Personalized Retail Promotions through a DAG-based Representation of Customer Preferences

Srikanth Jagabathula
Leonard N. Stern School of Business, New York University, New York, NY 10012, sjagabat@stern.nyu.edu

Dmitry Mitrofanov
Leonard N. Stern School of Business, New York University, New York, NY 10012, dm3537@stern.nyu.edu

Gustavo Vulcano
School of Business, Universidad Torcuato di Tella, and Consejo Nacional de Investigaciones Científicas y Técnicas, Buenos Aires, Argentina, gvulcano@utdt.edu

We propose a back-to-back procedure for running personalized promotions in retail operations contexts, from the construction of a nonparametric choice model where customer preferences are represented by directed acyclic graphs (DAGs) to the design of such promotions. The source data includes a history of purchases tagged by customer id, and product availability and promotion data for a category of products. In each customer DAG, nodes represent products and directed edges represent the relative preference order between two products. Upon arrival to the store, a customer samples a full ranking of products within the category consistent with her DAG, and purchases the most preferred option among the available ones.

We describe the construction process to obtain the DAGs and explain how to mount a parametric, multinomial logit model (MNL) over them. We provide new bounds for the likelihood of a DAG and show how to conduct the MNL estimation. We test our model to predict purchases at the individual level on real retail data and characterize conditions under which it outperforms state-of-the-art benchmarks. Finally, we illustrate how to use the model to run personalized promotions. Our framework leads to significant revenue gains that make it an attractive candidate to be pursued in practice.

Key words: retailing, choice models, multinomial logit, promotion optimization, rank-based choice model


1. Introduction

The availability of individual-level transaction data allows retailers to implement personalized operational decisions. While such decisions have been around for several years now in online platforms, where e-tailers can profile their shoppers towards personalized pricing according to geographical location, past purchases or device used for access (desktop vs. mobile), recent technological developments open new opportunities have extended 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 have 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 (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 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. They offer 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 personalized promotions in both online and offline retail settings. The required input data consists 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 supplies 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 pre-specified 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 output of this first phase is a DAG for each customer.
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 which are easy to compute and that 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 dataset 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 multi-class). 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 dataset), 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 to the existing promotion strategy in place.

The empirical validation of our model supports its use towards 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. While 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, and 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. While only a few papers report a positive effect of sales promotions on long-run brand preferences (Dekimpe et al. (1998), Fockens 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 on 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 which 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 that 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. (2018) 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 OM-related 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 towards the specialization of the rank-based choice model to capture and estimate individual preferences. In that paper, the authors propose to model 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 non-empty 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 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 underlying assumptions for our model, 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 non-promoted 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 $\{a_1, a_2, \ldots, a_n\}$ of non-promoted 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 (DAG). A DAG $D$ consists of $2n$ nodes, with two copies for each product (one for the non-promoted version, and one for its promoted counterpart), and a collection of directed edges (or pairwise preference relations) denoted by $E_D \subset \{(a_k, a_j) : 1 \leq k, j \leq 2n, k \neq j\}$, so that for any $(a_k, a_j) \in E_D$ 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 non-promoted counterpart, the DAG includes $n$ arcs of the form $(a_{j+n}, a_n)$.

The DAG captures the strong preferences the customer has over the products. These preferences remain constant from one purchase instance to the next one. 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). She then chooses the most preferred product according to the sampled ranking from a subset of products she would purchase if she were not constrained by the DAG.
considers from among the offered products. More formally, if \( \sigma \) denotes the sampled preference list, \( \sigma(a_j) \) indicates the preference rank of product \( a_j \). 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 \( a_k \succ \sigma a_j \), 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 \( D \), i.e., \( S_D = \{ \sigma : \sigma(a_k) < \sigma(a_\ell) \text{ whenever } (a_k,a_\ell) \in D \} \), so that the sampled ranking \( \sigma \) belongs to the set \( S_D \).

In the store, the customer is offered a subset of products \( S \subset \mathcal{N} \). Naturally, at most one element between \( a_j \) and \( a_j+n \) 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, i.e., \( a_k = \arg\min_{a \in C} \sigma(a) \). 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 \( C = S \), i.e., 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 \( a_j \) from offer set \( S \) if and only if the sampled ranking \( \sigma \) belongs to the set \( A_j(S) = \{ \sigma : \sigma(a_j) < \sigma(a_k) \forall a_k \in S \setminus \{a_j\} \} \). Because a customer with DAG \( D \) will only sample rankings \( \sigma \in S_D \), we have that the probability \( f(a_j,S,D) \) that the customer will purchase product \( a_j \) from offer set \( S \) is given by

\[
f(a_j,S,D) = \Pr_{\lambda}(\sigma \in A_j(S)|\sigma \in S_D) = \frac{\Pr_{\lambda}(\sigma \in A_j(S) \cap S_D)}{\Pr_{\lambda}(\sigma \in S_D)},
\]

where \( \Pr_{\lambda}(\cdot) \) 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: \( a_\ell \) and \( a_{n+\ell} \)).
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, i.e., 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 dataset 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 \((a_{j\!it}, S_{it})\), 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 \( a_{j\!it} \in S_{it} \) 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 dataset 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 discounts 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, and in our case it is done to compensate for the data sparsity (with the goal of having several observations involving each node in the dataset) 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 non-promoted 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 \(a \succ b\) is strong and so the edge \((a, b)\) must be part of the customer’s DAG; (ii) the preference \(a \succ b\) is not a strong preference and the customer simply sampled a ranking \(\sigma\) with \(\sigma(a) < \sigma(b)\) 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 \(\sigma\) with \(\sigma(a) < \sigma(b)\) would never be sampled because it would be inconsistent with the DAG); and (iii) product \(b\) was not considered by the customer and therefore, we cannot make any inferences about the preference relation between \(a\) and \(b\) in the DAG (in fact, it is perfectly possible for \(b \succ a\) 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, and 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.

**Phase 0: Initializing the preference graph with edges from promoted versions to corresponding non-promoted versions of products.** We start from an empty graph \( G \), and add \( 2n \) isolated nodes from the product universe \( N' \), where each node represents either a non-promoted or a promoted version of a product. Let \( E_G \) denote the set of edges in the graph \( G \). Starting from the empty set \( E_G \), we add \( n \) edges \((a_{j+n}, a_j)\) for each non-promoted item \( a_j \), \( j \in [n] \). These edges capture the fact that a promoted copy \( a_{j+n} \) of every product \( a_j \in \{a_1, \ldots, a_n\} \) is preferred to its non-promoted copy \( a_j \) since 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.

**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 \((a_{jt}, S_a)\) of individual \( i \), we draw edges from \( a_{jt} \) to the other items in the offer set, i.e., \( E_G \leftarrow E_G \cup \{(a_{jt}, a_\ell) : \forall a_\ell \in S_a \setminus \{a_{jt}\}\} \). 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 \((a_j, a_\ell)\) is added through the weight \( w_{j,\ell} \), defined as the number of times the customer chose product \( a_j \) when \( a_\ell \) was also offered.

**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 non-promoted (promoted) copies of two products, then the preference extends to also the corresponding promoted (non-promoted) copies. More precisely, for any \( 1 \leq j, \ell \leq n \), if \((a_j, a_\ell) \in E_G\), then we add the edge \((a_{j+n}, a_{\ell+n})\) to \( G \), i.e., \( E_G \leftarrow E_G \cup \{(a_{j+n}, a_{\ell+n})\}\). Similarly, if \((a_{j+n}, a_{\ell+n}) \in E_G\), then we add the edge \((a_j, a_\ell)\) to \( G \), resulting in \( E_G \leftarrow E_G \cup \{(a_j, a_\ell)\}\). To de-emphasize the implicit counterparts of these edges, we assign the weight \( w_{j+n,\ell+n} \leftarrow w_{j,\ell}/(T_in^2) \) when \((a_{j+n}, a_{\ell+n})\) is the implicit edge, and \( w_{j,\ell} \leftarrow w_{j+n,\ell+n}/(T_in^2) \) when \((a_j, a_\ell)\) is the implicit edge. In other words, the weights of the implicit counterparts are scaled down by a factor of \( T_in^2 \).

Intuitively, by scaling down the weights of the implicit edges, we prioritize candidate edges over implicit candidate edges, since 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.

**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 does not identify the DAG and 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 1 (Karp (1972) provides a reduction from the minimum vertex cover problem).

We formulate the above decycling procedure as a mixed integer linear program (MILP). For a given graph $G$, and for every edge $(a_k, a_\ell) \in E_G$, define the binary variable $x_{k\ell}$ that takes the value 1 if edge $(a_k, a_\ell)$ 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 $\{x_{k\ell} : (a_k, a_\ell) \in E_G\}$ does not contain cycles, we introduce auxiliary binary variables $y_{k\ell}$, 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 $y_{k\ell} = 1$ if $a_k$ is preferred over $a_\ell$, and $y_{k\ell} = 0$ otherwise. The following MILP enforces that the final DAG is a subset of some total order defined by the $y$ variables:

$$\max_{x, y} \sum_{(a_k, a_\ell) \in E_G \setminus \{(a_{j+n}, a_j) : 1 \leq j \leq n\}} w_{k\ell} x_{k\ell}$$

subject to:

$$(C1) \quad x_{k+n, k} = 1, \quad \forall 1 \leq k \leq n$$

$$(C2) \quad x_{k, \ell} = y_{k, \ell}, \quad \forall (a_k, a_\ell) \in E_G,$$

$$(C3) \quad y_{k\ell} + y_{\ell k} = 1, \quad \forall a_k, a_\ell \in N', \quad k \leq \ell,$$

$$(C4) \quad y_{k\ell} + y_{\ell p} + y_{pk} \leq 2, \quad \forall a_k, a_\ell, a_p \in N', \quad k \neq \ell \neq p,$$

$$(C5) \quad y_{k, \ell} \in \{0, 1\} \quad \forall a_k, a_\ell \in N', \quad k \leq \ell.$$
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 pre-specified 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 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 $\sigma(a) < \sigma(b)$ in the particular ranking $\sigma$ 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 that purchase instance, 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 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 de-cycling 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 \( a_\ell \), but she has not revealed if the same preference extends to their respective promoted copies \( a_{k+n} \) and \( a_{\ell+n} \). Yet, we make the assumption that the customer is likely to prefer \( a_{k+n} \) over \( a_{\ell+n} \). 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 \( T^2 n \). Proposition A1 in the Appendix A2.1 shows that with this scaling the MILP strictly prioritizes implicit candidate edges over the candidate edges for deletion. Overall, since 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 Appendix). We notice that by adding implicit edges 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 \( (a_{jt}, S_i) \) (which can be represented as a star graph with head \( a_{jt} \) and set of leaves \( S_i \setminus \{a_{jt}\} \)), is a subgraph of the corresponding DAG \( D_i \). In case a cycle was formed, say for customer \( i \), consider a transaction \( (a_{jt}, S_i) \) such that one of the edges \( (a_{jt}, a_k) \), with \( a_k \in S_i \setminus \{a_{jt}\} \), was deleted in the decycling procedure. Since \( (a_{jt}, a_k) \) was part of a cycle, it follows that there is a directed path from \( a_k \) to \( a_{jt} \) 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 \( a_{jt} \) 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, i.e.,

$$
\log L(\text{Panel Data}) = \sum_{i=1}^{m} \log \lambda(D_i) = \sum_{i=1}^{m} \log \left( \sum_{\sigma \in S_{D_i}} \lambda(\sigma) \right),
$$

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 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 the above MLE problem 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 size of set $S_D$) is a #P-hard problem (Birghtwell 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 $a \in N$ is associated with parameter (i.e., weight) $v_a > 0$. The probability of sampling ranking $\sigma$ is given by

$$
\lambda(\sigma) = \prod_{r=1}^{n} \frac{v_{\sigma_r}}{\sum_{j=r}^{n} v_{\sigma_j}}.
$$

For brevity of notation, we also use $v_j$ to refer to $v_{a_j}$, 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 $(v_a)_{a \in N}$. In other words, we have

$$
\Pr(a_i|S) = \frac{v_i}{\sum_{a_j \in S} v_j}.
$$
under both the PL and MNL models. Under the PL model, the choice probability \( \Pr(a_i|S) \) 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 dataset 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 \( \lambda(D) = \sum_{\sigma \in S_D} \lambda(\sigma) \) 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, \( \Psi_D(a) = \{b: \text{there is a directed path from } a \text{ to } b \text{ in } D\} \).

We assume that a node is reachable from itself, so \( a \in \Psi_D(a) \) for all \( a \), and \( \Psi_D \) is always non-empty. 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\(^2\) 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

\[
\lambda(D) = \prod_{a \in N} \frac{v_a}{\sum_{a' \in \Psi_D(a)} v_{a'}},
\]

whenever \( D \) is a forest of directed trees, each with a unique root.
Jagabathula and Vulcano (2018) propose to use (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 $D$, \[
\tilde{\lambda}(D) \leq \lambda(D), \quad \text{where } \tilde{\lambda}(D) := \prod_{a \in N} \frac{v_a}{\sum_{a' \in \Psi_D(a)} v_{a'}}.
\]
The inequality above is strict if $D$ has at least one v-node.

The proof is rather involved and the details are provided in the 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 $\tilde{D}[a_y]$ to create a new copy $a'_y$ such that one of the incoming edges into $a_y$ moves to the node $a'_y$ while 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 $\tilde{D} \subset D$, we have that $S_{\tilde{D}} \supset S_D$, where recall that $S_D$ is the set of all rankings that are consistent with $D$. It thus follows that $\lambda(\tilde{D}) \geq \lambda(D)$. We state this result formally in the following proposition and prove it in the Appendix A2.1.

**Proposition 2.** For any two DAGs $D$ and $\tilde{D}$ such that $\tilde{D} \subset D$, we must have that $\lambda(D) \leq \lambda(\tilde{D})$, 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 $\bar{D}$ 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 $\bar{D}$ 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 Appendix for details of the algorithm.

Next we explore the tightness of the developed lower and upper bounds of a DAG’s likelihood. Let $R(D, \bar{D}) = \lambda(\bar{D})/\tilde{\lambda}(D)$ denote the ratio between them for any DAG $\bar{D} \subset D$. It is clear that $R(D, \bar{D}) \geq 1$ for all $\bar{D} \subset D$, so we express our tightness guarantee by deriving a parametric upper bound for $R(D, \bar{D})$. For that, let $\ell$ denote the size of the largest reachability set in DAG $D$, i.e., $\ell = \max_{a \in N} |\Psi_D(a)|$, and let $p$ denote the number of nodes with $v$-nodes in their reachability sets, i.e., $p = |\{a \in N : \exists v$-node $b \in \Psi_D(a)\}|$. Further, let $\Delta := \max_{a \in N} \max_{b \in \Psi_D(a) \setminus \{a\}} v_b/v_a$ 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 $\bar{D} \subset D$ is obtained by deleting all, except one, of the incoming edges into each of the $v$-nodes. Then, we have that

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

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

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

$$f(a_j, S, D) = \Pr(S_{C(a_j, S)} \mid S_D) = \frac{\Pr(S_{C(a_j, S)} \cap S_D)}{\Pr(S_D)},$$

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 $S$. Since 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 $f(a_j, S, D) = 0$ for all $a_j \in S \setminus h_D(S)$. For the products in $h_D(S)$, the probability of choosing $a_j$ from $S$ depends on the probability of the DAG representing the collection of permutations $S_{C(a_j, S)} \cap S_D$, which corresponds to the merged DAG $D \uplus C(a_j, S)$ obtained by taking the union of the graphs $D$ and $C(a_j, S)$. We thus obtain

$$f(a_j, S, D) = \begin{cases} \frac{\lambda(D \uplus C(a_j, S))}{\lambda(D)}, & \text{if } a_j \in h_D(S), \\ 0, & \text{otherwise}, \end{cases}$$

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) shows that

$$f(a_j, S, D) = \frac{\hat{\lambda}(D \uplus C(a_j, S))}{\lambda(D)} = \frac{\bar{v}_{D(a_j)}}{\sum_{a_j \in h_D(S)} \bar{v}_{D(a_j)}},$$

where we define $\bar{v}_{D(a)} = \sum_{b \in \Psi_D(a)} v_b$. More generally, they propose to use 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

$$f(a_j, S, D) := \frac{\hat{\lambda}(D \uplus C(a_j, S))}{\lambda(D)} \quad \text{and} \quad f(a_j, S, D) = \frac{\lambda(D \uplus C(a_j, S))}{\lambda(D)},$$

where for any DAG $D$, we let $\overline{D}$ 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:

$$\underline{f}(a_j, S, D) \leq f(a_j, S, D) \leq \overline{f}(a_j, S, D), \quad \text{and}$$

$$\underline{f}(a_j, S, D) \leq \hat{f}(a_j, S, D) \leq \overline{f}(a_j, S, D),$$
where
\[
\hat{f}(a_j, S, D) = \frac{\hat{\lambda}(D \uplus C(a_j, S))}{\lambda(D)} = \frac{v_{\Psi_D(a_j)}}{\sum_{a_i \in h_D(S)} v_{\Psi_D(a_i)}}.
\] (5)

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 DAG \( D \uplus C(a_j, S) \). That is, \( \ell = \max_{a \in N} |\Psi_{D \uplus C(a_j, S)}| \) and \( p = |\{a \in N: \exists v\text{-node } b \in \Psi_{D \uplus C(a_j, S)}(a)\}| \). We also define \( \Delta = \max_{a \in N} \max_{b \in \Psi_{D \uplus C(a_j, S)} \backslash \{a\}} v_b / v_a \). We can then establish the following result:

**Proposition 4.** Given DAG \( D \), offer set \( S \), and product \( a_j \in h_D(S) \), we have that
\[
0 \leq \log \frac{\hat{f}(a_j, S, D)}{f(a_j, S, D)} \leq 2p \cdot \log(1 + \ell \cdot \Delta).
\]

Further, if \( \Delta \in o(n^{-2}) \), where \( n \) is the number of nodes, then we have that \( \lim_{n \to \infty} \hat{f}(a_j, S, D) = f(a_j, S, D) \).

