hard problem in general. In order to ease the estimation process and the posterior prediction, we provide lower and upper bounds for the likelihood of a DAG that are easy to compute and are then used as an approximation for the exact probability.
The predictive power of our method is illustrated through an extensive set of numerical experiments using real grocery panel data on purchases across two big U.S. markets in the year 2007. We split our data set for each of 27 product categories in two parts. On the first half (i.e., the training data), we perform the aforementioned two stages: DAG construction and MNL estimation (single and multiclass). Then, on the second half (i.e., the holdout sample), we predict what each customer would purchase under our model when confronted with the historical offer sets and products on promotion, and compare our prediction with the reported purchase. Our study demonstrates that our approach results in more precise, fine-grained predictions for customer choice behavior in comparison with state-of-the-art benchmarks that also incorporate promotion effects. Specifically, we obtain up to 14% improvement in prediction accuracy, on average, across the 27 product categories studied, on standard measures.

Confident about the predictive power of our model, we then use the outputs of the DAG construction and estimation stages as inputs to run personalized promotions. We formulate a mixed integer linear program (MILP) that decides which products to promote when a particular customer identified by her DAG faces a given offer set. We analyze two types of scenarios. The first one focuses on the setting in which the retailer is running personalized promotions in conjunction with the mass promotions already in place (as reported in the data set), so the retailer can personalize the promotion of only those products not already under mass promotion. The second one affords more flexibility to the retailer and assumes that the retailer can personalize the promotion of any product on offer. Our simulated results show average improvements of 16.7% and 23.9% across the 27 categories, respectively, for the two scenarios when compared with the existing promotion strategy in place.

The empirical validation of our model supports its use toward the implementation of customized promotions in a systematic, data-driven way. Another key advantage of our method is that the DAG-based representation of the preferences provides an intuitive and transparent interpretation of the personalized promotion decision. This is an appealing feature for the retail industry where several technically sophisticated grocery chains still rely on manual processes based on rule of thumb and past experience to decide price promotions (e.g., see Cohen et al. 2017).

1.2. Related Literature

Our paper touches upon two streams of literature: marketing and operations. Although the use of panel data as a source to estimate choice models is still limited in the operations literature, it has been around for a while in the marketing field. A pioneering work in this regard is the seminal paper by Guadagni and Little (1983), where the authors fit an MNL model to household panel data on regular ground coffee transactions, which has led the way for choice modeling in marketing using scanner panel data. Chandukala et al. (2008) and Wierenga et al. (2008) provide a detailed overview of choice modeling using panel data in marketing. Much of this research stream focuses on understanding how various panel covariates affect the individual choice process.

The effect of sales promotions on a retailer is also extensively studied in marketing (Blattberg and Neslin 1990). A very illustrative overview of different promotion mechanisms used by a retailer is provided by Gedenk et al. (2006), where they summarize different objectives and effects of promotions known in the literature. Although only a few papers report a positive effect of sales promotions on long-run brand preferences (Dekimpe et al. 1998, Foekens et al. 1998), most of the empirical studies conclude that (mass) sales promotions are good instruments to induce customer’s substitution behavior in the short run but with neutral or negative effect on the brand preference in the long run (Mela et al. 1997, Srinivasan et al. 2004, DelVecchio et al. 2006). Most of this work is focusing on empirical understanding of the overall impact of promotions, unlike our paper, which is focused on developing a methodology for carrying out personalized promotions.

Our paper is most closely related to the body of empirical research within marketing focused on developing a methodology for individual-level marketing policies. Zhang and Krishnamurthi (2004) provide a decision-support system to optimize the timing and the depth of promotions for a given brand. Their structural model accounts for three simultaneous components: interdependence in purchase incidence, brand choice, and purchase quantity, and assumes that preferences (even for a single customer) may vary over time. Like in our case, the building block for the model is the individual household level; but the likelihood function is based on a latent class market structure that captures unobserved consumer heterogeneity. This function can be specified in closed form but lacks convexity properties that would ease the estimation process and make it hard to scale to a large number of alternatives in the category (in fact, in their experiments, they report results based on two categories with only four options each). The promotion decision problem considered is also different. Whereas we focus on the optimal subset of brands to put on promotion, Zhang and Krishnamurthi (2004) assume that the retailer decides on a single brand or manufacturer to promote at a time. Given the brand, the decision
focuses on how to set discounted prices (both the timing and the depth of promotions) for the next few store visits of a given individual. This price promotion problem is highly nonlinear and lacks any structural property, which also makes it hard to scale for a large number of variants in the category.

Khan et al. (2009) develop a dynamic programming-based approach similar to the one in Zhang and Krishnamurthi (2004), but they use individual-level coefficients to evaluate the benefits of optimizing customized promotions at the level of each single customer. However, this twist makes their methodology even more computationally intractable as for estimation they need to use a Markov chain Monte Carlo procedure to simulate the posterior distribution of the model parameters and to compute household level estimates of preferences.

The existing work on choice-based demand in the operations literature has largely focused on using aggregate sales transaction data to estimate models in the presence of stockouts and offer sets that change over time and then to use these estimates as inputs to solve the assortment and pricing optimization problems. The paper by Berbeglia et al. (2021) provides an up-to-date overview of the retail operations and revenue management literature on choice-based demand models.

Our proposal here is rooted in a rank-based choice model of demand. This type of nonparametric choice model specifies customer classes defined by their rank orderings of all alternatives within the product category. When visiting the store, a customer is assumed to purchase the available product with the highest ranking in her preference list or to leave without making any purchase. This model, which provides the full flexibility of random utility models, has been gaining increasing attention in the Operations Management literature (Mahajan and van Ryzin 2001, Rusmevichientong et al. 2006, Farias et al. 2013, van Ryzin and Vulcano 2014). However, these references still assume a market-level, choice-based demand model.

In Jagabathula and Vulcano (2018) a first step is taken toward the specialization of the rank-based choice model to capture and estimate individual preferences. In that paper, the authors propose modeling individual preferences through DAGs; but their construction is guided by heuristic definitions of the consideration sets (e.g., see Hauser 2014). Therein, a customer samples a full preference list of items in the product universe (along with the no purchase alternative) in accordance with her partial order, forms a consideration set, and then buys the available product among the considered ones with the highest rank in the preference list. Three models based on different consideration set definitions were studied: (i) standard, where the consideration set is equal to the offer set, (ii) inertial, where the consideration set is a subset of the offer set given by the previous purchase and the current products on promotion, and (iii) censored, which is a slight generalization of the inertial model. Both (ii) and (iii) were designed to capture the inertia in choice (Jeuland 1979), a principle claiming that customers tend to stick to the same option when facing frequently purchased consumer goods.

Motivated by the promising predictive results of that model (which was also successfully applied recently to model preferences for virtual items in video games; Khandelwal et al. 2017), in this paper we leverage the performance of the DAG-based approach with the objective of designing customized promotions. Our contribution with respect to Jagabathula and Vulcano (2018) spans several dimensions. First, our consideration set formation is purely data-driven, providing greater flexibility without imposing any prior beliefs on bounded rationality of individuals (such as the stickiness principle for the aforementioned inertial model). This approach allows us to extend the coverage of the number of individuals whose behavior our model can explain with nonempty DAG structures. Second, in our new proposal we explicitly account for promotions as part of the DAG definition (and not indirectly through the heuristic formation of the consideration set). Our method of incorporating promotions forms the fundamental backbone of the proposal. It provides clean managerial insights about customer preferences (as explained later) and allows running promotion optimization in a transparent way. Third, from a theoretical perspective, we develop tractable analytical lower and upper bounds for the likelihood of DAGs under the MNL model, because it is known that computing the exact likelihood is a #P-hard problem. The lower bound is indeed the exact probability of a DAG when it is a forest of directed trees, as shown in Jagabathula and Vulcano (2018). Here, we show that under some technical conditions, the bounds are asymptotically tight. In addition, we derive tractable analytical lower and upper bounds for the MNL probability of a customer choosing a specific product conditioning on her DAG and the available offer set. Finally, we address the promotion optimization problem as a key distinguishing feature of our work, whereas in Jagabathula and Vulcano (2018) the focus was limited to establishing the predictive power of the behavioral-based DAG model.

2. Choice Model Description
This section formally introduces our general modeling framework, starting from some basic notation and explaining the choice process derived from the customers’ DAGs. We continue with the discussion of the modeling assumptions and the description of the data
model that serves as input for our choice model, followed by the presentation of the different phases involved in the DAG construction procedure. Next, we discuss the assumptions underlying our DAG construction procedure and close the section with the formulation of the associated maximum likelihood estimation problem.

### 2.1. Modeling Framework

Consider a category of $n$ substitutable products on which a set of $m$ individuals make purchases over a finite horizon. Both the set of customers and the set of products remain constant over time. Each product has two different versions: the nonpromoted version and its promoted counterpart. The promotion could be a price or display promotion or any form of product presentation that highlights its presence on the shelf. We denote by $\mathcal{N}$ the set \{a_1, a_2, ..., a_n\} of nonpromoted versions of the products. For any $j \in [n]$ (i.e., $1 \leq j \leq n$), we let $a_{j+n}$ denote the promoted version of product $a_j$. Furthermore, we let $\mathcal{N}' = \mathcal{N} \cup \{a_{n+1}, a_{n+2}, \ldots, a_{2n}\}$ denote the expanded product universe with the corresponding promoted counterparts.

The preferences of each customer over the product universe $\mathcal{N}'$ are described through a partial order, which could be visualized as a directed acyclic graph. A DAG $D$ consists of $2n$ nodes, with two copies for each product (one for the nonpromoted version and one for its promoted counterpart), and a collection of directed edges (or pairwise preference relations) denoted by $E_D \subseteq \{(a_j, a_j') : 1 \leq k, j \leq 2n, k \neq j\}$, so that for any $(a_j, a_j') \in E_D$, we have that item $a_k$ is preferred to item $a_j$. With the assumption that a customer always prefers the promoted version of a product over its nonpromoted counterpart, the DAG includes $n$ arcs of the form $(a_{j+n}, a_j)$.

The DAG captures the strong preferences the customer has over the products. These preferences remain constant from one purchase instance to the next. For instance, suppose that a customer always prefers caffeinated (regular) coffee over decaffeinated (decaf) coffee. Such a customer will be captured by a DAG with preference edges from every regular coffee brand (promoted or not) to every decaf coffee brand (promoted or not). The customer’s brand preferences may change from one purchase instance to another, but she will always purchase regular coffee over decaf coffee. Note that a customer may have no strong preferences, in which case her DAG would be rather sparse. At the other extreme, a customer may have very strong preferences over all the products, in which case her DAG would be a total ordering over the $2n$ products. The DAGs provide us with a flexible tool to capture customers between these two extremes.

We will describe the complete process for constructing the DAG from the observed transaction data in Section 2.4. But for now, given these DAGs, we describe the choice process. In general, DAGs can only specify what a customer will not purchase—rather than what she will purchase—in each store visit. For instance, in the example above, the DAG specifies that the customer will not purchase decaf coffee in the presence of regular coffee; but it remains silent on which of the regular coffee brands she will purchase. Because the preferences not present in the DAG may change between purchase instances, we capture them through a probabilistic model. We let $\lambda$ denote a distribution over all possible total orderings of the $2n$ products. A total ordering (unlike a partial order) specifies the pairwise preferences for all possible $\binom{2n}{2}$ pairs. Equivalently, a total ordering is a ranking (i.e., permutation or preference list) of the $2n$ products.

In each interaction with the retailer, the customer samples a ranking that is consistent with her DAG $D$ according to distribution $\lambda$ (to be estimated from data as explained below). The customer then chooses the most preferred product according to the sampled ranking from a subset of products she considers from among the offered products. More formally, if $\sigma$ denotes the sampled preference list, $\sigma(a_i)$ indicates the preference rank of product $a_i$. A lower ranking indicates a higher preference order; in other words, we have that $a_k$ is preferred to $a_j$ according to $\sigma$, written as $a_k >_\sigma a_j$, if and only if $\sigma(a_k) < \sigma(a_j)$. We say that a preference list $\sigma$ is consistent with partial order $D$ if and only if $\sigma(a_k) < \sigma(a_j)$ for each $(a_k, a_j) \in E_D$. We define $S_D$ as the set of rankings compatible with $D$, that is, $S_D = \{\sigma : \sigma(a_k) < \sigma(a_j) \text{ whenever } (a_k, a_j) \in D\}$, so that the sampled ranking $\sigma$ belongs to the set $S_D$.

In the store, the customer is offered a subset of products $S \subseteq \mathcal{N}'$. Naturally, at most one element between $a_i$ and $a_{i+n}$ is included in $S$. Let $C \subseteq S$ denote the subset of products the customer considers during this visit. Then, she purchases the most preferred product $a_k$ within the set of considered products, that is, $a_k = \arg \min_{a \in C} \sigma(a)$. The customer could sample a different ranking, independently, in each store visit; but the ranking is always consistent with her DAG $D$. Figure 1 illustrates the choice process for a particular store visit given a DAG $D$, a distribution over full rankings $\lambda$, and an offer set $S$. We do not impose any structural assumptions on how the consideration set is formed by the customer. In the absence of any additional information, in principle we assume that $\bar{C} = S$, that is, the customer considers everything on offer (this is indeed our approach in the numerics in Section 4).

