The Limit of Rationality in Choice Modeling: Formulation, Computation, and Implications

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

Paat Rusmevichientong
Marshall School of Business, University of Southern California, Los Angeles, CA 90089, rusmevic@marshall.usc.edu

Customer preferences may not be rational, so we focus on quantifying the *limit of rationality* (LoR) in choice modeling applications. We define LoR as the cost of approximating the observed choice fractions from a collection of offer sets with those from the best-fitting probability distribution over rankings. Computing LoR is intractable in the worst case. To deal with this challenge, we introduce two new concepts, *rational separation* and *choice graph*, through which we reduce the problem to solving a dynamic program on the choice graph and express the computational complexity in terms of the structural properties of the graph. By exploiting the graph structure, we provide practical methods to compute LoR efficiently for a large class of applications. We apply our methods to real-world grocery sales data and identify product categories for which going beyond rational choice models is necessary to obtain an acceptable performance.

Key words: limit of rationality, choice modeling, rank aggregation

1. Introduction

Firms routinely collect inventory and sales data that provide information on product availability and customer choices, respectively. Together, these data are used by firms to obtain the necessary demand estimates for inventory, assortment, and pricing decisions. Traditionally, fitting independent demand models, which assume that products receive independent demand streams, which are unaffected by the availability of other products, was common. However, this assumption introduces biases in demand estimates when products are close substitutes and their availability changes over time; customers may substitute an unavailable product with an available one, making product demand a function of the entire offer set. For this reason, moving from independent to choice-based demand models has been a trend both in the academic literature and in industry practice.

Among choice-based demand models, random utility maximization (RUM) models have, by far, received the most attention. These models are based on the utility maximization principle and assume that in each choice instance, a customer draws a random sample of utilities for the offered products and chooses the product with the highest utility. RUM models generalize the classic notion of a rational individual maximizing a utility function to a population of rational individuals, each maximizing a (potentially) different utility function. A variation in the utilities may arise because of preference heterogeneity across individuals or the stochastic elements within individual preferences (McFadden 2005). When the number of alternatives is finite (Mas-Colell et al. 1995), only the ordering of the products induced by the sampled utilities, not the actual utility values themselves, matters in determining customer choices. As a result, an RUM model is equivalently described by the distribution over preference lists induced by the model, and the choice probabilities under the RUM model are *stochastically rationalizable*: the probability of choosing a product from an offer set is equal to the sum of the probabilities associated with all the preference lists in which the product is the most preferred among all the products in the offer set.

However, stochastic rationalizability imposes regularity conditions on the choice probabilities that may not be satisfied in practice. For instance, stochastic rationalizability requires the probability of choosing an alternative to never increase when more products become available[1], but this condition may be violated because individuals may not be perfectly rational; see, for example, Tversky et al. (1990). The observed violations of rationality raise the following empirical questions: (a) Given a population of individuals and a collection of offer sets, are the observed choice probabilities (as computed from sales transactions) consistent with the RUM principle? (b) If not, what is the degree of inconsistency?

Motivated by these questions, this paper focuses on quantifying the *limit of stochastic rationality* (LoR), which is the degree to which the RUM principle is inconsistent with the given empirical choice probabilities observed from a collection of offer sets. More formally, suppose $N = \{1, 2, \ldots, n\}$ is the universe of n products, and we have collected choice observations for a collection $\mathcal{M}$ of subsets of N. For each subset $S \in \mathcal{M}$, let $f_{i,S} \in [0,1]$ denote the fraction of customers who purchased product i when S was offered. We then ask, how well does the RUM model explain the observed choice probabilities $\boldsymbol{f}_{\mathcal{M}} = (f_{i,S} \colon i \in S,\, S \in \mathcal{M})$?

We measure the degree of inconsistency by means of a non-negative, strictly convex loss function $\boldsymbol{x}_{\mathcal{M}} \mapsto \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}})$ that measures the distance between the observed choice probabilities $\boldsymbol{f}_{\mathcal{M}}$ and the choice probabilities $\boldsymbol{x}_{\mathcal{M}}$ that are consistent with an RUM model. We suppose that the loss function has the property in which $\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}}) = 0$ if and only if $\boldsymbol{f}_{\mathcal{M}} = \boldsymbol{x}_{\mathcal{M}}$; examples include log-loss and general norm-loss functions (Examples 2.1 and 2.2). We define the limit of (stochastic) rationality or simply the *limit of rationality* $\mathsf{LoR}(\mathcal{M})$ as the minimum distance $\min_{\boldsymbol{x}_{\mathcal{M}}} \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}})$ over all choice probabilities $\boldsymbol{x}_{\mathcal{M}}$ that are consistent with an RUM model. Note that the observed choice data are stochastically rationalizable if and only if $\mathsf{LoR}(\mathcal{M}) = 0$.

[1] Block and Marschak (1960) and Falmagne (1978) gave the necessary and sufficient conditions for a system of choice probabilities to be stochastically rationalizable.

This paper focuses on two key aspects related to the above optimization problem: (a) the computability of $\mathsf{LoR}(\mathcal{M})$ for a given collection of observed choice probabilities $\boldsymbol{f}_{\mathcal{M}}$ and (b) the application of $\mathsf{LoR}(\mathcal{M})$ for *model selection* to determine if a modeler should fit a more complex RUM model or must go beyond the RUM class to obtain an acceptable performance.

Computing the LoR requires us to find the choice probabilities $\boldsymbol{x}^*_{\mathcal{M}}$ that are consistent with an RUM model and are closest to the observed probabilities $\boldsymbol{f}_{\mathcal{M}}$. We formulate this problem as a constrained convex program whose feasible region is a polytope with $n!$ (n factorial) extreme points. By applying the conditional-gradient algorithm, we reduce the problem of solving the constrained convex program to solving a sequence of rank aggregation problems. Each rank aggregation problem requires finding the ranked list σ that minimizes a linear cost function: $\min_\sigma \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} \cdot \mathbb{1}[\sigma, i, S]$, where the cost coefficients $c_{i,S}$ are given, and $\mathbb{1}[\sigma, i, S]$ takes a value of 1, if i is the most preferred product in S according to the ranked list σ and 0 otherwise. The problem is known to be NP-hard (Dwork et al. 2001). As the key theoretical contribution of this paper, we characterize the computational complexity of the rank aggregation problem in terms of the structure of the corresponding *choice graph*, a novel concept that we introduce in Definition 3.3. A choice graph is defined for each collection $\mathcal{M}$ of subsets; every subset in $\mathcal{M}$ is a vertex in the choice graph, and the edges capture the relationships of the most preferred products among the subsets in $\mathcal{M}$. We show that for commonly occurring set collections in the retail and revenue management (RM) settings, the corresponding choice graphs have rich structures, leading to efficient solutions for rank aggregation problems.

From a practical standpoint, our methods provide researchers and practitioners with a diagnostic tool to quantify the impact of their rationality and parametric assumptions. In practice, one fits a parametric choice model (such as the multinomial logit model) from the RUM class to the observed choice data to predict the demand for various offer sets. This procedure is satisfactory only if the researcher can assess the performance *loss* from his/her modeling assumptions and deems the loss acceptable. Our methods allow the practitioner to decompose the loss as follows:

$$\underbrace{\text{Total loss}}_{\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}},\boldsymbol{y}^*_{\mathcal{M}})} = \underbrace{\text{Rationality loss}}_{\mathsf{LoR}(\mathcal{M})} + \underbrace{\text{Parametric loss}}_{\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}},\boldsymbol{y}^*_{\mathcal{M}}) - \mathsf{LoR}(\mathcal{M})},$$

where $\boldsymbol{y}^*_{\mathcal{M}}$ denotes the choice probability vector under the fitted parametric model used by the researcher. This decomposition provides the following immediate insights: If the rationality loss is acceptable but the total loss is not, then the researcher will benefit from fitting a more complex parametric model from the RUM class. If, on the other hand, the rationality loss is itself unacceptable, then the researcher must relax the rationality assumption and go beyond the RUM class.

1.1 Main Contributions

Our work makes the following three main contributions:

1. *Computing the LoR.* Our theoretical results express the computational complexity of the rank aggregation problem in terms of the structure of what we call the *choice graph*, which captures the *rational separation* structure among the subsets in $\mathcal{M}$ (see Section 3 for precise definitions). By formulating the rank aggregation problem as both a dynamic program (DP) and a linear program (LP) on the choice graph, we show the following (see Sections 4 and 5):

Choice Graph	DP complexity	LP (# of vars. and constrs.)
Tree	$O(n\,\lvert\mathcal{M}\rvert)$ [Thm. 4.1]	$O(n\,\lvert\mathcal{M}\rvert)$ [Thm. 4.2]
Bdd. tree width	exp. in tree width [Thm. 5.4]	exp. in tree width [Thm. 5.5]
Unbd. tree width but bdd. choice depth	exp. in choice depth [Thm. 5.8]	exp. in choice depth [Thm. 5.9]

The first column specifies the structure of the choice graph and whether tree width and choice depth are bounded (bdd.) or unbounded (unbd.), the second column reports the complexity of solving the DP, and the last column specifies the number of variables (vars.) and constraints (constrs.) in the LP. Where specified, the scaling of the DP and LP is exponential (exp.) in the tree width or choice depth, but polynomial in n and $|\mathcal{M}|$.

2. *Efficient computation of LoR for important set collections.* Basing on our theoretical results, we derive the structural properties of the choice graphs for the following four set collections that commonly arise in the retail and RM contexts, and show that the LoR can be computed efficiently for these collections (see Section 3):

Collection	Definition	Choice Graph	Compl.
Nested	$\{S_1, S_2, \ldots, S_L\}$ s.t. $S_\ell \subset S_{\ell+1}$ for all ℓ	line	linear
Laminar	Either $A \subseteq B$, or $B \subseteq A$, or $A \cap B = \emptyset \;\forall\; A, B \in \mathcal{M}$	tree	linear
Differentiated	$\{A_0\} \cup \{A_0 \cup B_\ell \colon 1 \le \ell \le L\}, B_\ell \cap B_{\ell'} = \emptyset \;\forall \ell \ne \ell'$	star	linear
k-deletion	$\{N \setminus A \colon\, \lvert A\rvert \le k\}$	$\mathsf{cd} \le \mathsf{k}$	$O(n^k)$

The last column, "Compl.," is the computational complexity of solving the rank aggregation problem, and cd denotes the choice depth of the graph. The nested collection arises in RM contexts when airlines adopt the nested fare policy. The laminar collection arises when customers use the elimination-by-aspects (EBA) screening process to form a consideration set before making the choice. The differentiated collection arises when a firm carries a common collection of products A_0 in all stores, along with a unique subset B_ℓ to differentiate each store ℓ from others. Finally, the k-deletion collection arises when a firm faces frequent replenishment and stock-outs so that, at any point, at most $k < n$ products are stocked out.

3. *Empirical findings and practical insights.* We applied our methods to the grocery sales transactions data on 29 product categories from the popular IRI Academic Dataset (Bronnenberg et al. 2008). For each category, we computed the corresponding total, rationality, and parametric losses under the multinomial logit (MNL) model. We obtained the following insights: (a) most of the total loss is from the rationality loss (supporting existing work for going beyond the RUM class; see Figure 2); (b) rationality loss provides a model selection tool for determining the number of latent classes or when to go outside the RUM class (Figure 3); (c) fitting a complex model within the RUM class, such as the latent-class MNL model, reduces the parametric loss but not the rationality loss, whereas fitting a model outside the RUM class, such as the latent-class generalized attraction model (Gallego et al. 2014) can reduce the rationality loss; and (d) rationality loss is correlated with the market concentration of a category, suggesting that when market shares are split about evenly across the offered brands, customers tend to be variety seeking, switching their purchases from week to the next, thereby resulting in nontransitive preferences and high rationality loss.

1.2 Literature Review

Our work is related to the following three streams of literature: (a) rationalization in econometrics, (b) fitting nonparametric choice models in operations, and (c) rank aggregation in machine learning.

The question of whether a collection of observed choice probabilities is rationalizable is a classic question that has received significant attention in economics. The majority of the existing work focuses on deriving the necessary and sufficient conditions to answer the binary yes/no question of whether the given data are stochastically rationalizable. Falmagne (1978) shows that a system of choice probabilities defined over all possible subsets is stochastically rationalizable if and only if all the Block–Marschak polynomials are non-negative; see also Barberá and Pattanaik (1986). McFadden and Richter (1990) show that under certain regularity conditions, the axiom of the revealed stochastic preference provides a necessary and sufficient condition for a collection of observed choice probabilities to be stochastically rationalizable. McFadden (2005) provides the additional necessary and sufficient conditions in the form of systems of linear inequalities, and shows how the different conditions relate to one another. However, existing studies do not address the computability question of how to efficiently check the conditions when the number of products is large. To the best of our knowledge, this is the first paper to exhibit interesting collections of subsets for which stochastic rationality can be verified efficiently.

Our work is related to the literature on estimating nonparametric choice models pioneered by Jagabathula (2011) and Farias et al. (2013). This body of work has primarily focused on estimating a probability mass function $\lambda(\cdot)$ from transaction data for the purpose of making assortment or pricing decisions (Jagabathula and Rusmevichientong 2017). It reduces the estimation problem to a sequence of rank aggregation problems and develops heuristics for obtaining good approximations that can be used in operations management decisions. For example, van Ryzin and Vulcano (2015) recently developed an integer programming heuristic to solve the rank aggregation problem. By contrast, our work is the first to study the problem of identifying the *source* of computational complexity in solving the rank aggregation problem. By introducing the concepts of *rational separation* and *choice graph*, we characterize computational complexity in terms of what we call the *choice depth* of the corresponding choice graph. This characterization provides a new understanding of what drives the computational difficulty in solving rank aggregation problems. Using our characterization, we show that broad classes of offer set collections that commonly occur in practice have a bounded choice depth and, therefore, allow efficient solutions for the rank aggregation problem.

Our work is also related to the extensive literature within the machine learning community on solving the rank aggregation problem. The majority of such studies focus on finding a single ranking that minimizes some metric. The most common metric is the total weight associated with each unmatched pair, and the resulting problem has been extensively studied as the Kemeny optimization problem; see Kenyon-Mathieu and Schudy (2007), Ali and Meliă (2012). These studies focus on the types of observations collected on the web, which comprise choice observations from the set collection consisting of all pairs of products. Instead, we focus on choice observations from *general* set collections, which arise frequently in RM, supply chain, and operational applications, and identify new offer set collections for which the rank aggregation problem can be solved efficiently.

2. Problem Formulation

We now formally introduce the setup for our problem. As above, let $N = \{1, 2, \ldots, n\}$ be the universe of products, and we assume that N already includes the no-purchase or outside option. Let $\mathcal{M}$ denote a collection of subsets of N, corresponding to the assortments of products offered to customers. For each subset $S \in \mathcal{M}$, we observe the fraction $f_{i,S} \in [0, 1]$ of customers who chose product $i \in S$ when offered a subset S, with $\sum_{i \in S} f_{i,S} = 1$. We let $\boldsymbol{f}_{\mathcal{M}} = (f_{i,S} : i \in S, S \in \mathcal{M})$. Our objective is to check if the observed choice probabilities are consistent with the RUM assumptions, and if not, to determine the degree of inconsistency. To measure the degree of inconsistency, we find the rational choice probabilities that are closest to the observed data. More formally, let $\mathscr{P}_n$ denote the set of all permutations (or linear preference orders) of n products. Each element $\sigma \in \mathscr{P}_n$ is a ranking of n products, and for all $i \in N$, $\sigma(i)$ denotes the *rank* of product i. We assume that if $\sigma(i) < \sigma(j)$, then product i is preferred over product j.

For any distribution over rankings $\lambda : \mathscr{P}_n \to [0,1]$ such that $\sum_{\sigma \in \mathscr{P}_n} \lambda(\sigma) = 1$, let $\boldsymbol{x}^\lambda_{\mathcal{M}} = \left(x^\lambda_{i,S} : i \in S, \, S \in \mathcal{M}\right)$, where $x^\lambda_{i,S} = \sum_{\sigma \in \mathscr{P}_n} \mathbb{1}[\sigma, i, S] \cdot \lambda(\sigma)$ and $\mathbb{1}[\sigma, i, S]$ is the indicator variable that takes a value of 1 if and only if product i is the most preferred product in S under σ; that is, $\mathbb{1}[\sigma, i, S] = 1$ if and only if $\sigma(i) < \sigma(j)$ for all $j \in S$, $j \neq i$.

The best-fitting distribution $\lambda(\cdot)$ is the one whose corresponding choice probabilities $\boldsymbol{x}^\lambda_{\mathcal{M}}$ are closest to the observed data. We measure the distances through a non-negative, strictly convex loss function $\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}^\lambda_{\mathcal{M}})$, with the properties that for each $\boldsymbol{f}_{\mathcal{M}}$, the function $\boldsymbol{x}_{\mathcal{M}} \mapsto \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}})$ is non-negative, strictly convex in $\boldsymbol{x}_{\mathcal{M}}$, and takes a value of 0 if and only if $\boldsymbol{x}_{\mathcal{M}} = \boldsymbol{f}_{\mathcal{M}}$. Common examples of loss functions that satisfy these properties include the following:

Example 2.1 (Log-likelihood/Kullback-Leibler (KL) divergence loss function) Here, $\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}}) = -\sum_{S \in \mathcal{M}} M_S \sum_{i \in S} \; f_{i,S} \log\left(x_{i,S}/f_{i,S}\right)$, where $M_S > 0$ is the weight associated with set $S \in \mathcal{M}$, representing the number of customers who were shown the assortment S. The non-negativity of the loss function follows because $-\sum_{i \in S} f_{i,S} \log(x_{i,S}/f_{i,S})$ is the KL divergence between the distributions $(f_{i,S} : i \in S)$ and $(x_{i,S} : i \in S)$. Therefore, the loss function is a weighted sum of KL divergences, so $\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}}) = 0$ if and only if $x_{i,S} = f_{i,S}$ for all $i \in S$ and $S \in \mathcal{M}$. Strict convexity follows from the strict concavity property of the logarithm function. Under this loss function, the best-fitting distribution λ is indeed the maximum likelihood estimate (MLE) because

$$\max_\lambda \; \log\left(\prod_{S \in \mathcal{M}} \prod_{i \in S} \left(x^\lambda_{i,S}\right)^{M_S f_{i,S}}\right) \quad \Leftrightarrow \quad \max_\lambda \; \sum_{S \in \mathcal{M}} M_S \sum_{i \in S} f_{i,S} \log x^\lambda_{i,S} \quad \Leftrightarrow \quad \min_\lambda \; \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}^\lambda_{\mathcal{M}}).$$

Example 2.2 (Squared Norm loss function) The squared norm loss function is defined as $\mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}_{\mathcal{M}}) = \left\| \boldsymbol{f}_{\mathcal{M}} - \boldsymbol{x}_{\mathcal{M}} \right\|^2$, where the Euclidean norm $\|\cdot\|$ is defined on $\mathbb{R}^{\sum_{S \in \mathcal{M}} |S|}$. With the use of the properties of the norm function, it may be verified that the squared norm loss function is strictly convex in $\boldsymbol{x}_{\mathcal{M}}$ for any fixed $\boldsymbol{f}_{\mathcal{M}}$ and takes a value of 0 if and only if $\boldsymbol{x}_{\mathcal{M}} = \boldsymbol{f}_{\mathcal{M}}$.

We define the limit of rationality $\mathsf{LoR}(\mathcal{M})$ associated with the collection $\mathcal{M}$ as the minimum loss that we incur when we restrict our attention to the RUM class:

$$\mathsf{LoR}(\mathcal{M}) \;\equiv\; \min\left\{ \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}^\lambda_{\mathcal{M}}) \;\middle|\; \lambda : \mathscr{P}_n \to [0,1], \; \sum_{\sigma \in \mathscr{P}_n} \lambda(\sigma) = 1 \right\}. \tag{1}$$

Indeed, if the observed choice probabilities are stochastically rationalizable, it follows that (1) has an optimal objective value of zero, resulting in $\mathsf{LoR}(\mathcal{M}) = 0$, as desired.

Our solution approach: Equation (1) is a convex minimization problem over a convex feasible region because the set of rational choice probabilities can be described as a polyhedron, as follows:

$$\left\{ \boldsymbol{x}_{\mathcal{M}} \;\middle|\; x_{i,S} = \sum_{\sigma \in \mathscr{P}_n} \mathbb{1}[\sigma, i, S]\lambda(\sigma) \quad \forall\, S \in \mathcal{M}, i \in S,\ \sum_{\sigma \in \mathscr{P}_n} \lambda(\sigma) = 1,\ \lambda(\sigma) \geq 0 \quad \forall \sigma \in \mathscr{P}_n \right\} .$$

In theory, the optimization problem in Equation (1) can be solved using standard methods in convex optimization theory. The challenge, however, is the curse of dimensionality: the feasible region is described by $n!$ (n factorial) variables, with one variable $\lambda(\sigma)$ for each ranking $\sigma \in \mathscr{P}_n$; this is intractable even for small values of n. We deal with the high dimensionality by applying the Frank–Wolfe (FW) method (Frank and Wolfe 1956), also called the conditional gradient method, which has become increasingly popular in the machine learning community as the method of choice for solving large-scale convex optimization problems (Jaggi 2013). The FW method has many variants, and the details of our implementation are deferred to Section 6. At its core, the FW algorithm is an iterative method that starts with an initial feasible solution and generates a sequence of feasible solutions that are guaranteed to converge to the optimal solution. In each iteration, the algorithm solves the following linear program:

$$\min \left\{ \sum_{S \in \mathcal{M}} \sum_{i \in S} c_{i,S}\, x_{i,S} \quad \middle| \quad x_{i,S} = \sum_{\sigma \in \mathscr{P}_n} \mathbb{1}[\sigma, i, S]\lambda(\sigma) \quad \forall\, i \in S,\ S \in \mathcal{M}, \right.$$
$$\left. \sum_{\sigma \in \mathscr{P}_n} \lambda(\sigma) = 1, \quad \lambda(\sigma) \geq 0 \quad \forall\, \sigma \in \mathscr{P}_n \right\} \quad \text{(Rank Aggregation LP)}$$

where the coefficient $c_{i,S}$ is the gradient of the loss function evaluated at the current iterate, and represents the first-order Taylor series approximation of the objective function. The FW method computes the new iterate by taking a convex combination of the current iterate and the solution to the above linear program by using an appropriate convex combination weight. Thus, the FW method converts the original convex optimization problem into a series of linear programs. The rate at which the FW method converges depends on the particular variant of the algorithm used and the properties of the objective function and the constraint set. Establishing the convergence rates of the FW method is an active research area, and existing work has shown that the FW method converges relatively quickly in a wide range of settings; see, for example, Lacoste-Julien and Jaggi (2015) and the references therein.

Given the above discussion, we introduce the following definition that characterizes the complexity of computing the LoR:

Definition 2.3 (Rational Complexity) The *computational complexity* associated with $\mathsf{LoR}(\mathcal{M})$ is the complexity of solving the Rank Aggregation LP. We say that $\mathsf{LoR}(\mathcal{M})$ can be computed efficiently if the Rank Aggregation LP can be solved in time that is polynomial in n and $|\mathcal{M}|$.

Nesterov (2013) and Bubeck (2015) show that if the Rank Aggregation LP can be solved efficiently, then the original convex optimization problem can also be solved efficiently. However, we note that the above Rank Aggregation LP still has $n!$ (n factorial) variables. Lemma 2.5 shows that the problem is equivalent to the combinatorial optimization problem of finding the ranking that minimizes the total cost across all subsets in $\mathcal{M}$, which, in turn, is equivalent to finding the most preferred product in each set that is consistent across the entire collection *and* minimizes the total cost. It also shows that the problem is indeed NP-hard. Before we can state the lemma, we introduce the following notation, which will be used frequently throughout the paper:

Definition 2.4 (Consistent Rankings) For any set S and $z_S \in S$, let $\mathscr{D}_S(z_S)$ be the set of rankings that make z_S the most preferred product in S; i.e., $\mathscr{D}_S(z_S) = \{\sigma : \sigma(z_S) < \sigma(i) \ \forall\ i \in S, i \neq z_S\}$. Similarly, for any collection of subsets $\mathcal{A}$, let $z_{\mathcal{A}} = (z_S \in S : S \in \mathcal{A})$ and $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) = \bigcap_{S\in\mathcal{A}} \mathscr{D}_S(z_S)$.

It follows from the above definition that if $\mathscr{D}_{\mathcal{M}}(z_{\mathcal{M}}) \neq \varnothing$, then $z_{\mathcal{M}}$ represents consistent top-ranked products. In other words, there exists a ranking σ of all products, such that for all $S \in \mathcal{M}$, z_S is the most preferred product in S under σ. The next lemma shows that the Rank Aggregation LP reduces to finding consistent top-ranked products for $\mathcal{M}$ that minimize the cost.

Lemma 2.5 (Three Equivalent Characterizations) *Let the polytope $\mathscr{M}$ be defined by*

$$\mathscr{M} \equiv \text{ Convex hull of the set } \left\{ \quad (\mathbb{1}[\sigma, i, S] : i \in S,\ S \in \mathcal{M}) \in \{0,1\}^{\sum_{S\in\mathcal{M}} |S|} \quad : \ \sigma \in \mathscr{P}_n \right\} \ .$$

The Rank Aggregation LP *is equivalent to the following three optimization problems:*

$$\min_{\boldsymbol{x}_{\mathcal{M}} \in \mathscr{M}} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} x_{i,S} = \min_{\sigma\in\mathscr{P}_n} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} \mathbb{1}[\sigma, i, S] = \min_{z_{\mathcal{M}} : \mathscr{D}_{\mathcal{M}}(z_{\mathcal{M}}) \neq \varnothing} \ \sum_{S\in\mathcal{M}} c_{z_S, S}, \ \text{and}$$

the Rank Aggregation LP *is NP-hard, even for the collection of all pairs* $\mathsf{Pairs} = \{\{i,j\} : i \neq j\}$.

Proof: By definition of $\mathscr{M}$, it follows that the Rank Aggregation LP is equivalent to $\min_{\boldsymbol{x}_{\mathcal{M}}\in\mathscr{M}} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} x_{i,S}$. By the standard result in linear programming, the optimal solution occurs at an extreme point of $\mathscr{M}$, corresponding to a binary vector $(\mathbb{1}[\sigma, i, S] : i \in S,\ S \in \mathcal{M})$ for some $\sigma \in \mathscr{P}_n$. Therefore, the problem is also equivalent to $\min_{\sigma\in\mathscr{P}_n} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} \mathbb{1}[\sigma, i, S]$. The final equivalence follows because for each $\sigma \in \mathscr{P}_n$, $\sum_{i\in S} \mathbb{1}[i, \sigma, S] \cdot c_{i,S} = c_{z_S,S}$, where z_S is the most preferred product in S under σ. A vector $z_{\mathcal{M}} = (z_S : S \in \mathcal{M})$ is feasible if $\mathscr{D}_{\mathcal{M}}(z_{\mathcal{M}}) = \cap_{S\in\mathcal{M}} \mathscr{D}_S(z_S) \neq \varnothing$; therefore, the problem $\min_{\sigma\in\mathscr{P}_n} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} \mathbb{1}[\sigma, i, S]$ is equivalent to the following problem:

$$\min \left\{ \ \sum_{S\in\mathcal{M}} c_{z_S,S} \ \middle| \ z_S \in S \ \ \forall\ S \in \mathcal{M} \text{ and } \bigcap_{S\in\mathcal{M}} \mathscr{D}_S(z_S) \neq \varnothing \ \right\} = \min_{z_{\mathcal{M}} : \mathscr{D}_{\mathcal{M}}(z_{\mathcal{M}}) \neq \varnothing} \ \sum_{S\in\mathcal{M}} c_{z_S,S}.$$

The NP-hardness result follows immediately from the result of Dwork et al. (2001). ■

It is clear from Lemma 2.5 that constructing the optimal ranking is equivalent to determining the choice z_S^* for each subset $S \in \mathcal{M}$ under the optimal ranking that minimizes the total cost. Naturally, we cannot determine the optimal choices for the subsets separately because the resulting choices may violate the transitivity of preferences; for instance, choosing 2 from $\{1,2,3\}$ and 3 from $\{2,3,4\}$ will imply that products 2 and 3 are simultaneously preferred, violating the transitivity of the preferences. As a result, the optimal choices across subsets must be determined *jointly* to ensure that the transitivity of preferences is satisfied; therefore, searching for the top-ranked product in each set is challenging, and as noted in Lemma 2.5, the corresponding optimization problem is NP-hard even for the collection just comprising all pairs of products.