The tightness of the bound above again follows from the case when \( p = 0 \). For other cases, the approximation ratio can be much better than that suggested by the bound above, as demonstrated on real-world data in Appendix A5.

### 4. Empirical study

We now test our proposal on the IRI Academic dataset (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 (LC-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 (CPG) purchase transaction data for year 2007 over a chain of grocery stores in two large Behavior Scan markets in USA. 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 four because of data sparsity.

The data consist of 1.2M records of weekly purchase transactions from 84K customers over 52 weeks. The transaction data is 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 64K customers and 1.1M 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 1 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 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 8 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 these models belong to the general class of random utility models (RUMs), so that in each purchase instance, a customer samples product utilities and then chooses the product with the highest value.
Table 1  Summary of the data. The column ‘Vend’ - the number of products obtained in the training data after aggregating different UPCs by vendor; ‘AvOS’ - the average # of vendors in the offer set; ‘Total’ - the total # of individuals in the training data; ‘≥2 sales’ - # of individuals with at least two purchases in the training data; ‘AvTr’ - the average # of transactions in the training data. Analyzing customers under PO-MNL Promotion model, we report ‘|woCyc|’ - # of individuals without cycles in the preference graph; ‘% Del’ - the average percentage of edges deleted after preference graph decycling; ‘Dens1’ - the average # of edges in the DAG for individuals without cycles in the preference graph; ‘Dens2’ - the average # 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.
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 non-promoted one, of each product. Following the notation introduced in Section 2.1, products \( a_1, a_2, \ldots, a_n \) are the non-promoted copies and \( a_{n+1}, a_{n+2}, \ldots, a_{2n} \) are the promoted copies. For any \( j \in [n] \), 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 \( a_j \in N' \):

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

where \( \beta_j^0 \) is the utility derived from the non-promoted 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:

\[
\max_{\beta, \beta_0} \sum_{i=1}^{m} \sum_{j=1}^{2n} \left[ \tau_j - \log \left( \sum_{a_j \in \Psi_{D_i}(a_j)} \exp(\tau) \right) \right] - \alpha(\|\beta^0\|_1 + \|\beta\|_1), \tag{6}
\]

where \( \Psi_{D_i}(a_j) \) is the set of nodes that are reachable from \( a_j \) in DAG \( D_i \) of customer \( i \). To arrive at the above approximation, we used the lower bound \( \hat{\lambda} \) 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 5-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:

\[
\hat{f}(a_j, S, D) = \begin{cases} 
\frac{v_{\Psi_{D}(a_j)}}{\sum_{a_j \in h_D(S)} v_{\Psi_{D}(a_j)}}, & \text{if } a_j \in h_D(S), \\
0, & \text{otherwise.}
\end{cases}
\tag{7}
\]

In 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 \( h \in \{1, \ldots, K\} \) latent classes. A customer from class \( h \) samples her DAGs according to the PO-MNL Promotion model with parameters \( \tau_{jh} \), defined as

\[
\tau_{jh} = \begin{cases} 
\beta_{jh}, & \text{if } 0 \leq j \leq n \\
\beta_{j-n, h}^0 + \beta_{j-n, h}, & \text{if } n + 1 \leq j \leq 2n.
\end{cases}
\]
We let the prior probability that a customer belongs to class $h$ by $\gamma_h \geq 0$, 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:

$$\max_{\beta, \beta_0, \gamma} \sum_{i=1}^{m} \log \left[ \sum_{h=1}^{K} \gamma_h \prod_{j=1}^{2n} \frac{\exp(\tau_{jh})}{\sum_{a \in \Psi_{D_i}(a_j)} \exp(\tau_{lh})} \right] - \alpha \sum_{h=1}^{K} (\|\beta_h\|_1 + \|\beta_0\|_1),$$

The above optimization problem is non-concave for $K > 1$, even with the value of $\alpha$ 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 $D_1, D_2, \ldots, D_K$, which form a partition of the collection of all the customers. Then we set $\gamma_h^{(0)} = |D_h|/(\sum_{d=1}^{K} |D_d|)$. 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 $h \in [1, \ldots, K]$:

$$\hat{\gamma}_{ih} = \frac{\gamma_h \prod_{j=1}^{2n} \frac{v_{jh}}{\sum_{a \in \Psi_{D_i}(a_j)} v_{th}}}{\sum_{d=1}^{K} \gamma_d \prod_{j=1}^{2n} \frac{v_{jd}}{\sum_{a \in \Psi_{D_i}(a_j)} v_{td}}},$$

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

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

where $\hat{f}_h(a_j, S, D)$ are the approximated probabilities in (7). We estimated the model for $K = 1, 2, \ldots, 10$, 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 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 $h \in \{1, \ldots, K\}$ 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. Similarly to the LC PO-MNL Promotion model, we estimated the model for $K = 1, 2, \ldots, 10$, 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 $\Sigma$, i.e., $\beta \sim N(\mu, \Sigma)$. 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.

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:

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

where $\hat{n}_{ij} = \sum_{t=1}^{T_i} f_i(j_{it}, t)$,

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 $f_i(j_{it}, t)$ takes value 1 if the product indexed $j$ has the highest choice probability for individual $i$ at time $t$, and 0 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:

$$
\text{miss rate} = \frac{1}{|U|} \sum_{t \in U} \frac{1}{|T_i|} \sum_{t=1}^{T_i} [f_i(j_{it}, t) = 1],
$$
where $\mathbb{I}[A]$ is the indicator function that takes value 1 if $A$ is true and 0 otherwise, and $j_{it}$ 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 method 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 (multi-class), 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 Appendix A11 reports the distribution of the 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 methods 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 $300 \times$ 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
Figure 3 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]. 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.

this case as compared to their performance in the left panels. Yet, we see that 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, while 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
Figure 4 Scatter plot of the average miss rate of all 27 product categories under the PO-MNL Promotion [single class] and LC PO-MNL Promotion [multiclass], vs. the best of up to 10 LC-MNL [benchmark] and RPL [benchmark]. 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.

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 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 incorpo-
rate 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 by DAGs.

In 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 bench-
mark 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 to including them. Finally, in 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 high loyalty index, and within them, for customers having non-empty
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, and these DAG structures with both
promoted and non-promoted nodes serve as 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 Dataset. 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 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 since it is dominated by both versions of product 2; (b) product 3 will not be purchased whether it is put on promotion or not since 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 non-promoted 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 non-promoted and promoted copies, respectively, of the same product. For each $j \in [n]$, we let $r_j$ denote the revenue from the non-promoted 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 non-promoted 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 on promotion. For any \(j \in [n]\), 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 \(S_A \subseteq N' \cup \{a_0\}\), 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 \(a_{j+n} \in S_A\). Then, 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 non-promoted 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 \(y\), used to determine which product version (promoted or non-promoted) is offered within the available set \(S_A\), i.e., \((y_j \in \{0, 1\}: a_j \in S_A)\), where \(y_j = 1\) means that product copy \(a_j\) is offered. This can be captured by the constraint:

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

Since 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, and 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 probabilities \(\tilde{f}(a_j, S(y), D)\) 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 \( a_j \in h_D(\mathbf{y}) \) such that

\[
\frac{v_{\Psi D(a_j)}}{\sum_{a_l \in h_D(\mathbf{y})} v_{\Psi D(a_l)}} \geq \frac{v_{\Psi D(a_k)}}{\sum_{a_l \in h_D(\mathbf{y})} v_{\Psi D(a_l)}} \quad \forall a_k \in h_D(\mathbf{y}) \setminus \{a_j\},
\]

where we define \( v_j = \exp(\beta_j^0) \) and \( v_{j+n} = \exp(\beta_j^0 + \beta_j) \) for all \( j \in [n] \). Since the denominators on both sides of the inequality are equal, the customer purchases product \( a_j \) only if \( v_{\Psi D(a_j)} \geq v_{\Psi D(a_k)} \) for all \( a_k \in h_D(S(\mathbf{y})) \setminus \{a_j\} \). These constraints can together be expressed as

\[
a_j \notin \arg \max_{a_k \in S_A : z_k = 1} v_{\Psi D(a_k)} \implies x_j = 0, \quad \text{(8)}
\]

\[
x_j \leq z_j \quad \forall a_j \in S_A, \quad \text{(9)}
\]

\[
\sum_{j: a_j \in S_A} x_j = 1, \quad x_j \in \{0, 1\} \quad \forall a_j \in S_A, \quad \text{(10)}
\]

where the first inequality 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 \( \mathbf{y} \), 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 \( k, j \in \{0, 1, 2, ..., 2n\} \). 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:

\[
z_j \leq y_j, \quad \forall a_j \in S_A, \quad \text{(11)}
\]

\[
z_j \leq 1 - B_{kj}y_k, \quad \forall a_k, a_j \in S_A, k \neq j, \quad \text{(12)}
\]

\[
z_j \geq y_j - \sum_{a_k \in S_A \setminus \{a_j\}} B_{kj}y_k, \quad \forall a_j \in S_A, \quad \text{(13)}
\]

\[
z_j, y_j \in \{0, 1\}, \quad \forall a_j \in S_A, \quad \text{(14)}
\]

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 \( \sum_{j: a_j \in S_A} R_j q_j x_j \) 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 recall that \( q_j \) represents the expected purchase quantity of product \( a_j \).
Combining the above, we can express the retailer’s optimization problem as follows:

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

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 non-head product. Given such \( p \)'s, constraint (8) can be expressed as

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

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 Appendix A8 that the following set of constraints ensure that \( p \)'s are the normalized attraction values:

\[
\begin{align*}
p_j & \leq z_j \quad \forall \ a_j \in S_A, \\
p_0 + \sum_{j: a_j \in S_A} p_j & = 1, \\
0 \leq p_j & \leq v_{\Psi,D(a_j)} p_0, \quad \forall \ a_j \in S_A, \\
p_0 + z_j - 1 & \leq p_j / v_{\Psi,D(a_j)} \quad \forall \ a_j \in S_A.
\end{align*}
\]

Putting everything together, we obtain the following MILP:

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

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 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. Our robustness checks (also reported in 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 to 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’ vs. 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 non-promoted 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 to 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. 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 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 dataset). 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 to 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: \( \sum_{j=n+1}^{2n} y_j \leq 1 \). However, since this constraint reduces the search space from \( O(2^n) \) to \( O(n) \), it could be executed via a simple search algorithm that
Figure 5  Revenue performance. Left panel: Scatter plot of the expected revenue per customer transaction taking the set of promoted items from the sales data when customers choose according to the PO-MNL Promotion model [Predicted revenue] vs. realized revenue per customer transaction obtained from the data [Realized revenue], across 27 product categories. Middle panel: Scatter plot of the expected revenue per customer transaction after promotion optimization [Promotion optimization revenue] vs. 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. Right panel: Scatter plot and linear regression of the percentage improvement of the retailer’s revenue after the promotion optimization over the purchase frequency across 27 product categories.

effectively sets $y_{j+n} = 1$ and $y_{k+n} = 0$ for all $k \in [n]\{j\}$, 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 by an estimated 23.88% on average across the 27 categories, when compared to 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 to 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 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 massive promotions, offering the same deal to all the customers due to its simple practical implementation, even at the expense of a neutral or a negative impact in the long run revenues. Since different customers are affected by promotions differently, it is worth 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.

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 non-promoted 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.

Figure 6  Revenue performance of promoting a single item. Scatter plot of the expected revenue per customer transaction after single item promotion optimization [Single-item promotion optimization revenue] vs. 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.
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 to state-of-the-art benchmarks. Theoretically, we derived tractable lower and upper bounds relative to the exact likelihood of partial orders and to the likelihood of purchasing a particular product from a given offer set.

The good performance of purchasing prediction results served as basis for the next phase: the implementation of customized promotions. We formulated a compact MILP to solve the personalized promotion optimization problem. On the same dataset, we verified via simulation studies that our personalized promotions provide revenue gains across the 27 categories of the order of 16% if run on top of the current mass promotions already in place and of the order of 23% if instead any subset of available products can be promoted. Similar revenue gains were observed even if after constraining the retailer to promote at most a single item. Overall, based on the results we obtain on real retail data, we believe that our methodology constitutes an interesting framework to be further tested in the retail operations practice.

An industry implementation of our proposal will need the fine-tuning of a few details. For instance, there are two MILPs that need to be solved: the decycling and the promotion optimization. The decycling procedure is run periodically for each customer (e.g., once every six months) and could be solved as an overnight batch process. However, the promotion optimization must be solved in real time upon each store visit since it depends on both the particular DAG of the customer and the subset of products on offer. Even though in our experience the problem is solved to optimality within a fraction of a second for up to 100 products, as the size of the product category scales the computational performance could suffer. As such, developing valid inequalities and designing a branch-and-cut procedure, or testing polynomial running time heuristics, could be fundamental for real applications.

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:

- **Stockpiling effects.** Our method focuses on the short-term objective of immediate profit 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 to 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 to 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 (c.f., Zhang and Krishnamurthi (2004), Khan et al. (2009)). A state variable that keeps track of last purchase incidence and corresponding quantity could help towards 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. 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 profit (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 non-promoted 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 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 84K. But we analyze categories separately, so we treat each “customer-category” combination as a separate customer.

5. See the details in Appendix A2.1.2 in Jagabathula and Vulcano 2018.

6. This quantity $q_j$ could be adjusted to become a customer-dependent $q_{ij}$ 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 Science* 25(7).


A1. Preliminaries on DAGs

For completeness, we summarize the relevant notation from the main body and also introduce additional notation. Let $\mathcal{N} = \{a_1, \ldots, a_n\}$ denote the universe of $n$ products. For the purposes of this Appendix, we ignore the no purchase option and information about product promotions in order to simplify exposition. This assumption is without loss of generality since we can explicitly account for promotions by expanding the product universe and include the no purchase option as one more item. This is further developed in Section 5 in the main body of the paper.

A DAG $D$ is a subset of pairwise preferences, $\{(a_j, a_{j'}) : 1 \leq j, j' \leq n\}$. We visualize a DAG $D$ as a directed graph with nodes as products and a directed edge from $a$ to $b$ if the ordered pair $(a, b) \in D$. We abuse notation and let $D$ denote both the directed graph and the collection of pairwise preferences. We let $E_D$ denote the set of pairwise preferences in the transitive reduction of $D$.