The above model assumptions lead to the following choice probability expression. Assuming that the
customer considers everything on offer, the customer purchases product \(a_i\) from offer set \(S\) if and only if the sampled ranking \(\sigma\) belongs to the set \(\mathcal{A}_i(S) = \{\sigma : \sigma(a_i) < \sigma(a_j) \; \forall \; a_j \in S \setminus \{a_i\}\}\). Because a customer with DAG \(D\) will only sample rankings \(\sigma \in S_D\), we have that the probability \(f(a_i, S, D)\) that the customer will purchase product \(a_i\) from offer set \(S\) is given by

\[
 f(a_i, S, D) = \frac{\Pr_{i}(\sigma \in \mathcal{A}_i(S) | \sigma \in S_D)}{\Pr_{i}(\sigma \in S_D)},
\]

where \(\Pr_{i}(\cdot)\) is the probability of an event under the distribution \(\lambda\). The computational complexity of computing the above probabilities depends on the structure of the underlying distribution \(\lambda\). We defer such issues to Section 3.

### 2.2. Discussion of Model Assumptions

The assumption that the product universe \(\mathcal{N}'\) and the set of customers remains constant over a finite horizon is needed to infer the customer DAGs, as described below. Echoing Jagabathula and Vulcano (2018), our approach can be run periodically to update the DAGs and incorporate new customers. In between updates, new products in the category can be considered as part of a family of products (say, products of the same brand represented by only two elements: \(a_i\) and \(a_{i+1}\)).

Note that in our presentation, without loss of generality, we do not provide a special treatment to the always available, no-purchase option. This option can be handled in the same way as other items in the product universe except that only one copy of this alternative would be part of the DAG; that is, the no-purchase option can be represented by a particular node in the DAG, say \(a_0\).

We also note that different customers may have different DAGs, but they all use the same distribution \(\lambda\) to sample the rankings. In other words, the distribution \(\lambda\) is a population attribute, whereas the DAG is an individual attribute. Even though the same distribution \(\lambda\) is being used by all the customers, our model easily captures preference heterogeneity. For instance, the distribution \(\lambda\) could be a latent-class multinomial logit (LC-MNL) model, which assumes that the population is comprised of \(K\) latent classes and the preferences of each class of customers is described by a different MNL model. In addition, the rankings sampled by customers must be consistent with their respective DAGs, so the effective distribution used for each customer is the conditional distribution \(\lambda\) given her DAG. Because DAGs differ across customers, these conditional distributions will also differ.

### 2.3. Data Model

Our data model is the same as the one used in Jagabathula and Vulcano (2018). We consider a data set with transactions tagged by the IDs from \(m\) customers. For a given customer \(i\), we consider a training horizon of \(T_i\) transactions of the form \((a_{ij}, S_{it})\), for \(t = 1, 2, \ldots, T_i\), that we use to infer her partial order of preferences. The offer set is \(S_{it} \subset \mathcal{N}'\), and \(a_{it} \in S_{it}\) denotes the product she purchased in period \(t\).

The subset \(P_{it} \subset S_{it}\) denotes the set of promoted products in period \(t\). In our data set, the promotion could be either display or price. In our numerics, we restrict our attention to price promotions and in particular we consider the promotion feature as a binary attribute of a product. That is, we do not distinguish between different levels of price promotions, although our model could be easily extended to account for a finite number of price discount points by simply adding a product copy (i.e., node) for each discount level in the discrete set.

In order to partially mitigate the data sparsity issue, in our implementation we aggregate products within a category by brand, so as to have at least a few observations of offerings and purchases for each of the items. It is important to note that this minimal level of aggregation is just a modeling decision; in our case, it is done to compensate for the data sparsity (with the goal of having several observations involving each node in the data set) and also avoid sparse DAG structures (i.e., DAGs with many nodes and few edges). However, our model allows for alternative product definitions. For instance, a node may represent a particular SKU (and therefore have several nodes of the same brand) or may represent different package sizes and then have two nonpromoted nodes per brand: one for large packages and one for small packages.

### 2.4. DAG Construction

The first stage in our framework is the construction of a DAG for each individual purchasing from a category. We process customer transactions one at a time...
to dynamically build the DAG. The whole process involves four steps; but at the core, it relies on a set of preference inferences made from each transaction. To illustrate the process and its challenges, consider a transaction in which the customer was offered products $a$ and $b$ and she purchased product $a$. Given this transaction, we can reason that there are three different possibilities: (i) the preference $a > b$ is strong and so the edge $(a, b)$ must be part of the customer’s DAG; (ii) the preference $a > b$ is not a strong preference and the customer simply sampled a ranking $σ$ with $σ(a) < σ(b)$ for this purchase instance, so neither edge $(a, b)$ nor $(b, a)$ should be part of the customer’s DAG (if the edge $(b, a)$ were part of the DAG, then the preference list $σ$ with $σ(a) < σ(b)$ would never be sampled because it would be inconsistent with the DAG); and (iii) product $b$ was not considered by the customer; therefore, we cannot make any inferences about the preference relation between $a$ and $b$ in the DAG. (In fact, it is perfectly possible to have $b > a$ in the DAG; but it would not matter because as far as the customer is concerned, product $b$ was never under consideration.) All three inferences are consistent with our model; therefore, more than one DAG is consistent with the given data. Our challenge lies in identifying the set of DAGs that are consistent with the given data and then using a reasonable criterion to pick one from this set.

At a high level, we deal with the above challenge in the DAG construction by first building a directed graph $G$ including what we call candidate edges. A candidate edge is an edge that we are unsure of, with the understanding that it may be removed at a later stage in the DAG construction process. As seen below, the graph $G$ allows us to keep track of the set of DAGs that are consistent with the given transaction data. We then make an identification assumption to pick a DAG that is a subgraph of $G$. To keep our presentation clean, we first describe all the steps involved in constructing a DAG. We then discuss in Section 2.5 all the assumptions implicit in our DAG construction process. Figure 2 illustrates the DAG construction process for a small running example with four products ($n = 4$) and three transactions ($T = 3$), following the sequence of four phases below.

### 2.4.1. Phase 0: Initializing the Preference Graph with Edges from Promoted Versions to Corresponding Nonpromoted Versions of Products

We start from an empty graph $G$ and add $2n$ isolated nodes from the product universe $\mathcal{N}$, where each node represents either a nonpromoted or a promoted version of a product. Let $E_C$ denote the set of edges in the graph $G$. Starting from the empty set $E_C$, we add $n$ edges $(a_{j+n}, a_j)$ for each nonpromoted item $a_j$, $j \in [n]$. These edges capture the fact that a promoted copy $a_{j+n}$ of

---

**Figure 2.** (Color online) Phases of DAG Construction

**Phase 0: Initialization**

$t=0$:

- $E = \emptyset$
- $G = \emptyset$
- $σ = \emptyset$

**Phase 1: Adding candidate edges**

$t=1$:

- $S_1 = \{1, 2, 7\}$
- $P_1 = \{7\}$
- $σ_1 = 2$

**Phase 2: Adding implicit candidate edges**

$t=2$:

- $S_2 = \{2, 4, 7\}$
- $P_2 = \{7\}$
- $σ_2 = 2$

- $S_3 = \{2, 7\}$
- $P_3 = \{7\}$
- $σ_3 = 2$

**Phase 3: Decycling**

$t=3$:

- $S_4 = \{2, 7\}$
- $P_4 = \{7\}$
- $σ_4 = 2$

---

**Notes.** There are $n = 4$ products and $T = 3$ transactions. Nodes $\{1, 2, 3, 4\}$ represent the nonpromoted version of the products, and nodes $\{5, 6, 7, 8\}$ represent their corresponding promoted version. We initialize the graph in Phase 0, starting from the addition of edges from the promotion copy of each node to its nonpromoted counterpart. In Phase 1, we go through the data and in each purchasing instance we draw candidate edges from the purchased item to all the other items in the offer set. Note that the addition of edge $(7, 2)$ when $t = 2$ creates a cycle. In Phase 2, we add implicit candidate edges. In this case, because we have the edge $(2, 1)$, we add the edge between the corresponding promotion counterparts: $(6, 5)$. Finally, in Phase 3, we decycle the graph. Note that $w_{2,7} = 2$ and $w_{7,2} = 1$. The decycling procedure will retain the edge $(2, 7)$.
every product \( a_i \in \{ a_1, \ldots, a_n \} \) is preferred to its non-promoted copy \( a_i \) because both products have the same attributes except the promotion feature. Note that these are not candidate edges because we are certain of their presence in the final DAG.

### 2.4.2. Phase 1: Adding Candidate Edges from Sales Transactions.** We incrementally add candidate edges to the preference graph \( G \) by processing the customer’s transactions one at a time. For each transaction \((a_{ij}, S_i)\) of individual \( i \), we draw edges from \( a_{ij} \) to the other items in the offer set, that is, \( E_G \leftarrow E_G \cup \{(a_{ij}, a_k) : \forall a_k \in S_i \setminus \{a_{ij}\}\}\). These edges signify that potentially all the offered products were considered by the individual and all preference edges were indeed “strong” preferences (i.e., they are all part of the DAG and not just sampled preferences). We keep track of the purchase events where each edge \((a_{ij}, a_k)\) is added through the weight \( w_{ij} \), defined as the number of times the customer chose product \( a_j \) when \( a_i \) was also offered.

### 2.4.3. Phase 2: Adding Implicit Candidate Edges.** To make the DAG denser, we enrich it with implicit candidate edges, based on the assumption that if a customer has a strong preference between the nonpromoted (promoted) copies of two products, then the preference extends also to the corresponding promoted (nonpromoted) copies. More precisely, for any \( 1 \leq j, \ell \leq n \), if \((a_{ij}, a_{k}) \in E_G\), then we add the edge \((a_{ij+n}, a_{k+n})\) to \( G \), that is, \( E_G \leftarrow E_G \cup \{(a_{ij+n}, a_{k+n})\}\). Similarly, if \((a_{ij+n}, a_{k+n})\) \in \( E_G \), then we add the edge \((a_{ij}, a_{k})\) to \( G \), resulting in \( E_G \leftarrow E_G \cup \{(a_{ij}, a_{k})\}\). To de-emphasize the implicit counterparts of these edges, we assign the weight \( w_{ij+n, k+n} \leftarrow w_{ij, k} / T_n \) when \((a_{ij+n}, a_{k+n})\) is the implicit edge and \( w_{ij} \leftarrow w_{ij+n, k+n} / T_n^2 \) when \((a_{ij}, a_{k})\) is the implicit edge. In other words, the weights of the implicit counterparts are scaled down by a factor of \( T_n \). Intuitively, by scaling down the weights of the implicit edges, we prioritize candidate edges over implicit candidate edges, because candidate edges inferred directly from the revealed preferences of customers are likely to be more informative. The reason for this precise choice of scaling factors will become clear below.

### 2.4.4. Phase 3: Graph Decycling.** This is a critical step in the DAG construction process, in which we attempt to eliminate the spurious edges added in \( G \) so far to arrive at the final DAG, where by spurious we mean that the edge contradicts the interpretation of other edges in the graph. The first indication that there are spurious edges in \( G \) is the presence of directed cycles. As discussed above, the data do not identify the DAG; therefore, we need to make an identification assumption to arrive at the final DAG from \( G \). We assume that the underlying DAGs of customers are large, so we find the largest-weight DAG that is supported by the choice observations. In other words, we assume that all candidate and implicit candidate edges are part of the underlying DAG, unless contradicted by data. This assumption translates to deleting cycles from \( G \) while maximizing the aggregate weight of the edges retained (or similarly, minimizing the total weight of the edges deleted). This problem is known in the graph theory literature as the minimum weight feedback arc set problem and is known to be NP-hard even when all weights are equal to one (Karp 1972 provides a reduction from the minimum vertex cover problem).