However, for important collections of subsets, we show that the optimal choices of some subsets can be determined "separately" from those of other subsets. When carefully done, this allows us to efficiently determine the optimal choices by decoupling their search. We formalize this concept next.

3. Rational Separation and Choice Graph

In this section, we introduce the notions of *rational separation* and *choice graph*. These concepts allow us to decouple the search for the optimal choices in certain subsets, provided that choices from other subsets are fixed. We start with the formal definition of rational separation and then provide an example to illustrate the concept.

Definition 3.1 (Rational Separation) Given three subsets, A, B, and C, we say that A and B are *rationally separable* given C, written as $A \perp\!\!\!\perp B \mid C$, if for all $z_A \in A$, $z_B \in B$, and $z_C \in C$, whenever $\mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing$ and $\mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$, we also have $\mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$. Similarly, for any three collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, we say that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$ if $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$, then $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

A consequence of rational separation between A and B given C is that

$$\mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing \quad \text{if and only if} \quad \mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing \text{ and } \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing\,.$$

In other words, if we ensure that the choices from subsets A and C are consistent (the choices do not violate the transitivity of preferences) and the choices from subsets B and C are also consistent, then the rational separation property of A and B given C immediately implies that the choices from A, B, and C are mutually consistent. The following simple example illustrates this property.

Example 3.2 Let $A = \{1,2,3\}$, $B = \{5,6,7\}$, and $C = \{2,3,4,5,6\}$. We will show $A \perp\!\!\!\perp B \mid C$; i.e., if we fix the choice from C, then the choices from A and B can be made independently of each other.

Consider the case when $z_C = 2$, i.e., product 2 is the most preferred among all the products in set C. Because set C includes product 3, it follows that 2 is preferred over 3. Therefore, the most preferred product in set A must be either 1 or 2, implying that $\mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing \iff z_A \in \{1,2\}$. Furthermore, because $z_C = 2$ does not constrain the preference ordering among products 5, 6, and 7, we conclude that $\mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing \iff z_B \in \{5,6,7\}$. The case when $z_C \in \{3,4,5,6\}$ can be argued similarly, resulting in the following table of consistent choices for z_A, z_B, for each z_C:

z_C	2	3	4	5	6
$\{z_A : \mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing\}$	$\{1,2\}$	$\{1,3\}$	$\{1,2,3\}$	$\{1,2,3\}$	$\{1,2,3\}$
$\{z_B : \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing\}$	$\{5,6,7\}$	$\{5,6,7\}$	$\{5,6,7\}$	$\{5,7\}$	$\{6,7\}$

Using the above table, we now show that $A \perp\!\!\!\perp B \mid C$. By construction, for any combination of choices (z_A, z_B, z_C) in the above table, we must have $\mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing$ and $\mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$. Therefore, it suffices to show that we also have $\mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$.

Suppose $z_C = 2$. There are two cases: $z_A = 1$ and $z_A = 2$. If $z_A = 1$, then for any $z_B \in \{5,6,7\}$, consider the ranking σ that ranks product 1 at the top; product 2, second; and product z_B, third. The ranking σ results in the choices z_A, z_B, and z_C from sets A, B, and C, respectively. Therefore, $\sigma \in \mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C)$. In a similar manner, when $z_A = 2$ and for any $z_B \in \{5,6,7\}$, the ranking τ that ranks product 2 at the top and product z_B, second, is such that $\tau \in \mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C)$. Therefore, in both cases, we have $\mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$, which is the desired result. Using a similar argument for other values of z_C shows that $A \perp\!\!\!\perp B \mid C$.

Computational Savings from Rational Separation: An important consequence of the rational separation property is that once we fix the best item in set C, the optimal choices in sets A and B can be determined separately. More precisely, if $A \perp\!\!\!\perp B \mid C$, then

$$\begin{aligned}
&\min\left\{c_{z_A,A} + c_{z_B,B} + c_{z_C,C} \;\middle|\; z_A \in A, z_B \in B, z_C \in C, \quad \mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing\right\} \\
&\quad = \min_{z_C \in C} \quad c_{z_C,C} + \left\{\min_{z_A \in A : \mathscr{D}_A(z_A) \cap \mathscr{D}_C(z_C) \neq \varnothing} c_{z_A,A}\right\} + \left\{\min_{z_B \in B : \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing} c_{z_B,B}\right\}.
\end{aligned}$$

The above equation shows that instead of a brute force search over all possible triples (z_A, z_B, z_C), searching over z_A and z_B separately, for each z_C, suffices. Note that for each triple (z_A, z_B, z_C), checking whether $\mathscr{D}_A(z_A) \cap \mathscr{D}_B(z_B) \cap \mathscr{D}_C(z_C) \neq \varnothing$ is an $O(1)$ operation. Therefore, a brute force search over all possible triples is an $O(n^3)$ operation. Instead, when we exploit rational separation, searching for the optimal z_A and z_B separately is an $O(n)$ operation for each value of z_C. Because z_C has $O(n)$ possible values, determining the optimal choices becomes an $O(n^2)$ operation. Thus, rational separation has allowed us to shave off a factor n from the computational complexity. When $\mathcal{M}$ contains many subsets, the savings become more substantial.

To facilitate its use, we represent the rational separation structure of a collection of subsets $\mathcal{M}$ in the form of a *choice graph*, which is defined as follows:

Definition 3.3 (Choice Graph Representation of Rational Separation) Given a collection of subsets $\mathcal{M}$, a graph $\mathsf{G} = (\mathcal{M}, \mathsf{E})$ is a *choice graph* of $\mathcal{M}$ if G is an undirected graph whose vertices correspond to the subsets in $\mathcal{M}$, and for any three collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, if $\mathcal{C}$ *separates* $\mathcal{A}$ and $\mathcal{B}$ in graph G, then it must be that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \,|\, \mathcal{C}$. Here, we say $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$ in the graph G if every path from a set in $\mathcal{A}$ to a set in $\mathcal{B}$ passes through some set in $\mathcal{C}$.

In Sections 4 and 5, we show how the structure of the choice graph is related to the computational complexity of computing LoR. Before that, in Sections 3.1–3.4, we identify the choice graphs for four important classes of set collections: nested, laminar, differentiated, and k-deletion. As discussed below, these collections occur frequently in inventory and RM applications.

3.1 Choice Graphs for Nested Collections

The collection $\mathcal{M} = \{S_1, S_2, \ldots, S_m\}$ is nested if and only if $S_1 \subset S_2 \subset \cdots \subset S_m$ for some indexing of the subsets; that is, smaller subsets are always contained within larger ones. Nested collections are common in retail and RM settings. In their classic paper, van Ryzin and Mahajan (1999) show that when the demand follows the multinomial logit (MNL) model and all products have the same price and cost, a profit-maximizing retailer should stock the top h products with the largest MNL weights (which measure the popularity of the corresponding products), and the optimal size (or variety) h is determined by store-specific factors, such as the profit margin or the market size. When there are no stockouts, the resulting collection of offer sets across different stores is nested. Cachon et al. (2005) extend this finding to the case in which consumers may decide to search, which results in non-purchase even if an offered product is preferred to the no-purchase option. Similarly, Talluri and van Ryzin (2004) show that in the single-leg revenue management problem, a *nested policy* in which the offer set is always chosen from a nested collection is profit maximizing. The nesting is by the fare order under the MNL and other common demand models. Nested collections also arise when airlines control seat inventories using *serial nested booking class controls*, in which a lower-value class product is removed from the offering whenever a higher-value class in the nested hierarchy is closed for sale[2].

Example 1 in Talluri and van Ryzin (2004) provides a concrete illustration of serial nesting control. Suppose the airline offers three fare products, Y, M, and K, listed in the order of decreasing prices and increasing restrictions. The airline serves five segments of customers differing in their willingness to pay and the restrictions they qualify for. Talluri and van Ryzin (2004) show that to maximize revenue, it is optimal for the airline to offer only subsets in $\mathcal{M} = \{\{Y\}, \{Y, K\}, \{Y, K, M\}\}$,

[2] Vinod (2006), a senior vice president of Sabre Corporation (the leading RM company in the travel industry), notes that airlines frequently use a serial nesting control.

even if there are $8 = 2^3$ possible offer sets to consider. When the airline follows the optimal policy, choice observations are collected on a nested collection of subsets. By Theorem 3.4, the choice graph for this example is: $\boxed{\{Y\}} — \boxed{\{Y,K\}} — \boxed{\{Y,K,M\}}$.

Theorem 3.4 *The choice graph for the nested collection $\mathcal{M}$ is the line graph $S_1 - S_2 - \cdots - S_m$.*

To prove the above result, we need to show that for any three disjoint collections of subsets $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathcal{M}$, if $\mathcal{A}$ and $\mathcal{B}$ are separated by $\mathcal{C}$ in the line graph, then $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \,|\, \mathcal{C}$. We provide a proof sketch here for the case when each collection, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, consists of a single set; the detailed proof is given in Appendix A. In particular, consider the sets S_i, S_j, and S_k with $i < j < k$, so that $S_i \subset S_j \subset S_k$, and, therefore, S_j separates S_i from S_k in the line graph. Let $z_i \in S_i$, $z_j \in S_j$, and $z_k \in S_k$ be the respective choices, such that $\mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \neq \varnothing$ and $\mathscr{D}_{S_k}(z_k) \cap \mathscr{D}_{S_j}(z_j) \neq \varnothing$. We now show that $\mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \cap \mathscr{D}_{S_k}(z_k) \neq \varnothing$ to establish that $S_i \perp\!\!\!\perp S_k \,|\, S_j$. For that, note that

$$\begin{aligned} \mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \neq \varnothing \quad &\text{if and only if} \quad z_j \in S_j \setminus S_i \text{ or } z_j = z_i, \quad \text{and} \\ \mathscr{D}_{S_k}(z_k) \cap \mathscr{D}_{S_j}(z_j) \neq \varnothing \quad &\text{if and only if} \quad z_k \in S_k \setminus S_j \text{ or } z_j = z_k \; . \end{aligned}$$

Consider a ranking $\sigma \in \mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j)$. We modify σ as follows: If $z_k \in S_k \setminus S_j$, we obtain the modified ranking $\tilde{\sigma}$ by moving z_k to the top of the list σ, so that z_k becomes the most preferred product under $\tilde{\sigma}$. Because $z_k \notin S_i \cup S_j$, this modification does not affect the choices from S_i and S_j; therefore, we have $\tilde{\sigma} \in \mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \cap \mathscr{D}_{S_k}(z_k)$. On the other hand, if $z_k = z_j$, we obtain $\tilde{\sigma}$ by moving all the elements in $S_k \setminus S_j$ to the bottom of σ. Again, because $(S_k \setminus S_j) \cap (S_j \cup S_i) = \varnothing$, the choices from both S_i and S_j are unaffected; therefore, we also have $\tilde{\sigma} \in \mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \cap \mathscr{D}_{S_k}(z_k)$. In both cases, we have shown that $\mathscr{D}_{S_i}(z_i) \cap \mathscr{D}_{S_j}(z_j) \cap \mathscr{D}_{S_k}(z_k) \neq \varnothing$, which is the desired result.

3.2 Choice Graphs for Laminar Collections

The laminar collection generalizes the nested collection. A collection $\mathcal{M}$ is laminar if it has the property that for any two distinct sets $A, B \in \mathcal{M}$, either $A \subset B$, or $B \subset A$, or $A \cap B = \varnothing$; in other words, any two sets are either disjoint or contain each other. Let $\mathsf{T} = (\mathcal{M}, \mathsf{E})$ be the graph representing the laminar collection $\mathcal{M}$ with vertices corresponding to the sets in $\mathcal{M}$ and an edge $\{S_1, S_2\} \in \mathsf{E}$ between the two sets S_1 and S_2 if and only if they are *adjacent*. Two subsets $A, B \in \mathcal{M}$ are *adjacent* if they are not disjoint and there is no other subset $X \in \mathcal{M}$, such that $A \subset X \subset B$ or $B \subset X \subset A$. The main result of this section is stated in the following theorem:

Theorem 3.5 *The choice graph for the laminar collection $\mathcal{M}$ is the forest (a collection of disjoint trees)* $\mathsf{T} = (\mathcal{M}, \mathsf{E})$.

The proof of this theorem is given in Appendix A. Here, we present a motivating example. In practice, the laminar collection arises naturally when customers use the EBA screening process (Tversky and Kahneman 1974) to form consideration sets before making a purchase. In the EBA screening process, customers essentially whittle down large product spaces by sequentially eliminating products that do not meet their criteria. Figure 1 provides an example from a RM application. Suppose a customer is looking for a one-way economy flight from Los Angeles, CA, (LAX) to Toronto, ON, (YYZ) on July 25, 2015. This example has a total of eight itineraries, with each one described by three attributes: departure time, number of connections, and airline. We label the itineraries $1, 2, \ldots, 8$, as shown at the bottom of Figure 1.

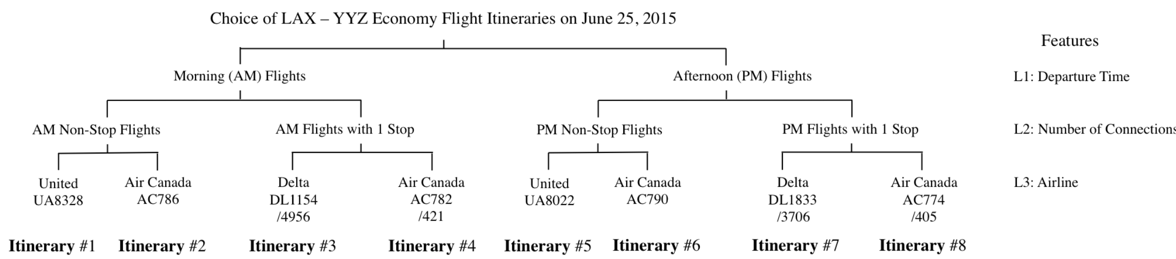
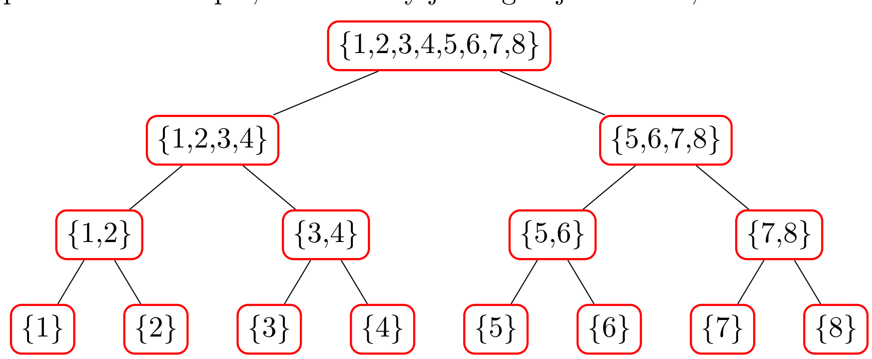


Figure 1 An illustration of the EBA screening process for a one-way flight itinerary from LAX to YYZ on June 25, 2015. The data were collected on March 1, 2015 from Orbitz.com.

`Orbitz.com` allows customers to filter their search results by various attributes. For this example, we assume that customers follow the EBA screening process to narrow their search results. For instance, a customer who prefers non-stop flights that depart in the morning would filter the search results on these attributes to arrive at the consideration set $\{1, 2\}$, and then make a choice from this set. In a similar way, customers who prefer afternoon flights would choose from the consideration set (or the *effective* offer set) $\{5, 6, 7, 8\}$. Because `Orbitz.com` can observe the filtered search results, the purchase data provides choice observations from effective offer sets from the following laminar collection: $\mathcal{M} = \{\{1,2,3,4,5,6,7,8\}, \{1,2,3,4\}, \{5,6,7,8\}, \{1,2\}, \{3,4\}, \{5,6\}, \{7,8\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$. The choice graph for this example, obtained by joining adjacent sets, is the following tree:

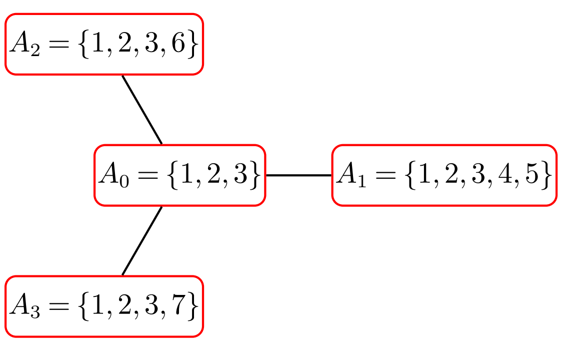


3.3 Choice Graphs for Differentiated Collections

The differentiated collection $\mathcal{M} = \{A_0, A_1, \ldots, A_\ell\}$ comprises sets that possess a common core but are otherwise disjoint. It has the property in which $A_i \cap A_j = A_0$ for all $1 \le i, j \le \ell$ so that the set A_0 forms the common core of products present in every set. Beyond the core products, there is no overlap in the subsets. Consider a star-shaped choice graph with A_0 at the center and edges from A_0 to A_i for each i; more formally, let $\mathsf{Star} = (\mathcal{M}, \mathsf{E})$ be the graph with node set $\mathcal{M}$ and edge set $\mathsf{E} = \{(A_0, A_i) \colon 1 \le i \le \ell\}$. Theorem 3.6 shows that Star is a choice graph of $\mathcal{M}$.

A differentiated collection arises when, in addition to common products, firms carry a unique sub-collection of products in each store to cater to local tastes. A chain of grocery stores often carries a common set of popular products, but each store differentiates itself by carrying locally sourced products, such as local cheese or beer. As an example, consider a retailer with stores in Indianapolis, Detroit, and St. Louis. By analyzing the actual grocery transaction data from the IRI academic dataset (Bronnenberg et al. 2008), we identified the following brands of soup that were sold at each of the three stores; the number in parentheses next to each brand is the product ID, which ranges from 1 to 7.

- Indianapolis: Healthy Choice (1), Campbell's (2), Progresso (3), Health Valley (4), Pacific (5)
- Detroit: Healthy Choice (1), Campbell's (2), Progresso (3), Hormel (6)
- St. Louis: Healthy Choice (1), Campbell's (2), Progresso (3), Private Label (7)



In this example, the common core set of products is given by $A_0 = \{1, 2, 3\}$, and we have the following collection of offer sets: $A_1 = \{1, 2, 3, 4, 5\}$ for the Indianapolis store, $A_2 = \{1, 2, 3, 6\}$ for the Detroit store, and $A_3 = \{1, 2, 3, 7\}$ for the St. Louis store. The figure on the left shows the choice graph for this example. Note that the choice graph of the differentiated collection is a tree, but the collection is not laminar. The main result of this section is stated in the following theorem, whose proof is given in Appendix A.

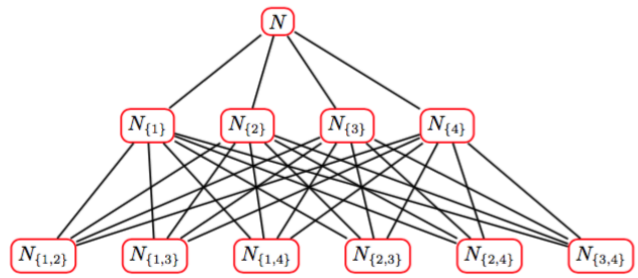
Theorem 3.6 *The star-graph* Star *is the choice graph for the differentiated collection* $\mathcal{M}$.

3.4 Choice Graphs for k-deletion Collections

The k-deletion collection, denoted by $\mathtt{Del}_k$, consists of subsets obtained from $N = \{1, 2, \ldots, n\}$ by deleting at most k products. For any subset $A \subseteq N$, let $N_A = N \setminus A$. Then, $\mathtt{Del}_k = \{N_A : |A| \le k\}$, so

$\texttt{Del}_0 = \{N\}$, $\texttt{Del}_1 = \{N, N_{\{i\}} : i \in N\}$, $\texttt{Del}_2 = \{N, N_{\{i\}}, N_{\{i,j\}} : i \in N, j \in N, i \neq j\}$, and so on. The k-deletion collection arises in retail contexts when a firm faces frequent stock-outs and replenishments. In practice, retailers often maintain high service levels, resulting in low stock-out rates, especially in the fast-moving consumer goods industries (Gruen and Corsten 2008). As a result, at most $k \ll n$ products are stocked out at any point (see the case study with real-world data in Section 6).

The choice graph for the k-deletion collection has a layered structure with $k+1$ layers (see the figure on the right, for which $k=2$ and $N=\{1,2,3,4\}$). For any $0 \leq \ell \leq k$, let $\mathcal{F}_\ell \equiv \{N_A : |A| = \ell\}$ denote the collection of subsets obtained by deleting exactly ℓ products from N. It is clear that $\texttt{Del}_k = \cup_{\ell=0}^{k} \mathcal{F}_\ell$. Let $\mathsf{G}_k = (\texttt{Del}_k, \mathsf{E}_k)$ be the layered graph whose vertices are the subsets in $\texttt{Del}_k$, and G_k has $k+1$ layers, with the subsets in $\mathcal{F}_\ell$ comprising layer ℓ for $0 \leq \ell \leq k$. The nodes in successive layers are connected to each other, resulting in the edge set $\mathsf{E}_k = \{\{X_\ell, X_{\ell+1}\} : X_\ell \in \mathcal{F}_\ell,\ X_{\ell+1} \in \mathcal{F}_{\ell+1},\ \ell = 0, 1, \ldots, k-1\}$. We have the following result, the detailed proof of which is presented in Appendix A:



Theorem 3.7 *For all $k \geq 0$, the graph G_k is a choice graph for the k-deletion collection $\texttt{Del}_k$.*

We conclude this section with a few remarks on the algorithmic verification of the rational separation property. Given a collection of subsets A, B, and C, we can verify if the triple (z_A, z_B, z_C) of choices are mutually consistent in $O(n)$ operations by forming a directed graph with nodes as products and edges from z_S to $S \setminus \{z_S\}$ for $S \in \{A, B, C\}$, and by checking for the presence of directed cycles. As a result, we can verify if the subsets A, B, and C satisfy the rational separation property in $O(n^4)$ complexity by checking Definition 3.1 for all possible triples (z_A, z_B, z_C). In a similar way, we can verify the rational separation property for any three collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, albeit with exponential complexity. The details are shown in Appendix B.

4. Limits of Rationality in Trees

We now derive the complexity of computing the LoR when the underlying choice graph $\mathsf{T} = (\mathcal{M}, \mathsf{E})$ is a tree. We first establish that the Rank Aggregation LP can be solved in linear time by using $O(n\,|\mathcal{M}|)$ operations. This complexity is the best possible bound (up to constant factors) because reading the problem inputs requires $\Omega(n\,|\mathcal{M}|)$ operations to read all the objective coefficients $\{c_{i,S} : i \in S, S \in \mathcal{M}\}$. We prove the result by designing an efficient DP to find the optimal solution.

We also describe a compact linear program with $O(n\,|\mathcal{M}|)$ variables and $O(n\,|\mathcal{M}|)$ constraints to determine the optimal solution using off-the-shelf tools. The main result of this section is stated in the following theorem:

Theorem 4.1 (Linear Time Computation of $\mathsf{LoR}(\mathcal{M})$) *If the underlying choice graph is a tree $\mathsf{T} = (\mathcal{M}, \mathsf{E})$, then the* Rank Aggregation LP *can be solved via a DP in linear time with $O(n\,|\mathcal{M}|)$ operations.*

Proof: By Lemma 2.5, the Rank Aggregation LP is equivalent to $Z^* = \min\left\{\sum_{S\in\mathcal{M}} c_{z_S,S} \;\middle|\; z_S \in S \;\; \forall S \in \mathcal{M}, \;\; \bigcap_{S\in\mathcal{M}} \mathscr{D}_S(z_S) \neq \varnothing\right\}$. Pick an arbitrary vertex in T, label it as root, and treat it as the root vertex in the tree. In addition, for any $S \in \mathcal{M}$, let $\mathsf{T}(S)$ denote the *collection* of subsets (aka vertices) in the subtree rooted at S; for mathematical convenience, we exclude S from $\mathsf{T}(S)$. Note that it follows from the tree structure of the choice graph that if $S_1 \neq S_2$, $S_1 \notin \mathsf{T}(S_2)$, and $S_2 \notin \mathsf{T}(S_1)$, then $\mathsf{T}(S_1) \cap \mathsf{T}(S_2) = \varnothing$. For each $S \in \mathcal{M}$, define the value function $V_S : S \to \mathbb{R}$ for the subtree rooted at S as follows:

$$V_S(z_S) = c_{z_S,S} + \min\left\{ \sum_{A\in\mathsf{T}(S)} c_{z_A,A} \;\middle|\; z_A \in A \;\; \forall A \in \mathsf{T}(S), \;\; \mathscr{D}_S(z_S) \cap \mathscr{D}_{\mathsf{T}(S)}(z_{\mathsf{T}(S)}) \neq \varnothing \right\},$$

where $\mathscr{D}_{\mathsf{T}(S)}(z_{\mathsf{T}(S)}) = \cap_{A\in\mathsf{T}(S)} \mathscr{D}_A(z_A)$. By definition, $Z^* = \min_{z_{\mathsf{root}}\in\mathsf{root}} V_{\mathsf{root}}(z_{\mathsf{root}})$. Note that

$$V_S(z_S) = c_{z_S,S} + \min\left\{ \sum_{A\in\,\mathsf{Children}(S)} \left[c_{z_A,A} + \sum_{B\in\mathsf{T}(A)} c_{z_B,B}\right] \;\middle|\; \mathscr{D}_S(z_S) \cap \left(\bigcap_{A\in\,\mathsf{Children}(S)} \left[\mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)})\right] \right) \neq \varnothing \right\}.$$

To obtain an efficient dynamic programming recursion, we now exploit the rational separation as described by the choice tree. Note that

$$\begin{aligned} &\mathscr{D}_S(z_S) \cap \left(\bigcap_{A\in\,\mathsf{Children}(S)} \left[\mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)})\right] \right) \neq \varnothing \\ &\Leftrightarrow \;\; \mathscr{D}_S(z_S) \cap \mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)}) \neq \varnothing \quad\; \forall\, A \in \mathsf{Children}(S) \\ &\Leftrightarrow \;\; \mathscr{D}_S(z_S) \cap \mathscr{D}_A(z_A) \neq \varnothing \quad \text{and} \quad \mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)}) \neq \varnothing \quad\; \forall A \in \mathsf{Children}(S)\;, \end{aligned}$$

where the first equivalence follows because S *rationally separates* its children and their descendants from one another; that is, for any two children A_1 and A_2 of S, $(A_1 \cup \mathsf{T}(A_1)) \perp\!\!\!\perp (A_2 \cup \mathsf{T}(A_2)) \mid S$ because each path from $A_1 \cup \mathsf{T}(A_1)$ to $A_2 \cup \mathsf{T}(A_2)$ must go through S. Similarly, the second equivalence follows because for each $A \in \mathsf{Children}(S)$, A rationally separates S from $\mathsf{T}(A)$; that is, $S \perp\!\!\!\perp \mathsf{T}(A) \mid A$ because all paths from S to $\mathsf{T}(A)$ must go through A. It now follows that

$$V_S(z_S) = c_{z_S,S} + \min\left\{ \sum_{A\in\,\mathsf{Children}(S)} \left[c_{z_A,A} + \sum_{B\in\mathsf{T}(A)} c_{z_B,B}\right] \;\middle|\; \mathscr{D}_S(z_S) \cap \mathscr{D}_A(z_A) \neq \varnothing, \right.$$

$$\text{and} \quad \mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)}) \neq \varnothing \ \forall A \in \mathsf{Children}(S), \Bigg\}$$

$$= c_{z_S,S} + \sum_{A \in \mathsf{Children}(S)} \min \left\{ c_{z_A,A} + \sum_{B \in \mathsf{T}(A)} c_{z_B,B} \;\middle|\; \mathscr{D}_S(z_S) \cap \mathscr{D}_A(z_A) \neq \varnothing, \right.$$

$$\left. \text{and} \quad \mathscr{D}_A(z_A) \cap \mathscr{D}_{\mathsf{T}(A)}(z_{\mathsf{T}(A)}) \neq \varnothing \right\}$$

$$= c_{z_S,S} + \sum_{A \in \mathsf{Children}(S)} \min_{z_A \in A \,:\, \mathscr{D}_S(z_S) \cap \mathscr{D}_A(z_A) \neq \varnothing} V_A(z_A) \ ,$$

where the last equality follows from the definition of $V_A(z_A)$. The above recursion allows us to start at the leaf vertices of T, traverse in a breadth-first search, and recursively compute the value functions until we reach root.