Let $\mathcal{S}_n$ denote the collection of all possible $n!$ rankings or permutations of the products in $\mathcal{N}$. For any ranking $\sigma \in \mathcal{S}_n$, we let $\sigma(a)$ denote the preference ranking of product $a$ under ranking $\sigma$. We adopt the convention that lower ranked products are preferred over higher ranked ones, which means that product $a_j$ is preferred over product $a_{j'}$ under $\sigma$ if and only if $\sigma(a_j) < \sigma(a_{j'})$.

Given a DAG $D$, let $S_D$ denote the subset of rankings that are consistent with $D$; that is, $S_D := \{\sigma \in \mathcal{S}_n : \sigma(a) < \sigma(b) \text{ whenever } (a, b) \in D\}$.

For any product $a_j$ and DAG $D$, the reachability set $\Psi_D(a)$ comprises the set of all nodes that can be reached from $a$ through a directed path in $D$. Formally, $\Psi_D(a) := \{b : \text{there is a directed path from } a \text{ to } b \text{ in } D\}$. We assume that $a$ is reachable from itself, so $a \in \Psi(a)$. The set $\Theta_D(a)$ comprises the nodes from which $a$ can be reached, i.e., $\Theta_D(a) := \{b \in \Psi_D(a) : \text{there is a directed path from } b \text{ to } a \text{ in } D\}$. 


Appendix

Srikanth Jagabathula
NYU Stern School of Business, New York, NY, 10012, sjagabat@stern.nyu.edu

Dmitry Mitrofanov
NYU Stern School of Business, New York, NY, 10012, dm3537@stern.nyu.edu

Gustavo Vulcano
School of Business, Universidad Torcuato di Tella, and CONICET,
Buenos Aires, C1428BCW, Argentina, gvulcano@utdt.edu
\{b: there is a directed path from \(b\) to \(a\) in \(D\)\}. To be consistent, we also include \(a\) in \(\Theta_D(a)\). When the DAG \(D\) is clear from the context, we drop \(D\) from the notation and simply write \(\Psi(a)\) and \(\Theta(a)\).

For any subset \(S \subseteq \mathcal{N}\), suppose \(\pi\) is a ranking of the products in \(S\) possibly including less than \(n\) products. Then, \(\sigma(\pi)\) denotes the set of all complete rankings of the products in \(\mathcal{N}\) that are consistent with \(\pi\), i.e., \(\sigma(\pi) = \{\sigma \in \mathcal{S}_n : \sigma(a) < \sigma(b)\ \text{whenever} \ \pi(a) < \pi(b)\}\).

### A2. Technical results

#### A2.1. Propositions and proofs in Section 2

##### A2.1.1. Preference graph decycling

Here we argue that the decycling procedure prioritizes retaining as many candidate edges as possible (measured by the sum of the weights involved). To that end, we show that the weight of the candidate edges in DAG \(D\) after preference graph \(G\) decycling is equal to the weight of the candidate edges in DAG \(D^*\) such that \(D^* \subseteq G\) has maximum weight of the candidate edges.

Let \(cw(D)\) denote the weight of the candidate edges in DAG \(D\) and let \(iw(D)\) denote the weight of the implicit candidate edges in DAG \(D\). Let \(tw(D)\) denote the total weight of DAG \(D\), i.e., \(tw(D) = cw(D) + iw(D)\). Let us define \(D^*\) as a DAG in \(G\) with the maximum weight of the candidate edges, i.e., \(D^* = \arg\max\{cw(D) : D \subseteq G\}\). Let \(D\) denote the DAG obtained from \(G\) after solving MIP (1). The next result follows.

**Proposition A1.** Candidate weight of DAG \(D\), obtained after MIP (1) decycling applied over \(G\) is equal to the candidate weight of DAG \(D^*\) such that \(D^* = \max\{cw(D) : D \subseteq G\}\), i.e.,

\[cw(D) = cw(D^*).\]

**Proof of Proposition A1:** Assume by contradiction that \(cw(D) < cw(D^*)\). Because \(cw(\cdot)\) is always an integer, it follows that \(cw(D^*) - cw(D) \geq 1\). Further, note that \(D^*\) is a feasible solution to the optimization problem MIP (1), which implies that \(tw(D) \geq tw(D^*)\). It now follows from the definition of \(tw(\cdot)\) that

\[cw(D) + iw(D) \geq cw(D^*) + iw(D^*) \implies iw(D) \geq cw(D^*) - cw(D) + iw(D^*).\]

Because \(iw(D^*) \geq 0\) and \(cw(D^*) - cw(D) \geq 1\), we obtain that \(iw(D) \geq 1\). This, however, is not possible because of the scaling factor \(1/(n^2T)\). Specifically, note that the maximum number of implicit candidate edges is \(n(n - 1)\) and the maximum possible weight of each implicit edge is \(T/(n^2T) = 1/n^2\). Therefore, the aggregate implicit weight is always bounded above by \(n(n - 1)/n^2\), which is strictly less than 1. We have thus arrived at a contradiction. \(\square\)
A2.1.2. Likelihood of the DAG-based choice model

**Proposition A2.** For a given set of parameters \( \beta \) that characterize the distribution \( \lambda \), the likelihood function of the DAG-based choice model is given by

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

**Proof of Proposition A2:** For each individual \( i \) and transaction \( t \), let \( C_{it} \subseteq S_{it} \) denote the consideration set of the individual. Because products outside the consideration set do not affect the individual’s choices, the data log-likelihood only depends on the choices and the consideration sets, rather than choices and offer sets. Letting \( f(a_{j_it}, C_{it}, D_i) \) denote the probability of purchasing product \( a_{j_it} \) from consideration set \( C_{it} \) for individual \( i \) with DAG \( D_i \), we can write

\[
\log \mathcal{L}(\text{Panel Data}|\beta) \triangleq \sum_{i=1}^{m} \log \Pr \left( (a_{j_it}, C_{it})_{t=1}^{T_i} | D_i, \beta \right)
\]

\[
= \sum_{i=1}^{m} \log \left( \Pr(D_i|\beta) \cdot \Pr \left( (a_{j_it}, C_{it})_{t=1}^{T_i} | \beta, D_i \right) \right)
\]

\[
= \sum_{i=1}^{m} \log \left( \Pr(D_i|\beta) \cdot \prod_{t=1}^{T_i} \Pr \left( (a_{j_it}, C_{it}) | \beta, D_i \right) \right)
\]

\[
= \sum_{i=1}^{m} \log \Pr(D_i|\beta) + \sum_{i=1}^{m} \sum_{t=1}^{T_i} \log f(a_{j_it}, C_{it}, D_i) \quad (A1)
\]

The second equality follows from a straightforward application of the conditional probability formula. The third equality follows because conditioning on the DAG \( D_i \), individual \( i \)’s purchase probabilities can be computed independently.

We now focus on the term \( f(a_{j_it}, C_{it}, D_i) \). Note that we only observe the offer sets \( S_{it} \). The consideration sets \( C_{it} \) are latent. Nevertheless, given \( D_i \), we can constrain \( C_{it} \) sufficiently to allow for the computation of \( f(a_{j_it}, C_{it}, D_i) \). There are two cases. First, we consider the case when none of the edges in the set \( \{(a_{j_it}, a_k): a_k \in S_{it} \setminus \{a_{j_it}\}\} \) was deleted in the de-cycling step. In this case, the customer always prefers product \( a_{j_it} \) over all the other products in the offer set \( S_{it} \), and therefore, chooses product \( a_{j_it} \) irrespective of the consideration set. This implies that \( f(a_{j_it}, C_{it}, D_i) = 1 \) for all \( C_{it} \subseteq S_{it} \) such that \( a_{j_it} \in C_{it} \).

Next, we consider the case when some of the edges in the set \( \{(a_{j_it}, a_k): a_k \in S_{it} \setminus \{a_{j_it}\}\} \) are deleted in the de-cycling step. Because the de-cycling procedure deletes the smallest possible weight of edges, the edge \((a_{j_it}, a_k)\) for some \( a_k \in S_{it} \setminus \{a_{j_it}\} \) is deleted only if there is a directed path from \( a_k \) to \( a_{j_it} \) in the final DAG \( D_i \). Now, because \( a_k \) is preferred over \( a_{j_it} \), the only way the customer would choose \( a_{j_it} \) when \( a_k \) was also on offer is if \( a_k \) was not considered. We can thus conclude that
with where \( \lambda \) nodes a directed tree; see Figure A1. Further, let The result of the proposition now follows. □

A2.2. Propositions and proofs in Section 3

The following auxiliary results will be used in the proofs of the results in the main body of the paper. We start quoting Lemma A1 from Jagabathula and Vulcano (2018):

**Lemma A1 in Jagabathula and Vulcano (2018)** Consider two subsets \( S_1, S_2 \subset N \) with \( S_1 \cap S_2 = \emptyset \). Let \( \pi_1 \) be a ranking over \( S_1 \), and \( \pi_2 \) be a ranking over \( S_2 \). Assume w.l.o.g. that \( \pi_1 = (a_{1,1}, a_{1,2}, \ldots, a_{1,k_1}) \) and \( \pi_2 = (a_{2,1}, a_{2,2}, \ldots, a_{2,k_2}) \). For a fixed \( i, 0 \leq i \leq k_1 \), let \( S_i(\pi_1, \pi_2) \) be the set of rankings in \( \mathcal{J}_n \) consistent with both \( \pi_1 \) and \( \pi_2 \), where the head of \( \pi_2 \) is located after the \( i \)th element of \( \pi_1 \), i.e.,

\[
S_i(\pi_1, \pi_2) = \{ \sigma \in \mathcal{J}_n : \sigma \in \sigma(\pi_1) \cap \sigma(\pi_2), \text{ with } \sigma(a_{2,1}) \geq i + 1 \}.
\]

Then,

\[
\lambda(S_i(\pi_1, \pi_2)) = \prod_{j=1}^{i} \frac{v_{1,j}}{\sum_{j' = j}^{k_1} v_{1,j'}} \prod_{j = i+1}^{k_1} \frac{v_{1,j}}{\sum_{j' = j}^{k_1} v_{1,j'}} \lambda(\pi_2).
\]

The first new result refers to an expression for the probability of DAG \( D \) that is split in two independent terms by breaking the bottom part of \( D \) in two pieces.

**LEMMA A1.** Given a DAG \( D \), let \( a_y \in N \) be a node such that every node in \( \Psi_D(a_y) \setminus \{a_y\} \) has at most one incoming edge and the subgraph \( D[a_y] \) induced in \( D \) by the set of nodes \( \Psi_D(a_y) \) is a directed tree; see Figure A1. Further, let \( \bar{D}[a_y] \) denote the subgraph induced in \( D \) by the set of nodes \( (N \setminus \Psi_D(a_y)) \cup \{a_y\} \). Then, under the MNL distribution \( \lambda \), we have that

\[
\lambda(D) = \lambda(D[a_y]) \cdot \lambda_y(\bar{D}[a_y]),
\]

where \( \lambda_y \) is the distribution over rankings obtained by replacing the MNL weight \( v_y \) of product \( a_y \) with \( v_{\Psi_D(a_y)} = \sum_{a_i \in \Psi_D(a_y)} v_i \).
Illustration for the proof in Lemma A1. The bottom part of a DAG is split in two independent terms. Note that DAG $D[a_y]$ is defined by nodes in $\Psi_D(a_y)$.

**Proof of Lemma A1:** Note that $D[a_y]$ is the tree “hanging” from the node $a_y$ in DAG $D$. We establish the result of this lemma by showing that the term $\lambda(D[a_y])$ factors out from the expression for $\lambda(D)$.

For that, let $S_1$ denote $\mathcal{N} \setminus \Psi_D(a_y)$ and $S_2$ denote $\Psi_D(a_y)$. It is clear that $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = \mathcal{N}$. For any ranking $\sigma$ and position $1 \leq r \leq n$, let $\sigma^{-1}(r)$ denote the product ranked at position $r$ under $\sigma$. Let $D_1$ and $D_2$ denote the subgraphs in $D$ induced by the sets $S_1$ and $S_2$, respectively. It follows from our notation that $D_2 = D[a_y]$. Let $E_1$ and $E_2$ denote the edges in the transitive reductions of $D_1$ and $D_2$, respectively.

With this notation, we first establish the following result.

**Claim:** The set $E_D$ of edges in the transitive reduction of $D$ can be partitioned as

$$E_D = E_1 \cup E_2 \cup E_3,$$

where $E_3 = \{(a, a_y) : (a, a_y) \in E_D\}$ and $E_i \cap E_j = \emptyset \forall 1 \leq i \neq j \leq 3$ \hspace{1cm} (A2)

**Proof.** We first note that $E_1 \cup E_2 \cup E_3 \subseteq E_D$ since it follows by definition that $E_i \subseteq E_D$ for all $1 \leq i \leq 3$. To show that $E_D \subseteq E_1 \cup E_2 \cup E_3$, consider an edge $(a, b) \in E_D$. We note that if $a \in S_2$, then $b$ must belong to $S_2$. The reason is that since $S_2 = \Psi_D(a_y)$, if $a \in S_2$, then $a$ is reachable from $a_y$ and because $b$ is reachable from $a$, it must be that $b$ is also reachable from $a_y$, which implies that $b \in \Psi_D(a_y) = S_2$. Therefore, there are two cases to consider: (i) both $a$ and $b$ belong to $S_1$ or $S_2$ and (ii) $a \in S_1$ and $b \in S_2$. In case (i), it follows by definition that $(a, b)$ belongs to $E_1$ or $E_2$. In case (ii), since every node in $S_2 \setminus \{a_y\}$ can have at most one incoming edge and every node in $S_2 \setminus \{a_y\}$ already has an incoming edge from a node in $S_2$, the only way there can be an edge from $a \in S_1$ to $b$ is if $b = a_y$. It now follows that $(a, b) = (a, a_y) \in E_3$. We have thus shown that $E_D = E_1 \cup E_2 \cup E_3$. The fact that the three sets $E_1$, $E_2$, and $E_3$ are mutually disjoint follows immediately from noting that the sets $S_1$ and $S_2$ are disjoint.
With the above decomposition of the edges of \( E_D \), we now show that the set of rankings \( S_D \) can be decomposed in a convenient manner. Consider the following definitions:

- Let \( \pi_1 \) (of length \( k_1 \)) and \( \pi_2 \) (of length \( k_2 \)) be rankings of products in the sets \( S_1 \) and \( S_2 \), respectively. Note that \( k_1 + k_2 = n \).
- Let \( X \) be the set of tuples \((\pi_1, \pi_2)\) such that \( \pi_1 \) and \( \pi_2 \) are consistent with DAGs \( D_1 \) and \( D_2 \), respectively. In other words, \( X = \{(\pi_1, \pi_2) : \sigma(\pi_1) \subseteq S_{D_1}, \sigma(\pi_2) \subseteq S_{D_2}\} \).
- For any \( 1 \leq i \leq k_1 \), let \( S_i(\pi_1, \pi_2) \) denote the set of rankings in \( \mathcal{S}_n \) consistent with both \( \pi_1 \) and \( \pi_2 \) where the head of \( \pi_2 \) is located after the \( i \)-th element of \( \pi_1 \), i.e., \( S_i(\pi_1, \pi_2) = \{\sigma \in \mathcal{S}_n : \sigma \subseteq \sigma(\pi_1) \cap \sigma(\pi_2), \text{ with } \sigma(\pi_2^{-1}(1)) \geq \sigma(\pi_1^{-1}(i)) + 1\} \).
- Further, let \( i(\pi_1) \) denote the position of the least preferred item in \( \Theta_D(\pi_1) \) in the ranking \( \pi_1 \), i.e., \( i(\pi_1) = \max\{\pi_1(a) : a \in \Theta_D(\pi_1)\} \).