We formulate the above decycling procedure as a mixed integer linear program. For a given graph \( G \), and for every edge \((a_{ij}, a_{k}) \in E_G\), define the binary variable \( x_{ij, k} \) that takes the value one if edge \((a_{ij}, a_{k})\) is finally retained in the induced acyclic subgraph \( D \subset G \) and takes value zero otherwise. To ensure that the DAG defined by the variables \( (x_{ij, k} : (a_{ij}, a_{k}) \in E_G) \) does not contain cycles, we introduce auxiliary binary variables \( y_{ij, k} \) for all \( 1 \leq k, \ell \leq 2n \) and \( k \neq \ell \). These variables represent a total order over all the products in \( N' \) with \( y_{ij, k} = 1 \) if \( a_k \) is preferred over \( a_i \) and \( y_{ij, k} = 0 \) otherwise. The following MILP enforces that the final DAG is a subset of some total order defined by the \( y \) variables:

\[
\max \sum_{x, y} w_{ij, k} x_{ij, k}
\]

s.t. : \( x_{k+\ell, k} = 1 \) \( \forall 1 \leq k \leq n \)
\( x_{ij, k} = y_{ij, k} \) \( \forall (a_{ij}, a_k) \in E_G \)
\( y_{ij, k} + y_{ij, k} = 1 \) \( \forall a_i, a_k \in N' \) \( k \leq \ell \)
\( y_{ij, k} + y_{j, k+\ell} \leq 2 \) \( \forall a_i, a_k, a_p \in N' \) \( k \neq \ell \neq p \)
\( y_{ij, k} \in \{0, 1\} \) \( \forall a_i, a_k \in N' \) \( k \leq \ell \)

The constraints guarantee that the induced subgraph \( D \) defined by those edges \((a_{ij}, a_k) \in E_G\) for which \( x_{ij, k} = 1 \) is a DAG. The first set of equalities (C1) ensures that all the edges added in Phase 0 are retained in the final DAG. The second set of equalities (C2) ensures that the DAG \( D \) defined by the variables \( x \) is a subset of the graph defined by the total order corresponding to the variables \( y \). The third (C3) and fourth (C4) constraints together ensure that \( y \) indeed defines a total order. Specifically, the third set of constraints ensures that either \( a_i \) is preferred over \( a_k \) or \( a_k \) is preferred over \( a_i \) but not both; and the fourth set of constraints imposes the total ordering among any three products. The correctness of the MILP is shown in Proposition A1 in Online Appendix A2. This proposition shows that the MILP maximizes the weight of the candidate edges in the resulting DAG, and it never deletes a candidate edge if an implicit candidate edge can be deleted to break a directed cycle.

The size of the MILP (1) scales quadratically with \( n \) in the number of variables and cubically in the number of constraints. In Section A.3 in the online appendix, we propose a greedy heuristic to approximately
solve the preference graph decycling in polynomial time. We compare the output DAGs of the heuristic and MILP (1) on our data set and observe that under the heuristic we obtain only 0.4% sparser DAGs (in aggregate), which indicates its promising applicability in other real size problems.

2.5. Discussion of the DAG Construction Procedure

We now discuss the most relevant assumptions we make for constructing the DAGs.

There are two key distinctions between the DAG construction process here and the one in Jagabathula and Vulcano (2018). First and foremost, our proposal here is purely data driven and in Phase 1 mirrors the standard consideration set definition therein (under which the consumer chooses among all the products in the offer set), though accounting explicitly for promoted products, here represented by node entities. The other two models presented in Jagabathula and Vulcano (2018), inertial and censored, which actually showed the best predictive performance, are based on behavioral rules to build the consideration sets.

The second key distinction is the way we address apparent inconsistencies in the purchasing behavior of a customer. According to Jagabathula and Vulcano (2018), during the DAG construction process, as soon as the addition of arcs into a customer DAG $D_i$ implies the creation of a cycle or the customer’s transaction cannot be explained by the prespecified behavioral assumptions, the process stops and all the arcs are deleted, keeping $D_i$ as the empty DAG (i.e., a collection of isolated nodes). In such a case, no structure is superimposed and the customer could be described by a standard choice-based demand model (e.g., a typical, single-class MNL). In our new proposal, Phase 2 could end with a graph with cycles, which are then deleted in Phase 3. In the context of our model, cycles could originate because of spurious edges introduced for reasons (ii) and (iii) laid out in Section 2.4. Under this interpretation, consumers are fully rational and the modeler incorrectly added edge $(a, b)$ either because $a(σ) < b(σ)$ in the particular ranking $σ$ sampled in this particular store visit (although $(a, b)$ is not a strong preference, case (ii)), or because the modeler incorrectly assumed that $b$ was part of the consideration set (case (iii)). As discussed in the description of Phase 3, our decycling procedure deletes a minimum number of spurious edges added along the way.

Another possible way to rationalize the decycling process is model misspecification. Customers exhibit a bounded rational behavior, including possible inconsistencies in their purchases. In this case, the addition of candidate edges in Phase 1 assuming that the consideration set of the customer is indeed the entire offer set is correct for those purchase instances, but the customer is inconsistent over time. The decycling step in Phase 3 provides the largest DAG that is sustained by the customer’s inconsistent purchase behavior, although strictly speaking, that behavior over time cannot be explained by a DAG.

The identification assumption that the underlying customer DAGs are large is driven by our desire to retain a rich representation of the customers’ strong preferences. This assumption is reasonable for product categories in which customers make repeated purchases, which increases their familiarity of the product category, allowing them to develop strong preferences. Grocery categories are a good example of that, as also evidenced by our empirical study. One can imagine other assumptions that are appropriate in particular settings. Such assumptions will result in a different decycling step and adjustments in the corresponding MILP, while keeping the rest of our framework intact.

Another point that deserves discussion is the addition of implicit candidate edges. These edges are not directly revealed in the customer’s choices. So the customer may have revealed that $a_k$ is preferred over $a_ℓ$, but the customer has not revealed if the same preference extends to the respective promoted copies $a_{k+n}$ and $a_{ℓ+n}$. Yet, we make the assumption that the customer is likely to prefer $a_{k+n}$ over $a_{ℓ+n}$. The reason is that strong preferences of the customer (those that are part of the DAG and do not change from one purchase instance to next) are likely to be driven by characteristics other than promotion activity, which does vary from one visit to another. As an example, a customer may prefer regular coffee to decaf coffee because of taste. Such a customer would continue to prefer regular coffee over decaf coffee even if both of them are on promotion. There is still the possibility that our assumption is wrong in specific cases because of which we scale down the weights of implicit edges by $T_n$. Proposition A1 in Online Appendix A2.1 shows that with this scaling, the MILP strictly prioritizes implicit candidate edges over the candidate edges for deletion. Overall, because we do not observe whether the pairwise comparisons in the DAG are correct or spurious, we test empirically whether it is effective to add those implicit edges in the preference DAG (see Section A6 in the online appendix). We notice that by adding them in the DAG construction process and obtaining denser DAGs, the improvements in the prediction performance are significant.

2.6. Maximum Likelihood Estimation of the DAG-Based Choice Model

Once we infer the customers’ DAGs, we use maximum likelihood estimation (MLE) to calibrate a probability distribution over the full rankings consistent with these DAGs. In order to compute the panel data log-likelihood function, we consider only revealed
preferences that are consistent with the inferred DAGs. That is, if during the DAG construction process no cycle was formed, then every transaction pair \((a_{ij}, S_h)\) (which can be represented as a star graph with head \(a_{ij}\) and set of leaves \(S_h \setminus \{a_{ij}\}\), is a subgraph of the corresponding DAG \(D_i\). In case a cycle was formed, say for customer \(i\), consider a transaction \((a_{ij}, S_h)\) such that one of the edges \((a_{ij}, a_k)\), with \(a_k \in S_h \setminus \{a_{ij}\}\), was deleted in the decycling procedure. Because \((a_{ij}, a_k)\) was part of a cycle, it follows that there is a directed path from \(a_k\) to \(a_{ij}\) in the final DAG \(D_i\). This implies that conditioned on the customer having DAG \(D_i\), she did not consider product \(a_k\) when choosing \(a_{ij}\) even though \(a_k\) was on offer. Therefore, product \(a_k\) can be ignored for computing data log-likelihood.

Once we filter out these inconsistent preferences, the likelihood function that we maximize to calibrate the model is just the sum of likelihoods of customers’ partial orders, that is,

\[
\log \mathcal{L}(\text{Panel Data}) = \sum_{i=1}^{m} \log \lambda(D_i) \\
= \sum_{i=1}^{m} \log \left( \sum_{\sigma \in S_i} \lambda(\sigma) \right), \tag{1}
\]

where \(\lambda(D)\) denotes the probability of DAG \(D\) under distribution \(\lambda\). We interpret any partial order \(D\) as a censored representation of the underlying full rankings \(\sigma\) that a customer could sample. Therefore, the probability of DAG \(D\) under \(\lambda\) is equal to the probability of sampling a ranking \(\sigma \in S_D\) that is consistent with \(D\). We thus obtain that

\[
\lambda(D) = \sum_{\sigma \in S_D} \lambda(\sigma). \tag{2}
\]

Proposition A2 in Online Appendix A2.1 provides a formal justification of the log-likelihood expression.

The tractability of the MLE problem depends on the structure of the distribution \(\lambda\) over preference lists. In the next section, we discuss how we resolve the computational issues that arise in solving (1) and computing the choice probabilities laid out in Section 2.1.

### 3. Theoretical Analysis of the DAG-Based MNL Model

We now focus on two computational problems that arise in using our model with data: computing (a) the probability of a DAG \(D\) and (b) the choice probability given an offer set \(S\) conditioned on a DAG \(D\). The first computation is needed to solve the estimation problem discussed above, and the second one is needed to predict the purchase of a customer with DAG \(D\). Both computations are difficult for a general DAG \(D\). In fact, even the problem of counting the number of total orders that are consistent with a given DAG \(D\) (i.e., the cardinality of set \(S_D\)) is a #P-hard problem (Birgwell and Winkler, 1991). For that reason, we limit our attention to the standard Plackett-Luce (PL) (Marden, 1995) model for the underlying distribution \(\lambda\) over rankings, for which at least there is a closed form expression for the likelihood of a ranking. In the PL model, each product \(a \in \mathcal{N}\) is associated with parameter (i.e., weight) \(v_a > 0\). The probability of sampling ranking \(\sigma\) is given by

\[
\lambda(\sigma) = \prod_{i=1}^{n} \frac{v_{\sigma_i}}{\sum_{j=1}^{n} v_{\sigma_j}}. 
\]

For brevity of notation, we also use \(v_i\) to refer to \(v_{\sigma_i}\), for a given indexing of the products. As shown by Jagabathula and Vulcano (2018), the choice probabilities under the PL model are consistent with those under a standard MNL model with the same parameters \((v_a)_{a \in \mathcal{N}}\). In other words, we have

\[
\Pr(a|S) = \frac{v_i}{\sum_{j \in S} v_j}. 
\]

Under both the PL and MNL models. Under the PL model, the choice probability \(\Pr(a|S)\) is equal to the probability of a star DAG with edges from product \(a_i\) to all the products in the set \(S \setminus \{a_i\}\) because this DAG always results in the choice of \(a_i\) from \(S\). The choice probability under the MNL model, on the other hand, can be derived from its random utility specification (Ben-Akiva and Lerman, 1985). Because of this equivalence of both models on the choice probabilities, we use the terms PL and MNL interchangeably.

In the rest of this section, we first obtain easy-to-compute, closed-form analytical bounds for the likelihood of a DAG under the PL model. The lower bound stems from treating the DAG as a forest of directed trees. The upper bound is obtained by deleting some edges from the DAG. We derive a closed-form upper bound for the error made when using any of these two bounds as an approximation to the true probability. In particular, we use the lower bound to solve the MLE problem for training our model in Section 4. We then focus on computing the choice probabilities conditioned on a DAG, for which we also develop both lower and upper bounds. But for practical matters, we use the approximation to the true choice probabilities proposed by Jagabathula and Vulcano (2018), whose value is also provably in between the same bounds of the true choice probabilities. We derive a closed-form upper bound for the error under this approximation. Moreover, using this approximation, we show that we are able to obtain a promising performance on the data set studied in Section 4.
3.1. Tractable Analytical Bounds for the Likelihood of a DAG

We first focus on the problem of computing the likelihood $\lambda(D) = \sum_{v \in S_D} \lambda(\sigma)$ of a DAG $D$ under the PL model. Jagabathula and Vulcano (2018) derive a closed-form expression for $\lambda(D)$ when $D$ satisfies a special structure. To state the result, we introduce the concept of reachability. The reachability function $\Psi_D$ of DAG $D$ maps each node $a$ to the set of nodes that can be reached from $a$ through the edges in $D$. More precisely, $\Psi_D(a) = \{b: \text{ there is a directed path from } a \text{ to } b \in D\}$. We assume that a node is reachable from itself, so $a \in \Psi_D(a)$ for all $a$, and $\Psi_D$ is always nonempty. The DAG $D$ is equivalently described by the reachability function $\Psi_D(\cdot)$ of its nodes. Without loss of generality, we represent the DAG $D$ by its unique transitive reduction, which is the unique graph with the fewest number of edges possible and the same reachability function as $D$. We start from DAGs that are forests of directed trees* with unique roots, where root is any node with no incoming edges. It is then shown in Jagabathula and Vulcano (2018, proposition 3.2) that

$$\lambda(D) = \prod_{a \in N} \frac{v_a}{\sum_{b \in \Psi_D(a)} v_b},$$

(3)

whenever $D$ is a forest of directed trees, each with a unique root.