Complexity: By traversing the choice graph in a breadth-first search manner, the DP recursion can be computed in linear time using $O(n\,|\mathcal{M}|)$ operations. The details are in Appendix C.1. ■

An immediate corollary of Theorem 4.1 is that the complexity of computing the LoR for the nested, laminar, and differentiated collections (introduced in Section 3) is $O(n\,|\mathcal{M}|)$. We emphasize the the linear running time in Theorem 4.1 is the best one possible because the number of parameters $|\{c_{i,S} : i \in S,\, S \in \mathcal{M}\}|$ for the problem is already at least $\Omega(n\,|\mathcal{M}|)$ in the worst case.

An alternative computational method based on linear programming (LP): In practical applications, having algorithms that exploit off-the-shelf optimization packages is often desirable. For a tree choice graph, the Rank Aggregation LP can also be reformulated as a compact linear program with just $O(n\,|\mathcal{M}|)$ variables and constraints. This result is stated in the Theorem 4.2, whose proof is in Appendix C.2, which also gives specific examples of the compact LP representation for the nested, laminar, and differentiated collections introduced in Section 3.

Theorem 4.2 (Compact LP for Tree) *When the choice graph* $\mathsf{T} = (\mathcal{M}, \mathsf{E})$ *is a tree, the* Rank Aggregation LP *can be equivalently reformulated as the following LP with* $O(n\,|\mathcal{M}|)$ *variables and constraints:*

$$\min \left\{ \sum_{S \in \mathcal{M}} \sum_{i \in S} c_{i,S}\, y_{i,S} \;\middle|\; y_{r,A} \le z_{r,\{A,B\}} \text{ and } y_{r,B} \le z_{r,\{A,B\}} \ \forall\ r \in A \cap B, \{A,B\} \in \mathsf{E}, \right.$$

$$\left. \sum_{i \in S} y_{i,S} = 1 \ \forall\ S \in \mathcal{M}, \sum_{r \in A \cap B} z_{r,\{A,B\}} = 1 \ \forall\ \ \{A,B\} \in \mathsf{E},\ \boldsymbol{y} \ge \mathbf{0}, \boldsymbol{z} \ge \mathbf{0} \right\}$$

5. Limits of Rationality for General Graphs

We now extend the results of Section 4 to arbitrary choice graphs that are not necessarily trees. Our extension uses the concept of *tree decomposition* of a graph and the corresponding *tree width.*

These concepts are well known in classic graph theory and are used to describe the computational complexity of many algorithms on graphs. In Section 5.1, we present the concepts of tree decomposition and tree width. Then, in Section 5.2, we consider the problem of computing the limit of rationality in general choice graphs. We show that if the tree width is bounded, then the RANK AGGREGATION LP can be solved efficiently in polynomial time by formulating the rank aggregation problem as a DP on the tree decomposition of the choice graph. We then develop methods to deal with the graphs of unbounded tree width by using the new concept of choice depth, which is unique to choice graphs.

5.1 Tree Decomposition and Tree Width

We first review the concepts of *tree decomposition* and *tree width* by using the definition given by Robertson and Seymour (1986); see Halin (1976) and Diestel (2005) for equivalent definitions.

Definition 5.1 A *tree decomposition* of a choice graph $\mathsf{G} = (\mathcal{M}, \mathsf{E})$ is a connected tree T_G with nodes indexed by $\{\mathcal{X}_b : b \in \mathsf{B}\}$. For all $b \in \mathsf{B}$, the node $\mathcal{X}_b \subseteq \mathcal{M}$ is a **subset** of the vertices of G and is referred to as a bag of vertices. The collection of bags $\{\mathcal{X}_b : b \in \mathsf{B}\}$ satisfies the following properties:

1. $\mathcal{M} = \cup_{b \in \mathsf{B}} \mathcal{X}_b$.
2. For every edge $\{S_1, S_2\} \in \mathsf{E}$ in graph G, there exists a bag $\mathcal{X}_b$, such that $\{S_1, S_2\} \subseteq \mathcal{X}_b$.
3. Running intersection property: If $\mathcal{X}_b$ and $\mathcal{X}_c$ contain a set $S \in \mathcal{M}$, then every bag in the unique path between $\mathcal{X}_b$ and $\mathcal{X}_c$ must also contain S.

The width of the tree decomposition T_G, denoted by $\mathsf{width}(\mathsf{T}_G)$, is equal to the size of the largest bag minus one; that is, $\max_{b \in \mathcal{B}} |\mathcal{X}_b| - 1$. The *tree width* of a graph G, denoted by $\mathsf{tw}(\mathsf{G})$, is the minimum width among all tree decompositions of G.

Note that G has many tree decompositions. A trivial one is a tree with a single bag consisting of all vertices in the original graph. The problem of computing the tree width of a graph is NP-complete (Arnborg et al. 1987), but extensive research efforts to develop efficient algorithms for finding tree decompositions with small widths have been made (Fomin and Kratsch 2010). We do not address this particular issue here, as it is beyond the scope of our paper.

Each bag $\mathcal{X}_b \subseteq \mathcal{M}$ in a tree decomposition T_G corresponds to a collection of subsets. To distinguish between the vertices of the original graph G and those of T_G, we will refer to a vertex in T_G as a bag of subsets. It is important to note that if the choice graph G is already a tree, then G is **NOT** its tree decomposition; rather, a tree decomposition of G will be a graph, where each bag $\mathcal{X}_b = \{S_1, S_2\}$ consists of exactly two subsets, such that $\{S_1, S_2\} \in \mathsf{E}$ corresponds to an edge in the original tree. Therefore, if the original choice graph is a tree, then its tree width is equal to $2-1=1$. The following lemma shows that our concept of rational separation from Definition 3.1 is preserved under any tree decomposition.

Lemma 5.2 (Preservation of Rational Separation) *Every tree decomposition T_G of a choice graph G preserves the rational separation encapsulated in G; that is, for any bag $\mathcal{X}_k$ that lies on the unique path between bags $\mathcal{X}_i$ and $\mathcal{X}_j$, we have $\mathcal{X}_i \perp\!\!\!\perp \mathcal{X}_j \mid \mathcal{X}_k$.*

Proof: It suffices to show that whenever $\mathcal{X}_k$ separates $\mathcal{X}_i$ from $\mathcal{X}_j$ in T_G, then $\mathcal{X}_k$ also separates $\mathcal{X}_i$ from $\mathcal{X}_j$ in the original graph G. To arrive at a contradiction, we suppose on the contrary that there exist $A \in \mathcal{X}_i$ and $B \in \mathcal{X}_j$, such that for some path $A = C_0 - C_1 - \cdots - C_m - C_{m+1} = B$ in G, we have $C_\ell \notin \mathcal{X}_k$ for all $0 \le \ell \le m+1$.

Now, consider the edges $\mathsf{e}^\ell = \{C_\ell, C_{\ell+1}\}$ for $0 \le \ell \le m$ in the original graph G. Property 2 in Definition 5.1 of the tree decomposition implies that for $0 \le \ell \le m$, there exists a bag $\mathcal{Y}_\ell$ in the tree decomposition T_G, such that $\mathsf{e}^\ell = \{C_\ell, C_{\ell+1}\} \subseteq \mathcal{Y}_\ell$. By our construction, $A = C_0 \in \mathcal{Y}_0$ and $B = C_{m+1} \in \mathcal{Y}_m$.

Claim 1: For each $0 \le \ell \le m-1$, the bag $\mathcal{X}_k$ does not lie on the path from $\mathcal{Y}_\ell$ to $\mathcal{Y}_{\ell+1}$. To prove this, we note that by construction, $C_{\ell+1} \in \mathcal{Y}_\ell$ and $C_{\ell+1} \in \mathcal{Y}_{\ell+1}$. Then, by the running intersection property in Definition 5.1, every path connecting $\mathcal{Y}_\ell$ and $\mathcal{Y}_{\ell+1}$ in T_G must contain $C_{\ell+1}$. Because we are assuming that $C_{\ell+1} \notin \mathcal{X}_k$, it follows that $\mathcal{X}_k$ cannot lie on the path from $\mathcal{Y}_\ell$ to $\mathcal{Y}_{\ell+1}$ in graph T_G. The claim thus follows.

Claim 2: There is a path in the tree decomposition T_G from $\mathcal{X}_i$ to $\mathcal{Y}_0$ that does not contain $\mathcal{X}_k$, and there is a path in the tree decomposition T_G from $\mathcal{Y}_m$ to $\mathcal{X}_j$ that does not contain $\mathcal{X}_k$. To prove this, note that both bags $\mathcal{X}_i$ and $\mathcal{Y}_0$ contain C_0, so by the running intersection property of the tree decomposition, it must be that every bag in the path from $\mathcal{X}_i$ to $\mathcal{Y}_0$ also contains C_0. Since $C_0 \notin \mathcal{X}_k$, it must be that $\mathcal{X}_k$ does not lie on this path, giving the desired result. The proof for the path from $\mathcal{Y}_m$ to $\mathcal{X}_j$ is exactly the same. The claim thus follows.

It then follows from Claims 1 and 2 that there is a path in the tree decomposition T_G from $\mathcal{X}_i$ to $\mathcal{X}_j$ through the bags $\mathcal{Y}_0, \mathcal{Y}_1, \ldots, \mathcal{Y}_m$ that does not contain $\mathcal{X}_k$. This contradicts the fact that $\mathcal{X}_k$ separates $\mathcal{X}_i$ and $\mathcal{X}_j$ in T_G. Therefore, we must have $\mathcal{X}_i \perp\!\!\!\perp \mathcal{X}_j \mid \mathcal{X}_k$. ■

Because every tree decomposition preserves rational separation, we can solve the Rank Aggregation LP on the tree decomposition. As the first step, the following lemma shows that we can always find a tree decomposition with at most $|\mathcal{M}|$ bags. The result follows immediately from the vertex elimination algorithm of Rose et al. (1976), and we omit the details.

Lemma 5.3 (Theorem 9 in Rose et al. 1976) *We can always construct a tree decomposition of graph G with at most $|\mathcal{M}|$ bags by using $O(|\mathcal{M}| + |\mathsf{E}|)$ operations.*

5.2 Computing $\mathsf{LoR}(\mathcal{M})$ in General Choice Graphs

We now consider the problem of computing the limit of rationality in a general choice graph. When the tree width of a choice graph is bounded, the following theorem shows that the Rank Aggregation LP can be solved in polynomial time:

Theorem 5.4 (Complexity in Tree Width) *The* Rank Aggregation LP *can be solved in* $O\left(|\mathcal{M}|\, n^{4+2\mathsf{tw}(\mathsf{G})}\right)$ *operations.*

The proof of Theorem 5.4 makes use of a dynamic programming equation that is similar to the one used in the proof of Theorem 4.1; the detail is in Appendix D. In addition, for a general choice graph with a bounded tree width, we can also solve the Rank Aggregation LP by using a linear program with $O\left(|\mathcal{M}|\, n^{2(\mathsf{tw}(\mathsf{G})+1)}\right)$ variables and $O(|\mathcal{M}|\, n^{\mathsf{tw}(\mathsf{G})+1})$ constraints. This result is stated in the following theorem, whose proof is given in Appendix D.

Theorem 5.5 (LP for Bounded Tree Width) *For any collection $\mathcal{M}$ of subsets and the associated choice graph G, the* Rank Aggregation LP *can be formulated as a linear program with* $O\left(|\mathcal{M}|\, n^{2(\mathsf{tw}(\mathsf{G})+1)}\right)$ *variables and* $O(|\mathcal{M}|\, n^{\mathsf{tw}(\mathsf{G})+1})$ *constraints.*

Dealing with Choice Graphs with Large Tree Widths: The results thus far encompass a broad range of set collections with tree choice graphs, including the nested, laminar, and differentiated collections. Recall from Section 3.4 that G_k denotes the choice graph for the k-deletion collection, $\mathtt{Del}_k$. Note that G_1 is a tree, so its tree width is one. However, for $k \geq 2$, G_k is not a tree, and as shown in Lemma 5.7, its tree width is at least n.

When the tree width of the choice graph increases with the number of products, the methods from the previous sections become intractable because the computational complexity scales exponentially in the tree width. Fortunately, for many collections of subsets, including the k-deletion collection, we can still solve the Rank Aggregation LP efficiently. To describe this result, we introduce the following new concept that is unique to the choice graph:

Definition 5.6 (Choice Depth) *Given a tree decomposition T_G of a choice graph $\mathsf{G} = (\mathcal{M}, \mathsf{E})$, the choice depth under T_G – denoted by $\mathsf{cd}(\mathsf{G}, \mathsf{T}_\mathsf{G})$ – is defined by*

$$\mathsf{cd}(\mathsf{G}, \mathsf{T}_\mathsf{G}) = \max_{b \in \mathsf{B}} \left\{ \left| \bigcup_{S \in \mathcal{X}_b} S \right| - \min_{A \in \mathcal{X}_b} |A| \right\} ,$$

and the choice depth of G, denoted by $\mathsf{cd}(\mathsf{G})$, is defined as the minimum of the choice depth under all possible tree decompositions of G; i.e., $\mathsf{cd}(\mathsf{G}) = \min\{\mathsf{cd}(\mathsf{G}, \mathsf{T}_\mathsf{G}) : \mathsf{T}_\mathsf{G}$ is a tree decomposition of $\mathsf{G}\}$.

The choice depth concept is unique to the choice graph and is fundamentally different from the classic tree width concept. As shown in the following lemma (proved in Appendix D), the choice depth of a graph can be much smaller than the tree width.

Lemma 5.7 (Unbounded Tree Width but Small Choice Depth) *For all $2 \le k < n$, the choice graph G_k (as defined in Section 3.4) of the k-deletion collection $\mathtt{Del}_k$ has tree width of at least n and choice depth of at most k; that is, $\mathsf{cd}(\mathsf{G}_k) \le k < n \le \mathsf{tw}(\mathsf{G}_k)$.*

The main result of this section is stated in the following theorem, which characterizes the complexity of the LoR in terms of choice depth. The proof uses the same DP equation, as in the proof of Theorem 5.4, but exploits the special structure of the choice graph to show that instead of maintaining the top-ranked product in each set in each bag, maintaining the ranking of the top $\mathsf{cd}(\mathsf{G})+1$ products in each bag suffices. The proof of this result is given in Appendix D.

Theorem 5.8 (Complexity in Choice Depth) *For any collection of subsets $\mathcal{M}$ and the associated choice graph G, the* Rank Aggregation LP *can be solved in $O\left(|\mathcal{M}|\, n^{4+2\mathsf{cd}(\mathsf{G})}\right)$.*

LP for graphs with bounded choice depth: As a companion to the LP in Theorem 5.5, we now describe a compact LP to solve the rank aggregation problem for graphs with bounded choice depth. Suppose that $\mathsf{cd}(\mathsf{G}) \le k-1$. Let T_G be a tree decomposition of the choice graph G with the collection of bags $\{\mathcal{X}_b : b \in \mathtt{B}\}$ and choice depth less than or equal to k. For any set $S \subset N$, let $\mathscr{O}_k(S)$ denote the set of all possible top k orderings of the elements in S; that is, $\mathscr{O}_k(S) = \{\boldsymbol{a} \in S^k : a_i \neq a_j \ \forall\ i \neq j, 1 \le i,j \le k\}$, where $S^k = S \times S \times \cdots \times S$ denotes the k-fold Cartesian product of the set S. For each $b \in \mathtt{B}$, let $U_b = \cup_{S \in \mathcal{X}_b} S$ denote the collection of all products comprising the bag $\mathcal{X}_b$. For any $b \in \mathtt{B}$ and $\boldsymbol{a} \in \mathscr{O}(U_b)$, let $\mathscr{P}_k(\boldsymbol{a}, U_b) \subset \mathscr{P}_n$ denote the set of rankings σ that result in $\boldsymbol{a}$ as the top k ordering among the elements in U_b; that is,

$$\mathscr{P}_k(\boldsymbol{a}, U_b) = \left\{ \sigma \in \mathscr{P}_n \ \middle|\ \text{the top } k \text{ products under } \sigma \text{ correspond to the vector } \boldsymbol{a} \right\} .$$

Let $z_{\mathcal{X}_b} = (z_S : S \in \mathcal{X}_b)$ denote the vector of choices from all the subsets in the bag $\mathcal{X}_b$. We let $\mathscr{O}_k(U_b, z_{\mathcal{X}_b})$ denote the top k orderings $\boldsymbol{a} \in \mathscr{O}_k(U_b)$ that are consistent with the choices $z_{\mathcal{X}_b}$; that is,

$$\mathscr{O}_k(U_b, z_{\mathcal{X}_b}) = \left\{ \boldsymbol{a} \in \mathscr{O}_k(U_b) \ \middle|\ z_S = a_{j_S^*} \text{ where } j_S^* = \min_{1 \le \ell \le k \,:\, a_\ell \in S} \ell \quad \forall\ S \in \mathcal{X}_b \right\} .$$

For each $b \in \mathtt{B}$ and $\boldsymbol{a} \in \mathscr{O}_k(U_b)$, we introduce the variables $x_{b,\boldsymbol{a}}$ that correspond to the probability that $\boldsymbol{a}$ is the top k ordering among the elements in U_b. For any $\boldsymbol{a}_1 \in \mathscr{O}_k(U_{b_1})$ and $\boldsymbol{a}_2 \in \mathscr{O}_k(U_{b_2})$ that are consistent with each other, that is, $\mathscr{P}_k(\boldsymbol{a}_1, U_{b_1}) \cap \mathscr{P}_k(\boldsymbol{a}_2, U_{b_2}) \neq \varnothing$, let $w_{b_1,\boldsymbol{a}_1,b_2,\boldsymbol{a}_2}$ be the joint probability that $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are the top k orderings among U_{b_1} and U_{b_2}, respectively. We now define the polyhedron $\mathscr{H}$ as follows:

$$\mathscr{H} = \Bigg\{ \left(y^b_{z_{\mathcal{X}_b}} : b \in \mathcal{B},\ \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing \right) \ \Bigg| $$

$$y^b_{z_{\mathcal{X}_b}} = \sum_{\boldsymbol{a} \in \mathscr{O}_k(U_b, z_{\mathcal{X}_b})} x_{b,\boldsymbol{a}} \qquad \forall\, b \in \mathsf{B}, z_{\mathcal{X}_b}, \text{ such that } \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing$$

$$x_{b_1,\boldsymbol{a}_1} = \sum_{\boldsymbol{a}_2 \in \mathscr{O}_k(U_{b_2}) :\ \mathscr{P}_k(\boldsymbol{a}_2, U_{b_2}) \cap \mathscr{P}_k(\boldsymbol{a}_1, U_{b_1}) \neq \varnothing} w_{b_1,\boldsymbol{a}_1,b_2,\boldsymbol{a}_2} \qquad \forall \text{ edge } \{\mathcal{X}_{b_1}, \mathcal{X}_{b_2}\} \in \mathsf{T}_{\mathsf{G}} \text{ and } \boldsymbol{a}_1 \in \mathscr{O}_k(U_{b_1})$$

$$x_{b_2,\boldsymbol{a}_2} = \sum_{\boldsymbol{a}_1 \in \mathscr{O}_k(U_{b_1}) :\ \mathscr{P}_k(\boldsymbol{a}_2, U_{b_2}) \cap \mathscr{P}_k(\boldsymbol{a}_1, U_{b_1}) \neq \varnothing} w_{b_1,\boldsymbol{a}_1,b_2,\boldsymbol{a}_2} \qquad \forall \text{ edge } \{\mathcal{X}_{b_1}, \mathcal{X}_{b_2}\} \in \mathsf{T}_{\mathsf{G}} \text{ and } \boldsymbol{a}_2 \in \mathscr{O}_k(U_{b_2})$$

$$\sum_{\boldsymbol{a} \in \mathscr{O}_k(U_b)} x_{b,\boldsymbol{a}} = 1 \qquad \forall\, b \in \mathsf{B}$$

$$\text{All variables are non-negative} \ \Bigg\}.$$

The number of variables and constraints in the above LP is determined by the cardinality of the set $\left\{ z_{\mathcal{X}_b} \ :\ \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing \right\}$, for each bag $\mathcal{X}_b$. As noted above and as shown in the proof of Theorem 5.8, the vector of top choices $z_{\mathcal{X}_b}$ from all subsets $S \in \mathcal{X}_b$ can be recovered from the top $(1 + \mathtt{cd}(\mathsf{G}))$-ranked products in the bag $\mathcal{X}_b$. Therefore, we must have

$$\left| \left\{ z_{\mathcal{X}_b} \ :\ \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing \right\} \right| = O(n^{1+\mathtt{cd}(\mathsf{G})}) = O(n^k), \text{ for each } b,$$

because $\mathtt{cd}(\mathsf{G}) \leq k - 1$. Now, because the number of bags is at most $|\mathcal{M}|$ (cf. Lemma 5.3), it follows that the polyhedron $\mathscr{H}$ is described by $O(|\mathcal{M}|\, n^{2k})$ variables and $O(|\mathcal{M}|\, n^k)$ constraints. The main result of this section is that $\mathscr{M}$ is equal to $\mathscr{H}$, as stated below. The proof is in Appendix D.

Theorem 5.9 (LP for Small Choice Depth) *Given a collection of subsets $\mathcal{M}$ and the associated choice graph G with $\mathtt{cd}(\mathsf{G}) \ \leq \ k-1$, we have $\mathscr{M} = \mathscr{H}$.*

For the $(k-1)$-deletion collection from Section 3.4, we can express the polyhedron $\mathscr{H}$ even more compactly. Note that $\mathtt{cd}(\mathtt{Del}_{k-1}) = k - 1$. For the $(k-1)$-deletion collection, $U_b = \cup_{S \in \mathcal{X}_b} S = N$ for all $b \in \mathsf{B}$, which simplifies the above polyhedron because we can omit the variables $w_{b_1,\boldsymbol{a}_1,b_2,\boldsymbol{a}_2}$. Moreover, the dependence on the bag index $b \in \mathsf{B}$ can be dropped. With $y_{i,S}$ denoting the probability that i is chosen from subset S, the polyhedron for the k-deletion collection – denoted by $\mathscr{H}(\mathtt{Del}_{k-1})$ – can be compactly described as follows:

$$\mathscr{H}(\mathtt{Del}_{k-1}) = \left\{ (y_{i,S} : i \in S,\ S \in \mathcal{M}) \ \middle| \ \sum_{\boldsymbol{a} \in \mathscr{O}_k(N)} x_{\boldsymbol{a}} = 1,\ y_{i,S} = \sum_{\boldsymbol{a} \in \mathscr{O}_k(N, z_S)} x_{\boldsymbol{a}} \quad \forall\, i \in S, S \in \mathcal{M} \right\}.$$

This result is summarized in the following corollary.

Corollary 5.10 (LP for the $(k-1)$-Deletion Collection) *For the $(k-1)$-deletion collection, the associated* RANK AGGREGATION LP *requires only $O(n^k)$ variables and $O(n^k)$ constraints.*

Thus far, we have shown that for any set collection whose choice graph admits a small choice depth, the corresponding LoR can be computed efficiently. Complementing this result, we now demonstrate a set collection whose choice depth increases with the number of products n, and the corresponding RANK AGGREGATION LP is NP-hard. By Lemma 2.5, the RANK AGGREGATION LP is NP-hard for the collection of all pairs of products $\mathsf{Pairs} = \{\{i,j\} : i \neq j\}$. The next proposition shows that for this collection of subsets, both tree width and choice depth increase with the number of products n. The proof is given in Appendix D.

Proposition 5.11 (Unbounded Choice Depth and Tree Width) *For every choice graph* $\mathsf{G} = (\mathsf{Pairs}, \mathsf{E})$ *associated with the collection* Pairs*, we have* $\mathsf{tw}(\mathsf{G}) \geq 2(n-2)$ *and* $\mathsf{cd}(\mathsf{G}) \geq n-2$*.*

5.3 Extension to Allow an Additional Structure on Rankings

The development, thus far, has considered distributions over all possible rankings. In some contexts, rationality imposes an additional structure, which constrains the space of possible rankings. For example, in several retail contexts, products are offered on price or display promotion, and rationality dictates that the promoted copy of a product is always preferred over its corresponding non-promoted copy. This structure can be captured through a constrained set of rankings of over $2n$ items, where each item is either a promoted or a non-promoted copy of a product and each ranking is constrained to prefer the promoted copies over the corresponding non-promoted copies of all n products (see Section 6.2). In a similar way, price promotions with multiple levels of discounts may be captured by creating a copy for each level of discount. Our framework, including all the concepts and the theoretical results, that we developed above extends almost immediately when considering distributions over a constrained set of rankings. The details are shown in Appendix D.7.