**Claim:** The set of rankings \( S_D \) is obtained by taking a tuple \((\pi_1, \pi_2) \in X\) and combining them such that the head of \( \pi_2 \) occurs after the \( i(\pi_1) \)-th element of \( \pi_1 \). More precisely, we claim that

\[
S_D = \bigcup_{(\pi_1, \pi_2) \in X} S_{i(\pi_1)}(\pi_1, \pi_2), \quad \text{where } S_{i(\pi_1)}(\pi_1, \pi_2) \cap S_{i'(\pi_1)}(\pi_1, \pi_2') = \emptyset \text{ for } (\pi_1, \pi_2) \neq (\pi_1', \pi_2') \tag{A3}
\]

**Proof.** We first show that \( S_{i(\pi_1)}(\pi_1, \pi_2) \subseteq S_D \) for all \((\pi_1, \pi_2) \in X \). For that, consider \( \sigma \in S_{i(\pi_1)}(\pi_1, \pi_2) \) and consider an edge \((a, b) \in E_D\). It is sufficient to show that \( \sigma(a) < \sigma(b) \). It follows from (A2) that \((a, b)\) is in either \( E_1 \) or \( E_2 \) or \( E_3 \). If \((a, b)\) is in \( E_1 \), then we must have that \( \pi_1(a) < \pi_1(b) \) because \( \pi_1 \) is consistent with \( D_1 \). Since \( \sigma \) is consistent with \( \pi_1 \), we have that \( \sigma(a) < \sigma(b) \). Using a symmetric argument, we can similarly show that \( \sigma(a) < \sigma(b) \) when \((a, b) \in E_2 \).

Now suppose that \((a, b) \in E_3 \). We then have that \( b = a_y \). Since \( \pi_2 \) is consistent with \( D_2 \) and \( a_y \) is preferred over every product in \( S_2 \setminus \{a_y\} \) under the partial order \( D_2 \), it follows that \( a_y \) must be the head of \( \pi_2 \), i.e., \( \pi_2(a_y) = 1 \). Now, let \( a^* \) denote the least preferred element under \( \pi_1 \) in the set \( \Theta_D(a_y) \). Since \((a, a_y) \in E_D \), we have that \( a \in \Theta_D(a_y) \), implying that \( \sigma(a) = \pi_1(a) \leq \pi_1(a^*) = \sigma(a^*) \), with equality when \( a = a^* \). Note that both rankings \( \sigma \) and \( \pi_1 \) must coincide until position \( i(\pi_1) \). It also follows by our definitions that \( \pi_1(a^*) = i(\pi_1) \) and \( \sigma(a_y) = \sigma(\pi_2^{-1}(1)) > \sigma(\pi_1^{-1}(i(\pi_1))) = \sigma(a^*) \), where the inequality follows from the definition of \( S_{i(\pi_1)}(\pi_1, \pi_2) \) and the fact that \( \sigma \in S_{i(\pi_1)}(\pi_1, \pi_2) \). We have thus shown that \( \sigma(a) \leq \sigma(a^*) < \sigma(a_y) = \sigma(b) \), establishing the result that \( S_{i(\pi_1)}(\pi_1, \pi_2) \subseteq S_D \) for all \((\pi_1, \pi_2) \in X \), which implies that \( \bigcup_{(\pi_1, \pi_2) \in X} S_{i(\pi_1)}(\pi_1, \pi_2) \subseteq S_D \).

We now show that \( S_D \subseteq \bigcup_{(\pi_1, \pi_2) \in X} S_{i(\pi_1)}(\pi_1, \pi_2) \). For that consider \( \sigma \in S_D \) and let \( \pi_1 \) and \( \pi_2 \) denote the rankings \( \sigma \) induced on the set of products \( S_1 \) and \( S_2 \), respectively. We show that \( \sigma \in S_{i(\pi_1)}(\pi_1, \pi_2) \). It follows by the definitions of \( \pi_1 \) and \( \pi_2 \) that \( \sigma \in \sigma(\pi_1) \cap \sigma(\pi_2) \). Therefore, it is sufficient to show that \( \sigma(\pi_2^{-1}(1)) \geq \sigma(\pi_1^{-1}(i(\pi_1))) + 1 \). Using the arguments above, it readily follows that \( \pi_2^{-1}(1) = a_y \). Since \( \sigma \) is consistent with \( D \), we have that \( \sigma(a) < \sigma(a_y) \) for all \( a \in \Theta_D(a_y) \),
and in particular, $\sigma(a^*) < \sigma(a_y)$, where $a^*$ is the least preferred product under $\pi_1$ from the set $\Theta_D(a_y)$. Since $i(\pi_1) = \pi_1(a^*)$ by definition, we have shown that $\sigma(\pi_1^{-1}(i(\pi_1))) < \sigma(a_y)$. We have thus established that $S_D \subseteq \bigcup_{(\pi_1, \pi_2) \in X} S_i(\pi_1)(\pi_1, \pi_2)$.

The arguments above establish that $S_D = \bigcup_{(\pi_1, \pi_2) \in X} S_i(\pi_1)(\pi_1, \pi_2)$. The disjointness of $S_i(\pi_1)(\pi_1, \pi_2)$ and $S_{i(\pi_1)}(\pi_1, \pi_2)$ for $(\pi_1, \pi_2) \neq (\pi_1', \pi_2')$ readily follows from the disjointness of $\sigma(\pi_1) \cap \sigma(\pi_2)$ and $\sigma(\pi_1') \cap \sigma(\pi_2')$ for $(\pi_1, \pi_2) \neq (\pi_1', \pi_2')$. We have thus established the claim in (A3).

We can then write from (A3) that

$$\lambda(D) = \sum_{\sigma \in S_D} \lambda(\sigma) = \sum_{(\pi_1, \pi_2) \in X} \lambda(S_i(\pi_1)(\pi_1, \pi_2)) = \sum_{\pi_1: \sigma(\pi_1) \subseteq S_D_1} \sum_{\pi_2: \sigma(\pi_2) \subseteq S_D_2} \lambda(S_i(\pi_1)(\pi_1, \pi_2)).$$

Now consider $(\pi_1, \pi_2) \in X$. Without loss of generality, suppose that $\pi_1 = (a_1, \ldots, a_{k_1})$ and $\pi_2 = (a_2, \ldots, a_{k_2})$. Then, invoking (Jagabathula and Vulcano 2018, Lemma A1), we can write

$$\lambda(S_i(\pi_1)(\pi_1, \pi_2)) = \left[ \prod_{j=1}^{i(\pi_1)} \frac{v_{1,j}}{\sum_{j'=j}^{k_1} v_{1,j'} + \sum_{j'=i(\pi_1)} v_{2,j'} + \sum_{j'=1}^{k_1} v_{1,j'} + \sum_{j'=1}^{k_2} v_{2,j'}} \right] \cdot \lambda(\pi_2)$$

$$= g(\pi_1) \cdot \lambda(\pi_2),$$

where the second equality follows from the fact that $S_2 = \Psi_D(a_y)$, and where we define

$$g(\pi_1) = \prod_{j=1}^{i(\pi_1)} \frac{v_{1,j}}{\sum_{j'=j}^{k_1} v_{1,j'} + \sum_{j'=i(\pi_1)+1} v_{2,j'} + \sum_{j'=1}^{k_1} v_{1,j'} + \sum_{j'=1}^{k_2} v_{2,j'}}.$$

We now have

$$\lambda(D) = \sum_{\pi_1: \sigma(\pi_1) \subseteq S_D_1} \sum_{\pi_2: \sigma(\pi_2) \subseteq S_D_2} \lambda(S_i(\pi_1)(\pi_1, \pi_2)) = \left[ \sum_{\pi_1: \sigma(\pi_1) \subseteq S_D_1} g(\pi_1) \right] \cdot \left[ \sum_{\pi_2: \sigma(\pi_2) \subseteq S_D_2} \lambda(\pi_2) \right].$$

Noting that

$$\sum_{\pi_2: \sigma(\pi_2) \subseteq S_D_2} \lambda(\pi_2) = \sum_{\sigma \in S_D_2} \lambda(\sigma) = \lambda(D[a_y]),$$

we have shown that

$$\lambda(D) = \lambda(D[a_y]) \cdot \left[ \sum_{\pi_1: \sigma(\pi_1) \subseteq S_D_1} g(\pi_1) \right]. \tag{A4}$$

It now suffices to show that

$$\lambda_y(D[D[a_y]]) = \sum_{\pi_1: \sigma(\pi_1) \subseteq S_D_1} g(\pi_1).$$

For that, consider the distribution $\lambda_y$ in which the weight $v_y$ is replaced by $v_{\Psi_D(a_y)}$, and repeat the above set of arguments for the DAG $D[D[a_y]]$ with the nodes of the DAG decomposed into sets $S_1$ as...
defined above and $S_3 = \{a_y\}$. For any ranking $\pi_1$ of the products in set $S_1$, note that $g(\pi_1)$ remains the same under both distributions $\lambda$ and $\lambda_y$. As a result, following (A4), we can write

$$\lambda_y(\bar{D}[a_y]) = \lambda_y(D_3) \cdot \left[ \sum_{\pi_1: \sigma(\pi_1) \subset S_{D_1}} g(\pi_1) \right],$$

where $D_3$ is the DAG induced in $D$ by $S_3$. Since $S_3$ is a singleton, the DAG $D_3$ will be empty, implying that $\lambda_y(D_3) = 1$. We have thus shown that

$$\lambda_y(\bar{D}[a_y]) = \sum_{\pi_1: \sigma(\pi_1) \subset S_{D_1}} g(\pi_1).$$

The result of the lemma now follows. $\square$

To establish the next result, we first quote the so-called internal consistency property referred to in Jagabathula and Vulcano (2018), i.e., the probability of a ranking in a subset of products only depends on the relative order of the elements in the subset.

**Proposition A1.2 in Jagabathula and Vulcano (2018).** For any subset $S$ of $k$ products, let $\pi$ be a ranking over the elements of $S$. Let $\sigma(\pi)$ denote the set of full rankings over $\mathcal{N}$ defined as

$$\sigma(\pi) = \{\sigma: \sigma(a_i) < \sigma(a_j) \text{ whenever } \pi(a_i) < \pi(a_j) \text{ for all } a_i, a_j \in S\}.$$

Then, it must hold that

$$\lambda(\pi) = \sum_{\sigma \in \sigma(\pi)} \lambda(\sigma) = \prod_{r=1}^{k} \frac{v_{\pi_r}}{\sum_{j=r}^{k} v_{\pi_j}}.$$

We also need the following notation. Given a DAG $D$ and product $a_y$, let $\lambda^\text{aug}_{y}$ denote the distribution of rankings on the expanded product universe $\mathcal{N} \cup \{a_y'\}$, where $a_y'$ is a copy of the product $a_y$, with the weight $v_y$ of product $a_y$ replaced with $v_{\Psi_D(a_y)}$ and the copy $a_y'$ also assigned the weight $v_{\Psi_D(a_y)}$.

We say that a node $a$ in DAG $D$ is a $v$-node if it has more than one incoming edge. We define the $v$-degree of a DAG $D$ as $\sum_{a_j} a_j$ is a $v$-node $(d^\text{in}_j - 1)$, where $d^\text{in}_j$ is the in-degree of node $a_j$. We now establish the following result.

**Lemma A2.** Suppose that a leaf node $a_y$ in the DAG $D$ has at least two incoming edges. Then there exists DAG $D^\text{split}$ whose $v$-degree is one less than that of $D$, such that

$$\lambda^\text{aug}_{y}(D^\text{split}) \leq \lambda(D).$$

The inequality is strict when all the parameters under the MNL model are positive.

Furthermore the approximate likelihoods of the DAGs $D$ and $D^\text{split}$ are equal, i.e., $\hat{\lambda}^\text{aug}_{y}(D^\text{split}) = \hat{\lambda}(D) = \prod_{a \in \mathcal{N}} \frac{\hat{v}_a}{\sum_{a' \in \Psi_D(a)} \hat{v}_{a'}}.$
Proof of Lemma A2: Since the leaf node $a_y$ in DAG $D$ has at least two incoming edges, suppose w.l.o.g. that $(a_1, a_y), (a_2, a_y) ∈ E_D$. Let $D_1$ denote the DAG obtained by adding the isolated copy $a'_y$ to $D$. Let $D^{\text{split}}$ denote the DAG obtained by erasing the edge $(a_1, a_y)$ and adding the edge $(a_2, a'_y)$ to $D_1$; in other words, $E_{D^{\text{split}}} = E_D \setminus \{(a_1, a_y)\} \cup \{(a_2, a'_y)\}$. Figure A2 illustrates these DAGs. Note that by construction, the $v$-degree of $D^{\text{split}}$ is one less than that of $D$ because the in-degree of node $a_y$ has been reduced by 1.

We need the following intermediate result.

**Claim:** $\lambda(D) = \lambda_y^{\text{aug}}(D_1)$.

**Proof.** Note that since $a_y$ is a leaf node in $D$, we have that $v_{\Psi_D}(a_y) = v_y$. Therefore, the distribution $\lambda_y^{\text{aug}}$ is defined on the expanded universe $\mathcal{N} \cup \{a'_y\}$ with the weights of the products in $\mathcal{N}$ remaining the same as in $\lambda$ and the weight $v_y$ assigned to product $a'_y$.

Now, for any ranking $\pi ∈ S_D$ (including only products in set $\mathcal{N}$), let $\sigma(\pi) ⊂ S_{D_1}$ denote the set of rankings of the products in the set $\mathcal{N} \cup \{a'_y\}$ that are consistent with $\pi$. By invoking (Jagabathula and Vulcano 2018, Proposition A1.2), it holds that $\lambda(\pi) = \sum_{\sigma(\pi)} \lambda_y^{\text{aug}}(\sigma)$. We can now write

$$\lambda_y^{\text{aug}}(D_1) = \sum_{\sigma ∈ S_{D_1}} \lambda_y^{\text{aug}}(\sigma) = \sum_{\pi ∈ S_D} \sum_{\sigma(\pi)} \lambda_y^{\text{aug}}(\sigma) = \sum_{\pi ∈ S_D} \lambda(\pi) = \lambda(D).$$

Therefore, in order to establish that $\lambda_y^{\text{aug}}(D^{\text{split}}) ≤ \lambda(D)$, it is sufficient to show that $\lambda_y^{\text{aug}}(D^{\text{split}}) ≤ \lambda_y^{\text{aug}}(D_1)$. For that, consider the three DAGs $I_1$, $I_2$, and $I_3$ (see Figure A3), defined over the set of products in $\mathcal{N} \cup \{a'_y\}$ such that

$$E_{I_1} = E_D \setminus \{(a_2, a_y)\}, E_{I_2} = E_D \setminus \{(a'_y, a_2)\}, \text{ and } E_{I_3} = (E_{D_1} \setminus \{(a_1, a_y)\}) \cup \{(a_y, a_2), (a_2, a'_y)\}.$$

It is then follows by the definitions that

$$\lambda_y^{\text{aug}}(D_1) = \lambda_y^{\text{aug}}(I_1) + \lambda_y^{\text{aug}}(I_2)$$

$$\lambda_y^{\text{aug}}(D^{\text{split}}) = \lambda_y^{\text{aug}}(I_1) + \lambda_y^{\text{aug}}(I_3).$$

Figure A2: Bottom parts of DAGs from Lemma A2.
Thus, to show that $\lambda^\text{aug}_y(D_{\text{split}}) \leq \lambda^\text{aug}_y(D_1)$, it is sufficient to show that $\lambda^\text{aug}_y(I_3) \leq \lambda^\text{aug}_y(I_2)$.