Jagabathula and Vulcano (2018) propose using (3) to approximate the probability of a general DAG, even if it is not a directed tree. For the general case, they do not provide any guarantees for this approximation, suggesting that computing the probability of a DAG is difficult in the presence of v-nodes, defined as the nodes with at least two incoming edges. We now show that (3) provides a lower bound approximation for the probability of a general DAG. In particular, we establish the following result:

**Proposition 1.** Under the PL model, we have that for any DAG $D$,

$$\bar{\lambda}(D) \leq \lambda(D),$$

where $\bar{\lambda}(D) = \prod_{a \in N} \frac{v_a}{\sum_{b \in \Psi_D(a)} v_b}$. The inequality above is strict if $D$ has at least one v-node.

The proof is rather involved and the details are provided in Online Appendix A2.1. Here, we provide a sketch. The proof uses induction on v-degree, $k$, of $D$, defined as the sum of the degrees of the v-nodes in $D$ minus the number of v-nodes. The base case of $k = 0$ follows from (3) because the v-degree of $D$ is zero if and only if $D$ is a forest of directed trees, each with a unique root. To establish the induction step, we consider a DAG with v-degree of $k + 1$ and carry out the following “splitting” operations to create a DAG with v-degree of at most $k$, in order to apply the induction hypothesis. We pick a v-node $a_y$ with the property that the subgraph $D[a_y]$ "hanging" from node $a_y$ —which is induced by $D$ on the set of nodes $\Psi_D(a_y)$—is a directed tree. Such a v-node always exists (at the minimum, it is a leaf in the DAG). Then, we split $D$ into DAG $D[a_y]$ and the remaining DAG $\tilde{D}[a_y]$, which is induced by $D$ on the set of nodes $\{N \setminus \Psi_D(a_y)\} \cup \{a_y\}$. We then split the node $a_y$ in DAG $\tilde{D}[a_y]$ to create a new copy $a_y'$ such that one of the incoming edges into $a_y$ moves to the node $a_y'$, whereas the other incoming edges remain with node $a_y$, resulting in the DAG $D_y^{\text{split}}$. This splitting operation results in new nodes for which the PL parameters values must be appropriately defined. With these parameter values, we show that our splitting operation can only reduce the probability of the resulting collection of DAGs. We then establish the result by invoking the induction hypothesis on $D_y^{\text{split}}$, which by construction has a v-degree of at most $k$.

An upper bound for the likelihood of a DAG $D$ can be readily obtained by deleting some edges in $D$. Deleting an edge strictly increases the set of permutations that are consistent with the DAG, so for any $\hat{D} \subset D$, we have that $S_{\hat{D}} \supsetneq S_D$, where recall that $S_D$ is the set of all rankings that are consistent with $D$. It thus follows that $\lambda(D) \geq \lambda(\hat{D})$. We state this result formally in the following proposition and prove it in Online Appendix A2.1.

**Proposition 2.** For any two DAGs $D$ and $\hat{D}$ such that $\hat{D} \subset D$, we must have that $\lambda(D) \leq \lambda(\hat{D})$, with strict inequality under the PL model if all the parameter values are strictly positive.

Note that the above result is true for any distribution $\lambda$ and not just for the PL model. To obtain a tractable upper bound under the PL model, we choose a DAG $\hat{D}$ that is a forest of directed trees, each with a unique root. Multiple such DAGs may exist, and we can pick the one that provides the tightest upper bound. Finding the optimal DAG $\hat{D}$ is a hard problem, so we propose a greedy heuristic that recursively deletes all, except one, of the incoming edges to each of the v-nodes in the DAG. See Section A5 in the online appendix for details of the algorithm.

Next, we explore the tightness of the developed lower and upper bounds of a DAG’s likelihood. Let $R(D, D) = \lambda(D) / \bar{\lambda}(D)$ denote the ratio between them for any DAG $D \subset D$. It is clear that $R(D, D) \geq 1$ for all $\hat{D} \subset D$, so we express our tightness guarantee by deriving a parametric upper bound for $R(D, \hat{D})$. For that, let $\ell$ denote the size of the largest reachability set in DAG $D$, that is, $\ell = \max_{a \in A} |\Psi_D(a)|$, and let $p$ denote the number of nodes with v-nodes in their reachability sets, that is, $p = |\{a \in N : \exists v\text{-node } b \in \Psi_P(a)\}|$. Further, let $\Delta := \max_{a \in A} \max_{b \in \Psi_D(a)} \left|\frac{v_b}{e_a}\right|$ be the maximum
ratio between the weights of nodes within the same directed path in the DAG. We can derive the following guarantee:

**Proposition 3.** Consider DAGs $D$ and $\bar{D}$ such that $\bar{D} \subseteq D$ is obtained by deleting all, except one, of the incoming edges that the bound above applies to each of the $v$-nodes. Then, we have that

\[
0 \leq \log R(D, \bar{D}) \leq p \cdot \log(1 + \ell \cdot \Delta).
\]

Moreover, if $\Delta \in o(n^{-2})$, where $n$ is the number of nodes, then we have that \(\lim_{n \to \infty} \lambda(D) = \lambda(\bar{D})\).

The proof is given in Online Appendix A2.1. Note that the bound above applies to any DAG $\bar{D}$ that satisfies the conditions stated in the proposition; in particular, it applies to the DAG $\bar{D}$ that is constructed using the heuristic described in Section A5 in the online appendix. The above result shows that our approximation guarantee depends on the number of $v$-nodes in the DAG (more precisely, on the number of nodes with $v$-nodes in their reachability sets) and the ratio $\Delta$ of PL parameters. The bound is derived in the most general setting and, in this generality, it is tight. For instance, when there are no $v$-nodes, then $p = 0$ and we obtain the guarantee $R(D, \bar{D}) = 1$, as expected.\(^3\) In several other cases, however, the bound can be weak. In fact, we show on the actual sales data that the approximation ratio $R(D, \bar{D})$ can be much smaller than what is suggested by our theoretical bound.

### 3.2. Tractable Analytical Bounds for the Purchase Probability Prediction

We now turn to the prediction problem, i.e., that of predicting the probability that a customer with DAG $D$ purchases product $a_i$ from offer set $S$. Recall that a customer with DAG $D$ always samples a preference list $\sigma$ that is consistent with $D$, that is, $\sigma \in S_D$. The probability that such a customer will purchase product $a_i$ from offer set $S$ is then equal to the conditional probability that the sampled permutation is consistent with the star DAG $C(a_i, S)$, in which there are edges only from $a_i$ to all the products in $S \setminus \{a_i\}$. Thus, the probability $f(a_i, S, D)$ that the customer will purchase $a_i$ from $S$ is given by

\[
f(a_i, S, D) = \Pr(S_{C(a_i, S)} \cap S_D) = \frac{\Pr(S_{C(a_i, S)} \cap S_D)}{\Pr(S_D)},
\]

where the second equality follows from the Bayes rule. Now, given any offer set $S$, let $h_D(S) \subset S$ denote the subset of “heads” (i.e., the subset of nodes without parents) in the subgraph of the transitive closure of $D$ restricted to set $S$. Because it follows by definition that every node in $S \setminus h_D(S)$ has at least one incoming edge from a node in $h_D(S)$, the customer with DAG $D$ will never purchase the products in $S \setminus h_D(S)$. Therefore, we obtain that $f(a_i, S, D) = 0$ for all $a_i \in S \setminus h_D(S)$. For the products in $h_D(S)$, the probability of choosing $a_i$ from $S$ depends on the probability of the DAG representing the collection of permutations $S_{C(a_i, S)} \cap S_D$, which corresponds to the merged DAG $D \cup C(a_i, S)$ obtained by taking the union of the graphs $D$ and $C(a_i, S)$. We thus obtain

\[
f(a_i, S, D) = \frac{\lambda(D \cup C(a_i, S))}{\lambda(D)}, \quad \text{if } a_i \in h_D(S),
\]

In computing the choice probabilities for the products in $h_D(S)$, we run into similar #P-hardness issues as mentioned above. To deal with this challenge, Jagabathula and Vulcano (2018) focus on the special case when $D$ is a forest of directed trees, each with a unique root, and all the nodes in $h_D(S)$ are roots in $D$. With these assumptions, Jagabathula and Vulcano (2018, proposition 3.3) show that

\[
f(a_i, S, D) = \frac{\tau(D \cup C(a_i, S))}{\tau(D)}, \quad \text{for } a_i \in h_D(S),
\]

where we define $\tau(D) = \sum_{a_i \in h_D(S)} \Pr(a_i)$. More generally, they propose using the above expression as an approximation but do not provide any performance guarantee. We can now use the results of Propositions 1 and 2 to obtain bounds for the choice probability prediction. For that, we define

\[
\tilde{f}(a_i, S, D) := \frac{\lambda(D \cup C(a_i, S))}{\lambda(D)} \quad \text{and} \quad \hat{f}(a_i, S, D) = \frac{\tau(D \cup C(a_i, S))}{\lambda(D)},
\]

where for any DAG $D$, we let $\bar{D}$ denote the DAG with the properties described in Proposition 3. We can now establish the following:

**Corollary 1.** For a given DAG $D$, under the Plackett-Luce model, the following tractable bounds of purchase probabilities apply

\[
\tilde{f}(a_i, S, D) \leq f(a_i, S, D) \leq \hat{f}(a_i, S, D), \quad \text{and} \quad \tilde{f}(a_i, S, D) \leq \hat{f}(a_i, S, D),
\]

where

\[
\hat{f}(a_i, S, D) = \frac{\lambda(D \cup C(a_i, S))}{\lambda(D)} = \frac{\Pr(a_i \in h_D(S))}{\Pr(S_D)} \sum_{a_i \in h_D(S)} \Pr(a_i).
\]

This corollary follows immediately from our definitions and the results of Propositions 1 and 2. We are also able to provide a parametric approximation guarantee similar to the one in Proposition 3. We define the parameters $\ell$ and $p$ as above, but now for the merged
The average number of edges in the DAG for individuals with cycles in the preference graph; Height

We now test our proposal on the IRI Academic data set (Bronnenberg et al. 2008), which consists of real-world purchase transactions from grocery and drug stores. We compare the predictive power of our method against standard benchmarks, such as the latent-class MNL and the random parameters logit (RPL) models. We show that our method significantly outperforms the benchmarks on holdout data on standard performance metrics for measuring predictive accuracy. In the next section, we show how our method can be used to personalize product promotions.

4. Empirical Study

We now test our proposal on the IRI Academic data set (Bronnenberg et al. 2008), which consists of real-world purchase transactions from grocery and drug stores. We compare the predictive power of our method against standard benchmarks, such as the latent-class MNL and the random parameters logit (RPL) models. We show that our method significantly outperforms the benchmarks on holdout data on standard performance metrics for measuring predictive accuracy. In the next section, we show how our method can be used to personalize product promotions.

4.1. Data Analysis

We analyze consumer packaged goods purchase transaction data for year 2007 over a chain of grocery stores in two large Behavior Scan markets in the United States. For every purchase instance in the data set, we have the week and the store ID of the purchase, the universal product code (UPC) of the purchased item, the panel ID of the purchasing customer, quantity purchased, price paid, and an indicator of whether the purchased item is on promotion or not. Overall, we considered 27 categories (see Table 1) of products out of the available 31 categories, skipping 4 because of data sparsity.