6. Case Study on the IRI Academic Dataset

In this section, we present the results from applying our methodology to compute the rationality loss and the parametric losses from fitting commonly used choice models to the IRI Academic Dataset (Bronnenberg et al. 2008). This dataset consists of real-world purchase transactions of consumer packaged goods for grocery and drug store chains, collected from 47 markets across the US. The purpose of our case study is twofold: (a) quantify the magnitude of the rationality loss vis-à-vis the total loss incurred from using common parametric models for different grocery product categories and (b) demonstrate a diagnostic tool based on our methodology that provides guidance for model selection—both for choosing the model complexity within the RUM family and for determining when going beyond the RUM family is necessary. For our analysis, we focused on the multinomial logit (MNL) and the 15-class mixture of logits (LCMNL) models, both of which are known to be consistent with the RUM assumption.

6.1 Data Analysis

The data consist of weekly sales transactions aggregated over all customers. Because of the large volume of transactions, we focused on data from calendar year 2007, specifically the first two weeks of the year. The resulting transactions span 29 product categories (cf. Table EC.1 in Appendix E.1).

We processed the data separately for each category to obtain choice data in the form of a collection of offer sets and corresponding aggregated sales fractions. For this, we first need to define our unit of analysis, the "product," for each category. In the dataset, items are identified at the most granular level in terms of their respective universal product codes (UPCs). We cannot work directly with each UPC because many UPCs have very few sales records. To deal with data sparsity, we aggregated the items with the same vendor code (comprising digits 4 to 8 in the UPC) into a single "vendor". Of the resulting vendors, we focused our analysis on the top nine purchased vendors (across all stores during the two-week period), labeling each of them as a product. We then combined the remaining vendors into one (the 10th) product, yielding a mapping from each UPC to one of the 10 product IDs ($n = 10$). Aggregating UPCs with the same vendor code is a common technique used in data pre-processing (Bronnenberg and Mela 2004, Ailawadi et al. 2006, Nijs et al. 2007).

Converting the raw transaction data to choice observations: With the products defined as above, we converted the sales transactions for each category into choice observations, as described next. Different categories were processed separately, and we followed the same processing steps for all the categories. For a given category, let $\mathcal{T}$ denote[3] the collection of purchase transactions (see Table EC.1 for the numbers of transactions $|\mathcal{T}|$ across categories), where each transaction consists of the following information: the week of the purchase, the store ID where the purchase occurred, the UPC of the purchased product, the quantity purchased, the price paid, and an indicator of whether the product was on price or display promotion. For each transaction $t \in \mathcal{T}$, let $j_t \in \{1, \ldots, n\}$ denote the product ID of the purchased UPC; $I_t \in \{0, 1\}$, the promotion indicator (1 if promoted and 0 otherwise); w_t, the week of purchase; and s_t, the store of purchase. We processed these transactions in two steps: (i) identifying the offer and promotion sets and (ii) computing the corresponding sales fractions.

[3] For notational brevity, we dropped the dependence of the quantities on the category label.

Step 1: Identifying the offer and promotion sets: We associated each store-week combination in our dataset with an offer and a promotion set. The offer sets of products vary across stores because of differences in the assortments carried, and across weeks within each store because of operational reasons, such as stock-outs. With store s and week w, we associated the offer set $S_{s,w} = \bigcup_{t \in \mathcal{T}} \{j_t : s_t = s, w_t = w\} \subseteq \{1, \ldots, n\}$, defined as the union of all products, in the category, that were purchased at least once during week w at store s. This construction provides only an approximation because a product might have been stocked out during the middle of the week, resulting in an over-estimation of the offer set, or might not have been purchased at all, resulting in an under-estimation of the offer set. The estimation errors, however, tend to be small in this dataset because each product is obtained by aggregating tens or hundreds of UPCs, and a stock out or non-purchase of a product will happen only when *all* the UPCs corresponding to the product are stocked out or not purchased, which is rare.

The purchase behavior is also affected by promotion activity. Therefore, we associated with each store s and week w the set of promoted products $P_{s,w} \subseteq S_{s,w}$, defined as $P_{s,w} = \{j \in S_{s,w} : \exists\ t \in \mathcal{T} \text{ s.t. } j_t = j, I_t = 1, s_t = s, w_t = w\}$, containing all the products that were purchased on promotion at least once during week w at store s[4]. As is the case for the offer sets, our construction only yields an approximation because the promotion status of a product might have changed during the week, and we did not observe it. Without additional data, our construction results in reasonable approximations, particularly because when a product is promoted at least once during a week, we observe that most of its purchases in the week occurred during the promotion.

The above procedure resulted in the collection of offer set and promotion set combinations $\mathcal{M} = \{(S_{s,w}, P_{s,w}) : (s,w) \text{ is a store-week combination}\}$ for each category. Note that having the same offer set at a store for two different weeks but with different promotion sets is possible because the store is running different promotion campaigns. Therefore, we keep track of the ordered pairs (S,P).

Step 2: Computing the sales fractions: For each offer set and promotion set combination $(S,P) \in \mathcal{M}$, we computed the corresponding sales fractions $f_{j,S,P}$ as the fraction of times j was purchased when the offer set was S and the promotion set was P. More precisely, we set $M_{S,P} = \sum_{t \in \mathcal{T}} \mathbb{1}[S_{s_t,w_t} = S, P_{s_t,w_t} = P]$, the number of times (S,P) was offered and

$$f_{j,S,P} = \frac{1}{M_{S,P}} \sum_{t \in \mathcal{T}} \mathbb{1}[j_t = j \text{ and } S_{s_t,w_t} = S, P_{s_t,w_t} = P]$$

[4] Since a product corresponds to a vendor, a product is considered to be on promotion even if only *one* UPC of that vendor was purchased on promotion. This encoding does not capture the number of UPCs per vendor that are on promotion, yielding only an approximation. An alternate encoding would create multiple promotion labels to capture different promotion intensities, measured as the fraction of UPCs on promotion for each vendor. A similar encoding can also be used to capture different levels of price discounts associated with price promotions.

Table EC.1 in Appendix E.1 reports the data summary; in particular, $|\mathcal{T}|$, $|\mathcal{M}|$, and $\frac{1}{|\mathcal{M}|}\sum_{(S,P)\in\mathcal{M}} M_{S,P}\cdot|S|$ for each of the 29 product categories.

Preparing promotion data for rationality loss computation: Because both the offer and promotion sets vary in our dataset, we used the extension of our framework that can handle constrained rankings. Specifically, we encoded the promotion information by creating two copies of each product—a non-promoted copy and a promoted copy—and by enlarging the product space to $2n$ items, consisting of the promoted and non-promoted copies of all n products. The key observation we made is that customers always prefer the promoted copy of a product over its non-promoted copy. Therefore, we modeled the preferences of a population of customers through a distribution over rankings of $2n$ items that are constrained, such that the promoted copies of products are always preferred over their corresponding non-promoted copies. In other words, customers were modeled through a distribution over a constrained set of rankings, and the rationality loss was computed by finding the distribution that minimizes the loss.

More precisely, let $(j,0)$ and $(j,1)$ denote the non-promoted and promoted copies of j, respectively, and let $\mathscr{P}_{2n}^{\mathsf{con}} \subseteq \mathscr{P}_{2n}$ denote the set of constrained rankings defined as $\mathscr{P}_{2n}^{\mathsf{con}} = \{\sigma\in\mathscr{P}_{2n}\colon \sigma(j,1)<\sigma(j,0)\ \forall\ j\in N\}$. Correspondingly, we mapped each tuple (S,P) to the set $A(S,P)=\{(j,0)\colon j\in S\setminus P\}\cup\{(j,1)\colon j\in P\}$, comprising the appropriate copy of each product. Let $\tilde{\mathcal{M}}$ denote $\{A(S,P)\colon (S,P)\in\mathcal{M}\}$, and $M_{\tilde{S}}$ and $f_{a,\tilde{S}}$ respectively denote $M_{S,P}$ and $f_{a,S,P}$, where $\tilde{S}=A(S,P)$ and $a=(j,1)$ if $j\in P$ and $(j,0)$, otherwise.

6.2 Computing the Rationality and Parametric losses

For each category, we used the choice observations to compute the rationality loss and the parametric losses from fitting the single-class MNL (henceforth, simply MNL) and 15-class latent class MNL (henceforth, simply LCMNL) models. The LCMNL model class subsumes the MNL model; therefore, the parametric loss under the former will be smaller than that under the latter model. We focused on computing the log-likelihood loss. For the MNL and LCMNL models, this corresponds to solving the MLE problems.

Computing the rationality loss. We used our proposed framework to compute the rationality loss. With the notation from the previous section, the rationality loss is given by

$$\begin{aligned}\text{Rationality loss} = \min\Bigg\{ -\frac{1}{|\mathcal{T}|}\sum_{\tilde{S}\in\tilde{\mathcal{M}}} M_{\tilde{S}} \sum_{a\in\tilde{S}} f_{a,\tilde{S}}\log(x^{\lambda}_{a,\tilde{S}}/f_{a,\tilde{S}}) \;\Bigg|\; \sum_{\sigma\in\mathscr{P}_{2n}^{\mathsf{con}}}\lambda(\sigma)=1, \lambda(\sigma)\geq 0\ \forall\ \sigma\in\mathscr{P}_{2n}^{\mathsf{con}}, \\ \text{and}\quad x_{a,\tilde{S}} = \sum_{\sigma\in\mathscr{P}_{2n}^{\mathsf{con}}}\lambda(\sigma)\cdot\mathbb{1}[\sigma,a,\tilde{S}]\ \forall\ a\in\tilde{S},\tilde{S}\in\tilde{\mathcal{M}}\Bigg\} \qquad (2)\end{aligned}$$

The above optimization problem is similar to the LoR problem in (1) but with the constraint that the distribution λ is over the set of constrained rankings $\mathscr{P}_{2n}^{\mathsf{con}}$. As detailed in Section 5.3, our theoretical results extend naturally for solving (2).

We solved the above constrained convex program by using the FW algorithm. The FW algorithm is a classic algorithm for solving constrained convex programs. We implemented the following standard variant (see Algorithms 1, 2, and 3 in Jaggi 2013): let $\boldsymbol{x}^{(k)}$ be the solution of (2) at the end of iteration k. We update the solution in iteration $k+1$ by solving the Rank Aggregation LP with $c_{a,\tilde{S}} = \frac{\partial}{\partial x_{a,\tilde{S}}} \mathsf{loss}(\boldsymbol{f}_{\tilde{\mathcal{M}}}, \boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}^{(k)}}$ for all $a \in \tilde{S}$, $\tilde{S} \in \tilde{\mathcal{M}}$ but over the constrained set of rankings $\mathscr{P}_{2n}^{\mathsf{con}}$ to obtain the optimal solution $\boldsymbol{s}^*_{(k)}$, and then by performing the update $\boldsymbol{x}^{(k+1)} := (1-\gamma^*)\boldsymbol{x}^{(k)} + \gamma^* \boldsymbol{s}^*_{(k)}$, where γ^* is obtained through a line search: $\gamma^* = \arg\min_{\delta \in [0,1]} \mathsf{loss}(\boldsymbol{f}_{\mathcal{M}}, \boldsymbol{x}^{(k)} + \delta \cdot (\boldsymbol{s}^*_{(k)} - \boldsymbol{x}^{(k)}))$. We started the algorithm with the solution obtained from fitting the LCMNL model (as detailed below) as the initial solution, and determined $\boldsymbol{s}^*_{(k)}$ by solving the constrained Rank Aggregation LP over the k-deletion polyhedron $\mathscr{H}(\mathtt{Del}_k)$, described in Corollary 5.10, with $k=4$. We used this approximation because we observed that at most four (out of ten) products were not offered in most of the offer sets across the categories. By carrying out the iterations until an appropriate stopping condition is met, we obtained the optimal loss.

Computing the parametric losses for the MNL and LCMNL models: For the MNL and LCMNL models, we computed the total loss for each category by maximizing the log-likelihood function. The resulting optimization problems are standard for these models, and we defer the details to Appendix E.2. The parametric losses under the MNL and LCMNL models are then obtained by subtracting the rationality loss from their respective total losses.

6.3 Results and Discussion

For each category, we reported the rationality loss and the parametric losses we computed. In this section, we will discuss (a) the relationship between rationality and parametric losses (Section 6.3.1), (b) the use of the rationality loss as a diagnostic tool for model selection (Section 6.3.2), (c) the benefits of going beyond the RUM family, and (d) the relationship between rationality loss and market concentration. Appendix E.6 also presents an additional discussion on the robustness of our results under different loss metrics.

6.3.1 Rationality loss vis-à-vis parametric losses. Figure 2 shows a bar graph of the different categories, arranged in descending order of their respective total losses under the MNL model. Each bar is composed of three losses: the rationality loss, the LCMNL parametric loss, and the difference between the parametric losses of the LCMNL and MNL models, represented in this order. For each category, the total length of the three bars corresponds to the parametric loss under the MNL model. What is immediately striking from the bar graph in Figure 2 is that the rationality loss accounts for most of the total losses across the categories, with the parametric loss being relatively small. Fitting more complex RUM models, such as the LCMNL model, can reduce or even eliminate (for some categories) the parametric loss; the rationality loss cannot be reduced unless one goes beyond the RUM class. This finding strongly supports and encourages work on relaxing the rationality assumption. In Appendix E.3, we provide a detailed breakdown of the parametric losses under different models.

6.3.2 Diagnostic tool for model selection. The rationality losses provide guidance for model selection, particularly in (a) choosing the number of mixture components and in (b) going beyond rationality. Specifically, the bar graph provides a bound on the number of mixture components for some of the categories. For instance, fitting the 15-class LCMNL model clearly eliminates (essentially) all the parametric losses for the categories Spaghetti/Italian Sauce, Peanut Butter, Toilet Tissue, Beer, and Blades. Consequently, 15 is an upper bound on the number of mixture components for these models because any further increase will increase model complexity (thereby potentially hurting its out-of-sample predictive power) without any benefit to the in-sample fit. The exact number of mixture components can then be determined using penalized likelihood methods (see Budanova 2016), which require a bound on the number of mixture components as an input. The bounds for other categories can be determined by increasing the number of mixture components until their respective parametric losses become zero.

Our method also provides guidance on when a practitioner should go beyond rationality. To aid this, we plotted the rationality loss and the total loss under the MNL model for all the categories on a scatter plot. Figure 3 represents each category as a point, with the vertical axis value denoting the rationality loss and the horizontal axis denoting the total loss under the MNL model. As expected, all the points are found below the 45° line, indicating that the rationality loss is always less than or equal to the total loss. Model selection, in practice, often means choosing a model that meets a certain performance threshold. For the purposes of illustration, suppose it is specified that a log-likelihood loss of less than or equal to 0.03 is acceptable; this loss corresponds to a mean absolute error of less than 2% between the predicted and the observed sales fractions. The acceptable threshold naturally splits into three regions, as follows:

1. *Region 1: MNL acceptable.* This region is found to the left of the vertical line passing through Total loss $=0.03$ and lying below the 45° line. It consists of all the product categories for which the total loss under the MNL model is acceptable; therefore, a practitioner may stop at the MNL model for these categories.

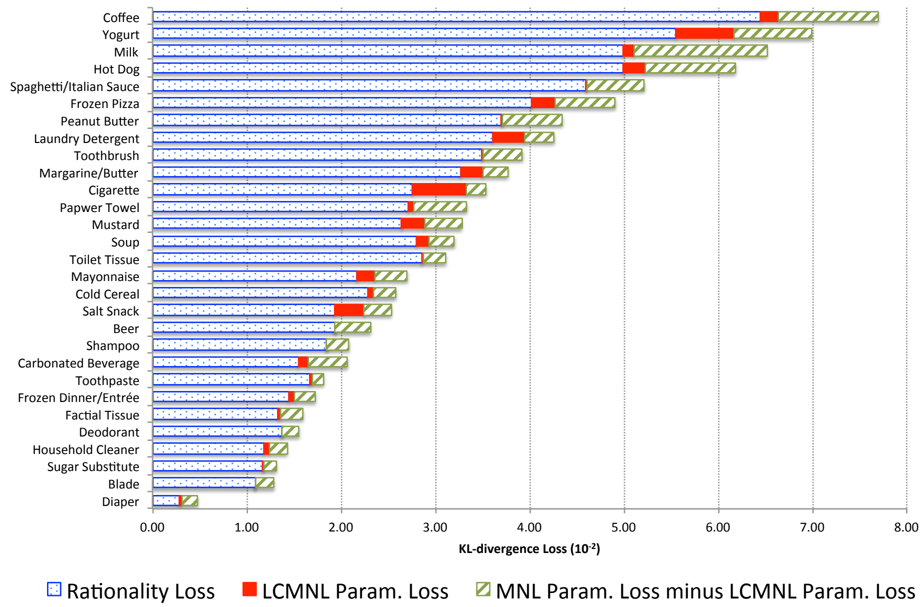


Figure 2 The rationality loss, the parametric losses under the LCMNL model, and the differences between the parametric losses of the MNL and LCMNL models. The categories are arranged in decreasing order of the total loss under the MNL model. The rationality loss accounts for the largest portion of total losses across the categories.

2. *Region 2: fit more complex RUM models.* This region is located below the horizontal line through Rationality loss $= 0.03$ and to the right of the vertical line through Total loss $= 0.03$. It consists of categories for which the total losses (under the MNL model) are not acceptable, *but* the rationality losses are acceptable, so the practitioner can obtain acceptable levels of performance by fitting more complex RUM models, such as K-class LC-MNL models for some value of $K > 1$.
3. *Region 3: RUM unacceptable.* This region is found below the 45° line and above the horizontal line through Rationality loss $= 0.03$, and it consists of 10 (out of 29) categories for the chosen value of the threshold. For these categories, fitting more complex RUM models does not result in an acceptable performance. The observed choices are significantly irrational for these categories, and one must go beyond the RUM class to attain acceptable performance levels.

We also ran the same analysis using transactions data from weeks 27 and 28 (about six months later) of the same year 2007. We found that about 79% of product categories were classified into the same regions in both two-week time periods; see Appendix E.7 for more details. This analysis suggests that our product classification in Figure 3 is robust over time.

6.3.3 Benefits of going beyond the RUM family. The key benefit of our approach is that it allows us to identify the product categories in Region 3 of Figure 3. For these categories, we now illustrate the benefits of going beyond the RUM model. Notably, there is *no* de facto model outside the RUM class that can be readily fit to our data. We selected the generalized attraction model (GAM) class, which was proposed by Gallego et al. (2014) as a generalization

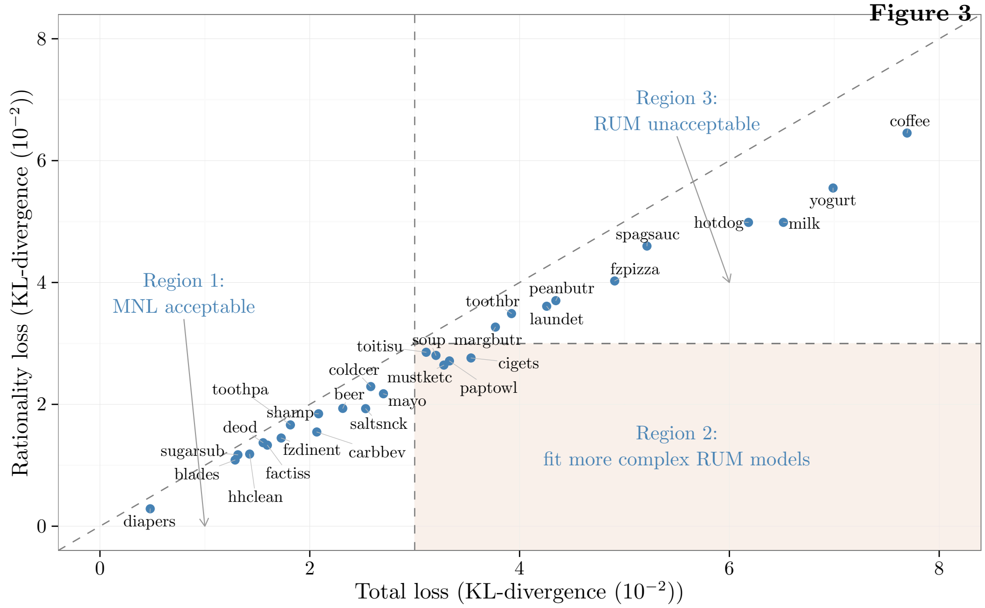


Figure 3 Scatter plot of the rationality loss and the total loss under the MNL model for the 29 product categories. Assuming that a loss of < 0.03 is acceptable, we identified the categories (in Region 3) for which going beyond the RUM class is necessary to obtain an acceptable performance.

of the basic attraction model, of which the MNL model is a special case, to address the issue of overestimation of demand recapture in RM applications. For particular values of the parameters (also called “shadow” attractions), the GAM model can be shown to be outside the RUM class. We extended the GAM model class for our purposes and fitted a latent-class GAM (LC-GAM) model with five classes to our choice data. We verified that the resulting model instances are indeed outside the RUM class. We found that the LC-GAM model allows us to breach the LoR for most of the 29 product categories (see Figure 4). Indeed, the LC-GAM model attains an acceptable performance (loss ≤ 0.03) for the Margarine/Butter (margbutr) category, which belongs to Region 3. See Appendix E.4 for details. We note that while the threshold of 0.03 was chosen for illustration; the appropriate threshold will depend on the application, and correspondingly, the classification of the categories into different regions may be different. At the end of Appendix E.4, we also discuss the benefits and downside risks of going beyond the RUM family.

6.3.4 Factors that might influence rationality loss. Another observation we make from the bar graph in Figure 2 is that a large variation exists in the rationality losses across product categories. With the rationality loss interpreted as a measure of the degree to which customers are irrational, this variation indicates that customer purchase behavior is highly irrational for some categories (e.g., Coffee and Yogurt) and less so for the others (e.g., Diapers and Blades). The data to which we had access are insufficient to provide a convincing explanation for why customers are more irrational when purchasing from some categories than from others. Nevertheless, to obtain a partial explanation, we computed the correlation between the rationality losses and the various category-level features.

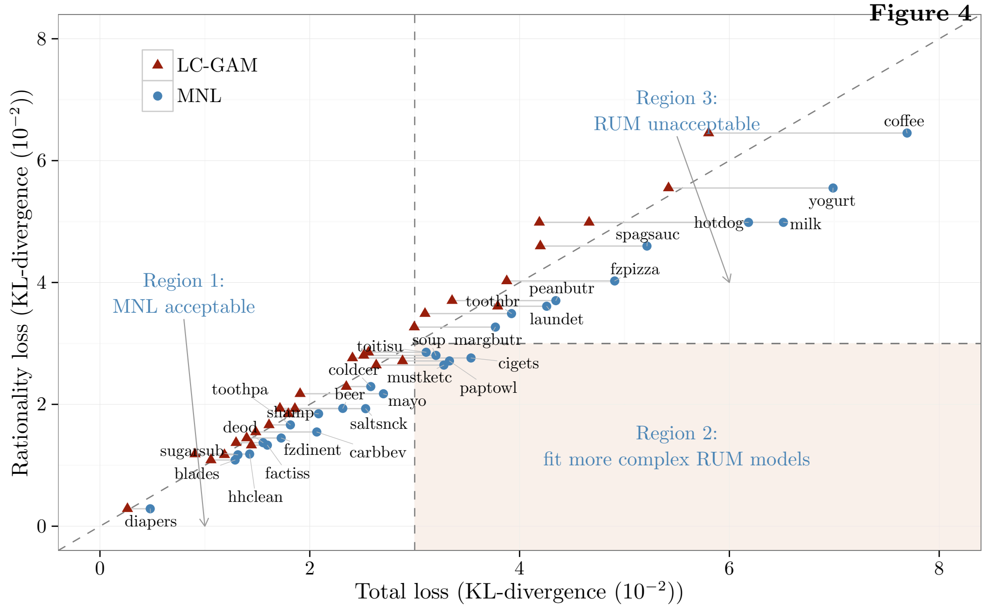


Figure 4 Scatter plot of the rationality loss and the total losses under the MNL and five class LC-GAM models for the 29 product categories. The fitted LC-GAM models are outside the RUM class and are able to breach the LoR for most categories. For margbutr, LC-GAM obtains an acceptable loss (≤ 0.03).

We found that the rationality loss is strongly correlated with the market concentration of the category, defined as the KL-divergence between the uniform distribution and the market share distribution across the different products in the category. A category in which a small number of products capture most of the market will have a market share distribution that is far from a uniform distribution, and thus a high market concentration. On average, product categories with a low market concentration (e.g., Yogurt) are more irrational than those with a large market concentration (e.g., Diapers). This observation is consistent with actual practice. A category with low market concentration, such as Yogurt, often has frequent product introductions, and customers tend to be variety seeking (Baltas et al. 2017), switching their purchases from one week to the next, resulting in non-transitive preferences for each customer. On the other hand, purchases within the Diapers category, with high market concentration, tend to be concentrated over a few products, with customers generally remaining brand loyal across different purchase instances. This purchase behavior results in preferences that are generally transitive for each customer and with a lower rationality loss.

In addition to the market concentration variable, we also explored the correlation between the rationality loss and a hedonic indicator[5], taking the value 1 if a product category possesses hedonic features (those that are strongly associated with affective sensation such as food and soft drinks; see, e.g., Hoyer and Ridgway 1984, Kahn and Lehmann 1991) and 0 otherwise. Using the rule that product categories corresponding to food/snacks (excl. condiments), drinks, or cigarettes are hedonic, while the rest are not, we observed a strong positive correlation between rationality loss and the hedonic indicator variable. Existing research establishes that consumers exhibit complex choice

[5] We thank an anonymous referee for pointing out the connection between the rationality loss and hedonic products.

behaviors for hedonic product categories, including preference reversal, which violates traditional utility theory (Okada 2005). These complex choice behaviors are not sufficiently well-captured by transitive preference orderings, leading to high rationality losses. The details of our analysis are presented in Appendix E.5.

Our findings indicate that going beyond the RUM class may be particularly fruitful for product categories with low market concentrations (or high variety seeking) and hedonic attributes.

7. Conclusion and Extensions

Motivated by widely accepted observations that customer preferences may not be rational or consistent, we considered the problem of quantifying the LoR in choice modeling. We formulated the problem as an optimization problem to find the probability distribution over rankings that is closest to the observed choice data. Our key theoretical contributions include the concepts of *rational separation* and *choice graph* that have allowed us to characterize the source of computational complexity in terms of the structural properties of a graph. We applied our methodology to real-world grocery sales data to identify product categories for which going beyond rational choice models to obtain an acceptable performance is necessary.

Our work also lays the groundwork for a number of interesting future research directions. The immediate extensions include developing approximation techniques to solve the rank aggregation problem for choice graphs with unbounded choice depth, reconciling the LP formulation when the choice graph is a tree with the general LP on the tree composition of a choice graph, and building algorithms to automatically construct choice graphs from data.

Acknowledgments: We are grateful to Professor Vishal Gaur, the associate editor, and the three anonymous referees for their detailed, thoughtful, and insightful comments, which helped substantially improve our manuscript.

References

Ailawadi, K. L., B. A. Harlam, J. Cesar, D. Trounce. 2006. Promotion profitability for a retailer: the role of promotion, brand, category, and store characteristics. *Journal of Marketing Research* **43**(4) 518–535.