To show that $\lambda^\text{aug}_y(I_3) \leq \lambda^\text{aug}_y(I_2)$, define the mapping $h: I_3 \to I_2$ such that for any $\sigma \in I_3$, which is of the form $\sigma = (\ldots, a_1, \ldots, a_y, \ldots, a_2, \ldots, a'_{y}, \ldots)$, we map it to $\sigma'$ in $I_2$, of the form $\sigma' = (\ldots, a_1, \ldots, a'_{y}, \ldots, a_2, \ldots, a_y, \ldots)$ obtained by swapping the positions of the products $a_y$ and $a'_{y}$. Now, it can be verified that the mapping $h(\cdot)$ is an injection, i.e., $h(\sigma) \neq h(\sigma')$ whenever $\sigma \neq \sigma'$. Then, since attraction parameters for nodes $a_y$ and $a'_{y}$ are the same, i.e., $v_{a_y} = v_{a'_{y}}$, it follows that for any $\sigma \in I_3$, $\lambda^\text{aug}_y(\sigma) = \lambda^\text{aug}_y(h(\sigma))$. As a result, we obtain

$$\lambda^\text{aug}_y(I_3) = \sum_{\sigma \in I_3} \lambda^\text{aug}_y(\sigma) = \sum_{\sigma \in I_3} \lambda^\text{aug}_y(h(\sigma)) \leq \sum_{\sigma' \in I_2} \lambda^\text{aug}_y(\sigma') = \lambda^\text{aug}_y(I_2),$$

(A5)

where the inequality holds because there could be additional rankings $\sigma' \in I_2$ that do not have a pre-image in $I_3$ through $h(\cdot)$. We have thus shown that $\lambda^\text{aug}_y(D_{\text{split}}) \leq \lambda^\text{aug}_y(D_1)$, which implies that $\lambda^\text{aug}_y(D_{\text{split}}) \leq \lambda(D)$.

The inequality in $\lambda^\text{aug}_y(D_{\text{split}}) \leq \lambda(D)$ is strict when all the parameters under the MNL model are positive. We establish this result by showing that there exists $\sigma' \in I_2$ such that $h(\sigma) \neq \sigma'$ for all $\sigma \in I_3$. It then follows that the inequality in (A5) is strict, implying that $\lambda^\text{aug}_y(D_{\text{split}}) < \lambda(D)$. Consider the ranking $\sigma' = (\ldots, a'_{y}, \ldots, a_1, \ldots, a_2, \ldots, a_y, \ldots)$ such that $\sigma' \in I_2$. As noted above, any $\sigma \in I_3$ is of the form $\sigma = (\ldots, a_1, \ldots, a_y, \ldots, a_2, \ldots, a'_{y}, \ldots)$, so that it gets mapped to $h(\sigma) = (\ldots, a_1, \ldots, a'_y, \ldots, a_2, \ldots, a_y, \ldots)$. Therefore, we have $h(\sigma)(a_1) < h(\sigma)(a'_{y})$ for all $\sigma \in I_3$, whereas $\sigma'(a_1) > \sigma'(a'_y)$. Thus, we have that $\sigma' \neq h(\sigma)$ for all $\sigma \in I_3$, establishing the claim.

We are now left with showing that $\lambda^\text{aug}_y(D_{\text{split}}) = \hat{\lambda}(D)$. Since the reachability weights $v_{\Psi_D(a)}$ for all $a \in \mathcal{N}$ under $\lambda$, and $v_{\Psi_{D_{\text{split}}}(a)}$ for all $a \in \mathcal{N} \cup \{a'_{y}\}$ under $\lambda^\text{aug}$, are equal by definition, and the approximations $\tilde{\lambda}$ and $\lambda^\text{aug}$ only depend on the reachability weights, then the equality $\lambda^\text{aug}_y(D_{\text{split}}) = \hat{\lambda}(D)$ immediately follows.

We can now proceed to prove the results in Section 3.
Proof of Proposition 1

We show the result, $\lambda(D) \leq \lambda(D)$, by induction on the $v$-degree, $k$, of DAG $D$.

Base case: $k = 0$. When $k = 0$, DAG $D$ does not have any $v$-nodes. Then, $D$ is a forest of directed trees each with a unique root. It follows from (Jagabathula and Vulcano 2018, Proposition 3.2) that $\lambda(D) = \lambda(D) = \prod_{a \in \mathcal{N}} \sum_{a' \in \Psi_D(a)} v_{a'}$, establishing the base case.

Induction hypothesis: Suppose $\mu(D) \leq \mu(D)$ for any DAG $D$ with $v$-degree less than or equal to $p$, for some $p \geq 0$, for all distributions $\mu$ under the PL model.

Induction step: Assuming that the induction hypothesis is true, we prove the result for $k = p + 1$. It is clear that there exists a $v$-node $a_y \in \mathcal{N}$ satisfying the conditions in Lemma A1, i.e., every node in $\Psi_D(a_y) \setminus \{a_y\}$ has at most one incoming edge and the subgraph $D[a_y]$, induced in $D$ by the set of nodes $\Psi_D(a_y)$ is a directed tree with unique root. As in Lemma A1, let $\bar{D}[a_y]$ denote the subgraph induced in $D$ by the set of nodes $(\mathcal{N} \setminus \Psi_D(a_y)) \cup \{a_y\}$. Now consider

$$
\tilde{\lambda}(D) = \left( \prod_{j \in \mathcal{N}} \sum_{j' \in \Psi_D(a_j)} v_{j'} \right) \cdot \left( \prod_{j \in \mathcal{N} \setminus \Psi_D(a_j)} v_j \right)
$$

where the fourth equation follows because $D[a_y]$ is a directed tree with a unique root, which implies that $\lambda(D[a_y]) = \tilde{\lambda}(D[a_y])$ (Jagabathula and Vulcano 2018, Proposition 3.2), and the fact that $\Psi_D(a_j) = \Psi_D[a_y](a_j)$ for all $j \in \mathcal{N}$ such that $a_y \notin \Psi_D(a_j)$. We now have

$$
\tilde{\lambda}(D) = \lambda(D[a_y]) \cdot \tilde{\lambda}_y(\bar{D}[a_y])
$$

$$
= \lambda(D[a_y]) \cdot \tilde{\lambda}_y(a_y)
$$

$$
\leq \lambda(D[a_y]) \cdot \lambda_y(\bar{D}[a_y]) \quad \text{[by the induction hypothesis]}
$$

$$
\leq \lambda(D[a_y]) \cdot \lambda_y(\bar{D}[a_y]) \quad \text{[by Lemma A2]}
$$

$$
= \lambda(D) \quad \text{[by Lemma A1]},
$$
where the second equality holds by Lemma A2 taking $\tilde{D}[a_y]$ here as $D$ there, and $D^\text{split}_y$ here as $D^\text{split}$ there; and the first inequality follows from induction hypothesis with distribution $\mu = \lambda^\text{aug}_y$ since the $v$-degree of $D^\text{split}_y$ is equal to $p$. The result of the proposition now follows. \[\square\]

**Proof of Proposition 2**

We must have that $S_D \subset S_{\tilde{D}}$ since if $\sigma$ is consistent with $D$, i.e., $\sigma \in S_D$, then it must also be consistent with $\tilde{D}$, i.e., $\sigma \in S_{\tilde{D}}$. It now follows that

$$\lambda(D) = \sum_{\sigma \in S_D} \lambda(\sigma) \leq \sum_{\sigma \in S_{\tilde{D}}} \lambda(\sigma) = \lambda(\tilde{D}).$$

We now show that $\lambda(D) < \lambda(\tilde{D})$ when all the PL parameters are strictly positive by exhibiting a ranking $\sigma \in S_{\tilde{D}}$ such that $\sigma \notin S_D$. Suppose, by contradiction, we have that $S_{\tilde{D}} = S_D$. Then any subgraph of $\tilde{D}$ which has less edges than $D$, is a transitive reduction of $D$, which results in contradiction (recall that DAG $D$ is assumed to be its unique transitive reduction). As a result, there is $\sigma \in S_D$ such that $\sigma \notin S_{\tilde{D}}$. Then we have

$$\lambda(\tilde{D}) - \lambda(D) \geq \lambda(\sigma) > 0,$$

which holds because all the parameters of $\lambda$ are positive. \[\square\]

For the proof of Proposition 3 we recall here some notation. Let $R(D, \tilde{D})$ denote the ratio of the upper bound to the lower bound $\lambda(\tilde{D}) / \lambda(D)$, for any DAG $\tilde{D} \subset D$. Let $\ell$ denote the size of the largest reachability set in DAG $D$, i.e., $\ell = \max_{a \in \mathcal{N}} |\Psi_D(a)|$, and let $p$ denote the number of nodes with $v$-nodes in their reachability sets, i.e., $p = |\{a \in \mathcal{N} : \exists v\text{-node } b \in \Psi_D(a)\}|$. Further, let $\Delta := \max_{a} \max_{b \in \Psi_D(a) \setminus \{a\}} v_b / v_a$ be the maximum ratio between the weights of nodes within the same directed path in the DAG.

**Proof of Proposition 3**

Define $\Phi(D) \subset D$ as a DAG with each node having a unique parent, such that for any distribution $\lambda$, $\lambda(\Phi(D)) \geq \lambda(D)$. Recall that $\tilde{D}$ is a forest of directed trees obtained by deleting arcs from $D$ in order to break the $v$-nodes. Thus, set $\tilde{D} = \Phi(D)$ so that each node has a unique parent, verifying

$$\lambda(\tilde{D}) = \prod_{a \in \mathcal{N}} \frac{v_a}{\sum_{a_j \in \Psi_{\Phi(D)}(a)} v_{a_j}}.$$ 

In turn, $\lambda(D)$ is the lower bound obtained by treating $D$ as a forest of directed trees with unique root. That is,

$$\tilde{\lambda}(D) = \prod_{a \in \mathcal{N}} \frac{v_a}{\sum_{a_j \in \Psi_D(a)} v_{a_j}}.$$
We have from Propositions 1 and 2:

\[ \tilde{\lambda}(D) \leq \lambda(D) \leq \lambda(\tilde{D}). \quad (A6) \]

Then,

\[ \log R(D, \tilde{D}) = \log \frac{\lambda(\tilde{D})}{\lambda(D)} = \log \left( \prod_{a \in N} \frac{\sum_{j \in \Psi_D(a)} v_j}{\sum_{j \in \Psi_{\Phi(D)}(a)} v_j} \right) \]

Let \( F_D \) be the set of nodes in \( D \) with more than one incoming edge. Continuing the sequence of equalities above:

\[ \log R(D, \tilde{D}) = \log \left( \prod_{a \in N} \frac{\sum_{j \in \Psi_D(a)} v_j}{\sum_{j \in \Psi_{\Phi(D)}(a)} v_j} \right) = \sum_{a \in N} \log \left( 1 + \frac{\sum_{j \in \Psi_D(a) \setminus \Psi_{\Phi(D)}(a)} v_j}{\sum_{j \in \Psi_{\Phi(D)}(a)} v_j} \right) \]

\[ = \sum_{a \in N} \| [ F_D \cap \Psi_D(a) \neq \emptyset ] \| \cdot \log \left( 1 + \frac{\sum_{j \in \Psi_D(a) \setminus \Psi_{\Phi(D)}(a)} v_j}{v_a} \right) \]

\[ \leq \sum_{a \in N} \| [ F_D \cap \Psi_D(a) \neq \emptyset ] \| \cdot \log \left( 1 + \sum_{j \in \Psi_D(a) \setminus \Psi_{\Phi(D)}(a)} \frac{v_j}{v_a} \right) \]

\[ \leq \sum_{a \in N} \| [ F_D \cap \Psi_D(a) \neq \emptyset ] \| \cdot \log \left( 1 + \sum_{j \in \Psi_D(a) \setminus \Psi_{\Phi(D)}(a)} \Delta \right) \]

where the fourth equality follows since the surviving terms are the once where \( \Psi_D(a) \setminus \Psi_{\Phi(D)}(a) \neq \emptyset \), i.e., there are nodes in \( \Psi_D(a) \) with more than one incoming edge; and the first inequality holds because \( a \in \Psi(a) \), and we add terms in the numerator and take out terms from the denominator.

The last three inequalities follow from the definitions of \( \Delta \), \( \ell \), and \( p \), respectively.

From (A6), we have that

\[ 0 \leq \lim_{n \to \infty} \log \frac{\lambda(D)}{\lambda(\tilde{D})} \leq \lim_{n \to \infty} p \cdot \log(1 + \ell \cdot \Delta) \leq \lim_{n \to \infty} n \cdot \log(1 + n \cdot \Delta) \]

\[ = \lim_{n \to \infty} \log(1 + n \cdot \Delta)^{\frac{2^\Delta}{\Delta}} = \lim_{n \to \infty} (\Delta n^2) \cdot \log(1 + n \cdot \Delta)^{\frac{1}{n}} = 0, \]

since as \( n \to \infty \), it can be shown that \( n\Delta \in o(n^{-1}) \) and \( n^2 \Delta \in o(1) \). \( \Box \)
Proof of Proposition 4

Recall that the merged DAG $D \cup C(a_j, S)$ is obtained by taking the union of the graphs $D$ and $C(a_j, S)$. From the definitions of the bounds in (4), it must hold that

$$
\log \frac{f(a_j, S, D)}{f(a_j, S, D)} = \log \left( \frac{\lambda(D \cup C(a_j, S))}{\lambda(D)} \cdot \frac{\lambda(D)}{\lambda(D \cup C(a_j, S))} \right)
$$

$$
= \log \left( \frac{\lambda(D \cup C(a_j, S))}{\lambda(D)} \right) + \log \left( \frac{\lambda(D)}{\lambda(D \cup C(a_j, S))} \right)
$$

$$
\leq 2 \cdot p \cdot \log(1 + \ell \cdot \Delta),
$$

where the last inequality follows from Proposition 3. As argued in the proof of Proposition 3, $p \log(1 + \ell \Delta) \to 0$ as $n \to \infty$ when $\Delta n^2 \in o(1)$, as $n \to \infty$. \qed

A3. Heuristic for preference graph decycling

In Section 2.4 we formulated MILP (1) for preference graph decycling (Phase 3) in the DAG construction process. Since solving the MILP to optimality could be challenging (e.g., if we have thousands of products or brands), we propose a tractable, greedy heuristic to decycle the preference graph.

Let $FindPath(a_k, a_j, G)$ denote the output of Dijkstra’s algorithm on a directed graph $G$, which finds the shortest path between nodes $a_k$ and $a_j$ and returns the set of weighted edges comprising this path (potentially, the empty set). The Dijkstra’s algorithm runs in $O(|V_G|^2)$ time, where $V_G$ is the set of nodes in $G$.

Taking advantage of the polynomial running time of Dijkstra’s, our heuristic proceeds as follows: For a directed graph $G$, we run Dijkstra’s between all pair of nodes $a_k$ and $a_j$, in both directions. In case both paths exist, then there is a cycle containing $a_k$ and $a_j$, and the edge with minimum weight is removed. As a result, since there are $O(|V_G|^2)$ pairs of nodes, the preference graph decycling can be implemented with $O(|V_G|^4)$ computational complexity. The steps are described in Algorithm 1 below.

In order to validate the effectiveness of Algorithm 1, next we compare empirical prediction results on the actual sales dataset obtained from DAGs decycled via MILP (1) and the analogous results from DAGs decycled via Algorithm 1. A description of the sales data is provided in Section 4.1 in the main body of the paper.

In Figure A4 (left panels), we observe that using MILP (1) we delete 6.27% fewer edges than when using Algorithm 1 and obtain 0.4% denser DAGs on average across 27 product categories. In the middle and right panels we represent the scatter plot over 27 product categories of the average miss rate and $\chi^2$ scores, respectively. It follows that by using MILP (1) we obtain 1.17% lower miss
Algorithm 1 Preference graph $G$ decycling

1: procedure Decycle($G$) \Comment{Where $G$ is a graph with set of nodes $V_G$ and set of weighted edges $E_G$.}

2: \hspace{1em} for $a_k$ in $V_G$ do

3: \hspace{2em} for $a_j$ in $V_G \setminus \{a_k\}$ do

4: \hspace{3em} while $\text{FindPath}(a_k, a_j, G) \neq \emptyset$ \& $\text{FindPath}(a_j, a_k, G) \neq \emptyset$ do

5: \hspace{4em} $Cycle \leftarrow \text{FindPath}(a_k, a_j, G) \cup \text{FindPath}(a_j, a_k, G)$

6: \hspace{3em} Remove the edge $(a_x, a_y)$ with minimum weight in $Cycle$ from the set $E_G$

7: \hspace{1em} return DAG $D = G$

rate and 2.75% lower $\chi^2$ score than by using Algorithm 1. These results provide good support for the use of the greedy heuristic as an alternative to the exact solution of MIP (1) in cases where the number of products is large. We highlight here though that in all our experiments reported in the main body of the paper we used MIP (1) limited to a max time of 30 seconds, and retaining the best feasible solution when optimality was not reached. According to Table 1, the largest category contains 95 products in our case.