<table>
<thead>
<tr>
<th>Category</th>
<th>expanded name</th>
<th>Vend</th>
<th>AvOS</th>
<th>Total</th>
<th>≥2 sales</th>
<th>AvTr</th>
<th>wOCyc</th>
<th>% Del</th>
<th>Dens1</th>
<th>Dens2</th>
<th>Height1</th>
<th>Height2</th>
</tr>
</thead>
<tbody>
<tr>
<td>beer</td>
<td>Beer</td>
<td>67</td>
<td>43.87</td>
<td>1,796</td>
<td>1,154</td>
<td>7.11</td>
<td>466</td>
<td>1.10</td>
<td>208.11</td>
<td>319.51</td>
<td>2.35</td>
<td>4.20</td>
</tr>
<tr>
<td>carbev</td>
<td>Carbonated beverages</td>
<td>46</td>
<td>15.36</td>
<td>4,677</td>
<td>3,837</td>
<td>17.63</td>
<td>666</td>
<td>3.38</td>
<td>92.45</td>
<td>148.11</td>
<td>2.22</td>
<td>4.43</td>
</tr>
<tr>
<td>cigets</td>
<td>Cigarettes</td>
<td>13</td>
<td>7.14</td>
<td>452</td>
<td>307</td>
<td>10.39</td>
<td>233</td>
<td>2.95</td>
<td>35.37</td>
<td>47.53</td>
<td>2.45</td>
<td>3.53</td>
</tr>
<tr>
<td>coffee</td>
<td>Coffee</td>
<td>59</td>
<td>19.80</td>
<td>3,101</td>
<td>2,255</td>
<td>5.59</td>
<td>1,098</td>
<td>1.93</td>
<td>130.37</td>
<td>186.81</td>
<td>2.85</td>
<td>4.66</td>
</tr>
<tr>
<td>colder</td>
<td>Cold cereal</td>
<td>39</td>
<td>17.66</td>
<td>4,438</td>
<td>3,998</td>
<td>10.94</td>
<td>687</td>
<td>3.69</td>
<td>88.41</td>
<td>158.14</td>
<td>2.47</td>
<td>4.81</td>
</tr>
<tr>
<td>deod</td>
<td>Doordarant</td>
<td>32</td>
<td>14.55</td>
<td>1,345</td>
<td>653</td>
<td>3.47</td>
<td>347</td>
<td>2.63</td>
<td>98.48</td>
<td>198.74</td>
<td>2.67</td>
<td>4.06</td>
</tr>
<tr>
<td>factiss</td>
<td>Facial tissue</td>
<td>10</td>
<td>4.17</td>
<td>2,967</td>
<td>2,063</td>
<td>4.96</td>
<td>1,148</td>
<td>7.64</td>
<td>22.48</td>
<td>28.36</td>
<td>2.79</td>
<td>4.44</td>
</tr>
<tr>
<td>fzpizza</td>
<td>Frozen pizza</td>
<td>38</td>
<td>15.50</td>
<td>3,460</td>
<td>2,946</td>
<td>7.83</td>
<td>1,213</td>
<td>3.88</td>
<td>179.24</td>
<td>349.34</td>
<td>2.65</td>
<td>5.51</td>
</tr>
<tr>
<td>hhclean</td>
<td>Household cleaners</td>
<td>68</td>
<td>31.42</td>
<td>2,725</td>
<td>1,699</td>
<td>4.14</td>
<td>388</td>
<td>1.58</td>
<td>170.89</td>
<td>269.30</td>
<td>2.94</td>
<td>4.72</td>
</tr>
<tr>
<td>hotdog</td>
<td>Hot dogs</td>
<td>41</td>
<td>16.81</td>
<td>3,318</td>
<td>2,187</td>
<td>3.82</td>
<td>1,190</td>
<td>1.89</td>
<td>89.95</td>
<td>127.86</td>
<td>2.84</td>
<td>4.37</td>
</tr>
<tr>
<td>laudent</td>
<td>Laundry deterrent</td>
<td>18</td>
<td>10.08</td>
<td>3,196</td>
<td>2,181</td>
<td>4.04</td>
<td>1,367</td>
<td>3.62</td>
<td>49.28</td>
<td>69.71</td>
<td>2.60</td>
<td>4.19</td>
</tr>
<tr>
<td>margbutr</td>
<td>Margarine/Butter</td>
<td>16</td>
<td>10.35</td>
<td>3,474</td>
<td>2,750</td>
<td>5.65</td>
<td>1,405</td>
<td>4.38</td>
<td>47.00</td>
<td>71.07</td>
<td>2.85</td>
<td>4.45</td>
</tr>
<tr>
<td>mayo</td>
<td>Mayonnaise</td>
<td>14</td>
<td>6.86</td>
<td>3,761</td>
<td>2,586</td>
<td>3.28</td>
<td>1,853</td>
<td>3.41</td>
<td>35.10</td>
<td>44.24</td>
<td>2.46</td>
<td>3.59</td>
</tr>
<tr>
<td>milk</td>
<td>Milk</td>
<td>33</td>
<td>11.69</td>
<td>4,851</td>
<td>4,652</td>
<td>14.90</td>
<td>1,674</td>
<td>3.12</td>
<td>78.84</td>
<td>118.78</td>
<td>2.88</td>
<td>4.70</td>
</tr>
<tr>
<td>mustketc</td>
<td>Mustard</td>
<td>52</td>
<td>17.07</td>
<td>3,728</td>
<td>2,515</td>
<td>3.66</td>
<td>895</td>
<td>1.83</td>
<td>107.38</td>
<td>142.27</td>
<td>2.77</td>
<td>4.06</td>
</tr>
<tr>
<td>paptowel</td>
<td>Paper towels</td>
<td>11</td>
<td>6.94</td>
<td>3,072</td>
<td>2,051</td>
<td>5.20</td>
<td>977</td>
<td>7.20</td>
<td>29.82</td>
<td>42.74</td>
<td>2.79</td>
<td>4.44</td>
</tr>
<tr>
<td>peanbutw</td>
<td>Peanut butter</td>
<td>19</td>
<td>7.99</td>
<td>3,153</td>
<td>1,923</td>
<td>3.89</td>
<td>1,232</td>
<td>3.13</td>
<td>43.38</td>
<td>56.88</td>
<td>2.55</td>
<td>3.70</td>
</tr>
<tr>
<td>saltsocks</td>
<td>Salt snacks</td>
<td>95</td>
<td>26.79</td>
<td>4,727</td>
<td>4,446</td>
<td>15.09</td>
<td>629</td>
<td>2.58</td>
<td>179.08</td>
<td>320.94</td>
<td>2.38</td>
<td>5.26</td>
</tr>
<tr>
<td>champ</td>
<td>Shampoo</td>
<td>41</td>
<td>18.74</td>
<td>1,466</td>
<td>738</td>
<td>3.73</td>
<td>357</td>
<td>1.67</td>
<td>127.54</td>
<td>172.51</td>
<td>2.62</td>
<td>4.16</td>
</tr>
<tr>
<td>soup</td>
<td>Soup</td>
<td>90</td>
<td>32.87</td>
<td>4,636</td>
<td>3,422</td>
<td>12.02</td>
<td>988</td>
<td>1.71</td>
<td>207.27</td>
<td>353.24</td>
<td>2.67</td>
<td>5.07</td>
</tr>
<tr>
<td>spagausc</td>
<td>Spaghetti/Italian sauce</td>
<td>52</td>
<td>17.85</td>
<td>3,473</td>
<td>2,698</td>
<td>5.46</td>
<td>1,363</td>
<td>1.98</td>
<td>120.92</td>
<td>178.83</td>
<td>2.76</td>
<td>4.62</td>
</tr>
<tr>
<td>sugarsub</td>
<td>Sugar substitutes</td>
<td>10</td>
<td>5.05</td>
<td>750</td>
<td>308</td>
<td>3.30</td>
<td>258</td>
<td>3.76</td>
<td>28.45</td>
<td>33.74</td>
<td>2.41</td>
<td>3.62</td>
</tr>
<tr>
<td>totissu</td>
<td>Toilet tissue</td>
<td>11</td>
<td>7.66</td>
<td>3,760</td>
<td>2,817</td>
<td>5.10</td>
<td>1,552</td>
<td>6.86</td>
<td>32.25</td>
<td>47.75</td>
<td>2.71</td>
<td>4.52</td>
</tr>
<tr>
<td>toothbr</td>
<td>Toothbrushes</td>
<td>36</td>
<td>15.86</td>
<td>1,115</td>
<td>499</td>
<td>3.06</td>
<td>260</td>
<td>1.96</td>
<td>90.46</td>
<td>124.38</td>
<td>2.70</td>
<td>4.15</td>
</tr>
<tr>
<td>toothpa</td>
<td>Toothpaste</td>
<td>25</td>
<td>12.05</td>
<td>2,110</td>
<td>1,186</td>
<td>3.58</td>
<td>708</td>
<td>2.01</td>
<td>64.15</td>
<td>86.54</td>
<td>2.46</td>
<td>3.72</td>
</tr>
<tr>
<td>yogurt</td>
<td>Yogurt</td>
<td>26</td>
<td>9.84</td>
<td>3,766</td>
<td>3,491</td>
<td>19.81</td>
<td>1,349</td>
<td>5.41</td>
<td>55.55</td>
<td>86.79</td>
<td>2.57</td>
<td>4.79</td>
</tr>
</tbody>
</table>

Notes: Vend, the number of products obtained in the training data after aggregating different UPCs by vendor; AvOS, the average number of vendors in the offer set; Total, the total number of individuals in the training data; ≥2 sales, the number of individuals with at least two purchases in the training data; AvTr, the average number of transactions in the training data. Analyzing customers under PO-MNL Promotion model, we report [wOCyc], the number of individuals without cycles in the preference graph; % Del, the average percentage of edges deleted after preference graph decycling; Dens1, the average number of edges in the DAG for individuals without cycles in the preference graph; Dens2, the average number of edges in the DAG for individuals with cycles in the preference graph; Height1, the average height of the DAG for individuals without cycles in the preference graph; Height2, the average height of the DAG for individuals with cycles in the preference graph.
The data consist of 1.2 million records of weekly purchase transactions from 84 thousand customers over 52 weeks. The transaction data are split into the training set, consisting of the first 26 weeks of purchase observations, and the test set, consisting of the last 26 weeks. We considered only customers with two or more transactions over the training period. After filtering out customers with less than two observations over the training data within each category, we were left with a total of 64 thousand customers and 1.1 million purchase transactions. To alleviate data sparsity, we aggregated all the items with the same vendor code (comprising digits 3 through 7 in 13-digit-long UPC code) into a unique “product.” For each transaction, we know the purchased product, say \( a_j \), but we do not have explicit knowledge of the offer set. As a result, we approximately constructed the offer set \( S \) by taking the union of all the products that were purchased in the same category as \( a_j \), in the same week and in the same store. The transaction also contains a promotion indicator, which is set to one if product \( a_j \) was on display or price promotion at the time of purchase. Using this information, we also approximately constructed the set of promoted products consisting of all the products in \( S \) that were on promotion at least once during the week.

Using the purchase transactions within training data, we constructed a DAG for each of the customers according to our model, which we label partial order MNL (PO-MNL) Promotion model from here onwards. The construction resulted in the set \( \text{woCyc} \) of customers with preference graphs without cycles and the set \( \text{Cyc} \) of customers with preference graphs with cycles. For the customers in \( \text{Cyc} \), we decycled their preference graphs using MILP (1) formulated in Section 2.4 to obtain DAGs. We implemented this decycling procedure in Python (version 2.7.2) using Gurobi (version 7.0) as the optimization engine and ran it on a 3.0Ghz processor with 16GB of RAM. We set the time limit to 30 seconds. The mean running time was eight seconds, with most instances solved to optimality.

On average, we deleted only 3.4% of edges during this process. As shown in Table 1, the preference graphs of 40% of customers had no cycles. Customers with cycles in their preference graphs generally had more purchases and, hence, denser graphs than customers without cycles, for all categories of products. In particular, the DAGs of customers in \( \text{Cyc} \) on average had 61% more edges and 74% larger height (defined as the length of the longest directed path in the graph after decycling) than the customers in \( \text{woCyc} \). On average, each category had 38 vendors and about 45.5% of vendors were offered in each store and week combination.

4.2. Models Compared

We fitted our PO-MNL Promotion model to the data above and compared its predictive performance against two widely used benchmarks: the LC-MNL and the RPL models. Both of these models belong to the general class of random utility models, so that in each purchase instance, a customer samples product utilities and then chooses the product with the highest value.

4.2.1. LC PO-MNL Promotion Model. First, we fitted a single class PO-MNL Promotion model to the DAGs. Recall that to deal with promoted products we expanded our product universe to consist of two copies, a promoted one and a nonpromoted one, of each product. Following the notation introduced in Section 2.1, products \( a_1, a_2, \ldots, a_n \) are the nonpromoted copies and \( a_{n+1}, a_{n+2}, \ldots, a_{2n} \) are the promoted copies. For any \( f \in [n] \), product \( a_{j+n} \) is the promoted copy corresponding to product \( a_j \). We let \( \tau_j \) denote the MNL parameter of product \( a_j \), so that \( v_j = \exp(\tau_j) \). We parameterize the model as follows: for any \( a_j \in N^n \),

\[
\tau_j = \begin{cases} \beta_j^0, & \text{if } 1 \leq j \leq n \\ \beta_{j-n} + \beta_j, & \text{if } n+1 \leq j \leq 2n, \end{cases}
\]

where \( \beta_j^0 \) is the utility derived from the nonpromoted copy of product \( j \in [n] \) and \( \beta_j \) is the additional utility from the promotion feature. We estimate the parameters by solving the following approximated regularized likelihood problem:

\[
\max_{\beta, \beta_0} \sum_{j=1}^{2n} \left[ \tau_j - \log\left( \sum_{a_i \in \Psi_D'(a_j)} \exp(\tau_j) \right) \right] - \alpha \left( \|\beta^0\|_1 + \|\beta\|_1 \right)
\]

(6)

where \( \Psi_D'(a_j) \) is the set of nodes that are reachable from \( a_j \) in DAG \( D \), of customer \( j \). To arrive at the above approximation, we used the lower bound \( \lambda \) for computing the likelihood of a DAG, as discussed in Section 3. When the value of \( \alpha \) is fixed, it can be shown that the optimization problem in (6) is globally concave and therefore can be solved efficiently (Train 2009). We tuned the value of \( \alpha \) by five-fold cross-validation. The above likelihood problem is exact only if every DAG \( D_i \) is a forest of directed trees, each with a unique root. Otherwise, as shown in Proposition 1, it provides a lower bound. Once we estimated the parameters, we predicted purchase probabilities on holdout data using the following approximation:

\[
\hat{f}(a_j, S, D) = \frac{\sum_{i \in h_D(S)} \Psi_D'(a_j)}{\sum_{i \in h_D(S)} \Psi_D'(a_i)}, \quad \text{if } a_j \in h_D(S),
\]

(7)

otherwise.