Ali, A., M. Meliă. 2012. Experiments with Kemeny ranking: What works when? *Mathematical Social Sciences* **64**(1) 28–40.

Arnborg, S., D. Corneil, A. Proskurowski. 1987. Complexity of finding embeddings in a k-tree. *SIAM Journal on Matrix Analysis and Applications* **8**(2) 277–284.

Baltas, G., F. Kokkinaki, A. Loukopoulou. 2017. Does variety seeking vary between hedonic and utilitarian products? the role of attribute type. *Journal of Consumer Behaviour* .

Barberá, S., P. K. Pattanaik. 1986. Falmagne and the rationalizability of stochastic choices in terms of random orderings. *Econometrica* **54**(3) 707–715.

Block, H. D., J. Marschak. 1960. Random orderings and stochastic theories of responses. Olkin, Ghurye, Hoeffding, Madow, Mann, eds., *Contributions to Probability and Statistics,*. Stanford University Press, Stanford, CA, 97–132.

Bodlaender, H. L., A. M. C. A. Koster. 2011. Treewidth computations II. Lower bounds. *Information and Computation* **209**(7) 1103–1119.

Bodlaender, H. L., T. Wolle, A. M. C. A. Koster. 2006. Contraction and treewidth lower bounds. *Journal of Graph Algorithms and Applications* **10**(1) 5–49.

Bronnenberg, B. J., M. W. Kruger, C. F. Mela. 2008. Database paper: The IRI marketing data set. *Marketing Science* **27**(4) 745–748.

Bronnenberg, B. J., C. F. Mela. 2004. Market roll-out and retailer adoption for new brands. *Marketing Science* **23**(4) 500–518.

Bubeck, S. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning* **8**(3-4) 231–357.

Budanova, S. 2016. Penalized maximum likelihood estimation of finite mixture models Nothwestern University Working Paper.

Cachon, G. P., C. Terwiesch, Y. Xu. 2005. Retail assortment planning in the presence of consumer search. *Manufacturing & Service Operations Management* **7**(4) 330–346.

Chernev, A., U. Böckenholt, J. Goodman. 2015. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology* **25**(2) 333–358.

Crowley, A. E., E. R. Spangenberg, K. R. Hughes. 1992. Measuring the hedonic and utilitarian dimensions of attitudes toward product categories. *Marketing Letters* **3**(3) 239–249.

Davis, J., G. Gallego, H. Topaloglu. 2011. Assortment optimization under variants of the nested logit model. Working Paper, Cornell University.

Dhar, R., K. Wertenbroch. 2000. Consumer choice between hedonic and utilitarian goods. *Journal of marketing research* **37**(1) 60–71.

Diestel, R. 2005. *Graph Theory*. Springer, New York.

Dwork, C., R. Kumar, M. Naor, D. Sivakumar. 2001. Rank aggregation methods for the web. *Proceedings of the 10th International Conference on World Wide Web*. ACM, New York, NY, 613–622.

Falmagne, J. C. 1978. A representation theorem for finite random scale systems. *Journal of Mathematical Psychology* **18**(1) 52–72.

Farias, V. F., S. Jagabathula, D. Shah. 2013. A non-parametric approach to modeling choice with limited data. *Management Science* **59**(2) 305–322.

Fomin, F. V., D. Kratsch. 2010. *Exact exponential algorithms*. Springer Science & Business Media, Berlin, Heidelberg.

Frank, M., P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Research Logistics Quarterly* **3**(1) 95–110.

Gallego, G., R. Ratliff, S. Shebalov. 2014. A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research* **63**(1) 212–232.

Givon, M. 1984. Variety seeking through brand switching. *Marketing Science* **3**(1) 1–22.

Gruen, T. W., D. Corsten. 2008. A comprehensive guide to retail out-of-stock reduction in the fast-moving consumer goods industry. Tech. rep., Grocery Manufacturers of America, Washington, DC.

Halin, R. 1976. S-functions for graphs. *Journal of Geometry* **8**(1) 171–186.

Holbrook, M. B., E. C. Hirschman. 1982. The experiential aspects of consumption: Consumer fantasies, feelings, and fun. *Journal of consumer research* **9**(2) 132–140.

Hoyer, W. D., N. M. Ridgway. 1984. Variety seeking as an explanation for exploratory purchase behavior: A theoretical model. Thomas C. Kinnear, ed., *Advances in Consumer Research*, vol. 11. Association for Consumer Research, Provo, UT, 114–119.

Iyengar, S. S., M. R. Lepper. 2000. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology* **79**(6) 995.

Jagabathula, S. 2011. Nonparametric choice modeling: Applications to operations management. Ph.D. thesis, MIT.

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

Jaggi, M. 2013. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. S. Dasgupta, D. McAllester, eds., *Proceedings of the 30th International Conference on Machine Learning*. Atlanta, GA, 427–435.

Kahn, B. E., M. U. Kalwani, D. G. Morrison. 1986. Measuring variety-seeking and reinforcement behaviors using panel data. *Journal of Marketing Research* **23**(2) 89–100.

Kahn, B. E., D. R. Lehmann. 1991. Modeling choice among assortments. *Journal of retailing* **67**(3) 274–299.

Kenyon-Mathieu, C., W. Schudy. 2007. How to rank with few errors. *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*. STOC '07, New York, NY, 95–103.

Kivetz, R., I. Simonson. 2002a. Earning the right to indulge: Effort as a determinant of customer preferences toward frequency program rewards. *Journal of Marketing Research* **39**(2) 155–170.

Kivetz, R., I. Simonson. 2002b. Self-control for the righteous: Toward a theory of precommitment to indulgence. *Journal of Consumer Research* **29**(2) 199–217.

Lacoste-Julien, S., M. Jaggi. 2015. On the global linear convergence of Frank-Wolfe optimization variants. *Proceedings of the 28th International Conference on Neural Information Processing Systems*. NIPS'15, MIT Press, Cambridge, MA, USA, 496–504.

Mas-Colell, A., M. D. Whinston, J. R. Green. 1995. *Microeconomic Theory*, vol. 1. Oxford University Press, New York.

McFadden, D. L. 2005. Revealed stochastic preference: A synthesis. *Economic Theory* **26**(2) 245–264.

McFadden, D. L., M. K. Richter. 1990. Stochastic rationality and revealed stochastic preference. *Preferences, Uncertainty, and Optimality, Essays in Honor of Leo Hurwicz*. Westview Press, Boulder, CO, 161–186.

Nesterov, Y. 2013. *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer Science & Business Media.

Nijs, V. R., S. Srinivasan, K. Pauwels. 2007. Retail-price drivers and retailer profits. *Marketing Science* **26**(4) 473–487.

O'curry, S., M. Strahilevitz. 2001. Probability and mode of acquisition effects on choices between hedonic and utilitarian options. *Marketing Letters* **12**(1) 37–49.

Okada, E. M. 2005. Justification effects on consumer choice of hedonic and utilitarian goods. *Journal of marketing research* **42**(1) 43–53.

Robertson, N., P. D. Seymour. 1986. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms* **7**(3) 309–322.

Rose, D., G. Lueker, R. E. Tarjan. 1976. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing* **5**(2) 266–283.

Talluri, K., G. J. van Ryzin. 2004. Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50**(1) 15–33.

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM journal on computing* **1**(2) 146–160.

Train, K. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge University Press.

Tversky, A., D. Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science* **185**(4157) 1124–1131.

Tversky, A., P. Slovic, D. Kahneman. 1990. The causes of preference reversal. *American Economic Review* **80**(1) 204–217.

van Ryzin, G. J., S. Mahajan. 1999. On the relationship between inventory costs and variety benefits in retail assortments. *Management Science* **45** 1496–1509.

van Ryzin, G. J., G. Vulcano. 2015. A market discovery algorithm to estimate a general class of non-parametric choice model. *Management Science* **61**(2) 281–300.

van Trijp, H. C. M. 1995. Variety-seeking in product choice behavior. theory with applications in the food domain. Ph.D. thesis, Wageningen University.

van Trijp, H. C. M., W. D. Hoyer, J. J. Inman. 1996. Why switch? product category-level explanations for true variety-seeking behavior. *Journal of Marketing Research* **33**(3) 281–292.

Vinod, B. 2006. Advances in inventory control. *Journal of Revenue and Pricing Management* **4**(4) 367.

Wertenbroch, K. 1998. Consumption self-control by rationing purchase quantities of virtue and vice. *Marketing science* **17**(4) 317–337.

This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.

Online Appendix

The Limit of Rationality in Choice Modeling: Formulation, Computation, and Implications

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

Paat Rusmevichientong
Marshall School of Business, University of Southern California, Los Angeles, CA 90089, rusmevic@marshall.usc.edu

Appendix A: Proofs for Section 3

A.1 Proof of Theorem 3.4

We prove this result by induction on the number of sets in $\mathcal{M}$. If $|\mathcal{M}| = 1$ or 2, the result is trivially true. So, suppose that the result is true for $|\mathcal{M}| = m-1$. Consider the case where $|\mathcal{M}| = m$, with $\mathcal{M} = \{S_1, \ldots, S_m\}$ and $S_1 \subset S_2 \subset \cdots \subset S_m$. Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be arbitrary disjoint sub-collections such that $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$ in the line graph; that is, every path from a set $A \in \mathcal{A}$ to a set $B \in \mathcal{B}$ must passes through some set in $\mathcal{C}$. We now show that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$. There are four cases to consider:

Case 1: $S_m \notin \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. The result then follows immediately from the inductive hypothesis applied to $\mathcal{M} \setminus \{S_m\}$.

Case 2: $S_m \in \mathcal{C}$. Suppose that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Let $\bar{\mathcal{C}} = \mathcal{C} \setminus \{S_m\}$. Note that $\bar{\mathcal{C}}$ still separates $\mathcal{A}$ and $\mathcal{B}$ in the line graph because S_m is the rightmost vertex. Also, $\mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) = \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \cap \mathscr{D}_{S_m}(z_{S_m})$, which implies that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \subset \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}})$. Therefore, it follows that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$. Similarly, we must have that $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$. Then, by invoking the induction hypothesis for $\mathcal{M} \setminus \{S_m\}$, we obtain that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$. This is almost what we want, but we need one additional argument to complete the proof.

Let X be the rightmost vertex among $\mathcal{A} \cup \mathcal{B} \cup \bar{\mathcal{C}}$, which corresponds to the vertex that is closest to S_m; this is well-defined because G is a line graph. The fact that $D_{\mathcal{A}}(z_{\mathcal{A}}) \cap D_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $D_{\mathcal{B}}(z_{\mathcal{B}}) \cap D_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies that either (1) $z_{S_m} \in S_m \setminus X$ or (2) $z_{S_m} = z_X$. Now consider any ranking $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}})$ (such a σ exists because of our arguments above). We modify σ as follows. If $z_{S_m} \in S_m \setminus X$, then we obtain the modified list $\tilde{\sigma}$ by moving the product z_{S_m} to the top of the list σ. Because z_{S_m} is not present in any of the sets in $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$, their choices are not affected.

On the other hand, if $z_{S_m} = z_X$, we obtain $\tilde{\sigma}$ by moving all the products in $S_m \setminus X$ to the bottom of the list σ. Again, none of the other choices are affected because $S_m \setminus X$ does not overlap with any of the sets in $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. Therefore, in both cases, we must have $\tilde{\sigma} \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$, implying that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. This is the desired result.

Case 3: $S_m \in \mathcal{A}$. Suppose that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Let $\bar{\mathcal{A}} = \mathcal{A} \setminus \{S_m\}$. Note that $\mathcal{C}$ still separates $\bar{\mathcal{A}}$ and $\mathcal{B}$ in the line graph because S_m is the rightmost vertex. Also, $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) = \mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{S_m}(z_{S_m})$. As above, this implies that $\mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. Then, by invoking the induction hypothesis for the collection $\mathcal{M} \setminus \{S_m\}$, we have that $\mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. As before, let X be the rightmost vertex among $\bar{\mathcal{A}} \cup \mathcal{B} \cup \mathcal{C}$, which corresponds to the vertex that is closest to S_m; this is well-defined because G is a line graph. Note that it must be the case that $X \in \bar{\mathcal{A}} \cup \mathcal{C}$ because $S_m \in \mathcal{A}$ and $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$. The fact that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies that either (1) $z_{S_m} \in S_m \setminus X$ or (2) $z_{S_m} = z_X$. Once again, using the arguments as above, we can modify a ranking $\sigma \in \mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$ to obtain a ranking $\tilde{\sigma} \in \mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}})$, implying that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. This is the desired result.

Case 4: $S_m \in \mathcal{B}$. This case is the same as Case 3 by symmetry.

In all four cases, we see that we have $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$. By induction, the result holds for all nested collections.

A.2 Proof of Theorem 3.5

We first show that the graph $\mathsf{T} = (\mathcal{M}, \mathsf{E})$, as constructed in Section 3.2, is a forest, that is, a collection of mutually disjoint rooted trees, where the root is the vertex corresponding to the largest subset among all the subsets that are a part of that tree. For that, it suffices to show that T does not contain a cycle. Suppose, on the contrary, that T does contain a cycle, say $\langle S_0, S_1, \ldots, S_t, S_{t+1} = S_0 \rangle$, where $S_0, S_1, \ldots, S_t$ are the distinct subsets. Let S_ℓ be a minimal set among $\{S_0, \ldots, S_t\}$ so that for all $j \neq \ell$, $S_j \nsubseteq S_\ell$. Consider the edges $\{S_{\ell-1}, S_\ell\}$ and $\{S_\ell, S_{\ell+1}\}$ in the cycle. Because $\mathcal{M}$ is laminar and S_ℓ is minimal, it must be that $S_\ell \subseteq S_{\ell+1}$ and $S_\ell \subseteq S_{\ell-1}$; so $S_{\ell-1} \cap S_{\ell+1} \neq \varnothing$, and therefore, either $S_{\ell-1} \subseteq S_{\ell+1}$ or $S_{\ell+1} \subseteq S_{\ell-1}$. If $S_{\ell-1} \subseteq S_{\ell+1}$, then we have $S_\ell \subseteq S_{\ell-1} \subseteq S_{\ell+1}$, so there must not be an edge between S_ℓ and $S_{\ell+1}$ because they are not adjacent. However, this is a contradiction. Similarly, if $S_{\ell+1} \subseteq S_{\ell-1}$, then $S_\ell \subseteq S_{\ell+1} \subseteq S_{\ell-1}$, and we must not have an edge between S_ℓ and $S_{\ell-1}$, which is again a contradiction. Therefore, it must be the case that T has no cycle and is therefore a forest.

To show that T is a choice graph, we note that the unique path from the root to each leaf node consists of nested sets, and we are back to the case of the nested collection in Theorem 3.5. The formal proof requires induction on the number of subsets in $\mathcal{M}$, as shown below.

The case of one or two sets are trivially true. So, suppose that the result is true with $|\mathcal{M}| = m-1$. Now consider the case when $|\mathcal{M}| = m$. Without loss of generality, assume that $\mathcal{M}$ has a *unique* maximal set S^*. If it has multiple maximal sets, we treat each tree separately. Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be arbitrary disjoint sub-collections such that $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$ in T; that is, every path from a set $A \in \mathcal{A}$ to a set $B \in \mathcal{B}$ passes through some set in $\mathcal{C}$. We will now show that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$. There are four cases to consider:

Case 1: $S^* \notin \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. The result then follows immediately from the inductive hypothesis applied to $\mathcal{M} \setminus \{S^*\}$.

Case 2: $S^* \in \mathcal{C}$. Suppose that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Let $\bar{\mathcal{C}} = \mathcal{C} \setminus \{S^*\}$, $\bar{\mathcal{M}} = \mathcal{M} \setminus \{S^*\}$, and let $\bar{\mathsf{T}}$ be the tree obtained from T by removing the vertex S^*. Note that $\bar{\mathcal{C}}$ still separates $\mathcal{A}$ and $\mathcal{B}$ in $\bar{\mathsf{T}}$ because S^* is the root vertex of T. Also, $\mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) = \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \cap \mathscr{D}_{S^*}(z_{S^*})$, which implies that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \subset \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}})$. Therefore, it follows that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$. Similarly, we also have $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$. It now follows by invoking the induction hypothesis for the collection $\mathcal{M} \setminus \{S^*\}$ that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \neq \varnothing$.

In order to complete the proof, let $\mathcal{T}$ denote the collection of maximal sets **among** $\mathcal{A} \cup \mathcal{B} \cup \bar{\mathcal{C}}$. Note that every set in $\mathcal{T}$ is a subset of S^*. The fact that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies that either (1) $z_{S^*} \in S^* \setminus [\cup_{G \in \mathcal{T}} G]$ or (2) $z_{S^*} = z_G$ for some $G \in \mathcal{T}$. Using the arguments in the proof of Theorem 3.4, we can modify any ranking $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}})$ to obtain a ranking $\tilde{\sigma} \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \cap \mathscr{D}_{S^*}(z_{S^*})$, implying that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \cap \mathscr{D}_{S^*}(z_{S^*}) \neq \varnothing$. This completes the proof of this case.

Case 3: $S^* \in \mathcal{A}$. Suppose that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Let $\bar{\mathcal{A}} = \mathcal{A} \setminus \{S^*\}$, $\bar{\mathcal{M}} = \mathcal{M} \setminus \{S^*\}$, and let $\bar{\mathsf{T}}$ be the tree obtained from T by removing the vertex S^*. Note that $\mathcal{C}$ still separates $\bar{\mathcal{A}}$ and $\mathcal{B}$ in $\bar{\mathsf{T}}$. Also, $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) = \mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{S^*}(z_{S^*})$. As above, it follows that $\mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. Then, by invoking the inductive hypothesis for the collection $\mathcal{M} \setminus \{S^*\}$, we obtain that $\mathscr{D}_{\bar{\mathcal{A}}}(z_{\bar{\mathcal{A}}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Let $\mathcal{T}$ denote the collection of maximal sets **among** $\bar{\mathcal{A}} \cup \mathcal{B} \cup \mathcal{C}$. Note that every set in $\mathcal{T}$ is a subset of S^* and it must be in $\bar{\mathcal{A}} \cup \mathcal{C}$ because $S^* \in \mathcal{A}$ is the root vertex and $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$. The fact that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ implies that either (1) $z_{S^*} \in S^* \setminus [\cup_{G \in \mathcal{T}} G]$ or (2) $z_{S^*} = z_G$ for some $G \in \mathcal{T}$. In both these cases, we can argue as above that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\bar{\mathcal{C}}}(z_{\bar{\mathcal{C}}}) \cap \mathscr{D}_{S^*}(z_{S^*}) \neq \varnothing$

Case 4: $S^* \in \mathcal{B}$. This case is the same as Case 3 by symmetry.

In all four cases, we see that we have $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$. By induction, the result holds for all laminar collections.

A.3 Proof of Theorem 3.6

Consider any three disjoint sub-collections, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, such that $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$ in Star. It must then be the case that $A_0 \in \mathcal{C}$ because all the paths from A_i to A_j must go through A_0. Now suppose we are given choices $z_{\mathcal{A}}$, $z_{\mathcal{B}}$, and $z_{\mathcal{C}}$, such that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

Consider an arbitrary set $S \in (\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}) \setminus \{A_0\}$. Let z_S be the choice from set S, and let z_0 denote the choice from A_0. Note that $S \in \{A_1, \ldots, A_\ell\}$. Because we are given $\mathscr{D}_S(z_S) \cap \mathscr{D}_{A_0}(z_0) \neq \varnothing$ and $A_0 \subset S$, it must be that either $z_S = z_0$ or $z_S \in S \setminus A_0$. Now, consider a sub-collection $\mathcal{F} = \{S \in (\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}) \setminus \{A_0\} : z_S \neq z_0\}$. Define the ranking σ with $\{z_S : S \in \mathcal{F}\}$ as the top $|\mathcal{F}|$ products (the precise ordering does not matter) and with z_0 ranked at position $|\mathcal{F}| + 1$. It can be verified that σ is consistent with all the given choices $z_{\mathcal{A}}$, $z_{\mathcal{B}}$, and $z_{\mathcal{C}}$. This consistency implies that $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$, so $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$, which is the desired result.

A.4 Proof of Theorem 3.7

We prove the following slightly stronger result by induction: for any collections of subsets $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathtt{Del}_k$ such that $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$ in graph G_k, we have that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$ and all the choices are completely determined by a top-$(k+1)$ partial ranked list. The result is trivially true for $k = 0$ and $k = 1$.

Suppose that the result is true for $k - 1$. We will now show that it is also true for k. Without loss of generality, suppose that $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are disjoint, and $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$. We are given choices $z_{\mathcal{A}}$, $z_{\mathcal{B}}$, and $z_{\mathcal{C}}$ such that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. We want to show that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and there exists a top-$(k+1)$ partial ranked list such that all the choices are consistent with it.

Note that there is an edge (S_1, S_2) for *any* two subsets $S_1 \in \mathcal{F}_\ell$ and $S_2 \in \mathcal{F}_{\ell+1}$ for all $0 \leq \ell \leq k-1$. Therefore, for $\mathcal{C}$ to separate $\mathcal{A}$ from $\mathcal{B}$, it must be the case that $\mathcal{C}$ contains $\mathcal{F}_\ell$ for some ℓ. Define ℓ^* to be the "largest" such level, i.e., $\ell^* = \max\{\ell : \mathcal{F}_\ell \subseteq \mathcal{C}\}$. There are 3 cases to consider.

Case 1: $\ell^* = k$. In this case, we have that $\mathcal{F}_k \subseteq \mathcal{C}$. Since $z_{\mathcal{C}}$ is given, we are given the choices $z_{\mathcal{F}_k}$ for all subsets of size $n-k$. These choices completely determine the top-$(k+1)$ ranked elements, say, $(d_1, \ldots, d_{k+1})$, where d_i is the element ranked at position i. Let σ be any ranked list having $(d_1, \ldots, d_{k+1})$ as the top-$(k+1)$ ranked products. Because the top-$(k+1)$ ranked products also uniquely determine the choices from any set N_A such that $|A| \leq k$ and the choices $z_{\mathcal{A}}$ and $z_{\mathcal{C}}$ are consistent with each other, we must have that the choices $z_{\mathcal{A}}$ are also consistent with σ. This implies that $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$. Similarly, we must have that $\sigma \in \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$. Thus, we have shown that $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$, which implies that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$, which is the desired result.

Case 2: $\ell^* = k-1$. Let $\tilde{\mathcal{A}} = \mathcal{A} \setminus \mathcal{F}_k$, $\tilde{\mathcal{B}} = \mathcal{B} \setminus \mathcal{F}_k$, and $\tilde{\mathcal{C}} = \mathcal{C} \setminus \mathcal{F}_k$. Then, it must be the case that $\tilde{\mathcal{C}}$ separates $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ in graph G_{k-1}. Thus, it follows by the inductive hypothesis that $\tilde{\mathcal{A}} \perp\!\!\!\perp \tilde{\mathcal{B}} \mid \tilde{\mathcal{C}}$ and the choices $z_{\tilde{\mathcal{A}}}$, $z_{\tilde{\mathcal{B}}}$, and $z_{\tilde{\mathcal{C}}}$ are all consistent with a top-k partial ranked list $\sigma^{(k)} = (d_1, d_2, \ldots, d_k)$, where product d_i is ranked at position i.

We now extend $\sigma^{(k)}$ to a top-$(k+1)$ partial ranked list that is consistent with all of the choices. Pick an arbitrary set $S \in \mathcal{F}_k$ such that $S \in \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$. There are two sub-cases to consider:

Subcase 2.1: Suppose that $S \cap \{d_1, d_2, \ldots, d_k\} = \varnothing$, i.e., $S = N \setminus \{d_1, d_2, \ldots, d_k\}$. Since there is only one such subset S, we tackle it by extending $\sigma^{(k)}$ to obtain the top-$(k+1)$ partial list $(d_1, d_2, \ldots, d_k, z_S)$ by adding z_S as the $(k+1)^{\text{th}}$ element.

Subcase 2.2: Suppose that $S \cap \{d_1, d_2, \ldots, d_k\} \neq \varnothing$. This means that $S = N \setminus Q$ where $|Q| = k$ and $Q \neq \{d_1, d_2, \ldots, d_k\}$. In this case, we will show that $\sigma^{(k)}$ is already consistent with the choice z_S; that is, $z_S = d_{i^*}$ where $i^* = \min\{i : d_i \in S\}$.

To prove this, assume without loss of generality that $S \in \mathcal{A}$; the argument is the same if $S \in \mathcal{B}$ or $S \in \mathcal{C}$. Consider a product $j^* \in Q \setminus \{d_1, d_2, \ldots, d_k\}$. Note that j^* exists because Q and $\{d_1, \ldots, d_k\}$ have exactly k elements and are not equal. Because $S \in \mathcal{F}_k$ and $j^* \notin S$, we have that $S \cup \{j^*\} \in \mathcal{F}_{k-1} \subseteq \mathcal{C}$, so $z_{S \cup \{j^*\}}$ is given. It follows from our induction hypothesis that $\sigma^{(k)}$ is consistent with the top choice of the set $S \cup \{j^*\}$, which means that $z_{S \cup \{j^*\}} = d_{i^*}$ because d_{i^*} is the most preferred product in S under $\sigma^{(k)}$ and j^* is not part of $\sigma^{(k)}$.

Now, since the choices of $\mathcal{A}$ and $\mathcal{C}$ are consistent with each other because $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cup \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$, it must be the case that the choices $z_{S \cup \{j^*\}}$ and z_S must be consistent with each other. Furthermore, since $S \subset S \cup \{j^*\}$ and $z_{S \cup \{j^*\}} = d_{i^*} \in S$, dropping j^* does not alter the choice; that is, we must have $z_S = z_{S \cup \{j^*\}} = d_{i^*}$, which is the desired result.

Case 3: $\ell^* \le k-2$. In this case, it must be the case that in $\cup_{s=\ell^*}^k \mathcal{F}_s$ contains either sets in $\mathcal{A}$ or $\mathcal{B}$, but not both. Without loss of generality, assume that $\mathcal{B} \cap (\cup_{s=\ell^*}^k \mathcal{F}_s) = \varnothing$; the proof for the case where $\mathcal{A} \cap (\cup_{s=\ell^*}^k \mathcal{F}_s) = \varnothing$ is similar. This means that in levels $\ell^*, \ell^*+1, \ldots, k$, we will only find sets from $\mathcal{A}$ or $\mathcal{C}$.

Since $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ by our hypothesis, pick $\tau \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$. Let $z_{\mathcal{F}_{k-1}}$ be the most preferred product in $\mathcal{F}_{k-1}$ under τ. Let $\tilde{\mathcal{B}} = \mathcal{B} \setminus \mathcal{F}_k = \mathcal{B}$, $\tilde{\mathcal{C}} = \mathcal{C} \setminus \mathcal{F}_k$, and $\tilde{\mathcal{A}} = (\mathcal{A} \setminus \mathcal{F}_k) \cup (\mathcal{F}_{k-1} \setminus \mathcal{C})$. Note that $\tilde{\mathcal{A}}$, $\tilde{\mathcal{B}}$, and $\tilde{\mathcal{C}}$ are (disjoint) collection of subsets in $\cup_{s=0}^{k-1} \mathcal{F}_s$, and $\mathcal{F}_{k-1} \subset \tilde{\mathcal{A}} \cup \tilde{\mathcal{C}}$. Note that $z_{\tilde{\mathcal{A}}}$, $z_{\tilde{\mathcal{B}}}$, and $z_{\tilde{\mathcal{C}}}$ are fixed because $z_{\mathcal{F}_{k-1}}$ is determined from τ. Also, by our construction, we have that $\mathscr{D}_{\tilde{\mathcal{A}}}(z_{\tilde{\mathcal{A}}}) \cap \mathscr{D}_{\tilde{\mathcal{C}}}(z_{\tilde{\mathcal{C}}}) \neq \varnothing$ and $\mathscr{D}_{\tilde{\mathcal{B}}}(z_{\tilde{\mathcal{B}}}) \cap \mathscr{D}_{\tilde{\mathcal{C}}}(z_{\tilde{\mathcal{C}}}) \neq \varnothing$.