Figure A4 Comparison of the the performance of heuristic Algorithm 1 vs. MIP (1) for 27 product categories. Left panel: scatter plot of the average percentage of edges deleted. Middle panel: scatter plot of the average miss rate. Right panel: scatter plot of the average $\chi^2$ score.

A4. Benchmark models

A4.1. LC-MNL model

LC-MNL model captures heterogeneity among customers by allowing them to belong to $K$ different classes with some probability. Customers from class $h \in \{1, \ldots, K\}$ make choices according to the single class MNL model with a parameter value $\beta_{0j} + I_{jit} \beta_{hj}t$ of product $j_{it} \in \{1, 2, \ldots, n\}$, where $I_{jit} = 1$ if product $j_{it}$ is under promotion at time $t$ for individual $i$, and 0 otherwise. A prior
probability of a customer to belong to the class $h$ is $\gamma_h \geq 0$ such that $\sum_{h=1}^{K} \gamma_h = 1$. The regularized maximum likelihood problem under $K$ class LC-MNL model can be formulated as follows:

$$\max_{\beta, \gamma: \beta_{h1} = 0, h \in [K]} \sum_{i=1}^{m} \log \left( \sum_{h=1}^{K} \gamma_h \prod_{t=1}^{T_i} \frac{\exp(\beta_{hji}^0 + I_{jit}\beta_{hjt})}{\sum_{a \in S_{it}} \exp(\beta_{hia}^0 + I_{ita}\beta_{hia})} \right) - \alpha \sum_{h=1}^{K} (\|\beta_h^0\|_1 + \|\beta_h\|_1)$$

When the value of $\alpha$ is fixed and $K = 1$, it can be shown that this optimization problem is globally concave and therefore can be solved efficiently (Train 2009). Note that we tuned the value of $\alpha$ by 5-fold cross-validation. Since the problem is nonconcave for $K > 1$, the EM technique is used to fit the model (see Appendix A2.1.1 in Jagabathula and Vulcano 2018). Specifically, we initialize the EM with a random allocation of customers to one of the $K$ classes, resulting in an initial allocation $D_1, D_2, \ldots, D_K$, which form a partition of the collection of all the customers. Then we set $\gamma_h^{(0)} = |D_h|/(\sum_{d=1}^{K} |D_d|)$. In order to get a parameter vector $(\beta_{h1}^{(0)}, \beta_h)$, we fit a single class MNL model to each subset of customers. Based on each customer $i$’s purchase history $(a_{jit}, S_{it})$ for $1 \leq t \leq T_i$, we can estimate their posterior membership probabilities $\forall h \in \{1, \ldots, K\}$:

$$\hat{\gamma}_{ih} = \frac{\gamma_h \prod_{t \in T_i} \left[ \exp(\beta_{hji}^0 + I_{jit}\beta_{hjt})/\left( \sum_{a \in S_{it}} \exp(\beta_{hia}^0 + I_{ita}\beta_{hia}) \right) \right]}{\sum_{d=1}^{K} \gamma_d \prod_{t \in T_i} \left[ \exp(\beta_{dji}^0 + I_{jit}\beta_{djt})/\left( \sum_{a \in S_{it}} \exp(\beta_{hia}^0 + I_{ita}\beta_{hia}) \right) \right]},$$

and the prediction can be made as follows:

$$f(j_{it}, S_{it}) = \sum_{h=1}^{K} \hat{\gamma}_{ih} \frac{\exp(\beta_{hji}^0 + I_{jit}\beta_{hjt})}{\sum_{a \in S_{it}} \exp(\beta_{hia}^0 + I_{ita}\beta_{hia})},$$

where $f(j_{it}, S_{it})$ is a probability to choose an item $j_{it}$ from the offer set $S_{it}$.

A4.2. RPL model

In this model, we assume that $\beta$ is sampled from multivariate normal distribution, i.e, $\beta \sim N(\mu, \Sigma)$, where $\mu$ is the mean, and $\Sigma$ is the covariance matrix, which is assumed to be diagonal. Then the log-likelihood of the sequence of purchases of all individuals $i \in \{1, \ldots, m\}$ for $t = \{1, \ldots, T_i\}$ is equal to $\sum_{i=1}^{m} \log \left( \prod_{t=1}^{T_i} \frac{\exp(\beta_{j_{it}}^0 + I_{jit}\beta_{j_{it}})}{\sum_{a \in S_{it}} \exp(\beta_{a_{it}}^0 + I_{ita}\beta_{a_{it}})} \right) \phi(\beta) d\beta$ such that $\beta_{j_{it}}^0 + I_{jit}\beta_{j_{it}}$ is a parameter value of product $j_{it} \in \{1, 2, \ldots, n\}$, where $I_{jit} = 1$ if product $j_{it}$ is under promotion at time $t$ for individual $i$, and 0 otherwise. Model parameters are estimated through maximum simulated likelihood estimation (MSLE) where we use the simulated probabilities to approximate the following log-likelihood function:

$$\max_{\mu, \Sigma: \mu_{1} = 0} \sum_{i=1}^{m} \log \left( \prod_{t=1}^{T_i} \frac{\exp(\beta_{j_{it}}^0 + I_{jit}\beta_{j_{it}})}{\sum_{a \in S_{it}} \exp(\beta_{a_{it}}^0 + I_{ita}\beta_{a_{it}})} \right) \phi(\beta) d\beta,$$
where for any random draw \( r = 1, 2, \ldots, R \) of a random vector \( \xi^r \), that is sampled as \( 2n \)-dimensional multivariate standard normal, we have that \( \beta^r_\ell = \mu_\ell + \xi^r_\ell \sigma_\ell \), for any \( \ell = 1, 2, \ldots, n \), and \( \beta^r_{n+\ell} = \mu_\ell + \xi^r_\ell \sigma_\ell \), \( \ell = n + 1, n + 2, \ldots, 2n \). The above optimization problem is nonconcave. To solve the problem we choose \( R = 400 \) and use a general nonlinear solver to converge to a stationary point (see Appendix A2.1.1 in Jagabathula and Vulcano 2018). Then we make predictions as follows:

\[
f(j_{it}, S_{it}) = \int \frac{\exp(\beta^0_{j_{it}} + I_{j_{it}} \beta_{j_{it}})}{\sum_{a_j \in S_{it}} \exp(\beta^0_{a_j} + I_{a_j} \beta_{a_j})} \hat{\phi}(\beta|H_i; \mu, \Sigma) d\beta,
\]

where \( f(j_{it}, S_{it}) \) is the probability to choose an item \( j_{it} \) from the offer set \( S_{it} \) for individual \( i \), \( \hat{\phi}(\beta|H_i; \mu, \Sigma) \) is the posterior distribution of parameter vector \( \beta \) for customer \( i \), conditioning on population prior and observed choices of customer \( i \), i.e., \( H_i = \{(a_{j_{it}}, S_{it}) : 1 \leq t \leq T_i\} \).

### A5. Evaluation of analytical bounds

In this section, we focus on the PO-MNL Promotion model and illustrate the behavior and quality of the bounds proposed in Section 3.

#### A5.1. Bounds on the probability of a DAG

The presence of \( v \)-nodes (i.e., nodes with more than one incoming edge) in DAGs of individuals complicates the maximum likelihood estimation of parameter values under PO-based choice models. The left panels in Figure A5 illustrate that individuals without cycles in their preference graph have on average 17.09 \( v \)-nodes in their DAG whereas individuals with cycles in their preference graph have on average 19.93 \( v \)-nodes in their DAG.

A tractable approximation of the likelihood of a DAG \( D \) is given by

\[
\tilde{\lambda}(D) = \prod_{a_j \in N} \frac{v_j}{\sum_{a_k \in \Psi_D(a_j)} v_k},
\]

where \( v_j = \exp(\beta_j), \forall a_j \in N \) and \( \Psi_D(a_j) \) denotes the reachability function such that \( \Psi_D(a_j) = \{a_k : a_k \text{ is reachable from } a_j \text{ in } D\} \). Note that \( \Psi_D(a_j) \) is always nonempty, since we assume that each node \( a_j \) is reachable from itself. The approximation \( \tilde{\lambda}(D) \) of the likelihood of DAG \( D \) is exact when \( D \) is a forest of directed trees, each with a unique root. We show in Proposition 1 that \( \tilde{\lambda}(D) \) is a lower bound for the likelihood of DAG \( D \).

Next, in order to find the upper bound approximation of DAG \( D \) likelihood, let us denote \( \tilde{D} \) the DAG obtained from \( D \) where for every node with more than one incoming edge we delete all the incoming edges but one. Instead of deleting an arbitrary set of edges, we can determine edges to delete to make the approximation as tight as possible. Finding the tightest upper bound is challenging in general. In order to ease the computational process, we develop a greedy-type
heuristic $\Phi(D)$ (see Algorithm 2) to obtain a tight upper bound of DAG $D$ likelihood, i.e., $\bar{D} = \Phi(D)$ and $\lambda(D) \leq \lambda(\bar{D})$. 

Figure A5: Analysis of bounds for the probability of a DAG. Top panels: Analysis restricted to individuals without cycles in their preference graph. Bottom panels: Analysis restricted to individuals with cycles in their preference graph. Left panels: Population average number of $v$-nodes for every product category. Middle panels: Population average lower and upper bounds of the negative of the DAG log-likelihood, i.e., $-\log L(X, \beta)$ and $-\log L(X, \beta)$, for every product category. Right panels: Percentage of transactions when prediction of the item to be chosen using the upper bound posterior probability of purchase, i.e., $f(a_j, S, D)$, is different from the prediction of the item to be chosen using the lower bound posterior probability of purchase, i.e., $f(a_j, S, D)$, for every product category.
Algorithm 2 DAG $D$ transformation to find its upper bound likelihood

1: procedure $\Phi(D)$, \(\triangleright\) where $\Phi(D)$ is the DAG with each node having a unique parent s.t. $\lambda(\Phi(D)) \geq \lambda(D)$
2: \(A \leftarrow F_D\) \(\triangleright\) $F_D$ is the set of nodes in $D$ with more than one incoming edge
3: for $a_i$ in $F_D$ do
4: \(D'\) is obtained from $D$: $V_{D'} = V_D$, and $E_{D'} = E_D \setminus B_i$ \(\triangleright\) where $B_i$ is the set of incoming edges into node $a_i$
5: \(D \leftarrow D'\)
6: while $A \neq \emptyset$ do
7: \((a_x, a_y) = \arg \min_{(a_i, a_j) \in B_i} \lambda(D')\) s.t. $D'': V_{D''} = V_D$, $E_{D''} = E_D \cup (a_i, a_j)$ and $a_j \in A$
8: \(D \leftarrow D''\)
9: \(A \leftarrow A \setminus \{a_y\}\)
10: return DAG $\bar{D} = \Phi(D)$

For a collection of panel data represented by $X$ and a given set of parameters $\beta$, let $\log Z(X, \beta)$ denote the upper bound approximation of the log-likelihood function under PO-MNL Promotion model defined as $\log Z(X, \beta) = \sum_{i=1}^{m} \log \lambda(\bar{D}_i)$. Then, letting $\beta^*$ be the solution to the maximization problem of the upper bound of the likelihood function, i.e., $\beta^* = \arg \max_{\beta} \log Z(X, \beta)$, we have that the maximum value of the exact log-likelihood function $\log L(X, \beta^*)$ satisfies:

$$\log L(X, \beta^*) \leq \log Z(X, \beta^*) \leq \log Z(X, \beta^*)$$

Similarly, let $\log L(X, \beta) = \sum_{i=1}^{m} \log \tilde{\lambda}(D_i)$ denote the lower bound approximation of the log-likelihood function under PO-MNL Promotion model, with optimal values $\beta^*$. Then,

$$\log L(X, \beta^*) \leq \log L(X, \beta^*) \leq \log L(X, \beta^*)$$

A natural question that arises is about the size of the gap between both easy-to-compute bounds (lower and upper). The middle column of the panels in Figure A5 illustrates that the upper bound of the log-likelihood function (i.e., $\log Z(X, \beta)$) is higher than the lower bound of the log-likelihood function (i.e., $\log L(X, \beta)$) by 4.72% for individuals without cycles in their preference graph, and by 6.79% for individuals with cycles in their preference graph, on average across 27 product categories. This observation provides good support to use any of the bounds as an approximation for the estimation problem under the exact likelihood of the DAGs. In particular, we used the lower bound $\log L(X, \beta^*)$.

A5.2. Bounds on the probability of purchase

Next we illustrate the behavior and quality of the bounds we have developed for posterior probabilities of purchase when customers make choices consistently with their partial orders. In particular,
we propose the approximate probability of choosing product $a_j$ from offer set $S$ assuming that the sampled preference list is consistent with DAG $D$:

$$\hat{f}(a_j, S, D) = \begin{cases} \frac{\lambda(DuC(a_j, S))}{\lambda(D)}, & \text{if } a_j \in h_D(S), \\ 0, & \text{otherwise.} \end{cases}$$

Letting $\underline{f}(a_j, S, D)$ denote the lower bound of the purchase probability and $\bar{f}(a_j, S, D)$ denote the upper bound of the purchase probability such that $\underline{f}(a_j, S, D) = \frac{\lambda(DuC(a_j, S))}{\lambda(D)}$ if $a_j \in h_D(S)$ and $0$, otherwise; $\bar{f}(a_j, S, D) = \frac{\lambda(D)D(a_j, S)}{\lambda(D)}$ if $a_j \in h_D(S)$, and $0$, otherwise; we have the following inequalities (see Corollary 1 in Section 3):

$$\underline{f}(a_j, S, D) \leq \hat{f}(a_j, S, D) \leq \bar{f}(a_j, S, D),$$

and for the exact and hard-to-compute probability of purchase $f(a_j, S, D)$,

$$\underline{f}(a_j, S, D) \leq f(a_j, S, D) \leq \bar{f}(a_j, S, D).$$

The right column of Figure A5 illustrates that the percentage of transactions when the prediction of the item to be chosen is made using the upper bound posterior probability of purchase, i.e., $\bar{f}(a_j, S, D)$, is different from the prediction of the item to be chosen using the lower bound posterior probability of purchase, i.e., $\underline{f}(a_j, S, D)$, in only 4.04% of the instances for individuals without cycles in their preference graph, and in only 1.79% of the instances for individuals with cycles in their preference graph. In both cases, our prediction is the item with the highest probability of being purchased. This empirical observation provides good support for the use of $\hat{f}(a_j, S, D)$ as a proxy for the true and hard-to-compute probability of purchase $f(a_j, S, D)$.

In our reported results in Section 4 we use the following tractable formula to compute the posterior probabilities of purchase:

$$\tilde{f}(a_j, S, D) = \begin{cases} \frac{\upsilon_D(a_j)}{\sum_{a_k \in h_D(S)} \upsilon_D(a_k)}, & \text{if } a_j \in h_D(S), \\ 0, & \text{otherwise.} \end{cases}$$

This expression is intended to be a good approximation for the alternative approximation $\hat{f}(a_j, S, D)$, which we already know is a good approximation for the exact $f(a_j, S, D)$. We verify this in Figure A6. Therein, we compare the choice prediction results made with $\tilde{f}(a_j, S, D)$ vs. the choice prediction results made with $\hat{f}(a_j, S, D)$ for individuals with and without cycles under “chi-square” score and miss rate (see description of the metrics in Section 4.3). For all the panels in Figure A6 the average MAE (Mean Absolute Error) is below 0.5% which indicates that the posterior probability approximation $\tilde{f}(a_j, S, D)$ is very close to the posterior probability based on the lower bound of the DAG likelihood $f(a_j, S, D)$ in terms of predictive performance.
A6. Robustness check for the prediction results in Section 4

In this section we summarize the major empirical experiments conducted in order to check the robustness of the prediction results reported in Section 4.4. We start checking the robustness with respect to different data aggregation strategies, followed by the effect of accounting for no-purchase observations, the effect of different cutoff points between training and holdout sample data, and the effect of adding implicit candidate edges in Phase 2 of the DAG construction process.