In Online Appendix A5, we provide empirical evidence that this approximation is a good and easy-to-compute proxy for the exact \( f(a_j, S, D) \).
To account for heterogeneity among the customers, we also fitted a K latent class PO-MNL Promotion model, which assumes that each customer belongs to one of the h ∈ {1, ..., K} latent classes. A customer from class h samples her DAG according to the PO-MNL Promotion model with parameters τ_{jh} defined as

$$\tau_{jh} = \begin{cases} \beta_{jh}^\prime & \text{if } 1 \leq j \leq n \\ \beta_{j-nh}^\prime + \beta_{j-nh} & \text{if } n+1 \leq j \leq 2n. \end{cases}$$

We denote the prior probability that a customer belongs to class h by γ_h ≥ 0, so that \( \sum_{h=1}^{K} \gamma_h = 1 \). Then, similar to the PO-MNL model, we estimate the parameters by solving the following approximated regularized likelihood problem:

$$\max_{\beta, \gamma} \sum_{i=1}^{m} \log \left( \sum_{h=1}^{K} \gamma_h \prod_{j=1}^{2n} \frac{\exp(\tau_{jh})}{\sum_{d=1}^{K} \gamma_d \prod_{j=1}^{2n} \exp(\tau_{jd})} \right) - a \sum_{h=1}^{K} (||\beta_h^\prime||_1 + ||\beta_h^\prime||_1).$$

The above optimization problem is nonconcave for K > 1, even with the value of a fixed. Therefore, we use the standard expectation-maximization (EM)-based algorithm described in Train (2009) to obtain a stationary point. Specifically, we initialize the EM with a random allocation of customers to one of the K classes, resulting in an initial allocation D_1, D_2, ..., D_K, which form a partition of the collection of all the customers. Then we set \( \gamma_{h}^{(0)} = |D_h|/(\sum_{h=1}^{K} |D_h|) \). In order to get a parameter vector \( \tau_{h}^{(0)} \), we fit a PO-MNL Promotion model, described above, to each subset of customers. After calibrating the model, we make predictions in the following way. For each individual i with DAG D_i, we estimate the posterior membership probabilities \( \hat{\gamma}_{ih} \) for each class h ∈ {1, ..., K}:

$$\hat{\gamma}_{ih} = \frac{\gamma_h \prod_{j=1}^{2n} \left[ \frac{v_{jh}}{\sum_{d=1}^{K} \gamma_d \prod_{j=1}^{2n} \exp(\tau_{jd})} \right]^{a} \exp(\tau_{jh})}{\sum_{d=1}^{K} \gamma_d \prod_{j=1}^{2n} \left[ \frac{v_{dh}}{\sum_{d=1}^{K} \gamma_d \prod_{j=1}^{2n} \exp(\tau_{hd})} \right]}$$

where \( v_{jh} = \exp(\tau_{jh}) \), and then make the prediction

$$\hat{f}(a_j, S, D_i) = \sum_{h=1}^{K} \hat{\gamma}_{ih} \hat{f}_h(a_j, S, D),$$

where \( \hat{f}_h(a_j, S, D) \) are the approximated probabilities in (7). We estimate the model for K = 1, 2, ..., 10 and report the best performance measure from these 10 variants, for every performance metric that we introduce below.

### 4.2.2. Benchmark Models

We compare our models with two benchmark models succinctly described here (see Online Appendix A4 for details). The first benchmark is the LC-MNL choice model with K latent classes. In this model, each customer belongs to one unobservable class, and customers from class h ∈ {1, ..., K} make purchases according to the MNL model associated with that class. The model is described by the parameters of the MNL characterizing each class and by the prior probabilities of customers belonging to each of the classes. Once the model parameters are estimated, we make customer-level predictions by averaging the predictions from K single-class models, weighted by the posterior probability of class membership. Similar to the LC PO-MNL Promotion model, we estimated the model for K = 1, 2, ..., 10 and report the best performance measure from these 10 variants, for every performance metric that we introduce in the upcoming subsection.

The second benchmark model that also captures heterogeneity in customer preferences is the RPL model, which assumes that in each purchase instance, a customer samples the \( \beta \) parameters of the product utilities according to some distribution and then makes the choice according to a single-class MNL model with parameter vector \( \beta \). In comparison with LC-MNL benchmark, RPL model allows the parameter vectors \( \beta \) to take a continuum of values. Particularly, we assume that parameter vector \( \beta \) is sampled according to multivariate normal distribution with mean \( \mu \) and diagonal variance-covariance matrix \( \Sigma \), that is, \( \beta \sim N(\mu, \Sigma) \). Calibration of the RPL choice model is based on the sample average approximation approach, which is computationally intensive.

In both benchmark models, we account for product promotion status by introducing a product-specific parameter to capture the additional utility from this feature like we did for the PO-MNL Promotion model.

### 4.3. Prediction Performance Measures

Broadly, we want to predict the product purchased by customer \( i \) in period time \( t+1 \) given the sales transaction data of the customer up to period \( t \) and the set of offered and promoted items at time \( t+1 \). For that, we compare the models based on a one-step ahead prediction experiment for every category under two different metrics: \( \chi^2 \) and miss rate. Recall that in our case, a period corresponds to a week. For each category of products, we separately fit the benchmark models and the PO-MNL Promotion and LC PO-MNL Promotion models to the following three subsets of individuals: customers without cycles in their preference graph, customers with cycles in their preference graph, and the combination of all the customers. Then, for each category and each subset of customers, we report the comparisons of the different fitted models.

The “chi-square” score is computed as follows:

$$\chi^2 = \frac{1}{|U|} \sum_{i \in U} \frac{(n_{ij} - \hat{n}_{ij})^2}{0.5 + \hat{n}_{ij}},$$

where \( \hat{n}_{ij} = \frac{1}{T_i} \sum_{t=1}^{T_i} \hat{f}(j_{it}, t), \)
where $U$ is the set of all individuals, and $n_{ij}$ is the observed number of times individual $i$ purchased product $j$ during the time horizon of length $T_i$. The indicator function $f(j_i, t)$ takes value one if the product indexed $j$ has the highest choice probability for individual $i$ at time $t$ and zero otherwise. This score measures the ability of the model combinations to predict the aggregate market shares of the products purchased by every individual, where lower scores indicate better prediction accuracy. The 0.5 added in the denominator allows to deal with undefined instances.

The miss rate is computed as follows:

$$\text{miss rate} = \frac{1}{|U|} \sum_{i \in U} \frac{1}{T_i} \sum_{t=1}^{T_i} \mathbb{I}[f(j_i, t) = 1],$$

where $\mathbb{I}[A]$ is the indicator function that takes value one if $A$ is true and zero otherwise and $j_i$ is the index of the item purchased at time $t$ by individual $i$. Miss rate is a more stringent predictive measure than chi-square score, because it rewards or penalizes a model on every individual transaction assessment, as opposed to the long-term aggregate prediction of the chi-square score. Both scores are designed to reflect the types of prediction problems that would be relevant in practice.

### 4.4. Brand Choice Prediction Results

Figure 3 presents scatterplots of the chi-square scores of LC-MNL and RPL versus chi-square scores of PO-MNL Promotion (single class) and LC PO-MNL Promotion (multiclass), across the 27 product categories, for three subsets of customers. We conclude that the PO Promotion models outperforms both LC-MNL and RPL benchmarks to a big extent (i.e., most of the points lie above the 45-degree line). Note that both benchmark models also account for the promotion status of the products. First, consider the left two panels in Figure 3. Here, we calibrate the models on the subset of individuals who do not have cycles in their preference graph (i.e., up to Phase 2 in the DAG construction process). The chi-square score of PO-MNL Promotion model exhibits an average improvement of 10.25% over LC-MNL and 4.55% over RPL. This improvement in prediction performance can be explained by the effectiveness of the DAGs in capturing partial preferences of the customers. Table A2 in Online Appendix A11 reports the distribution of the

![Figure 3](image)

**Figure 3.** (Color online) Scatter Plot of the Average $\chi^2$ Scores of All 27 Product Categories Under the Best of Up to 10 LC-MNL (Benchmark) and RPL (Benchmark) vs. PO-MNL Promotion (Single Class) and LC PO-MNL Promotion (Multiclass)

**Notes.** Lower is better; the benchmarks are outperformed for points above the 45° line. Left panels: Estimation and prediction only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: Estimation and prediction only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: Estimation of each type of individuals separately and joint prediction over both types. Average improvements over the corresponding benchmark are reported within each of the panels.
number of unique brands purchased by customers across the training data. On average customers purchase no more than 4 unique brands in the training data, indicating that customers have relatively strong preferences. This brand loyal customer behavior also explains the significant gains in performance that our method obtained over the benchmark models given its ability to capture the prevalence of some products over the other ones, at the customer level, via the DAGs. These improvements are significant, especially considering the fact that both benchmark models have more parameters to estimate and require around 300 × more time than it takes to estimate the PO-MNL Promotion model. The key attribute of the PO-MNL Promotion model making it superior to the benchmarks is that it accounts for heterogeneous customer preferences through their partial orders, so that it makes more efficient use of the limited purchase transaction data. Using the LC PO-MNL Promotion model, we can further boost performance, resulting in average improvement of 14.72% over LC-MNL and 8.98% over RPL.

Second, consider the middle column in Figure 3, where we calibrated the models on the subset of individuals that have cycles in their preference graph. The formation of cycles in the preference graph is symptomatic of a less consistent choice behavior of the customers in the first place. In fact, by checking the y-scale we can observe that the performance of all the models deteriorate in this case as compared with their performance in the left panels. Yet, we see that the PO-MNL Promotion model exhibits an average improvement of 13.01% over LC-MNL and 3.58% over RPL, whereas LC PO-MNL Promotion model, capturing heterogeneity of customers to a greater extent, has an average improvement of 13.96% over LC-MNL and 4.52% over RPL.

Third, consider the right panels in Figure 3, where we use the previous separate calibrations but report the joint prediction over all the individuals for each category of products. The performance here is a weighted average between the two types of customers: with and without cycles in the preference graphs, achieving significant improvements overall: PO-MNL Promotion model exhibits an average improvement of 12.83% over LC-MNL and 3.89% over RPL, whereas the LC PO-MNL Promotion model shows an average improvement of 14.7% over LC-MNL and 5.75% over RPL.

Figure 4 presents scatterplots of the miss rates, using a display format similar to that of Figure 3. From it, we observe that our model combinations obtain improvements of between 0.05% and 4.01% under PO-MNL Promotion, and further improvements of between 2.36% and 6.48% under LC PO-MNL Promotion over the benchmarks in all six panels. Even though these numbers appear to be low, we emphasize here that this metric is a very stringent one and therefore it is expected that our PO MNL Promotion models obtain moderate (but still significant) improvements over state-of-the-art alternatives.

We make the following observations from the results. First, recall that the decycling in Phase 3 of the DAG construction process allows us to calibrate the PO-MNL Promotion model also for the subset of individuals that have cycles in their preference graph. As a result, it further boosts the improvement of PO-based models over the classical benchmarks by increasing the coverage of individuals to the maximum level of 100%; in other words, we can calibrate PO-MNL Promotion and make predictions for both subsets of the customers, those with and without cycles in their preference graph. Second, from all the panels it can be concluded that the RPL model outperforms the LC-MNL model on average across 27 categories of products. Third, for all the panels we have that the LC PO-MNL model boosts the performance of PO-MNL model by accounting for additional heterogeneity of customers. Fourth, we observe that PO-MNL Promotion model (or LC PO-MNL model) outperforms in most of the categories both LC-MNL and RPL benchmarks, which incorporate the same information on promotions. Therefore, this model can be used to measure customer response to product promotions even when we have very few observations for each customer by capturing partial preferences of the customers via DAGs.

In Online Appendix A6, we perform several robustness checks with respect to some of the assumptions that we made here, including (i) the way we aggregate data from customers to estimate the benchmark models, (ii) accounting explicitly for the no-purchase option, and (iii) the way we split data between training and holdout samples. The key insights remain the same. We also tested (iv) the impact of not adding the implicit candidate edges in Phase 2 of the DAG construction process and noticed a poorer performance of around 1.85% on average with respect to both \( \chi^2 \) and miss rates compared with including them. Finally, in Online Appendix A7, we report comparative statistics on the predictive performance of the behavioral models studied by Jagabathula and Vulcano (2018). We find that for categories with a high loyalty index, and within them, for customers having nonempty behavioral DAGs, practitioners may prefer to use the PO-MNL Inertial and Censored models. Other than these (category, individual) combinations, the use of the PO-MNL Promotion model proposed in this paper leads to more accurate predictions. Yet, in this paper, we apply the PO-MNL Promotion model to run customized promotions for all categories and individuals because it relies on a completely data-driven approach to build the DAGs; these DAG structures with both
promoted and nonpromoted nodes serve as the basis to design and run the personalized promotions discussed in the next section.