Moreover, we have that $\tilde{\mathcal{C}}$ separates $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ in the graph G_{k-1}. So, by the inductive hypothesis, we have that $\tilde{\mathcal{A}} \perp\!\!\!\perp \tilde{\mathcal{B}} \mid \tilde{\mathcal{C}}$ (so $\mathscr{D}_{\tilde{\mathcal{A}}}(z_{\tilde{\mathcal{A}}}) \cap \mathscr{D}_{\tilde{\mathcal{B}}}(z_{\tilde{\mathcal{B}}}) \cap \mathscr{D}_{\tilde{\mathcal{C}}}(z_{\tilde{\mathcal{C}}}) \neq \varnothing$) and there exists a top-k partial ranked list that is consistent with the choices $z_{\tilde{\mathcal{A}}}$, $z_{\tilde{\mathcal{B}}}$, and $z_{\tilde{\mathcal{C}}}$. Let us denote this top-k partial ranked list by $\sigma^{(k)} = (d_1, d_2, \ldots, d_k)$, so product d_1 is the most preferred product, d_2 is the second most, and so on.

As above, we now extend $\sigma^{(k)}$ to a top-$(k+1)$ partial ranked list that is consistent with all of the choices. Pick an arbitrary set $S \in \mathcal{F}_k$ such that $S \in \mathcal{A} \cup \mathcal{C}$. Note that $S \notin \mathcal{B}$ by our hypothesis. There are two sub-cases to consider:

Subcase 3.1: Suppose that $S \cap \{d_1, d_2, \ldots, d_k\} = \varnothing$, i.e., $S = N \setminus \{d_1, d_2, \ldots, d_k\}$. Again, note that there is only one such subset. As a result, we simply extend $\sigma^{(k)}$ to $(d_1, d_2, \ldots, d_k, z_S)$ by adding z_S as the $(k+1)^{\text{th}}$ element.

Subcase 3.2: Suppose that $S \cap \{d_1, d_2, \ldots, d_k\} \neq \varnothing$. This means that $S = N \setminus Q$ where $|Q| = k$ and $Q \neq \{d_1, d_2, \ldots, d_k\}$. In this case, we will show that $\sigma^{(k)}$ is already consistent with the choice z_S; that is, $z_S = d_{i^*}$ where $i^* = \min\{i : d_i \in S\}$.

The arguments are similar to the subcase 2.2 above. In particular, assume without loss of generality that $S \in \mathcal{A}$; the argument is the same if $S \in \mathcal{B}$ or $S \in \mathcal{C}$. Consider a product $j^* \in Q \setminus \{d_1, d_2, \ldots, d_k\}$. Note that j^* exists because Q and $\{d_1, \ldots, d_k\}$ have exactly k elements and are not equal. Because $S \in \mathcal{F}_k$ and $j^* \notin S$, we have that $S \cup \{j^*\} \in \mathcal{F}_{k-1} \subset \tilde{\mathcal{A}} \cup \tilde{\mathcal{C}}$, so $z_{S \cup \{j^*\}}$ is given. It follows from our induction hypothesis that $\sigma^{(k)}$ is consistent with the top choice of the set $S \cup \{j^*\}$, which means that $z_{S \cup \{j^*\}} = d_{i^*}$ because d_{i^*} is the most preferred product in S under $\sigma^{(k)}$ and j^* is not part of $\sigma^{(k)}$.

Now, note that by construction (of $z_{\mathcal{F}_{k-1}}$) above, the choices $z_{\mathcal{A}}$, $z_{\mathcal{C}}$, and $z_{\mathcal{F}_{k-1}}$ are consistent with each other. As a result, it must be the case that the choices $z_{S \cup \{j^*\}}$ (with $S \cup \{j^*\} \in \mathcal{F}_{k-1}$) and z_S (with $S \in \mathcal{A} \cup \mathcal{C}$) must be consistent with each other (in particular, with ranking τ). Furthermore, since $S \subset S \cup \{j^*\}$ and $z_{S\cup\{j^*\}} = d_{i^*} \in S$, dropping j^* does not alter the choice; that is, we must have $z_S = z_{S\cup\{j^*\}} = d_{i^*}$, which is the desired result.

The result now follows by induction.

Appendix B: Verification of Rational Separation

In this section, we show how to verify the rational separation property for any three collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. For any $z_{\mathcal{A}} = (z_S \in S : S \in \mathcal{A})$, $z_{\mathcal{B}} = (z_S \in S : S \in \mathcal{B})$, and $z_{\mathcal{C}} = (z_S \in S : S \in \mathcal{C})$, let the directed graph $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ be defined by:

$$\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}}) = \Big(N, \{(z_S, j) : S \in \mathcal{A}\cup\mathcal{B}\cup\mathcal{C}, j \in S, j \neq z_S\}\Big) ,$$

and thus, the set of nodes in $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ consists of all the products in N, and for each set $S \in \mathcal{A}\cup\mathcal{B}\cup\mathcal{C}$, we have a directed edge from z_S to all other products in S, indicating that z_S is preferred to all other products in S, i.e., z_S is the top ranked item in S. The following lemma shows that a consistent ranking exists if and only the graph has no directed cycle.

Lemma B.1 (Consistency Verification) *The graph $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ has no directed cycle if and only if $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. Moreover, verifying whether or not $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ can be done in linear time, with $O\left(n + \min\{n^2, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\}\right)$ operations*

Proof: By our construction, if there is a directed cycle in $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$, then it is clear that there is no consistent ranking, so $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) = \varnothing$. However, if there is no directed cycle, then it is easy to construct a consistent ranking, by first constructing a consistent ranking among $(z_S : S \in \mathcal{A}\cup\mathcal{B}\cup\mathcal{C})$, and putting all other products at the bottom of the ranking. This completes the first part of the lemma.

Note that a strongly connected component of $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ is a subgraph in which every vertex is reachable from every other vertex. Thus, $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ has no directed cycle if and only if every strongly connected component consists of a single vertex. The problem of finding all strongly connected components in a directed graph is a well-studied problem in computer science, and it can be solved in linear time (Tarjan 1972). Note that the number of edges in $\mathcal{H}(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ is $O(\min\{n^2, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\})$. So, verifying whether or not $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ can be done in linear time, using $O\left(n + \min\{n^2, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\}\right)$ operations. ■

The following corollary shows how to verify the rational separation property.

Corollary B.2 (Rational Separation Verification) *For any collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, we can verify whether or not $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$ using $O\Big(\big(\prod_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\big) \times \big(n + \min\{n^2\,,\, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\}\big) \Big)$*

Proof: To verify that $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$, for each $z_{\mathcal{A}} = (z_S \in S : S \in \mathcal{A})$, $z_{\mathcal{B}} = (z_S \in S : S \in \mathcal{B})$, and $z_{\mathcal{C}} = (z_S \in S : S \in \mathcal{C})$, we need whether or not thew following condition holds:

$$\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \,\cap\, \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \,\neq\, \varnothing \text{ and } \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing \quad\Longrightarrow\quad \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing\ .$$

By Lemma B.1, checking the above conditions takes a total of

$$O\left(\Big(n + \min\{n^2\,, \sum_{S\in\mathcal{A}\cup\mathcal{C}} |S|\}\Big) + \Big(n + \min\{n^2\,, \sum_{S\in\mathcal{B}\cup\mathcal{C}} |S|\}\Big) + \Big(n + \min\{n^2\,, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\}\Big) \right)$$

operations, which is $O\Big(n + \min\{n^2\,, \sum_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|\}\Big)$. To complete the proof, note that the number of possible combinations of $(z_{\mathcal{A}}, z_{\mathcal{B}}, z_{\mathcal{C}})$ is $\prod_{S\in\mathcal{A}\cup\mathcal{B}\cup\mathcal{C}} |S|$. ■

Appendix C: Proofs for Section 4

C.1 Proof of of the Computational Complexity of Dynamic Program for Tree Choice Graphs

We will establish that the DP recursion in Theorem 4.1 can be computed in linear time using just $O(n\,|\mathcal{M}|)$ operations. We will compute the value function at each vertex in a *breadth-first search* manner, starting at the leaf vertices. For each leaf vertex S, $V_S(z_S) = c_{z_S,S}$ for all $z_S \in S$. Once we compute the value function for every leaf vertex, we then move up one level. Consider an arbitrary vertex S whose set of children is denoted by $\mathsf{Children}(S)$. Computing the value function at S involves two steps:

Step 1: For each vertex $A \in \mathsf{Children}(S)$, compute two numbers: $\alpha_A \equiv \min_{z_A\in A} V_A(z_A)$ and $\beta_A \equiv \min_{z_A\in A\setminus S} V_A(z_A)$. Step 1 requires a total of $O(|\mathsf{Children}(S)|n)$ operations because for each $A \in \mathsf{Children}(S)$, we enumerate through all the products $z_A \in A$ to compute α_A and β_A.

Step 2: For each product $z_S \in S$, we compute $V_S(z_S)$ as follows:

$$\begin{aligned} V_S(z_S) &= c_{z_S,S} + \sum_{A\in\,\mathsf{Children}(S)} \ \min_{z_A\in A\,:\,\mathscr{D}_S(z_S)\cap\mathscr{D}_A(z_A)\neq\varnothing} V_A(z_A) \\ &= c_{z_S,S} + \sum_{A\in\,\mathsf{Children}(S)} \Big(\mathbb{1}[z_S \notin A]\alpha_A \quad+\quad \mathbb{1}[z_S\in A]\min\{\beta_A, V_A(z_S)\} \Big), \end{aligned}$$

where the second equality follows from noting that $\{z_A \in A \ :\ \mathscr{D}_S(z_S)\cap\mathscr{D}_A(z_A) \neq \varnothing\}$ is equal to A, if $z_S \notin A$, and $\{z_S\} \cup (A \setminus S)$, otherwise. Therefore,

$$\begin{aligned} \min_{z_A\in A\,:\,\mathscr{D}_A(z_A)\cap\mathscr{D}_S(z_S)\neq\varnothing} V_A(z_A) = &\quad \begin{cases} \min_{z_A\in A} V_A(z_A) & \text{if} \quad z_S\notin A, \\ \min\big\{V_A(z_S)\ ,\ \min_{z_A\in A\setminus S} V_A(z_A)\big\} & \text{if} \quad z_S \in A, \end{cases} \\ = &\quad \mathbb{1}[z_S\notin A]\alpha_A \quad+\quad \mathbb{1}[z_S\in A]\min\{\beta_A, V_A(z_S)\}. \end{aligned}$$

As we have already pre-computed α_A and β_A in step 1, for each $z_S \in S$, computing $V_S(z_S)$ takes $O(|\,\mathsf{Children}(S)|)$, so the entire step 2 requires $O(n|\,\mathsf{Children}(S)|)$ operations.

Therefore, the total computation for computing the value function at vertex S is $O(2n|\,\mathsf{Children}(S)|) = O(n|\,\mathsf{Children}(S)|)$. We continue this process until we reach the root vertex of the tree. Therefore, the total operations for computing the value function at every single vertex of the tree is equal to $O(n\sum_{S\in\mathcal{M}}|\,\mathsf{Children}(S)|) = O(n\,|\mathcal{M}|)$, which is the desired result.

C.2 Proof of Theorem 4.2

In this section, we give the proof of Theorem 4.2, showing that for tree choice graphs, we can reformulate the Rank Aggregation LP as an equivalent LP with just $O(n\,|\mathcal{M}|)$ variables and constraints. Recall from Lemma 2.5 that the feasible region of the Rank Aggregation LP is given by a polytope $\mathscr{M}$ defined as follows:

$$\mathscr{M} \equiv \text{ Convex hull of the set } \left\{ \left(\mathbb{1}[\sigma, i, S] : i \in S,\ S \in \mathcal{M}\right) \in \{0,1\}^{\sum_{S\in\mathcal{M}}|S|} \ : \ \sigma \in \mathscr{P}_n \right\} ,$$

and the optimization problem is equivalent to $Z^* = \min_{\boldsymbol{x}\in\mathscr{M}} \sum_{S\in\mathcal{M}} \sum_{i\in S} c_{i,S} x_{i,S}$.

Let a polytope $\mathscr{L}$ be defined by:

$$\mathscr{L} = \Bigg\{ \Big(y_{i,A} : i \in A,\ A \in \mathcal{M} \Big) \ \Bigg| \
\begin{array}{ll}
y_{r,A} \le z_{r,\{A,B\}} \quad \text{and} \quad y_{r,B} \le z_{r,\{A,B\}} & \forall\ r \in A \cap B,\ \{A,B\} \in \mathsf{E} \\
\sum_{i\in S} y_{i,S} = 1 & \forall\ S \in \mathcal{M} \\
\sum_{r\in A\cap B} z_{r,\{A,B\}} = 1 & \forall\ \{A,B\} \in \mathsf{E} \\
y_{i,A} \ge 0, \quad z_{r,\{A,B\}} \ge 0 & \forall\ i \in A,\ A \in \mathcal{M},\ r \in A\cap B,\ \{A,B\} \in \mathsf{E}
\end{array} \Bigg\},$$

where E denotes the edge set of the tree and we interpret $y_{i,A}$ as the probability of choosing i from offer set A and $z_{r,A\cap B}$ as the probability of choosing r when offered $A \cap B$. Note that the polytope $\mathscr{L}$ has $O(n\,|\mathcal{M}|)$ variables and $O(n\,|\mathsf{E}| + |\mathcal{M}|) = O(n\,|\mathcal{M}|)$ constraints. The main result of this section is stated in the following theorem; it shows that $\mathscr{M}$ is equivalent to $\mathscr{L}$.

Theorem C.1 (Compact Polytope) *When the choice graph $\mathsf{T} = (\mathcal{M}, \mathsf{E})$ is a tree, $\mathscr{M} = \mathscr{L}$.*

Note that Theorem 4.2 follows immediately from the above theorem. It is clear that $\mathscr{M} \subseteq \mathscr{L}$. We prove that $\mathscr{L} \subseteq \mathscr{M}$ by explicitly constructing a distribution over rankings that is consistent with the choice probabilities $y_{i,A}$ for all $i \in A$ and $A \in \mathcal{M}$. The details are provided in Appendix C. We conclude this section by describing the compact LPs for the nested, laminar, and differentiated collections introduced in Section 3.

LP for the nested collection (Section 3.1): Because the choice graph for a nested collection is the line graph (Theorem 3.4), it follows from Theorem 4.2 that the Rank Aggregation LP for the nested collection $\mathcal{M}=\{S_1,\ldots,S_\ell\}$ is equivalent to the following compact LP:

$$\min\left\{\sum_{h=1}^{\ell}\sum_{i\in S_h} c_{i,S_h}\, y_{i,S_h} \;\middle|\; y_{i,S_{h+1}} \le y_{i,S_h}\ \forall\ i\in S_h,\ 1\le h\le \ell-1,\ \sum_{i\in S_h} y_{i,S_h}=1\ \forall\ h,\ \boldsymbol{y}\ge 0\right\}.$$

Because $S_1\subset S_2\subset\cdots\subset S_\ell$, for any edge $\{S_h,S_{h+1}\}$ in the graph, $S_h\cap S_{h+1}=S_h$; therefore, we do not need additional variables $z_{r,A\cap B}$, and we can express the LP using only the variables y_{i,S_h}.

LP for the laminar collection (Section 3.2): Because the laminar collection has a choice graph $\mathsf{T}=(\mathcal{M},\mathsf{E})$ that is a disjoint collection of trees (Theorem 3.5), the Rank Aggregation LP reduces to the following compact LP:

$$\min\left\{\sum_{S\in\mathcal{M}}\sum_{i\in S} c_{i,S}\, y_{i,S} \;\middle|\; y_{r,A}\le y_{r,B},\ \forall\ \{A,B\}\in\mathsf{E}, B\subseteq A, r\in B,\ \sum_{i\in S} y_{i,S}=1\ \forall\ S\in\mathcal{M},\ \boldsymbol{y}\ge 0\right\}.$$

As in the nested collection setting, we do not need additional variables $z_{r,A\cap B}$ because if $A\cap B\neq\varnothing$, we have either $A\subset B$ or $B\subset A$.

LP for the differentiated collection (Section 3.3): Because the choice graph for the differentiated collection $\mathcal{M}=\{A_0,A_1,\ldots,A_\ell\}$ is star-shaped (Theorem 3.6) with edges from A_0 to A_i, $1\le i\le\ell$, where for $i=1,2,\ldots,\ell$ and $A_i\cap A_j=A_0$ for all $i\neq j$, the Rank Aggregation LP has the following compact representation:

$$\min\left\{\sum_{h=0}^{\ell}\sum_{i\in A_h} c_{i,A_h}\, y_{i,A_h} \;\middle|\; y_{r,A_h}\le y_{r,A_0}\ \forall\ r\in A_0,\ 1\le h\le\ell,\ \sum_{i\in A_h} y_{i,A_h}=1\ \ 0\le h\le\ell,\ \boldsymbol{y}\ge 0\right\}.$$

The above three examples demonstrate that the resulting LP is much more compact than the original Rank Aggregation LP.

The proof of Theorem C.1 is based on the induction on the number of vertices in the tree. The non-trivial base case is when the tree has two vertices; that is, when $\mathcal{M}=\{A,B\}$. The proof here is constructive and is given in the following lemma.

Lemma C.2 (Base Case of Two Vertices) *If $\mathcal{M}=\{A,B\}$, then $\mathscr{M}=\mathscr{L}$.*

Proof: Without loss of generality, we assume that $A\cap B\neq\varnothing$; otherwise, the result is trivially true. When $\mathcal{M}=\{A,B\}$, note that

$$\mathscr{L}=\Big\{(y_{i,S}\ :\ i\in S,\ S\in\mathcal{M})\ \Big|\ y_{r,A}\le z_{r,\{A,B\}}\quad\text{and}\quad y_{r,B}\le z_{r,\{A,B\}},\quad \forall\ r\in A\cap B,$$
$$\sum_{i\in A} y_{i,A}=1,\ \sum_{i\in B} y_{i,B}=1,\ \sum_{r\in A\cap B} z_{r,\{A,B\}}=1,\ \boldsymbol{y},\boldsymbol{z}\ge 0\Big\}$$

We will first show that $\mathscr{M} \subseteq \mathscr{L}$. By definition, for an arbitrary $\boldsymbol{x} \in \mathscr{M}$, $x_{i,S}$ is of the form

$$x_{i,S} = \sum_{\sigma} \tau(\sigma) \cdot \mathbb{1}[\sigma, i, S] \ ,$$

where $\tau : \mathscr{P}_n \to [0,1]$ is a probability mass function over the set of rankings, with $\sum_\sigma \tau(\sigma) = 1$ and $\tau(\sigma) \geq 0$ for all σ. For all $i \in A$, let $y_{i,A} = \sum_\sigma \tau(\sigma) \cdot \mathbb{1}[\sigma, i, A]$, and for all $i \in B$, $y_{i,B} = \sum_\sigma \tau(\sigma) \cdot \mathbb{1}[\sigma, i, B]$. Also, for each $r \in A \cap B$, let

$$z_{r,\{A,B\}} = \sum_{\sigma} \tau(\sigma) \cdot \mathbb{1}[\sigma, r, A \cap B]$$

It is easy to check that $y_{i,A}$, $y_{i,B}$, and $z_{r,\{A,B\}}$ satisfies all of the constraints in $\mathscr{L}$ because $\max\{\mathbb{1}[\sigma, r, A], \mathbb{1}[\sigma, r, B]\} \leq \mathbb{1}[\sigma, r, A \cap B]$ for any $r \in A \cap B$. Therefore, $\mathscr{M} \subseteq \mathscr{L}$.

We now prove that $\mathscr{L} \subseteq \mathscr{M}$. Consider an arbitrary element $(\boldsymbol{y}, \boldsymbol{z}) \in \mathscr{L}$. For each $i \in A \setminus B$, $r \in A \cap B$, and $j \in B \setminus A$, define the following subset of rankings:

$$\begin{aligned}
\mathscr{S}_1(r) &= \mathscr{D}_A(r) \cap \mathscr{D}_{A \cap B}(r) \cap \mathscr{D}_B(r) \\
\mathscr{S}_2(i,r) &= \mathscr{D}_A(i) \cap \mathscr{D}_{A \cap B}(r) \cap \mathscr{D}_B(r) \\
\mathscr{S}_3(j,r) &= \mathscr{D}_A(r) \cap \mathscr{D}_{A \cap B}(r) \cap \mathscr{D}_B(j) \\
\mathscr{S}_4(i,j,r) &= \mathscr{D}_A(i) \cap \mathscr{D}_{A \cap B}(r) \cap \mathscr{D}_B(j)
\end{aligned}$$

Note that $\mathscr{S}_1(r)$ is the set of all rankings for which $r \in A \cap B$ is the most preferred product in $A \cup B$; $\mathscr{S}_2(i,r)$ is the set of rankings for which $i \in A \setminus B$ is the most preferred product in $A \cup B$, followed by r as the second most preferred product; $\mathscr{S}_3(j,r)$ is the set of rankings for which $j \in B \setminus A$ is the most preferred product in $A \cup B$, followed by r as the second most preferred product; and finally, $\mathscr{S}_4(i,j,r)$ is the set of rankings for which i and j are the top two most preferred product in $A \cup B$, followed by r as the third most preferred product. By our construction, these four sets are disjoint. Then, define a distribution $\lambda(\cdot)$ over rankings as follow:

$$\lambda(\sigma) = \begin{cases}
\frac{1}{|\mathscr{S}_1(r)|} \cdot (y_{r,A} y_{r,B}) / z_{r,\{A,B\}} & \text{if } \sigma \in \mathscr{S}_1(r) \text{ for some } r, \\[2ex]
\frac{1}{|\mathscr{S}_2(i,r)|} \cdot \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B} y_{i,A} / z_{r,\{A,B\}}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} & \text{if } \sigma \in \mathscr{S}_2(i,r) \text{ for some } i, r, \\[2ex]
\frac{1}{|\mathscr{S}_3(j,r)|} \cdot \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A} y_{j,B} / z_{r,\{A,B\}}}{\sum_{k \in B \setminus A} y_{k,B}} & \text{if } \sigma \in \mathscr{S}_3(j,r) \text{ for some } j, r, \\[2ex]
\frac{1}{|\mathscr{S}_4(j,r)|} \cdot \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B}) y_{i,A} \, y_{j,B} / z_{r,\{A,B\}}}{\left(\sum_{\ell \in A \setminus B} y_{\ell,A}\right)\left(\sum_{k \in B \setminus A} y_{k,B}\right)} & \text{if } \sigma \in \mathscr{S}_4(i,j,r) \text{ for some } i, j, r, \\[2ex]
0 & \text{otherwise.}
\end{cases}$$

We will first show that $\lambda(\cdot)$ is a bonafide probability mass function. By the constraints of $\mathscr{L}$, $\lambda(\cdot)$ is non-negative, so, we will now show that it sums to one. Fix $r \in A \cap B$. Then,

$$\sum_{i \in A \setminus B} \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B} y_{i,A} / z_{r,\{A,B\}}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} = \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B}}{z_{r,\{A,B\}}} \tag{3}$$

$$\sum_{j \in B \setminus A} \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A} y_{j,B} / z_{r,\{A,B\}}}{\sum_{k \in B \setminus A} y_{k,B}} = \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A}}{z_{r,\{A,B\}}} \tag{4}$$

$$\sum_{i \in A \setminus B} \sum_{j \in B \setminus A} \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B}) y_{i,A} y_{j,B} / z_{r,\{A,B\}}}{\left(\sum_{\ell \in A \setminus B} y_{\ell,A}\right)\left(\sum_{k \in B \setminus A} y_{k,B}\right)} = \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B})}{z_{r,\{A,B\}}}$$

Therefore,

$$\begin{aligned}
&\sum_{\sigma} \lambda(\sigma) \\
&= \sum_{r \in A \cap B} \left\{ \frac{y_{r,A} y_{r,B}}{z_{r,\{A,B\}}} + \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B}}{z_{r,\{A,B\}}} + \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A}}{z_{r,\{A,B\}}} + \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B})}{z_{r,\{A,B\}}} \right\} \\
&= \sum_{r \in A \cap B} z_{r,\{A,B\}} \; = \; 1 \; ,
\end{aligned}$$

where the last inequality follows from the constraint in $\mathscr{L}$.