A6.1. Robustness with respect to different data aggregation strategies

We start by showing the robustness of the results to changes in how we calibrate the benchmarks. In particular, in Section 4.4 we calibrated all the models separately on (i) customers who do not have cycles in their preference graph under the PO-MNL Promotion, and (ii) customers who have cycles in their preference graph under the PO-MNL Promotion model. Then we used previous separate calibrations but presented the joint prediction over all the individuals for each category.
of products, i.e., the weighted average prediction performance between both types of customers: with and without cycles in their preference graph. Here we show the prediction performance of our model versus the benchmarks using a display format similar to that in Section 4.4, but when calibrating both benchmarks on the set of all individuals. Note that there is a tension about the benchmarks here since one side they are estimated on a larger volume of data, but at the same time this extra volume comes at the expense of higher customer heterogeneity (i.e., individuals without and with cycles pool together for the estimation process).

Similarly to Figure 3, Figure A7 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 (multi-class), across the 27 product categories. Note that in all the panels we calibrate the benchmarks on the set of all individuals and then separately make predictions for three subsets of customers: without and with cycles, and the entire population.

First, consider the left two panels in Figure A7. Here, we calibrate the PO-MNL Promotion and make predictions with all the models on the subset of individuals who do not have cycles in their preference graph. The “chi-square” score of PO-MNL Promotion model exhibits an average improvement of 9.77% over LC-MNL and 5.79% over RPL. Using the LC PO-MNL Promotion model, we can further boost performance, resulting in average improvement of 14.31% over LC-MNL and 10.28% over RPL.

Second, consider the middle column in Figure A7, where we calibrate the PO-MNL Promotion and make predictions with all the models on the subset of individuals that have cycles in their preference graph. We see that PO-MNL Promotion model exhibits an average improvement of 12.4% over LC-MNL and deterioration of 1.05% over RPL, while LC PO-MNL Promotion model leverages the performance to an average improvement of 13.47% over LC-MNL and 0.2% over RPL.

Third, consider the right panels in Figure A7, where we use the previous separate calibrations for all the models 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 14.32% over LC-MNL and 2.68% over RPL, while LC PO-MNL Promotion model shows an average improvement of 16.24% over LC-MNL and 4.74% over RPL.

Figure A8 presents scatterplots of the miss-rates, using a display format similar to that of Figure 4. From it, we observe that our model combinations obtain improvements of up to 8.92% under PO-MNL Promotion, and further improvements of up to 11.3% under LC PO-MNL Promotion over the benchmarks within the six panels.
Brand choice “chi-square” prediction results. 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. the PO-MNL Promotion [single class] and LC PO-MNL Promotion [multiclass]. Lower is better; the benchmark is outperformed for points above the $45^\circ$ line. In all the panels we estimate the benchmarks on the set of all individuals. Left panels: we predict all the models only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: we predict all the models only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: we combine both types of the predictions from left and middle columns such that we cover all the individuals.

The key observations can be summarized as follows: (i) The PO MNL Promotion model, especially the multiclass version of it, outperforms the state-of-the-art competitive benchmarks even when the latter ones are allowed to be estimated on a larger population of customers; (ii) the best results for our model are observed on the individuals without cycles in the first place, who are the ones displaying the most consistent behavior; and (iii) LC-MNL is more competitive than RPL with respect to $\chi^2$ scores, but is dominated by RPL in terms of miss rates. The last observation is different from what we report in Section 4.4, where RPL dominated LC-MNL with respect to both $\chi^2$ and miss rates. The larger volume of data to train the models seems to favor more LC-MNL than RPL.
Figure A8  Brand choice miss rate prediction results. 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. the PO-MNL Promotion [single class] and LC PO-MNL Promotion [multiclass]. Lower is better; the benchmark is outperformed for points above the $45^\circ$ line. In all the panels we estimate the benchmarks on the set of all individuals. Left panels: we predict all the models only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: we predict all the models only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: we combine both types of the predictions from left and middle columns such that we cover all the individuals.

A6.2. Robustness with respect to adding no-purchase observations

In order to streamline the comparison of the models in the empirical case study in Section 4.4 we did not include the no purchase observations in the prediction tasks since we did not have explicit data on the no-purchase alternatives in our dataset. Here, we demonstrate that the brand choice prediction results remain qualitatively the same when we include the no purchase option in our calibration and prediction tasks.

To this end, we approximately build the no-purchase observations from our data. In particular, we can approximately infer from the data the times the customer visits to the store to make at least one category purchase. Therefore, we can easily obtain instances when the customer visited the store but ended up not making a specific category purchase. However, these observations in the
data cannot be considered as the no-purchase instances since we do not know if the customer had the intent to make a category purchase and ended up choosing the no-purchase option. In fact, the number of store visits is around ten times higher than the number of purchases for some categories. As a result, in order to minimize the number of ”spurious” no-purchase observations inferred from the data, we first assume that the number of times a customer chooses the outside option is comparable to the number of times a customer makes a category purchase. In particular, we say that the number of no-purchases of every customer is equal to the number of times a customer buys her second most purchased product. It implies that the customers chose a no-purchase alternative on average $\alpha T_c$ times, where $T_c$ is the total number of times a customer made a category purchase and $\alpha = 20.5\%$. We also show below that the obtained prediction results are robust to other values of $\alpha$. As a result, we randomly sample the fixed portion of the no-purchase observations from the data on the store visits of the customers when they decided not to make a category purchase, and include these additional transactions into our dataset for every category. Then we use the same approach described in the main body of the paper to calibrate the models and test their predictive performance.

Analogously to Figure 3, Figure A9 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 (multi-class), across the 27 product categories, for three subsets of customers. First, consider the left two panels. Here, we calibrate the models on the subset of individuals that do not have cycles in their preference graph. The “chi-square” score of PO-MNL Promotion model exhibits a moderate average deterioration of 2.79% over LC-MNL and an average improvement of 7.43% over RPL. Second, consider the middle column in Figure A9, where we calibrated the models on the subset of individuals that have cycles in their preference graph. We see that PO-MNL Promotion model exhibits an average improvement of 12.88% over LC-MNL and 4.17% over RPL.

In the left two panels in Figure A9, it is demonstrated that using LC PO-MNL Promotion model, we can further boost performance of the proposed methodology, resulting in average improvement of 14.64% over LC-MNL and 22.8% over RPL. Similarly, as illustrated in the middle column of Figure A9, LC PO-MNL Promotion model, capturing heterogeneity of customers to a greater extent, has an average improvement of 16.07% over LC-MNL and 7.43% over RPL.

Third, consider the right panels in Figure A9, 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 12.59% over LC-MNL and 3.48% over RPL, while LC PO-MNL Promotion model shows an average improvement of 16.18% over LC-MNL and 7.27% over RPL.
Figure A9  Brand choice prediction results with no purchase option included. We assume that the number of no-purchases of every customer is equal to the number of times a customer buys her second most purchased product. 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. the PO-MNL Promotion [single class] and LC PO-MNL Promotion [multiclass]. Lower is better; the benchmark is outperformed for points above the 45° line. Left panels: we estimate and predict only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: we estimate and predict only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: we estimate each type of individuals separately and combine both types for the prediction.

Analogously to Figure 4, Figure A10 presents scatterplots of the miss-rates, using a display format similar to that of Figure A9. From it, we observe that we obtain improvements of between 1.28% and 10.04% under LC PO-MNL Promotion over the benchmarks in all six panels.

Like in the base case in Section 4.4 without the no-purchase option, we observe that the PO MNL Promotion model continues to outperform both benchmarks in most of the categories, with respect to $\chi^2$ and miss rate, especially for its multiclass version.

Next, we showcase the robustness of the predictive results for other values of $\alpha$. In particular, we test the predictive performance of the LC PO-MNL Promotion model versus the LC-MNL benchmark for $\alpha = 30\%$ and $\alpha = 40\%$ in Figures A11 and A12, respectively. Here we focus only on the LC-MNL benchmark provided its competitive performance and the heavy computational burden of RPL. We observe that the improvements of PO-MNL Promotion over the LC-MNL
Figure A10  Brand choice prediction results with no purchase option included. We assume that the number of no-purchases of every customer is equal to the number of times a customer buys her second most purchased product. 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. the PO-MNL Promotion [single class] and LC PO-MNL Promotion [multiclass]. Lower is better; the benchmark is outperformed for points above the 45° line. Left panels: we estimate and predict only over individuals who do not have cycles in the preference graph under the PO-MNL Promotion model. Middle panels: we estimate and predict only over individuals who have cycles in the preference graph under the PO-MNL Promotion model. Right panels: we estimate each type of individuals separately and combine both types for the prediction.

benchmark vary with $\alpha$ between 12.36% and 14.95% in terms of the $\chi^2$ score, and between 1.24% and 1.76% in terms of the miss rate.

Figures A11 and A12 confirm that the superior performance of PO MNL Promotion is robust to different fractions of no-purchases in the dataset.

A6.3. Robustness with respect to the split between test and training datasets

In all our experiments so far, the training set consists of the first 26 weeks and the test set consists of the last 26 weeks of transactions. Here we show the robustness of the results to changes in how we split the data into the training and test sets. We also include the no-purchase observations (with $\alpha = 20.5\%$).
Figure A11  Brand choice prediction results: with no purchase option included. Every customer is assumed to choose a no-purchase alternative on average $\alpha T_c$ times, where $T_c$ is the total number of times a customer made a category purchase and $\alpha = 30\%$. Scatter plot of the prediction scores of all 27 product categories under the best of up to 10 LC-MNL vs. LC PO-MNL Promotion. Lower is better; the benchmark is outperformed for points above the $45^\circ$ line. Left panel: miss rate. Right panel: $\chi^2$ score.

Figure A12  Brand choice prediction results: with no purchase option included. Every customer is assumed to choose a no-purchase alternative on average $\alpha T_c$ times, where $T_c$ is the total number of times a customer made a category purchase and $\alpha = 40\%$. Scatter plot of the prediction scores of all 27 product categories under the best of up to 10 LC-MNL vs. LC PO-MNL Promotion. Lower is better; the benchmark is outperformed for points above the $45^\circ$ line. Left panel: miss rate. Right panel: $\chi^2$ score.

When we reduce the volume of the training set to the first 21 weeks and enlarge the test set to the last 31 weeks of transactions, Figure A13 shows that the LC PO-MNL Promotion model
Figure A13  Brand choice prediction results: with no purchase option included. The training data consists of 21 weeks, and the test data consists of 31 weeks. Scatter plot of the prediction scores of all 27 product categories under the best of up to 10 LC-MNL [LC-MNL] vs. the LC PO-MNL Promotion [PO-MNL Promotion]. Lower is better; the benchmark is outperformed for points above the 45° line. Left panel: miss rate score. Right panel: $\chi^2$ score.

outperforms the LC-MNL benchmark by 16.14% and 1.02% based on the miss rate and $\chi^2$ scores, respectively.

When increasing the training set to the first 31 weeks and reducing the test set to the last 21 weeks of transactions, Figure A14 shows that the LC PO-MNL Promotion model outperforms the LC-MNL benchmark by 14.66% and 1.12% based on the miss rate and $\chi^2$ scores, respectively.

We observe a sustained relative performance of the PO MNL Promotion model over the LC-MNL model as we reduce or increase the training dataset.

A6.4. Robustness with respect to the addition of implicit candidate edges in the DAGs

Phase 2 in the DAG construction process is about the inclusion of implicit candidate edges in the DAG that identifies each individual. This is a heuristic step that assumes that the relative preference between two products is preserved regardless the promotion status of the products. That is, if from Phase 1 a full price version of $a_j$ is preferred over the full price version of $a_k$, then the promoted version $a_j+n$ is also preferred to the promoted version $a_k+n$. Similarly, if the Phase 1 preference is stated on the promoted versions, then the relative preference is extended to the corresponding full price versions. These Phase 2 edges have a low weight and some of them are the ones to be deleted in Phase 3 in case a cycle arises in the DAG. Of course, the heuristic could add spurious implicit candidate edges, and the final justification for the existence of the edges is their empirical performance.
Figure A14  Brand choice prediction results: with no purchase option included. The training data consists of 31 weeks, and the test data consists of 21 weeks. Scatter plot of the prediction scores of all 27 product categories under the best of up to 10 LC-MNL [LC-MNL] vs. the LC PO-MNL Promotion [PO-MNL Promotion]. Lower is better; the benchmark is outperformed for points above the 45° line. Left panel: miss rate score. Right panel: $\chi^2$ score.

To this end, Figure A15 illustrates that the LC PO-MNL Promotion model with implicit candidate edges outperforms the LC PO-MNL Promotion without implicit edges by 1.82% and 1.9% based on miss rate and $\chi^2$ scores, respectively. This observation provides enough support for their inclusion in the DAG. It allows us to conclude that adding implicit edges in the DAG construction process boosts the benefit in prediction performance because of the denser DAG which outweighs the biases from adding few spurious edges along the way.

A7. Comparison with the DAG-based behavioral models studied by Jagabathula and Vulcano (2018)

In this section we compare the predictive performance of the PO-MNL Promotion model with the PO-MNL Inertial and PO-MNL Censored models studied by Jagabathula and Vulcano (2018). Note that both PO-MNL Inertial and PO-MNL Censored models take into account the information about product promotions implicitly through modeling the consideration sets of customers via behavioral rules. Relying on the pre-specified behavioral assumptions this approach cannot consistently explain the purchasing behavior of all the customers. As a result, the DAGs of customers whose purchasing transactions are inconsistent with these assumptions are assumed to be empty (i.e., without edges), which reduce the representation of customer preferences to any standard random utility model such as the MNL or the LC-MNL. In particular, to run the prediction performance of PO-MNL Inertial and PO-MNL Censored for customers that have empty DAGs, we use the best of up to 10 LC-MNL [LC-MNL].
Figure A15  Brand choice prediction results with the no-purchase option included. Scatter plot of the prediction scores across all 27 product categories under LC PO-MNL Promotion without implicit edges [PO-MNL Promotion without implicit edges] vs. LC PO-MNL Promotion with implicit edges [PO-MNL Promotion with implicit edges]. Left panel: miss rate scores. Right panel: $\chi^2$ scores.

Recall that the approach taken in our paper for the DAG construction is different, since it is completely data-driven and accounts explicitly for promotion effects. The approach could still lead to cycles in the preference graph, which are then deleted in Phase 3 such that all customers are characterized by non-empty DAGs. Figure A16 illustrates the scatter plot of the $\chi^2$ scores of all 29 product categories under the LC PO-MNL Promotion vs. PO-MNL Inertial with clustering and PO-MNL Censored with clustering (see (Jagabathula and Vulcano 2018, Section 5)). In all the panels we calibrate the LC-MNL and LC PO-MNL Promotion models on the set of all individuals. First, consider the left two panels in Figure A16. Here, we calibrate the PO-MNL Inertial [top left panel] and PO-MNL Censored [bottom left panel] and represent the prediction performance of all the models over the subset of individuals that can be explained by behavioral assumptions. The $\chi^2$ score of LC PO-MNL Promotion model exhibits an average deterioration of 15.52% over PO-MNL Inertial with clustering and 18.68% over PO-MNL Censored with clustering. Second, consider the right column in Figure A16, where we calibrate the PO-MNL Inertial [top right panel] and PO-MNL Censored [bottom right panel] and represent the prediction performance of all the models over the subset of individuals that can not be explained by behavioral assumptions. In this case, both PO-MNL Inertial and PO-MNL Censored models are reduced to the LC-MNL model. We see that LC PO-MNL Promotion model exhibits an average improvement of 11.18% over PO-MNL Inertial and 2.14% over PO-MNL Censored.