5. Optimization of Personalized Promotions

Having established that our model provides a more faithful representation of customer choice behavior than existing competitive benchmarks, we now turn to the problem of personalizing promotions. We take the standpoint of a retailer who wants to decide which products to put on promotion for each customer visit in order to maximize the expected revenue. In our study, the offer set is already decided and the retailer can only change the promotion activity from one customer to another. As discussed in Section 1, this setup reflects the practical situation faced by brick-and-mortar retailers, who cannot customize the shelf display to each visiting customer, but can adjust the promotion activity by launching personalized coupons to different customers.

We start by illustrating some basic facts that the retailer can infer about the preferences of each customer from the structure of the corresponding DAGs. Then, we formulate the retailer’s decision problem under the PO-MNL Promotion model as an MILP, followed by a test of our proposed methodology using the DAGs trained as described in Section 4 on the IRI Academic Data set. For each purchase instance of the customer in the holdout sample, we use the MILP to determine the optimal promotion set. We then use the PO-MNL Promotion model to predict the purchase decisions of the customer under the optimal and the existing (i.e., those that are part of the holdout data) promotion sets in order to assess potential revenue improvements.

5.1. Inferences from the DAG Structures

Our DAG-based representation of the customers’ preferences has inherent value to a retailer reasoning about his promotion strategy; namely, the retailer can come to some key conclusions about the promotion decision purely from the nonparametric structure DAG.

To illustrate this, consider a customer whose preferences are described by DAG $D$ in Figure 2 (after Phase

Figure 4. (Color online) Scatter Plot of the Average Miss Rate of All 27 Product Categories Under the Best of up to 10 LC-MNL (Benchmark) and RPL (Benchmark) vs. PO-MNL Promotion (Single Class) and LC PO-MNL Promotion (Multi-Class)

Notes. Lower is better; the benchmark is outperformed for points above the 45° line. Left panels: Estimation and prediction only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: Estimation and prediction only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: Estimation of each type of individuals separately and joint prediction over both types. Average improvements over the corresponding benchmark are reported within each of the panels.
3) facing the full offer set including products 1 through 4, each of them in either its promoted or non-promoted version. From this DAG alone, the retailer can make the following inferences about what his promotion strategy should be for this customer: (a) product 4 will be purchased only on promotion because it is dominated by both versions of product 2; (b) product 3 will not be purchased whether it is put on promotion or not because it is also dominated by both versions of product 2; (c) the promotion strategy for product 1 depends on what is done for product 2—if product 2 is on promotion, product 1 will not be purchased whether it is on promotion or not because there is a directed path from promoted 2 to promoted 1 (i.e., node 5 therein) and hence to nonpromoted 1; but if product 2 is not promoted, then product 1 could be purchased if it is put on promotion—note there is no directed path between nodes 2 and 5.

Similar reasoning can be applied in other cases. In this way, our proposed DAG structures provide a visual, intuitive, and systematic way for retailers to reason about their promotion strategy on a per-customer basis.

5.2. Promotion Optimization: MILP Formulation

We now systematize the intuitive reasoning above through an MILP to formulate the retailer’s promotion optimization problem. The retailer must solve this problem each time a customer visits the store. The formal setup is as follows. Recall that the universe $N$ consists of $2n$ products, where for all $j \in [n]$, the products $a_j$ and $a_{j+n}$ are the nonpromoted and promoted copies, respectively, of the same product. For each $j \in [n]$, we let $r_j$ denote the revenue from the nonpromoted copy $a_j$ and $d_j$ the discount offered for the promoted copy $a_{j+n}$ for a total revenue of $r_j - d_j$. Also, for each $j \in [n]$, let $q_j$ and $q_{j+n}$ denote the expected purchase quantities when the customer purchases the nonpromoted copy $a_j$ and the promoted copy $a_{j+n}$, respectively. We assume that the no purchase option $a_0$ is always available and $r_0 = d_0 = 0$. Note that throughout the paper so far, the no-purchase option was included implicitly in our analysis because as far as our methodology is concerned, there is no distinction between the no-purchase option and any other product (except that there is no promoted version of the no-purchase option). We now make it explicit because the promotion decision of the retailer not only impacts brand switching but also affects the purchase propensity of the customer.

The retailer must decide which products to offer to a customer, for each $j \in [n]$, if the retailer decides to offer product $a_j$ on promotion, then we say that the retailer has decided to offer the promoted copy $a_{j+n}$, whereas if the retailer decides not to promote product $a_j$, then we say that the retailer has decided to offer the non-promoted copy $a_j$. As a result, the promotion decision of the retailer reduces to an assortment decision. To capture this, we let $S_A \subseteq \mathcal{N} \cup \{a_0\}$, with $a_0 \in S_A$ denote the subset of available products from which the retailer must select his offer set. To be consistent with our setup, $S_A$ has the property that if product $a_j \in S_A$ for $j \in [n]$, then product $a_{j+n} \in S_A$. The goal of the retailer is to decide the subset of products in $S_A$ to offer to a customer, with the constraint that exactly one of the promoted or nonpromoted copies of each product in $S_A$ is offered, as discussed in Section 2.1.

Our MILP model includes three sets of decision variables: $x, y,$ and $z$. We start from defining binary variables $y_j$ used to determine which product version (promoted or nonpromoted) is offered within the available set $S_A$, that is, $(y_j \in \{0, 1\} : a_j \in S_A)$, where $y_j = 1$ means that product copy $a_j$ is offered. This can be captured by the constraint

$$y_j \in \{0, 1\} \text{ and } y_j + y_{j+n} = 1 \quad \forall \ a_j \in S_A, \ j \in [n].$$

Because the no-purchase alternative is always available, we set $y_0 = 1$.

The binary variables $x$ are used to indicate the product that will be purchased. Let $(x_j \in \{0, 1\} : a_j \in S_A)$, with $x_j = 1$ if and only if the customer purchases product $a_j$. Of course, only available products could be purchased; the set of binary variables $y$ enforces this connection. Let $S(y) := \{a_j \in S_A : y_j = 1\}$ denote the specific assortment offered to the customer under the offer decision $y$. Further, let $(z_j \in \{0, 1\} : a_j \in S_A)$ denote auxiliary variables with $z_j = 1$ for all products $a_j$ in the set $h_D(S(y))$ of heads (i.e., the nodes without parents) in the subgraph of the transitive closure of $D$ restricted to the set $S(y)$.

Customers only purchase the head products (see Section 3.2); therefore, we have that $x_j = 1$ only if $z_j = 1$. To determine which of the head products the customer purchases, we use the approximate posterior or probabilities $f(a_j \mid S(y), D)$ from (7) and assume that the customer purchases the product with the highest posterior probability. That is, we assume that the customer purchases the product $a_j \in h_D(S(y))$ such that

$$\frac{v_{\psi_j(a_j)}}{\sum_{a_k \in h_D(S(y))} v_{\psi_j(a_k)}} \geq \frac{v_{\psi_j(a_k)}}{\sum_{a_k \in h_D(S(y))} v_{\psi_j(a_k)}} \quad \forall \ a_k \in h_D(S(y) \setminus \{a_j\}),$$

where we define $v_j = \exp(\beta_j)$ and $v_{j+n} = \exp(\beta_j + \beta_n)$ for all $j \in [n]$. Because the denominators on both sides of the inequality are equal, the customer purchases product $a_j$ only if $v_{\psi_j(a_j)} \geq v_{\psi_j(a_k)}$ for all $a_k \in h_D(S(y) \setminus \{a_j\})$. These constraints can together be expressed as

$$a_j \notin \text{arg max } v_{\psi_j(a_k)} \Rightarrow x_j = 0, \quad (8)$$

$$x_j \leq z_j \quad \forall \ a_j \in S_A, \quad (9)$$

$$\sum_{j \in S_A} x_j = 1, \ x_j \in \{0, 1\} \quad \forall \ a_j \in S_A, \quad (10)$$
where the first constraint ensures that product \( a_j \) will not be purchased if it does not have the maximum attraction value (i.e., probability of being purchased) and the second inequality ensures that only heads are purchased. The normalization Constraint (10) ensures that exactly one product is purchased.

To relate the head variables \( z \) to the offer variables \( y \), let \( B \in \{0,1\}^{(2n+1) \times (2n+1)} \) denote the adjacency matrix of the transitive closure of \( D \), so that \( B_{kj} = 1 \) if and only if there is a path from node \( a_k \) to node \( a_j \) in \( D \), for any \( k,j \in \{0,1,2,\ldots,2n\} \). Now, product \( a_j \) becomes a head if and only if it is offered and there is no other product preferred over \( a_j \) that is also offered. We can express this condition as the following set of linear constraints:

\[
\begin{align*}
z_j &\leq y_j, \quad \forall \ a_j \in S_A, \quad (11) \\
z_j &\leq 1 - B_{kj}y_k, \quad \forall \ a_k,a_j \in S_A, k \neq j, \quad (12) \\
z_j &\geq y_j - \sum_{a_k \in S_A} B_{kj}y_k, \quad \forall \ a_j \in S_A, \quad (13) \\
z_j, y_j &\in \{0,1\}, \quad \forall \ a_j \in S_A, \quad (14)
\end{align*}
\]

where the first constraint ensures that only offered products can become heads, the second constraint ensures that \( a_j \) is not a head if an offered product \( a_k \in S_A \) is preferred over \( a_j \) in DAG \( D \), and the third constraint ensures that \( a_j \) becomes a head if it is offered and there is no other offered product \( a_k \in S_A \) that is preferred over \( a_j \) in \( D \).

Finally, it remains to express the objective function of the retailer in terms of the decision variables. The objective of the retailer is to maximize the expected revenue \( \sum_{j \in S_A} R_j q_j x_j \) from the customer, where \( R_j = r_j \) and \( R_{j+n} = r_j - d_j \) for any \( j \in [n] \), with \( R_0 = 0 \), and where \( q_j \) represents the expected purchase quantity of product \( a_j \).

Combining the above, we can express the retailer’s optimization problem as follows:

\[
\begin{align*}
&\max_{x,y,z} \sum_{j \in S_A} R_j q_j x_j \\
&\text{subject to} \quad x, z \text{ satisfy (8)--(10)}, \\
&\quad z, y \text{ satisfy (11)--(14)}, \\
&\quad y_j + y_{j+n} = 1 \quad \forall \ a_j \in S_A, j \in [n], \\
&\quad y_0 = 1.
\end{align*}
\]

To convert the above optimization problem into an MILP, we need to formulate Constraint (8) as a linear constraint. For that, we introduce continuous variables \( \{0 \leq p_j \leq 1 : a_j \in S_A\} \) such that \( p_j \)’s are the attraction values of the head products, but normalized to sum to less than 1. Also, \( p_j = 0 \) for a nonhead product. Given such \( p \)’s, Constraint (8) can be expressed as

\[
x_j \leq 1 + p_j - p_k, \quad \forall \ a_k,a_j \in S_A, k \neq j, \quad (15)
\]

where it is clear that \( x_j = 0 \) whenever \( p_j < p_k \) for some \( a_k \in S_A \). We show in Lemma A3 in Online Appendix A8 that the following set of constraints ensure that \( p \)’s are the normalized attraction values:

\[
\begin{align*}
p_j &\leq z_j \quad \forall \ a_j \in S_A, \quad (16) \\
p_0 + \sum_{j \in S_A} p_j &= 1, \quad (17) \\
0 &\leq p_j - \frac{v_{D(A_j)}p_0}{\sum_{k \in S_A} v_{D(A_k)}}, \quad \forall \ a_j \in S_A, \quad (18) \\
p_0 + z_j - 1 &\leq p_j - \frac{v_{D(A_j)}p_0}{\sum_{k \in S_A} v_{D(A_k)}}, \quad \forall \ a_j \in S_A, \quad (19)
\end{align*}
\]

Putting everything together, we obtain the following MILP:

\[
\begin{align*}
&\max_{x,y,z,p} \sum_{j \in S_A} R_j q_j x_j \\
&\text{subject to} \quad x, z \text{ satisfy (9)--(10)}, \\
&\quad z, y \text{ satisfy (11)--(14)}, \\
&\quad x, p, z \text{ satisfy (15)--(19)}, \\
&\quad y_j + y_{j+n} = 1 \quad \forall \ a_j \in S_A, j \in [n], \\
&\quad y_0 = 1.
\end{align*}
\]

Conceptually, the formulation is determining, through its variables \( y \), which ones of the \( O(2^n) \) subsets of products should be put on promotion. Variables \( z \) determine the set of heads of the intersection between the customer DAG and the offer set that are candidates to be purchased, variables \( p \) are normalized attraction values for those heads, and variables \( x \) indicate the product to be purchased (by identifying the one with highest purchasing likelihood). The size of MILP (20) scales linearly in the number of variables and quadratically in the number of constraints with respect to the number of products \( n \).