To complete the proof, we will show that $\lambda(\cdot)$ has the desired choice probabilities. Note that for any $i \in A \setminus B$,

$$\begin{aligned}
\sum_{\sigma} \lambda(\sigma) \cdot \mathbb{1}[\sigma, i, A] &= \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B} y_{i,A} / z_{r,\{A,B\}}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} \\
&\quad + \sum_{j \in B \setminus A} \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B}) y_{i,A} y_{j,B} / z_{r,\{A,B\}}}{\left(\sum_{\ell \in A \setminus B} y_{\ell,A}\right)\left(\sum_{k \in B \setminus A} y_{k,B}\right)} \\
&\overset{\text{from (4)}}{=} \frac{y_{i,A}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,A}) y_{r,B} + (z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B})}{z_{r,\{A,B\}}} \\
&= \frac{y_{i,A}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} \sum_{r \in A \cap B} \left(z_{r,\{A,B\}} - y_{r,A}\right) \\
&= \frac{y_{i,A}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} \left(1 - \sum_{r \in A \cap B} y_{r,A}\right) = \frac{y_{i,A}}{\sum_{\ell \in A \setminus B} y_{\ell,A}} \left(\sum_{r \in A \setminus B} y_{r,A}\right) = y_{i,A} \; ,
\end{aligned}$$

where the last two equalities follow from the fact that $\sum_{r \in A \cap B} z_{r,\{A,B\}} = 1$ and $\sum_{r \in A \cap B} y_{r,A} + \sum_{r \in A \setminus B} y_{r,A} = 1$. Similarly, for any $j \in B \setminus A$,

$$\begin{aligned}
\sum_{\sigma} \lambda(\sigma) \mathbb{1}[\sigma, j, B] &= \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A} y_{j,B} / z_{r,\{A,B\}}}{\sum_{k \in B \setminus A} y_{k,B}} \\
&\quad + \sum_{i \in A \setminus B} \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B}) y_{i,A} y_{j,B} / z_{r,\{A,B\}}}{\left(\sum_{\ell \in A \setminus B} y_{\ell,A}\right)\left(\sum_{k \in B \setminus A} y_{k,B}\right)} \\
&\overset{\text{from (3)}}{=} \frac{y_{j,B}}{\sum_{k \in B \setminus A} y_{k,B}} \sum_{r \in A \cap B} \frac{(z_{r,\{A,B\}} - y_{r,B}) y_{r,A} + (z_{r,\{A,B\}} - y_{r,A})(z_{r,\{A,B\}} - y_{r,B})}{z_{r,\{A,B\}}}
\end{aligned}$$

$$= \frac{y_{j,B}}{\sum_{k\in B\setminus A} y_{k,B}} \sum_{r\in A\cap B} \left(z_{r,\{A,B\}} - y_{r,B}\right)$$

$$= \frac{y_{j,B}}{\sum_{k\in B\setminus A} y_{k,B}} \left(1 - \sum_{r\in A\cap B} y_{r,B}\right) = \frac{y_{j,B}}{\sum_{k\in B\setminus A} y_{k,B}} \left(\sum_{r\in B\setminus A} y_{r,B}\right) = y_{j,B} \ ,$$

which is the desired result. Finally, for any $r \in A \cap B$,

$$\sum_{\sigma} \lambda(\sigma)\mathbb{1}[\sigma, r, A] = \frac{y_{r,A}y_{r,B}}{z_{r,\{A,B\}}} + \sum_{j\in B\setminus A} \frac{(z_{r,\{A,B\}} - y_{r,B})y_{r,A}y_{j,B}/z_{r,\{A,B\}}}{\sum_{k\in B\setminus A} y_{k,B}} = y_{r,A} \ ,$$

and

$$\sum_{\sigma} \lambda(\sigma)\mathbb{1}[\sigma, r, B] = \frac{y_{r,A}y_{r,B}}{z_{r,\{A,B\}}} + \sum_{i\in A\setminus B} \frac{(z_{r,\{A,B\}} - y_{r,A})y_{i,A}y_{r,B}/z_{r,\{A,B\}}}{\sum_{\ell\in A\setminus B} y_{\ell,A}} = y_{r,B} \ ,$$

which completes the proof. ■

Using the result of Lemma C.2, we now prove Theorem C.1

Proof of Theorem C.1: We prove the result by induction on the number of vertices $|\mathcal{M}|$ in the tree. Without loss of generality, assume that T is connected; otherwise, we can treat each connected component separately. When $|\mathcal{M}| = 1$, the result is trivially true. The case where $|\mathcal{M}| = 2$ is established in Lemma C.2. Suppose the result is true for all trees with $1, 2, \ldots, k-1$ vertices. Consider the case when $|\mathcal{M}| = k > 2$. Pick a vertex in T with at least two children; such a vertex exists because $|\mathcal{M}| > 2$ and T is connected[6]. We label this vertex as the root vertex. Note that $\mathsf{root} \in \mathcal{M}$. Let $A^1, \ldots, A^h$ denote the children of root in the tree, with $h \geq 2$. For $j = 1, \ldots, h$, let $\mathsf{T}(A^j)$ denote the sub-tree rooted at A^j excluding the vertex A^j; equivalently, $\mathcal{A}^j = \{A^j\} \cup \mathsf{T}(A^j)$ denotes the sub-tree rooted at A^j. Since we have a tree, the collections $\mathcal{A}^1, \ldots, \mathcal{A}^h$ are mutually disjoint; that is, $\mathcal{A}^{j_1} \cap \mathcal{A}^{j_2} = \varnothing$ for all $j_1 \neq j_2$.

We want to show that $\mathscr{M} = \mathscr{L}$. It is clear that $\mathscr{M} \subseteq \mathscr{L}$. To show that $\mathscr{L} \subseteq \mathscr{M}$, we will show that for every feasible solution $\left(y_{i,S},\ z_{r,\{S,T\}} : i \in S \in \mathcal{M},\ r \in S \cap T,\ \{S,T\} \in \mathsf{E}\right) \in \mathscr{L}$, there exists a distribution $\lambda(\cdot)$ over rankings that matches the choice probabilities $y_{i,S}$ for all $i \in S \in \mathcal{M}$. By our inductive hypothesis, since $|\{\mathsf{root}\} \uplus \mathcal{A}^j| \leq k-1$, there exists a probability mass function $\lambda^j(\cdot)$ over rankings that matches the choice probabilities of every set in $\{\mathsf{root}\} \uplus \mathcal{A}^j$; that is, for all $S \in \{\mathsf{root}\} \uplus \mathcal{A}^j$ and $i \in S$,

$$\sum_{\sigma\in\mathscr{P}_n} \lambda^j(\sigma) \cdot \mathbb{1}[\sigma, i, S] = y_{i,S} \qquad j = 1, 2, \ldots, h \ . \tag{5}$$

To complete the result, we will combine the distributions $\lambda^1(\cdot), \ldots, \lambda^h(\cdot)$ to create a probability distribution $\lambda(\cdot)$ over all rankings with the desired property. For $j = 1, 2, \ldots, h$ $\vec{\boldsymbol{a}}^j =$

[6] For example, if T is a line graph with three nodes $S_1 - S_2 - S_3$, then S_2 has two children.

$\left(a_B^j \in B \ : \ B \in \mathcal{A}^j\right) \ \in \ \bigtimes_{B\in\mathcal{A}^j} B$ denote a vector of elements, each coming from a set in $\mathcal{A}^j$. Then, for all $\gamma \in \mathsf{root}$ and $\vec{\boldsymbol{a}}^j \in \bigtimes_{B\in\mathcal{A}^j} B$, let

$$w^j\left(\gamma, \vec{\boldsymbol{a}}^j\right) = \sum_\sigma \lambda^j(\sigma)\cdot \mathbb{1}[\sigma,\gamma,\mathsf{root}]\cdot \prod_{B\in\mathcal{A}^j} \mathbb{1}[\sigma, a_B^j, B] = \sum_{\sigma\in\mathscr{D}_{\mathsf{root}}(\gamma)\cap\mathscr{D}_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right)} \lambda^j(\sigma)\ ,$$

where, by definition 3.1, $\mathscr{D}_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right) = \cap_{B\in\mathcal{A}^j}\mathscr{D}_B\left(a^j\right)$. Note that $w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)$ denotes the mass under $\lambda^j(\cdot)$ of all rankings where γ is the most preferred product in root and for every $B\in\mathcal{A}^j$, a_B^j is the most preferred product in B; $w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)$ may be zero. Also, by our construction,

$$w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right) \le \sum_\sigma \lambda^j(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}] = y_{\gamma,\mathsf{root}}.$$

For all $\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right) \in \mathsf{root}\times(\bigtimes_{B\in\mathcal{A}^1}B)\times\cdots\times(\bigtimes_{B\in\mathcal{A}^h}B)$, let

$$\mathscr{P}(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h) = \mathscr{D}_{\mathsf{root}}(\gamma)\cap\left(\cap_{j=1}^h \mathscr{D}_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right)\right)$$

Also, let

$$w(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h) = \frac{\prod_{j=1}^h w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}$$

where we define $0/0$ to be 0. This is well-defined because we have shown that $w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)\le y_{\gamma,\mathsf{root}}$ for all j, so whenever the denominator is zero, so is the numerator. We also note that whenever $\mathscr{P}(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)=\varnothing$, we must have that $w(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)=0$. To see this, consider $(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)$ such that $\mathscr{P}(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)=\varnothing$. Then, we must have that $\mathscr{D}_{\mathsf{root}}(\gamma)\cap\mathscr{D}_{\mathcal{A}^j}=\varnothing$ for some $1\le j\le h$; otherwise, it will follow from the rational separation of $\mathcal{A}^1,\mathcal{A}^2,\ldots,\mathcal{A}^h$ by root that $\mathscr{D}_{\mathsf{root}}(\gamma)\cap\mathscr{D}_{\mathcal{A}^i}\ne\varnothing$ for all $1\le i\le h$ implies that $\mathscr{P}(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)\ne\varnothing$. It now follows by definition that $\mathscr{D}_{\mathsf{root}}(\gamma)\cap\mathscr{D}_{\mathcal{A}^j}=\varnothing$ implies that $w^j(\gamma,\vec{a}^j)=0$, which in turn implies that $w(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)=0$. Now define the function $\lambda:\mathscr{P}_n\to[0,1]$ as follows:

$$\lambda(\sigma) = \begin{cases} \frac{w(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h)}{\left|\mathscr{P}\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)\right|} & \text{if } \sigma\in\mathscr{P}\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right) \text{ for some } (\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h) \\ \\ 0 & \text{otherwise} \end{cases}$$

We will show that $\lambda(\cdot)$ is a probability mass function over the rankings with the desired property. Note that $\lambda(\sigma)$ is well-defined because it follows by our inductive hypothesis that for each j, there exist γ and a vector $\vec{\boldsymbol{a}}^j$ such that $D_{\mathsf{root}}(\gamma)\cap D_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right)\ne\varnothing$. It then follows by the fact that root separates the collections $\mathcal{A}^j$ for all j, whenever $D_{\mathsf{root}}(\gamma)\cap D_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right)\ne\varnothing$ for all j, it must be that $\mathscr{P}(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h) = D_{\mathsf{root}}(\gamma)\cap\left(\cap_{j=1}^h D_{\mathcal{A}^j}\left(\vec{\boldsymbol{a}}^j\right)\right)\ne\varnothing$ by rational separation. Moreover, by definition, $\mathscr{P}\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)\cap\mathscr{P}\left(\tilde{\gamma},\vec{\tilde{\boldsymbol{a}}}^1,\ldots,\vec{\tilde{\boldsymbol{a}}}^h\right)=\varnothing$ unless $\gamma=\tilde{\gamma}$, and $\vec{\boldsymbol{a}}^j=\vec{\tilde{\boldsymbol{a}}}^j$ for all j. So, for each σ belongs to at most a single set $\mathscr{P}\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)$. Therefore, $\lambda(\cdot)$ is well-defined.

We will now show that $\lambda(\cdot)$ is a valid probability mass function over rankings. Note that

$$\begin{aligned}\sum_{\sigma}\lambda(\sigma)&=\sum_{\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)} w\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)=\sum_{\gamma\in\mathsf{root}}\frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}\sum_{\vec{\boldsymbol{a}}^1}\cdots\sum_{\vec{\boldsymbol{a}}^h}\prod_{j=1}^{h} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)\\&=\sum_{\gamma\in\mathsf{root}}\frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}\prod_{j=1}^{h}\left(\sum_{\vec{\boldsymbol{a}}^j} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)\right),\end{aligned}$$

where the second equality follows by the interchange of sum and product. By definition, for all $j=1,\ldots,h$, we have

$$\begin{aligned}\sum_{\vec{\boldsymbol{a}}^j} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)&=\sum_{\vec{\boldsymbol{a}}^j}\sum_{\sigma}\lambda^j(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}]\prod_{B\in\mathcal{A}^j}\mathbb{1}[\sigma,a_B^j,B]\\&=\sum_{\sigma}\lambda^j(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}]\sum_{\vec{\boldsymbol{a}}^j}\prod_{B\in\mathcal{A}^j}\mathbb{1}[\sigma,a_B^j,B]\\&=\sum_{\sigma}\lambda^j(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}]\prod_{B\in\mathcal{A}^j}\sum_{a_B\in B}\mathbb{1}[\sigma,a_B^j,B]\\&=\sum_{\sigma}\lambda^j(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}]=y_{\gamma,\mathsf{root}}\ ,\end{aligned}$$

and thus,

$$\sum_{\sigma}\lambda(\sigma)=\sum_{\gamma\in\mathsf{root}}\frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}\left(y_{\gamma,\mathsf{root}}\right)^{h}=\sum_{\gamma\in\mathsf{root}}y_{\gamma,\mathsf{root}}=1\ ,$$

where the last equality follows from the fact that $\boldsymbol{y}\in\mathscr{L}$. Thus, $\lambda(\cdot)$ is a bonafide probability mass function over the rankings.

We now show that $\lambda(\cdot)$ has the desired choice probability. For any $\gamma\in\mathsf{root}$,

$$\begin{aligned}\sum_{\sigma}\lambda(\sigma)\mathbb{1}[\sigma,\gamma,\mathsf{root}]&=\sum_{\vec{\boldsymbol{a}}^1}\cdots\sum_{\vec{\boldsymbol{a}}^h} w\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)\\&=\frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}\sum_{\vec{\boldsymbol{a}}^1}\cdots\sum_{\vec{\boldsymbol{a}}^h}\prod_{j=1}^{h} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)\\&=\frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}}\prod_{j=1}^{h}\left(\sum_{\vec{\boldsymbol{a}}^j} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)\right)\quad=\quad y_{\gamma,\mathsf{root}}\ ,\end{aligned}$$

where the last inequality follows from the above argument that shows $\sum_{\vec{\boldsymbol{a}}^j} w^j\left(\gamma,\vec{\boldsymbol{a}}^j\right)=y_{\gamma,\mathsf{root}}$ for all j.

Similarly, consider an arbitrary set $S\in\mathcal{A}^\ell$ for some ℓ and an element $s\in S$. Since T is a tree, it follows that $S\notin\mathcal{A}^j$ for $j\neq\ell$. Then,

$$\sum_{\sigma}\lambda(\sigma)\mathbb{1}[\sigma,s,S]=\sum_{\gamma\in\mathsf{root}}\sum_{\vec{\boldsymbol{a}}^1}\cdots\sum_{\vec{\boldsymbol{a}}^{\ell-1}}\sum_{\vec{\boldsymbol{a}}^\ell:a_S^\ell=s}\sum_{\vec{\boldsymbol{a}}^{\ell+1}}\cdots\sum_{\vec{\boldsymbol{a}}^h} w\left(\gamma,\vec{\boldsymbol{a}}^1,\ldots,\vec{\boldsymbol{a}}^h\right)$$

$$= \sum_{\gamma \in \mathsf{root}} \frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}} \sum_{\vec{\boldsymbol{a}}^1} \cdots \sum_{\vec{\boldsymbol{a}}^{\ell-1}} \sum_{\vec{\boldsymbol{a}}^\ell : a_S^\ell = s} \sum_{\vec{\boldsymbol{a}}^{\ell+1}} \cdots \sum_{\vec{\boldsymbol{a}}^h} \prod_{j=1}^{h} w^j\left(\gamma, \vec{\boldsymbol{a}}^j\right)$$

$$= \sum_{\gamma \in \mathsf{root}} \frac{1}{\left(y_{\gamma,\mathsf{root}}\right)^{h-1}} \times \left(\sum_{\vec{\boldsymbol{a}}^\ell : a_S^\ell = s} w^\ell\left(\gamma, \vec{\boldsymbol{a}}^\ell\right) \right) \times \prod_{j=1,\ldots,h \;:\; j \neq \ell} \left(\sum_{\vec{\boldsymbol{a}}^j} w^j\left(\gamma, \vec{\boldsymbol{a}}^j\right) \right)$$

$$= \sum_{\gamma \in \mathsf{root}} \sum_{\vec{\boldsymbol{a}}^\ell : a_S^\ell = s} w^\ell\left(\gamma, \vec{\boldsymbol{a}}^\ell\right) \,.$$

By definition,

$$\sum_{\gamma \in \mathsf{root}} \sum_{\vec{\boldsymbol{a}}^\ell : a_S^\ell = s} w^\ell\left(\gamma, \vec{\boldsymbol{a}}^\ell\right) = \sum_{\gamma \in \mathsf{root}} \sum_{\vec{\boldsymbol{a}}^\ell : a_S^\ell = s} \sum_{\sigma} \lambda^\ell(\sigma) \mathbb{1}[\sigma, \gamma, \mathsf{root}] \times \mathbb{1}[\sigma, s, S] \times \prod_{B \in \mathcal{A}^\ell \,:\, B \neq S} \mathbb{1}[\sigma, a_B^j, B]$$

$$= \sum_{\sigma} \lambda^\ell(\sigma) \left(\sum_{\gamma \in \mathsf{root}} \mathbb{1}[\sigma, \gamma, \mathsf{root}] \right) \times \mathbb{1}[\sigma, s, S] \times \prod_{B \in \mathcal{A}^\ell \,:\, B \neq S} \left(\sum_{a_B^j \in B} \mathbb{1}[\sigma, a_B^j, B] \right)$$

$$= \sum_{\sigma} \lambda^\ell(\sigma) \mathbb{1}[\sigma, s, S] = y_{s,S} \;,$$

where the last equality follows from our inductive assumption that $\lambda^\ell(\cdot)$ matches the choice probabilities of every set in $\{\mathsf{root}\} \uplus \mathcal{A}^\ell$.

The result of the theorem now follows by induction. ∎

Appendix D: Proofs for Section 5

D.1 Proof of Theorem 5.4

The proof is similar to that for the case when the choice graph is a tree in that we formulate the Rank Aggregation LP as a DP on the tree decomposition of the choice graph.

Let T_G be the tree decomposition of the graph G with a collection of bags $\{\mathcal{X}_b \,:\, b \in \mathsf{B}\}$, with $|\mathsf{B}| \leq |\mathcal{M}|$ by Lemma 5.3. Let $\mathcal{X}_{\mathsf{root}}$ be a root node in T_G and $\mathcal{X}_b^-$ denote the collection $\mathcal{X}_b \setminus \mathcal{X}_{\mathsf{Parent}(\mathcal{X}_b)}$. With the root node fixed, the parent of a node is uniquely defined; we suppose that the parent of the root node is the empty set. Also, let $\mathsf{T}(\mathcal{X}_b)$ denote the sub-tree rooted at $\mathcal{X}_b$ but not including $\mathcal{X}_b$. Also, for all $z_{\mathcal{X}_b} = (z_S : S \in \mathcal{X}_b) \in \times_{S \in \mathcal{X}_b} S$, let $\mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) = \cap_{S \in \mathcal{X}_b} \mathscr{D}_S(z_S)$ and $c_{z_{\mathcal{X}_b}, \mathcal{X}_b} = \sum_{S \in \mathcal{X}_b^-} c_{z_S, S}$.

For each bag $\mathcal{X}_b$, we define a value function $J : \times_{S \in \mathcal{X}_b} S \to \mathbb{R}$ as follows: for all $z_{\mathcal{X}_b} = (z_S : S \in \mathcal{X}_b) \in \times_{S \in \mathcal{X}_b} S$ such that $\mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) = \cap_{S \in \mathcal{X}_b} \mathscr{D}_S(z_S) \neq \varnothing$,

$$J_{\mathcal{X}_b}\left(z_{\mathcal{X}_b}\right) = c_{z_{\mathcal{X}_b}, \mathcal{X}_b} + \min \left\{ \sum_{\mathcal{X}_a \in \mathsf{T}(\mathcal{X}_b)} c_{z_{\mathcal{X}_a}, \mathcal{X}_a} \;:\; \left(z_{\mathcal{X}_a} \,:\, \mathcal{X}_a \in \mathsf{T}(\mathcal{X}_b)\right) \right.$$
$$\left. \text{and } \mathscr{D}_{\mathcal{X}_b}\left(z_{\mathcal{X}_b}\right) \cap \left(\cap_{\mathcal{X}_a \in \mathsf{T}(\mathcal{X}_b)} \mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a})\right) \neq \varnothing \right\}.$$

We first claim that $Z^* = \min_{z_{\mathcal{X}_{\mathsf{root}}}} J(z_{\mathcal{X}_{\mathsf{root}}})$. To see this, note that $\mathcal{X}_{\mathsf{root}} \cup \left(\cup_{\mathcal{X}\in\mathsf{T}(\mathcal{X}_{\mathsf{root}})}\mathcal{X}\right) = \mathcal{M}$. Therefore, it follows from our definitions that $\mathscr{D}_{\mathcal{X}_{\mathsf{root}}}(z_{\mathcal{X}_{\mathsf{root}}}) \cap \left(\cap_{\mathcal{X}_a\in\mathsf{T}(\mathcal{X}_{\mathsf{root}})}\mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a})\right) = \cap_{S\in\mathcal{M}}\mathscr{D}_S(z_S)$.

It is now sufficient to show that $c_{z_{\mathcal{X}_{\mathsf{root}}},\mathcal{X}_{\mathsf{root}}} + \sum_{\mathcal{X}_a\in\mathsf{T}(\mathcal{X}_{\mathsf{root}})} c_{z_{\mathcal{X}_a},\mathcal{X}_a} = \sum_{S\in\mathcal{M}} c_{z_S,S}$. We show this by establishing that $\mathcal{X}_b^-$ for all $b\in\mathsf{B}$ forms a partition of $\mathcal{M}$. We first argue that $\mathcal{X}_b^- \cap \mathcal{X}_a^- = \varnothing$ for $a,b\in\mathsf{B}$ and $a\neq b$. Otherwise, there exists an S that is contained in $\mathcal{X}_b^-$ and $\mathcal{X}_a^-$. It follows from our definitions that S is also contained in $\mathcal{X}_b$ and $\mathcal{X}_a$. It now follows from the running intersection property of Definition 5.1 that every $\mathcal{X}$ on the path from $\mathcal{X}_b$ to $\mathcal{X}_a$ in T_G also contains S. Because every path from $\mathcal{X}_b$ to $\mathcal{X}_a$ should contain both the parents $\mathsf{Parent}(\mathcal{X}_b)$ and $\mathsf{Parent}(\mathcal{X}_a)$, it follows that $S\in\mathsf{Parent}(\mathcal{X}_b)$ and $S\in\mathsf{Parent}(\mathcal{X}_a)$. We have thus established that S must be contained by in $\mathcal{X}_b$, $\mathcal{X}_a$, $\mathsf{Parent}(\mathcal{X}_b)$, and $\mathsf{Parent}(\mathcal{X}_a)$. This contradicts the assumption that $S\in\mathcal{X}_b^- = \mathcal{X}_b\setminus\mathsf{Parent}(\mathcal{X}_b)$.

It now suffices to show that for every $S\in\mathcal{M}$, there exists a $b\in\mathsf{B}$ such that $S\in\mathcal{X}_b^-$. But this immediately follows from noting that there is at least one $b\in\mathsf{B}$ such that $S\in\mathcal{X}_b$ and by taking $b^*\in\mathsf{B}$ to be such that $S\in\mathcal{X}_{b^*}$ but $S\notin\mathsf{Parent}(\mathcal{X}_{b^*})$, we have that $S\in\mathcal{X}_{b^*}^-$; note that such a b^* must always exist.

We now show how we can exploit the rational separation property to solve the DP efficiently. For that, we follow the arguments in the proof of Theorem 4.1 to express the $J_{\mathcal{X}_b}(\cdot)$ as the following DP recursion:

$$J_{\mathcal{X}_b}(z_{\mathcal{X}_b}) = c_{z_{\mathcal{X}_b},\mathcal{X}_b} + \sum_{\mathcal{X}_a\in\mathsf{Children}(\mathcal{X}_b)} \min\left\{J_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \;:\; \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \neq \varnothing\right\}$$

The above dynamic programming allows us to recursively compute the value function at each bag in the tree decomposition, starting with the bags at the leaves and moving up the tree in a breadth first fashion. In computing the value function at $\mathcal{X}_b$, suppose that for each value of $z_{\mathcal{X}_b}$, the values $V_a(\mathcal{X}_b) := \min\left\{J_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \;:\; \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \neq \varnothing\right\}$ have been pre-computed. Then, to compute the value function for $J_{\mathcal{X}_b}(\cdot)$ at the bag $\mathcal{X}_b$, we need to perform $|\mathsf{Children}(\mathcal{X}_b)|$ additions for each state. Since there are $O(n^{\mathsf{tw}(\mathsf{G})+1})$ total number of states, it follows that computing the value function at bag $\mathcal{X}_b$ requires $|\mathsf{Children}(\mathcal{X}_b)|\, n^{\mathsf{tw}(\mathsf{G})+1}$ operations. Now, to pre-compute $V_b(\mathcal{X}_{\mathsf{Parent}(\mathcal{X}_b)})$, we need to search over all possible combinations $z_{\mathcal{X}_b}$ and $z_{\mathsf{Parent}(\mathcal{X}_b)}$ such that $\mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathsf{Parent}(\mathcal{X}_b)}(z_{\mathsf{Parent}(\mathcal{X}_b)}) \neq \varnothing$. For a given combination of $z_{\mathcal{X}_b}$ and $z_{\mathsf{Parent}(\mathcal{X}_b)}$, checking whether $\mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathsf{Parent}(\mathcal{X}_b)}(z_{\mathsf{Parent}(\mathcal{X}_b)}) \neq \varnothing$ requires us to check if there is a directed cycle in a graph with $O(n)$ nodes, which has a $O(n^2)$ complexity. Because there is $O(n^{2\mathsf{tw}(\mathsf{G})+2})$ possible combinations of $z_{\mathcal{X}_b}$ and $z_{\mathsf{Parent}(\mathcal{X}_b)}$, the pre-computation has a complexity of $O(n^{2\mathsf{tw}(\mathsf{G})+4})$.

Therefore, the complexity of total computation is $\sum_{b=1}^{m} |\,\mathsf{Children}(\mathcal{X}_b)|\, n^{\mathsf{tw}(\mathsf{G})+1} + O(|\mathsf{B}| \cdot n^{2\mathsf{tw}(\mathsf{G})+4}) \le n^{\mathsf{tw}(\mathsf{G})+1} \sum_{b=1}^{m} |\,\mathsf{Children}(\mathcal{X}_b)| + O(|\mathsf{B}| \cdot n^{2\mathsf{tw}(\mathsf{G})+4}) = O\left(|\mathcal{M}|\, n^{2\mathsf{tw}(\mathsf{G})+4}\right)$, where the last equality follows because $\sum_{b=1}^{m} |\,\mathsf{Children}(\mathcal{X}_b)|$ is equal to the number of bags in the tree decomposition, which is at most $|\mathcal{M}|$ by Lemma 5.3.