Figure A17 presents scatterplots of the miss rates, using a display format similar to that of Figure A16. The insights are the same as in Figure A16. In the left column we calibrate the PO-MNL Inertial [top left panel] and PO-MNL Censored [bottom left panel] models and represent the
Figure A16  The layout is the same as Figure 3 in the paper by Jagabathula and Vulcano (2018). Scatter plot of the $\chi^2$ scores of all 29 product categories under the PO-MNL Inertial and PO-MNL Censored vs. the PO-MNL Promotion. In all the panels we calibrate the LC-MNL and LC PO-MNL Promotion models on the set of all individuals. Left panels: we estimate PO-MNL Inertial and PO-MNL Censored and make predictions for the models over the individuals whose purchasing transactions can be explained by the behavioral assumptions of PO-MNL Inertial [top left panel] or PO-MNL Censored [bottom left panel]. Right panels: Here, both PO-MNL Inertial and PO-MNL Censored models are reduced to LC-MNL model. We estimate PO-MNL Inertial and PO-MNL Censored and make predictions for the models over the individuals whose purchasing transactions can not be explained by the behavioral assumptions of PO-MNL Inertial [top right panel] and PO-MNL Censored [bottom right panel].

prediction performance of all the models over the subset of individuals who can be explained by behavioral assumptions. We observe that LC PO-MNL Promotion obtains an average deterioration of 1.36% over PO-MNL Inertial with clustering and of 3.66% over PO-MNL Censored with clustering. Then, in the right column in Figure A17, we calibrate the PO-MNL Inertial [top right panel] and PO-MNL Censored [bottom right panel] and represent the prediction performance of all the models over the subset of individuals that can not be explained by behavioral assumptions. In this case, both PO-MNL Inertial and PO-MNL Censored models are reduced to LC-MNL model. We notice that LC PO-MNL Promotion model exhibits an average improvement of 2.13% over PO-MNL Inertial and 2.53% over PO-MNL Censored.

Even though the results in Figures A16 and A17 show an average dominance of the behavioral models over the PO MNL Promotion optimization model with respecto to both $\chi^2$ and miss rates, the presence of points above the diagonal indicates that for some categories PO MNL Promotion
Figure A17  The layout is the same as Figure 5 in the paper by Jagabathula and Vulcano (2018). Scatter plot of the miss rate scores of all 29 product categories under the PO-MNL Inertial and PO-MNL Censored vs. the PO-MNL Promotion. In all the panels we calibrate the LC-MNL and LC PO-MNL Promotion models on the set of all individuals. Left panels: we estimate PO-MNL Inertial and PO-MNL Censored and make predictions for the models over the individuals whose purchasing transactions can be explained by the behavioral assumptions of PO-MNL Inertial [top left panel] or PO-MNL Censored [bottom left panel]. Right panels: Here, both PO-MNL Inertial and PO-MNL Censored models are reduced to LC-MNL model. We estimate PO-MNL Inertial and PO-MNL Censored and make predictions for the models over the individuals whose purchasing transactions can not be explained by the behavioral assumptions of PO-MNL Inertial [top right panel] and PO-MNL Censored [bottom right panel].

still dominates. In order to characterize those categories, in Figure A18 we report the loyalty score of each category computed on the training data (left panel). Then, in the middle and right panels we explore possible correlations between the percentage of $\chi^2$ improvement of the behavioral models with respect to PO MNL Promotion (vertical axis) vs. loyalty score (horizontal axis). We note a negative correlation for PO MNL Promotion improvements with respect to both PO MNL Inertial and Censored models, meaning that the behavior of customers for most of the categories with low loyalty index (which exhibit the least stickiness in customers’ preferences) are better explained by the PO MNL Promotion model, as it is the case for customers represented by empty DAGs in the Jagabathula and Vulcano’s approach.

These findings suggest that the practitioners might use the PO-MNL Inertial and Censored models for categories with high loyalty index, and within them, for customers having non-empty
DAGs. Other than this, the use of the PO-MNL Promotion model proposed in this paper leads to more effective predictions.

### A8. Optimization of personalized promotions

We now show that the set of constraints (16)–(19) ensures that \( p \)'s are normalized attraction values. Recall that \( z_j = 1 \) for all products \( a_j \) in the set \( h_D(S(y)) \) of heads in the subgraph of the transitive closure of \( D \) restricted to the set \( S(y) \).

**Lemma A3.** Suppose \( 0 \leq p_j \leq 1 \): \( a_j \in S_A \) satisfy (16)–(19), then

\[
p_j = \begin{cases} 
0, & \text{if } z_j = 0, \\
\frac{v_{\Psi_D(a_j)}}{1 + \sum_{k: a_k \in S_A} v_{\Psi_D(a_k)}}, & \text{if } z_j = 1.
\end{cases}
\]

**Proof:** To simplify notation, let \( w_j \) denote \( v_{\Psi_D(a_j)} \) for each \( a_j \in S_A \). For convenience, we reproduce the set of constraints (16)–(19) below:

\[
p_j \leq z_j \quad \forall \; a_j \in S_A,
\]
Let $S$ denote the set $\{a_j: z_j = 1\}$, consisting of all the product indices such that $z_j = 1$. It immediately follows from (A7) that $p_j = 0$ for all $a_j \in S_A \setminus S$ (since $z_j = 0$ therein). Then, for all $a_j \in S_A \setminus S$, (A9) trivially holds and (A10) reduces to $p_0 \leq 1$ (which also trivially holds).

Now, for any $j$ such that $a_j \in S$ (and hence, $z_j = 1$), we have from (A9) and (A10) that

$$0 \leq p_j \leq w_j p_0 \quad \forall a_j \in S_A.$$ (A9)

$$p_0 + z_j - 1 \leq p_j / w_j \quad \forall a_j \in S_A.$$ (A10)

It thus follows that $p_j = p_0 w_j$ for all $a_j \in S$. We now obtain from (A8) that

$$p_0 + \sum_{a_j \in S_A} p_j = 1 \implies p_0 + \sum_{a_j \in S} p_j = 1 \implies p_0 + \sum_{a_j \in S} p_0 w_j = 1 \implies p_0 = 1 / \left(1 + \sum_{a_j \in S} w_j\right),$$

where the first implication follows because $p_j = 0$ for all $a_j \in S_A \setminus S$ and the second implication follows because $p_j = p_0 w_j$ for all $a_j \in S$. Since $w_j > 0$ for all $a_j$, it follows that $p_0 \leq 1$, as needed.

We have thus obtained that

$$p_j = 0 \text{ for all } a_j \in S_A \setminus S \text{ and } p_j = w_j p_0 = w_j / \left(1 + \sum_{a_k \in S} w_k\right) \text{ for all } a_j \in S.$$ (A10)

In other words, we have that $p_j = w_j z_j / \left(1 + \sum_{a_k \in S_A} w_k z_k\right)$, which follows from the definition of $S$. The result of the lemma now holds.

**A9. Robustness check for the promotion optimization**

In the revenue results in Section 5 our prediction assumed that when arriving at the store every customer would buy the product with highest probability of being purchased under PO-MNL Promotion model. An alternative objective would be to compute expected revenues accounting for the probabilities of purchase of every single product on offer. Figure A19 illustrates the results of running personalized promotions under the modified objective function of the optimization problem. Since the new promotion optimization results, illustrated in Figure A19, almost exactly resemble the ones in Figure 5, we conclude that all the insights remain qualitatively the same under the updated formulation of the promotion optimization problem.
Figure A19  Promotion optimization problem when we optimize the expected revenue for every purchasing transaction across 27 product categories. Left panel: Scatter plot of the expected revenue per customer transaction taking the set of promoted items from the sales data [Predicted revenue] when customers choose according to the PO-MNL Promotion model vs. the realized revenue per customer transaction obtained from the data [Realized revenue]. Middle panel: Scatter plot of the expected revenue per customer transaction after promotion optimization [Promotion optimization revenue] vs. the realized revenue per customer transaction obtained from the data [Realized revenue]. Right panel: Scatter plot and linear regression of the log of the percentage improvement of the retailer’s revenue after the promotion optimization vs. purchase frequency.

A10. Managerial insights: Factors affecting improvements from personalization of promotions

The revenue improvements from personalization vary across different customers in each category. To explain this variation, for each category, we consider three different customer-level characteristics:

1. **Brand loyalty**, measured as the percentage of store visits a customer buys her most frequently purchased brand from the category;

2. **Purchase frequency**, measured as the number of purchases a customer makes from the category over the training data (26 weeks); and

3. **Promotion sensitivity**, measured as the percentage of store visits a customer buys a promoted product from the category.

We regressed the revenue improvement for each customer and category combination against the brand loyalty, purchase frequency and promotion sensitivity variables. Table A1 reports the results from fitting four different models:

\[
\text{RevImpr}_{i,c} = \beta_0 \cdot \text{Cat}_c + \beta_1 \cdot \text{Bloyalty}_{i,c} + \epsilon_{i,c} \\
\text{RevImpr}_{i,c} = \beta_0 \cdot \text{Cat}_c + \beta_2 \cdot \text{PurFreq}_{i,c} + \epsilon_{i,c} \\
\text{RevImpr}_{i,c} = \beta_0 \cdot \text{Cat}_c + \beta_3 \cdot \text{PromSens}_{i,c} + \epsilon_{i,c} \\
\text{RevImpr}_{i,c} = \beta_0 \cdot \text{Cat}_c + \beta_1 \cdot \text{Bloyalty}_{i,c} + \beta_2 \cdot \text{PurFreq} + \beta_3 \cdot \text{PromSens}_{i,c} + \epsilon_{i,c} \quad \text{(Model 4),}
\]
where the variables $Bloyalty_{i,c}$, $PurFreq_{i,c}$, and $PromSens_{i,c}$ respectively denote the brand loyalty, purchase frequency, and promotion sensitivity computed for customer $i$ under category $c$. The brand loyalty and purchase frequency variables were computed using the training data. To ensure exogeneity, the promotion sensitivity variable was computed using the data from the previous year (2006). The variable $Cat_c$ is an indicator variable denoting category $c$ to capture category fixed effects. Finally, $RevImpr_{i,c}$ is the average revenue improvement from personalization for customer $i$ under category $c$, computed over the holdout sample.

The results from the regressions are consistent and intuitive. The benefits from personalization are negatively correlated with brand loyalty (Model 1) and purchase frequency (Model 2) but positively correlated with promotion sensitivity (Model 3). In other words, customers who purchase infrequently and concentrate their purchases only on a few brands are harder to persuade to switch to more profitable brands through personalized promotions. On the other hand, customers who frequently purchase promoted items are easier to be influenced by personalizing promotions. These findings are consistent in a multiple regression of the revenue improvement against all three variables (Model 4). The coefficients in the multiple regression are all statistically significant, indicating that all three factors together influence the brand switching behavior of customers in response to promotions.

<table>
<thead>
<tr>
<th>Dependent variable: Revenue Improvement (%)</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand loyalty</td>
<td>-14.415***</td>
<td>-20.514***</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purch. frequency</td>
<td></td>
<td>-1.108***</td>
<td>-1.218***</td>
<td></td>
</tr>
<tr>
<td>Prom. sensitivity</td>
<td></td>
<td></td>
<td>14.575***</td>
<td>13.891***</td>
</tr>
<tr>
<td>Category FE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No. Observations:</td>
<td>57,059</td>
<td>57,059</td>
<td>57,059</td>
<td>57,059</td>
</tr>
<tr>
<td>R-squared:</td>
<td>0.001</td>
<td>0.003</td>
<td>0.001</td>
<td>0.006</td>
</tr>
</tbody>
</table>

$t$ statistics in parentheses

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

**Table A1** Individual-level regressions with product category fixed effects.

**A11. Additional insights on brand loyalty and the number of promoted items**

In this section, we provide additional descriptive statistics to gain insights into the extent of brand loyalty of the customers and the number of promoted items in our dataset. Table A2 presents these statistics.
The second to sixth columns in the table report the statistics describing the distribution of the number of unique brands purchased by the customers in each category; specifically, the columns report the mean, the standard deviation, and the first, second, and the third quartiles of the distribution, respectively. We note that on average customers purchase no more than 4 unique brands in the training data, indicating that customers have strong preferences. This observation complements the loyalty scores reported in Figure A18 and explains the significant performance gains that our model obtains over the benchmark models as reported in Figure 3.

Columns seven and eight report the average number of products our method offers on promotion across all the transactions in the holdout sample. Column seven reports this number when existing mass promotions are ignored and column eight reports the number of promoted products on top of the products that are already on mass promotion. We note that most of these numbers are less than 1, indicating that at optimality, our method offers only a small number of products on promotion. This observation explains why our method is able to extract most of the revenues even with the constraint of offering at most one product on promotion; see Section 5.3.

In order to provide a partial explanation for the small number of products that are put on promotion, the last two columns of the table report the average offer set size and the average number of products that can be potentially promoted (NonDom) across all the transactions. To calculate the number of products that can be potentially promoted, consider a transaction in which the offer set is $S$ and the customer has DAG $D$. We note that product $a_j$ will not be promoted if there is another product $a_i \in S$ such that the non-promoted copy of $a_i$ (and consequently, the promoted copy of $a_i$) is preferred over the promoted copy of $a_j$ in DAG $D$. The reason is that product $a_j$ will not be purchased whether promoted or not because either the promoted or the non-promoted copy of $a_i$ will be offered to the customer. We call such a product $a_j$ a dominated product and any product that is not dominated, a non-dominated product. Given this, NonDom reports the average number of non-dominated products across all the transactions in the holdout sample.

We observe from the table that customer DAGs are such that the average number of non-dominated products is far smaller than the average offer set size. Because the number of non-dominated products is an upper bound on the number of products that will promoted, this table provides a partial explanation as to why at optimality, only a small number of products are promoted.
<table>
<thead>
<tr>
<th>Category</th>
<th># Unique brands purchased</th>
<th># prom. items</th>
<th>AvOS</th>
<th>NonDom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>stdev</td>
<td>1st quart.</td>
<td>2nd quart.</td>
</tr>
<tr>
<td>beer</td>
<td>2.09</td>
<td>1.29</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>carbbev</td>
<td>2.91</td>
<td>1.31</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>cigets</td>
<td>1.29</td>
<td>0.52</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>coffee</td>
<td>2.17</td>
<td>1.19</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>colder</td>
<td>3.03</td>
<td>1.37</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>deod</td>
<td>2.03</td>
<td>1.04</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>factiss</td>
<td>1.90</td>
<td>0.87</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>fzdinent</td>
<td>3.36</td>
<td>2.10</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>fzpizza</td>
<td>2.52</td>
<td>1.33</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>hhclean</td>
<td>2.90</td>
<td>1.40</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>hotdog</td>
<td>2.09</td>
<td>1.04</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>laundet</td>
<td>1.96</td>
<td>1.07</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>margbutr</td>
<td>1.93</td>
<td>1.02</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>mayo</td>
<td>1.36</td>
<td>0.55</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>milk</td>
<td>2.36</td>
<td>1.17</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>mustketc</td>
<td>2.29</td>
<td>0.92</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>paptowl</td>
<td>2.15</td>
<td>1.08</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>peanbutr</td>
<td>1.65</td>
<td>0.75</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>saltsnck</td>
<td>3.67</td>
<td>1.89</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>champ</td>
<td>2.15</td>
<td>1.14</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>soup</td>
<td>2.94</td>
<td>1.46</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>spagsauc</td>
<td>2.19</td>
<td>1.25</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>sugarsub</td>
<td>1.23</td>
<td>0.47</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>toitisu</td>
<td>2.01</td>
<td>1.06</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>toothbr</td>
<td>2.06</td>
<td>1.00</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>toothpa</td>
<td>1.65</td>
<td>0.74</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>yogurt</td>
<td>2.41</td>
<td>1.30</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table A2 Relevant summary statistics from the data. The column '# Unique brands purchased' reports the distribution of the number of unique brands bought by a customer over the training period; specifically, the columns mean, stdev, 1st quart., 2nd quart., and 3rd quart. report the mean, the standard deviation, the 1st, 2nd, and 3rd quartiles, respectively. The column 'w./o. mass' reports the average number of products offered on promotion by our method, across all the transactions in the holdout sample when existing mass promotions are ignored. Similarly, the column 'w. mass' reports the average number of products offered on promotion by our method across all the transactions in the holdout sample on top of the products that are already on mass promotion. The columns ‘AvOS’ and ‘NonDom’ report the average number of vendors and the average number of non-dominated vendors in the offer set, respectively, across all the transactions in the holdout sample; see text for the definition of non-dominated vendors.