In our experience, the implementation of the promotion optimization ran very fast, taking just 0.17 seconds on average and always solving to optimality.

### 5.3. Customized Promotions: Performance Evaluation

We now evaluate the performance of the MILP proposed above to personalize promotions. We carry out our analysis using the DAGs that were trained as described in Section 4, but with the no-purchase option added. As noted above, the data do not consist of no-purchase observations. Therefore, as described in Online Appendix A6, we combine the purchases of a panelist across all categories to approximately infer customer visits to the store and then use a simple heuristic to infer which of these visits ended with no-purchase within a particular category. Our robustness checks (also reported in Online Appendix A6) show that our results are persistent to this specific heuristic.

Because of our aggregation, each product was purchased at different prices in the training data. Therefore, in order to arrive at the price \( r_j \) for each product \( a_j \), and to the discounted price \( r_j - d_j \) for its promotion counterpart \( a_{j+n} \), we averaged the full price and the discounted...
price, respectively, across different customers and stores in the training data. Similarly, to find the expected full price purchase quantity $q_j$ and the discounted price purchase quantity $q_{j;n}$ for each product, we averaged the full price purchase quantity and discount price purchase quantity, respectively, in the training data.

In order to assess the potential gains from our promotion strategy, we need a way to determine the purchases of the customers under different promotion strategies. We showed in Section 4 that our DAG model provides the best accuracy for predicting individual customer purchases, when compared with existing benchmarks. Therefore, the DAG-based model is a promising candidate to anticipate purchases. As an extra check, we also first verified its accuracy in predicting revenues from each customer. To this end, for each customer and each hold-out time period, we compared the revenue from the purchase predicted by the PO-MNL Promotion model to the revenue from the actual purchase. The left panel in Figure 5 illustrates a scatterplot of the predicted revenue versus the realized revenue from actual purchases, where by ‘predicted revenue’ we mean the predicted revenue when customers choose according to the PO-MNL Promotion model for a given set of promoted and nonpromoted items. Each point on the plot represents the revenues from one of the 27 product categories, averaged over all the customers and all the hold-out time periods. We found that the absolute revenue prediction errors are relatively small across all 27 product categories with a mean absolute error (MAE) of only 6.34%. This observation builds confidence on the predictive power of our model in terms of revenue assessment on top of the already verified purchase instance predictive power.

The revenue gains from customizing promotions are depicted in the middle panel in Figure 5. First, we consider the impact of the personalized promotions while ignoring the existing mass in-store promotions already offered in the store. Therefore, the retailer can set any subset of available products on promotion for each customer. We find that the retailer can increase the overall revenue by an estimated 23.93% on average across the 27 categories, when compared with the existing promotion strategy.

We notice from the crosses depicted in the middle panel that the revenue gains from personalizing promotions vary significantly from category to category. To better explain this variation, we regressed the percentage improvement in revenue from personalization for each category against the average purchase frequency for items in the category. We measure the purchase frequency as the average number of times a customer makes a category purchase. The right panel of Figure 5 illustrates the regression. We see a clear negative correlation between the percentage revenue improvement and purchase frequency, suggesting the personalization could have the biggest impact for less frequently bought categories of products. Online Appendix A10 provides further analysis on the factors that explain the variation in the gains from personalization at the individual customer level. The main takeaway is that personalization is more beneficial for customers who are sensitive to promotions and who purchase frequently and is less beneficial to customers who are brand loyal; see the online appendix for precise definitions of these terms.

In reality, personalized promotions need to coexist with the mass promotions already in place in the store (as reported in the data set). To capture this, we impose the additional constraint that a retailer can
personalize the promotions of only those products that are not already on mass promotions. The middle panel of Figure 5 illustrates with small circles that if the personalized promotions are mounted on top of existing in-store mass promotions the retailer can still increase the overall revenues by an estimated 16.61%, on average, across the 27 categories, when compared with the existing promotion strategy. Thus, under PO-MNL Promotion model, personalization boosts the flexibility of the promotion implementation, providing extra flexibility and enhancing the strategic promotion space.

Sometimes, a particular brand will impose a constraint to the retailer about not being promoted jointly with a competitive brand. In what follows, we empirically study the case where at most one item could be put on promotion at the personalized level. This could be implemented by taking the MILP (20) and adding the constraint \( \sum_{j=1}^{n} y_j \leq 1 \). However, because this constraint reduces the search space from \( O(2^n) \) to \( O(n) \), it could be executed via a simple search algorithm that effectively sets \( y_j = 1 \) and \( y_k = 0 \) for all \( k \in [n] \setminus \{j\} \), for each \( j \in [n] \), and finally retains the value assignment that leads to the highest objective function. Analogous to our previous analysis, Figure 6 illustrates this limited promotion situation under two cases: no mass promotions simultaneously present (left panel), and the case where personalized promotions are run on top of the mass ones (right panel).

If we promote at most one item for every customer arriving to the store, in the absence of mass promotions (left panel), the retailer can increase the overall revenues an estimated 23.88% on average across the 27 categories, when compared with the existing promotion strategy. The right panel illustrates that if the personalized single-item promotions are mounted on top of the existing in-store mass promotions, then the retailer can increase the overall revenues by an estimated 16.42% on average across the 27 categories, when compared with the existing promotion strategy. These results indicate that by promoting just one item for every customer arriving to the store the retailer can get close to all the additional revenue extractable through personalization; see Online Appendix A11 for a partial explanation of why a small number of items on promotion is sufficient to extract most of the benefit from personalization. Consequently, the strategy of customized promotions where we promote at most one item for every customer visit, might help the retailer to mitigate the negative effects of running mass promotions and still lead to near optimal revenues.

6. Conclusions and Future Work

Sales promotions planning is an important part of day-to-day operations in the retail industry, where a big proportion of products is sold under discounted prices. For many years grocery retailers have been running mostly mass promotions, offering the same deal to all the customers because of its simple practical implementation, even at the expense of a neutral or a negative impact in the long-run revenues. Because different customers are affected by promotions differently, it is worth it for the retailer to offer personalized deals, which become feasible nowadays given the unprecedented volume of panel data on sales transactions that businesses are able to collect, and provided the availability of new technology to personalize the customer experience. As a result, customization can mitigate the negative effects of promotions and be used as an appealing means for price discrimination.

Figure 6. (Color online) Revenue Performance of Promoting a Single Item

Notes. Scatter plot of the expected revenue per customer transaction after single-item promotion optimization (single-item promotion optimization revenue) versus the realized revenue per customer transaction obtained from the data (realized revenue) across 27 product categories, for all individuals based on PO-MNL Promotion model. Left panel: items available in the offer are set at full price (i.e., there are no mass promotions). Right panel: items follow the promotion status as reported in the data.
6.1. Key Findings
In this paper, we considered a back-to-back methodology to run personalized promotions with the objective of increasing retailers’ revenues by inducing the brand switching effect. Naturally, an important step in personalized promotion planning is to understand the individual preferences for different products within a category. The building block of our proposal identifies each customer with a nonparametric DAG that explicitly accounts for promotions by creating two copies for every item in the product category: promoted and nonpromoted versions. Edges in the DAG of an individual reflect the relative preference between two products (or, more precisely, between the two versions of each product). We described how to build each customer DAG for a given category in a purely data-driven way, and explained how to calibrate a parametric (multiclass) MNL model over the collection of customer DAGs. We demonstrated its ability to make more precise and fine-grained predictions of customers’ responses to price promotions on real retail data compared with state-of-the-art benchmarks. Theoretically, we derived tractable lower and upper bounds relative to the exact likelihood of partial or- 
derations need. Such stockpiling behavior impacts future purchase incidences from the customer and the long-term revenue for the retailer. We argue that when compared with mass promotions, personalized promotions mitigate the negative effect of stockpiling. The reason is that stockpiling typically occurs when a brand is promoted to a customer who would purchase the product at full price anyway; the price discounts only end up enticing the customer to stockpile, shifting her future purchases to the current period. Such erroneous price discounts are more likely to occur as part of mass promotions when compared with personalized promotions. Nevertheless, personalization may not completely eliminate stockpiling, in which case our formulation can be embedded within a dynamic programming (DP) framework to incorporate the effects of stockpiling and any other long-term effects of current period promotion decision (see Zhang and Krishnamurthi 2004, Khan et al. 2009). A state variable that keeps track of the last purchase incidence and its corresponding quantity could help toward the optimal timing of future promotions.

6.2. Limitations and Future Research
Our work opens up several ripe avenues for future work, each of which aimed at addressing certain limitations of our proposed method. Some of these directions include the following:

- **Stockpiling effects.** Our method focuses on the short-term objective of immediate revenue maximization in deciding the set of promoted products. As such, it ignores the stockpiling behavior of customers, whereby customers take advantage of the discounted prices to “stockpile” or purchase more than their immediate consumption need. Such stockpiling behavior impacts future purchase incidences from the customer and the long-term revenue for the retailer. We argue that when compared with mass promotions, personalized promotions mitigate the negative effect of stockpiling. The reason is that stockpiling typically occurs when a brand is promoted to a customer who would purchase the product at full price anyway; the price discounts only end up enticing the customer to stockpile, shifting her future purchases to the current period. Such erroneous price discounts are more likely to occur as part of mass promotions when compared with personalized promotions. Nevertheless, personalization may not completely eliminate stockpiling, in which case our formulation can be embedded within a dynamic programming (DP) framework to incorporate the effects of stockpiling and any other long-term effects of current period promotion decision (see Zhang and Krishnamurthi 2004, Khan et al. 2009). A state variable that keeps track of the last purchase incidence and its corresponding quantity could help toward the optimal timing of future promotions.

- **Reference price effect.** Another long-term effect of price promotions is the reference price effect. Repeated price reductions have the potential for lowering the reference price of the brands for the customers (Kalyanaram and Winer 1995). Such a recalibration of the reference price reduces the future impact of price discounts because customers now evaluate the discounts with respect to the lower reference price as opposed to the full price of the product. Our promotion optimization framework allows to control for the negative effect of repeated promotions through the use of business constraints, such as capping the number of times a product is promoted for a customer over subsequent store visits. Such constraints can be readily added to our promotion optimization formulation. This is routinely done in practice and is commonly referred to as frequency capping. In Online Appendix A12, we conduct an experiment over the IRI data set along these lines.
Alternatively, the reference price effect can be incorporated by embedding our promotion optimization framework within a DP. The state variable of the DP keeps track of the reference price, and the current promotion decision affects the future revenue (value-to-go) through its effect on the reference price.

• **Complementary products.** In our model, we treat each category independently. However, we can take advantage of the information about the customer being interested in a particular category during a store visit (for instance, by identifying the location of the customer in the store) in order to launch personalized promotions on a category of complementary products.

• **Purchase quantity estimation.** In our implementation, we estimated the purchase quantity of a brand conditional on its purchase (for both promoted and nonpromoted copies) by taking a simple average of the observed purchase quantities in the training data. This estimation technique might suffer from endogeneity bias. For instance, retailers may strategically promote products expecting higher quantity purchases, say, during holidays. In practice, this endogeneity bias may be corrected through detailed structural modeling to obtain a more precise estimate of the purchase quantity of a brand as a function of its promotion status. Such a correction will most certainly improve the predictive performance of our model.

**Endnotes**

1 By consistent we mean that all the pairwise relationships between products represented in the DAG are also satisfied by the total order sampled by the customer in a store visit.

2 A directed tree is a connected and directed graph that would still remain acyclic if the directions of the arcs are ignored.

3 Note that in the absence of v-nodes, then any connected component must have a single root.

4 The number of unique customers/panelists across the 27 product categories is far less than 84 thousand. But we analyze categories independently. However, we can take advantage of the information about the customer being interested in a particular category during a store visit (for instance, by identifying the location of the customer in the store) in order to launch personalized promotions on a category of complementary products.

5 See the details in appendix A2.1.2 in Jagabathula and Vulcano (2018).

6 This quantity $q_i$ could be adjusted to become a customer-dependent $q_i$ to capture an additional heterogeneity in the purchase quantity across customers.

**References**


Jeuland A (1979) Brand choice inertia as one aspect of the notion of brand loyalty. Management Sci. 25(7):671–682.


**Srikanth Jagabathula** is associate professor of operations management at the NYU Stern School of Business. This paper is part of his stream of research on the modeling and estimation of customer preferences and making assortment and price decisions. He has received a number of awards, including the National Science Foundation CAREER Award, the Wickham Skinner Early-Career Research Accomplishments Award, best paper awards, and the President of India Gold Medal.

**Dmitry Mitrofanov** is assistant professor of business analytics at the Carroll School of Management, Boston College. His research interests fall into the intersection of data-driven operations, choice-based demand modeling, and decision analytics.

**Gustavo Vulcano** is professor of operations management at the School of Business, Universidad Torcuato Di Tella, Argentina. The work described in this paper is part of his line of research related to choice-based revenue management, with focus on modeling, and the development of estimation and assortment optimization algorithms.