D.2 Proof of Theorem 5.5

We now describe an LP with $O\left(|\mathcal{M}|\, n^{2(\mathsf{tw}(\mathsf{G})+1)}\right)$ variables and $O(|\mathcal{M}|\, n^{\mathsf{tw}(\mathsf{G})+1})$ constraints to solve the Rank Aggregation LP under a general choice graph with bounded tree width. Let T_G be the minimal tree decomposition of G with a collection of bags $\{\mathcal{X}_b : b \in \mathsf{B}\}$. To facilitate our description of the LP feasible region, recall that we write $z_{\mathcal{X}_b} = (z_S : S \in \mathcal{X}_b)$ to denote the vector of choices from the subsets in the bag $\mathcal{X}_b$ and $\mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) = \cap_{S \in \mathcal{X}_b} \mathscr{D}_S(z_S)$. Consider the polytope $\mathscr{K}$, defined as follows:

$$
\begin{aligned}
\mathscr{K} = \Bigg\{ \left(y^b_{z_{\mathcal{X}_b}} : b \in \mathcal{B},\ \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing \right) \ \Bigg| \\
& \sum_{z_{\mathcal{X}_b} : \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing} y^b_{z_{\mathcal{X}_b}} = 1 && \forall\ b \in \mathcal{B}, \\
& y^b_{z_{\mathcal{X}_b}} = \sum_{z_{\mathcal{X}_a} : \mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \cap \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing} w^{a,b}_{z_{\mathcal{X}_a}, z_{\mathcal{X}_b}} && \forall \text{ edge } \{\mathcal{X}_a, \mathcal{X}_b\} \text{ in } \mathsf{T}_G \text{ and } z_{\mathcal{X}_b} \\
& y^a_{z_{\mathcal{X}_a}} = \sum_{z_{\mathcal{X}_b} : \mathscr{D}_{\mathcal{X}_a}(z_{\mathcal{X}_a}) \cap \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing} w^{a,b}_{z_{\mathcal{X}_a}, z_{\mathcal{X}_b}} && \forall \text{ edge } \{\mathcal{X}_a, \mathcal{X}_b\} \text{ in } \mathsf{T}_G \text{ and } z_{\mathcal{X}_a} \\
& \text{All variables are non-negative} \ \Bigg\}.
\end{aligned}
$$

In the above polytope, for any $z_{\mathcal{X}_b} = (z_S : S \in \mathcal{X}_b)$, we interpret the variable $y^b_{z_{\mathcal{X}_b}}$ as the probability that for all $S \in \mathcal{X}_b$, the product z_S is the highest-ranked product in set S. The terms $w^{a,b}_{z_{\mathcal{X}_a}, z_{\mathcal{X}_b}}$ correspond to the joint probabilities on the edge $\{\mathcal{X}_a, \mathcal{X}_b\}$ in T_G. Note that the above polytope has $O\left(|\mathcal{M}|\, n^{2(\mathsf{tw}(\mathsf{G})+1)}\right)$ variables and $O(|\mathcal{M}|\, n^{\mathsf{tw}(\mathsf{G})+1})$ constraints. Recall from Lemma 2.5 that the feasible region of the Rank Aggregation LP is given by the following polytope:

$$
\mathscr{M} \equiv \text{ Convex hull of the set } \left\{ \ (\mathbb{1}[\sigma, i, S] : i \in S,\ S \in \mathcal{M}) \in \{0,1\}^{\sum_{S \in \mathcal{M}} |S|} \ : \ \sigma \in \mathscr{P}_n \right\}.
$$

To complete the proof of Theorem 5.5, it suffices to show that $\mathscr{M} = \mathscr{K}$. It is clear that $\mathscr{M} \subseteq \mathscr{K}$, so it remains to show that $\mathscr{K} \subseteq \mathscr{M}$. We will prove the result by induction on the number of bags in the tree decomposition T_G of G. Consider the case where T_G has a single bag, corresponding to $\mathcal{M}$. In that case, the variables in $\mathscr{K}$ consists of $y_{z_\mathcal{M}}$ for all $z_\mathcal{M}$ such that $D_\mathcal{M}(z_\mathcal{M}) \neq \varnothing$. Define a probability mass function $\lambda : \mathscr{P}_n \to [0,1]$ as follows:

$$
\lambda(\sigma) = \begin{cases} \frac{y_{z_\mathcal{M}}}{|\mathscr{D}_\mathcal{M}(z_\mathcal{M})|} & \text{if } \sigma \in \mathscr{D}_\mathcal{M}(z_\mathcal{M}) \text{ for some } z_\mathcal{M}, \\ 0 & \text{otherwise} \end{cases}
$$

It is easy to check that $\lambda(\cdot)$ is a well-defined probability mass function over all rankings. Moreover, it has the desired choice selection probability.

Now, consider the case where the tree decomposition has exactly two bags, say $\mathcal{X}_1$ and $\mathcal{X}_2$ with an edge between $\mathcal{X}_1$ and $\mathcal{X}_2$. We define a distribution over rankings as follows:

$$\lambda(\sigma) = \begin{cases} \frac{w_{z_{\mathcal{X}_1}, z_{\mathcal{X}_2}}}{\left|\mathscr{D}_{\mathcal{X}_1}(z_{\mathcal{X}_1}) \cap \mathscr{D}_{\mathcal{X}_2}(z_{\mathcal{X}_2})\right|} & \text{if } \sigma \in \mathscr{D}_{\mathcal{X}_1}(z_{\mathcal{X}_1}) \cap \mathscr{D}_{\mathcal{X}_2}(z_{\mathcal{X}_2}) \text{ for some } z_{\mathcal{X}_1}, z_{\mathcal{X}_2}, \\ 0 & \text{otherwise} \end{cases}$$

Once again, it is easy to that $\lambda(\cdot)$ is a well-defined probability mass function with the desired choice probabilities.

Consider the case $|\mathcal{B}| = k > 2$. Pick a bag in T_G with at least two children; such a vertex exists because $|\mathcal{B}| > 2$ and T_G is connected. We label this vertex as the root bag. Let $\mathcal{C}^1, \ldots, \mathcal{C}^h$ denote the children of the root bag in the tree, with $h \geq 2$. For $j = 1, \ldots, h$, let $\mathcal{F}^j = \{\mathcal{C}^j\} \cup \mathsf{T}(\mathcal{C}^j)$ denote the collection of bags in the sub-tree rooted at $\mathcal{C}^j$.

To show that $\mathscr{K} \subseteq \mathscr{M}$, we will show that for every feasible solution of $\mathscr{K}$, there exists a distribution $\lambda(\cdot)$ over rankings that matches the desired choice probabilities. By our inductive hypothesis, since $|\{\mathsf{root}\} \cup \mathcal{F}^j| \leq k-1$, there exists a probability mass function $\lambda^j(\cdot)$ over rankings that matches the choice probabilities of every bag in $\{\mathsf{root}\} \cup \mathcal{F}^j$; that is, for all $\mathcal{X}_b \in \{\mathsf{root}\} \cup \mathcal{F}^j$,

$$\sum_{\sigma \in \mathscr{P}_n} \lambda^j(\sigma) \prod_{S \in \mathcal{X}_b} \mathbb{1}[\sigma, z_S, S] = y^b_{z_{\mathcal{X}_b}} \qquad j = 1, 2, \ldots, h\ .$$

To complete the result, we will combine the distributions $\lambda^1(\cdot), \ldots, \lambda^h(\cdot)$ to create a probability distribution $\lambda(\cdot)$ over all rankings with the desired property. The construction is similar to that given in the proof of Theorem 4.2. Specifically, for each $j = 1, 2, \ldots, h$, let $\mathcal{A}^j$ denote the collection of subsets $\cup_{\mathcal{X}_b \in \mathcal{F}^j} \mathcal{X}_b$. Let $\vec{\boldsymbol{a}}^j = (a^j_B \in B \colon B \in \mathcal{A}^j) \in \times_{B \in \mathcal{A}^j} B$ denote the vector of elements, each coming from a set in the collection $\mathcal{A}^j$. Further, let $\boldsymbol{\gamma} = (\gamma_B \in B \colon B \in \mathsf{root})$ be the vector of elements from sets in the root bag. Let

$$w^j\left(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^j\right) = \sum_{\sigma} \lambda^j(\sigma) \cdot \prod_{B \in \mathsf{root}} \mathbb{1}[\sigma, \gamma_B, B] \cdot \prod_{B \in \mathcal{A}^j} \mathbb{1}[\sigma, a^j_B, B] = \sum_{\sigma \in \mathscr{D}_{\mathsf{root}}(\boldsymbol{\gamma}) \cap \mathscr{D}_{\mathcal{A}^j}(\vec{\boldsymbol{a}}^j)} \lambda^j(\sigma).$$

It follows from our construction that

$$w^j(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^j) \leq \sum_{\sigma} \lambda^j(\sigma) \cdot \prod_{B \in \mathsf{root}} \mathbb{1}[\sigma, \gamma_B, B] = y^{\mathsf{root}}_{\boldsymbol{\gamma}}$$

Now, for all $(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) \in \mathsf{root} \times (\times_{B \in \mathcal{A}^1} B) \times \cdots \times (\times_{B \in \mathcal{A}^h} B)$, let

$$\mathscr{P}(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) = \mathscr{D}_{\mathsf{root}}(\boldsymbol{\gamma}) \cap \left(\cap_{j=1}^h \mathscr{D}_{\mathcal{A}^j}(\vec{\boldsymbol{a}}^j)\right)$$

and

$$w(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) = \frac{\prod_{j=1}^h w^j(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^j)}{\left(y_{\boldsymbol{\gamma}}^{\mathsf{root}}\right)^{h-1}},$$

where we define 0/0 to be 0. This is well-defined because we have shown that $w^j(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^j) \le y_{\boldsymbol{\gamma}}^{\mathsf{root}}$ for all j, so whenever the denominator is zero, so is the numerator. One difference from the proof of Theorem 4.2 is that the collections $\mathcal{A}^1, \ldots, \mathcal{A}^h$ may be overlapping. Specifically, suppose $B \in \mathcal{A}^{j_1} \cap \mathcal{A}^{j_2}$ for some $j_1 \neq j_2$. Then, it follows from the running intersection property that $B \in \mathsf{root}$. Now, consider vectors $\vec{\boldsymbol{a}}^{j_1}$ and $\vec{\boldsymbol{a}}^{j_2}$ such that $a_B^{j_1}, a_B^{j_2} \in B$ but $a_B^{j_1} \neq a_B^{j_2}$. It then follows that for any $\boldsymbol{\gamma} \in \bigtimes_{C \in \mathsf{root}} C$, we have that $\gamma_B \neq a_B^{j_1}$ or $\gamma_B \neq a_B^{j_2}$. Without loss of generality, suppose that $\gamma_B \neq a_B^{j_1}$. Then, the vector of choices $(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^{j_1})$ is inconsistent, resulting in $w^{j_1}(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^{j_1}) = 0$, which in turn implies that $w(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) = 0$. We have thus argued that even though the same subset may appear in different collections, our definition of $w(\cdot, \ldots, \cdot)$ ensures that inconsistent choices receive zero weight. Finally, following the arguments of the proof of Theorem 4.2, we can also show that whenever $\mathscr{P}(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) = \varnothing$, we have that $w(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) = 0$.

Now define the distribution $\lambda \colon \mathscr{P}_n \to [0,1]$:

$$\lambda(\sigma) = \begin{cases} \frac{w(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h)}{\left|\mathscr{P}(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h)\right|} & \text{if } \sigma \in \mathscr{P}\left(\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h\right) \text{ for some } (\boldsymbol{\gamma}, \vec{\boldsymbol{a}}^1, \ldots, \vec{\boldsymbol{a}}^h) \\ 0, & \text{otherwise.} \end{cases}$$

The rest of the proof follows the arguments of the proof of Theorem 4.2 with the summation $\sum_{\gamma \in \mathsf{root}}$ replaced by $\sum_{\boldsymbol{\gamma} \in \bigtimes_{B \in \mathsf{root}} B}$ because root is now a bag of sets as opposed to simply being a set. We omit the details.

D.3 Proof of Lemma 5.7

We first show that $\mathsf{tw}(\mathsf{G}_2) \ge n$, where G_2 is the choice graph of $\mathtt{Del}_2$, the collection of offer sets with at most two products stocked out. We prove this result by showing that a complete graph K_{n+1} with $n+1$ vertices is a minor of G_2; that is, K_{n+1} can be obtained from G_2 by deleting edges and vertices and by contracting edges. It then follows from a standard result (Robertson and Seymour 1986) that $\mathsf{tw}(\mathsf{G}_2) \ge \mathsf{tw}(K_{n+1}) = n$, where the equality follows from the standard fact that the tree width of the complete graph K_{n+1} is equal to n.

We show K_{n+1} is a minor of G_2 by establishing that K_{n+1} can be obtained from G_2 through edge contraction. Recall that the graph G_2 has $1 + n + \binom{n}{2}$ vertices consisting of sets N, $N_{\{1\}}$, $N_{\{2\}}, \ldots, N_{\{n\}}$, $N_{\{1,2\}}, \ldots, N_{\{1,n\}}$, $N_{\{2,3\}}, \ldots, N_{\{n-1,n\}}$. There is an edge between N and $N_{\{i\}}$, and each $N_{\{i,j\}}$ (with $i \neq j$) is connected to $N_{\{\ell\}}$ for all ℓ. Consider the edge between $N_{\{\ell\}}$ and $N_{\{i,j\}}$. If we contract this edge, it is equivalent to removing the node $N_{\{i,j\}}$ and adding an edge between $N_{\{\ell\}}$ and $N_{\{s\}}$ for all $s \neq \ell$ because $N_{\{s\}}$ is also connected to $N_{\{i,j\}}$. Thus, by repeatedly contracting the edge between $N_{\{\ell\}}$ and $N_{\{i,j\}}$, the vertices $N_{\{1\}},\ldots,N_{\{n\}}$ will be all connected to each other, so we a complete graph among the vertices $N, N_{\{1\}},\ldots,N_{\{n\}}$, which is the desired result. Thus, we have that $\mathsf{tw}(\mathsf{G}_2)\geq n$. Since G_k is a subgraph of G_{k+1}, it follows from a standard result in graph theory (Bodlaender et al. 2006) that $\mathsf{tw}(\mathsf{G}_{k+1})\geq \mathsf{tw}(\mathsf{G}_k)$, which implies that $\mathsf{tw}(\mathsf{G}_k)\geq \mathsf{tw}(\mathsf{G}_2)\geq n$, which is the desired result.

We will now show that the choice depth of the k-deletion collection is at most k. Consider the following tree decomposition $\mathsf{T}_{\mathsf{G}_k}$ of G_k. The tree $\mathsf{T}_{\mathsf{G}_k}$ is comprised of the k bags $\mathcal{X}_1,\ldots,\mathcal{X}_k$ defined as

$$\mathcal{X}_\ell = \{N_A : |A| = \ell \text{ or } \ell-1\}$$

In other words, $\mathcal{X}_1=\{N, N_{\{1\}},\ldots,N_{\{n\}}\}$ and $\mathcal{X}_2$ consists of all subsets that is obtained by deleting one or two products, and so on. We connect $\mathcal{X}_\ell$ to $\mathcal{X}_{\ell+1}$ for all $\ell<k$, resulting in the following line graph:

$$\mathcal{X}_1 \;—\; \mathcal{X}_2 \;—\; \cdots\cdots \;—\; \mathcal{X}_{k-1} \;—\; \mathcal{X}_k$$

The above line graph is a tree decomposition of G_k because: 1) $\cup_{\ell=1}^{k}\mathcal{X}_\ell=\mathcal{M}$, 2) every edge in $\mathtt{Del}_k$ is contained in some bag $\mathcal{X}_\ell$ by our construction, and 3) the running intersection property is true by default because the only bags that contain common sets are adjacent nodes $\mathcal{X}_\ell$ and $\mathcal{X}_{\ell-1}$, and there no bag in between them.

Now note that for any $\ell\leq k$, $\left|\bigcup_{S\in\mathcal{X}_\ell} S\right| - \min_{S\in\mathcal{X}_\ell}|S| = n-(n-\ell)=\ell$, because $\cup_{S\in\mathcal{X}_\ell}S=N$ for all ℓ. Therefore,

$$\min\left\{|\mathcal{X}_b|-1\ ,\ \left|\bigcup_{S\in\mathcal{X}_b} S\right| - \min_{S\in\mathcal{X}_b}|S|\right\}\leq \ell$$

This implies that $\mathsf{cd}(\mathsf{G}_k,\mathsf{T}_{\mathsf{G}_k})\leq k$, which gives the desired result.

D.4 Proof of Theorem 5.8

To establish the computational complexity, we revisit the dynamic programming equation in Theorem 5.4. Following the arguments in the proof of Theorem 5.4, we obtain the following DP recursion:

$$J_{\mathcal{X}_b}\left(z_{\mathcal{X}_b}\right) = c_{z_{\mathcal{X}_b},\mathcal{X}_b} + \sum_{\mathcal{X}_a\in\,\mathsf{Children}(\mathcal{X}_b)} \min\left\{J_{\mathcal{X}_a}\left(z_{\mathcal{X}_a}\right)\ :\ \mathscr{D}_{\mathcal{X}_b}\left(z_{\mathcal{X}_b}\right)\cap\mathscr{D}_{\mathcal{X}_a}\left(z_{\mathcal{X}_a}\right)\neq\varnothing\right\}.$$

All the terms are as defined in the proof of Theorem 5.4. Now, the state $(z_S : S\in\mathcal{X}_b)$ of the value function at bag $\mathcal{X}_b$ corresponds to the set of top ranked products for each set in the bag $\mathcal{X}_b$. In the case when the number of sets in each bag is "small," it can be maintained directly in a an efficient manner. However, in situations where the number of sets in each bag is large, maintaining the state directly can be inefficient. Instead, we make the observation that the top-ranked product for many of the sets within the same bag will be the same, especially when the graph has a small choice depth. We exploit this property to maintain state more efficiently.

In order to obtain an efficient state representation, consider an arbitrary bag $\mathcal{X}_b$. Let $U = \bigcup_{S\in\mathcal{X}_b} S$ and $d_{\mathcal{X}_b} = \left|\bigcup_{S\in\mathcal{X}_b} S\right| - \min_{S\in\mathcal{X}_b}|S|$. We claim that maintaining the top-$(1+d_{\mathcal{X}_b})$ ranked products within the universe U is sufficient to recover to the top-choices z_S from all subsets $S\in\mathcal{X}_b$. To see this, suppose $A = \left\{a_1, a_2, \ldots a_{1+d_{\mathcal{X}_b}}\right\}$ are the top-$(1+d_{\mathcal{X}_b})$ ranked products in the set U such that a_r is the rth-ranked product. Consider an arbitrary subset $S\in\mathcal{X}_b$. We claim that $S\cap A \neq \varnothing$. Otherwise, we must have that $|A|+|S| \geq 1+d_{\mathcal{X}_b}+\min_{B\in\mathcal{X}_b}|B| = 1+d_{\mathcal{X}_b}+\left|\bigcup_{S\in\mathcal{X}_b} S\right| - d_{\mathcal{X}_b} = 1+|U|$, which is a contradiction because both A and S are subsets of U by definition.

Because A consists of the top-ranked products, it must be that the products in $S\setminus A$ are never chosen. In fact, the choice z_S from S must be the top-ranked products in $S\cap A$, i.e.,

$$z_S = a_{i^*} \text{ where } i^* = \min\left\{1\leq i\leq d_{\mathcal{X}_b}+1 \colon a_i\in S\right\} \ .$$

Therefore, we maintain the vector of the top $d_{\mathcal{X}_b}+1$ ranked products in $\bigcup_{S\in\mathcal{X}_b} S$. With this representation, the number of states is $|U|^{d_{\mathcal{X}_b}+1} = O(n^{\mathsf{cd}(\mathsf{G})+1})$, where we assume that we have a tree decomposition whose choice depth is the minimum.

By following the rest of the arguments from the proof of Theorem 4.1 with the tree width replaced by the choice depth, we obtain the computational complexity $O\left(|\mathcal{M}|\, n^{4+2\mathsf{cd}(\mathsf{G})}\right)$.

The result of the theorem now follows.

D.5 Proof of Theorem 5.9

The proof is similar to that of Theorem 5.5. First, it is clear that $\mathscr{M}\subseteq\mathscr{H}$. So, we focus on establishing that $\mathscr{H}\subseteq\mathscr{M}$. For that, suppose we are given variables $\boldsymbol{y}$, $\boldsymbol{x}$, and $\boldsymbol{w}$ belonging to $\mathscr{H}$. We will construct variables $\tilde{\boldsymbol{w}}$ such that $\boldsymbol{y}, \tilde{\boldsymbol{w}} \in \mathscr{K}$, the polyhedron defined in Theorem 5.5. Since $\mathscr{K}=\mathscr{M}$ by Theorem 5.5, it follows that there exists a distribution over rankings that is consistent with the choice probabilities $\boldsymbol{y}$, establishing that $\mathscr{H}\subseteq\mathscr{M}$.

To construct $\tilde{\boldsymbol{w}}$ from $\boldsymbol{y}$, $\boldsymbol{x}$, and $\boldsymbol{w}$, we proceed as follows. For any $z_{\mathcal{X}_b} = (z_S \colon S\in\mathcal{X}_b)$, recall that $\mathscr{O}_k(U_b, z_{\mathcal{X}_b}) \subset \mathscr{O}_k(U_b)$ denotes the top-k orderings of elements in U_b that are consistent with the choices $z_{\mathcal{X}_b}$. In other words,

$$\mathscr{O}_k(U_b, z_{\mathcal{X}_b}) = \left\{\boldsymbol{a}\in\mathscr{O}_k(U_b) \colon z_S = a_{j^*}, j^* = \min\left\{1\leq j\leq k \colon a_j\in S\right\},\ \forall\ S\in\mathcal{X}_b\right\}.$$

Now for any $b, b'\in\mathsf{B}$ such that $\{\mathcal{X}_b, \mathcal{X}_{b'}\}$ is an edge in T_G, define

$$\tilde{w}^{b,b'}_{z_{\mathcal{X}_b}, z_{\mathcal{X}_{b'}}} = \sum_{\boldsymbol{a}\in\mathscr{O}_k(U_b, z_{\mathcal{X}_b}), \boldsymbol{a}'\in\mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}}) \colon \mathscr{P}_k(\boldsymbol{a}', U_{b'})\cap\mathscr{P}_k(\boldsymbol{a}, U_b)\neq\varnothing} w_{b,\boldsymbol{a},b',\boldsymbol{a}'}.$$

To show that $\tilde{\boldsymbol{w}}$ and $\boldsymbol{y}$ belong to $\mathscr{K}$, it is sufficient to show that

$$y^{b'}_{z_{\mathcal{X}_{b'}}} = \sum_{z_{\mathcal{X}_b}:\, \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \neq \varnothing} \tilde{w}^{b,b'}_{z_{\mathcal{X}_b}, z_{\mathcal{X}_{b'}}} \quad \text{and} \quad y^{b}_{z_{\mathcal{X}_b}} = \sum_{z_{\mathcal{X}_{b'}}:\, \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \neq \varnothing} \tilde{w}^{b,b'}_{z_{\mathcal{X}_b}, z_{\mathcal{X}_{b'}}} \tag{6}$$

We focus on establishing the first equality. For that, we proceed as follows. Fix an ordering $\boldsymbol{a}' \in \mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}})$. We first note that $\mathscr{O}_k(U_b, z_{\mathcal{X}_b}) \cap \mathscr{O}_k(U_b, \tilde{z}_{\mathcal{X}_b}) = \varnothing$ whenever $z_{\mathcal{X}_b} \neq \tilde{z}_{\mathcal{X}_b}$ because the same top-k ordering cannot result in distinct choices from the same subset. We now claim that

$$\begin{aligned} &\{\boldsymbol{a} \in \mathscr{O}_k(U_b) \colon \mathscr{P}_k(\boldsymbol{a}, U_b) \cap \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \neq \varnothing\} \\ &\quad = \bigcup_{z_{\mathcal{X}_b}\,:\, \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \cap \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing} \cdot \left\{\boldsymbol{a} \in \mathscr{O}_k(U_b, z_{\mathcal{X}_b}) \colon \mathscr{P}_k(\boldsymbol{a}, U_b) \cap \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \neq \varnothing\right\} \end{aligned} \tag{7}$$

It is clear that RHS $\subseteq$ LHS, so we show that LHS $\subseteq$ RHS. For that, consider an arbitrary top-k ordering $\bar{\boldsymbol{a}} \in$ LHS. Let $z_{\mathcal{X}_b}$ be the choices from the offer sets in $\mathcal{X}_b$ according to the ordering $\bar{\boldsymbol{a}}$. Now, because $\mathscr{P}_k(\bar{\boldsymbol{a}}, U_b) \cap \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \neq \varnothing$, we can pick a ranking $\sigma \in \mathscr{P}_k(\bar{\boldsymbol{a}}, U_b) \cap \mathscr{P}_k(\boldsymbol{a}', U_{b'})$. Because the ranking σ is consistent with the top-k ordering $\boldsymbol{a}'$ and $\boldsymbol{a}'$ is consistent with the choices $z_{\mathcal{X}_{b'}}$, it must be that σ is consistent with the choices $z_{\mathcal{X}_{b'}}$; that is, $\sigma \in \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}})$. Moreover, by construction, $\sigma \in \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b})$. We have thus shown that $\mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \cap \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \neq \varnothing$. By construction, we also have that $\bar{\boldsymbol{a}} \in \left\{\boldsymbol{a} \in \mathscr{O}_k(U_b, z_{\mathcal{X}_b}) \colon \mathscr{P}_k(\boldsymbol{a}, U_b) \cap \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \neq \varnothing\right\}$, so $\bar{\boldsymbol{a}} \in$ RHS. Since $\bar{\boldsymbol{a}}$ is arbitrary, we have that LHS $\subseteq$ RHS, as desired.

It now follows that

$$\begin{aligned} &\sum_{z_{\mathcal{X}_b}:\, \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \neq \varnothing} \tilde{w}^{b,b'}_{z_{\mathcal{X}_b}, z_{\mathcal{X}_{b'}}} \\ &= \sum_{z_{\mathcal{X}_b}:\, \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}}) \neq \varnothing} \quad \sum_{\boldsymbol{a} \in \mathscr{O}_k(U_b, z_{\mathcal{X}_b}), \boldsymbol{a}' \in \mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}}) \colon \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \cap \mathscr{P}_k(\boldsymbol{a}, U_b) \neq \varnothing} w_{b,\boldsymbol{a},b',\boldsymbol{a}'} \\ &= \sum_{\boldsymbol{a}' \in \mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}})} \left[\sum_{z_{\mathcal{X}_b}:\, \mathscr{D}_{\mathcal{X}_b}(z_{\mathcal{X}_b}) \cap \mathscr{D}_{\mathcal{X}_{b'}}(z_{\mathcal{X}_{b'}})} \quad \sum_{\boldsymbol{a} \in \mathscr{O}_k(U_b, z_{\mathcal{X}_b}) \colon \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \cap \mathscr{P}_k(\boldsymbol{a}, U_b) \neq \varnothing} w_{b,\boldsymbol{a},b',\boldsymbol{a}'} \right] \\ &\overset{\text{from (7)}}{=} \sum_{\boldsymbol{a}' \in \mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}})} \left[\sum_{\boldsymbol{a} \in \mathscr{O}_k(U_b) \colon \mathscr{P}_k(\boldsymbol{a}', U_{b'}) \cap \mathscr{P}_k(\boldsymbol{a}, U_b) \neq \varnothing} w_{b,\boldsymbol{a},b',\boldsymbol{a}'} \right] \\ &= \sum_{\boldsymbol{a}' \in \mathscr{O}_k(U_{b'}, z_{\mathcal{X}_{b'}})} x_{b',\boldsymbol{a}'} \\ &= y^{b'}_{z_{\mathcal{X}_{b'}}}, \end{aligned}$$

where the last two equalities follow from the fact that $\boldsymbol{x}$, $\boldsymbol{w}$, and $\boldsymbol{y}$ belong to $\mathscr{H}$. The second equality in (6) can be established in a similar fashion.

The result of the theorem now follows.

D.6 Proof of Proposition 5.11

The proof of Proposition 5.11 makes use of the following two lemmas. The first lemma establishes that the minimum degree of each vertex in any choice graph associated with the collection $\mathsf{Pairs} = \{\{i,j\} : i \neq j\}$ is at least $2(n-2)$. The second lemma shows that, for the purposes of computing the choice depth and the tree width, it suffices to consider tree decompositions whose adjacent bags do not contain each other.

Lemma D.1 *Every vertex* $\{i,j\}$ *of any choice graph associated with the collection* Pairs *is connected by an edge to* $\{i,\ell\}$ *and* $\{\ell,j\}$ *for every* $\ell \notin \{i,j\}$. *Therefore, every vertex has a degree of at least* $2(n-2)$.

Proof: Let $\mathsf{G} = (\mathsf{Pairs}, \mathsf{E})$ be an arbitrary choice graph associated with the collection $\mathsf{Pairs} = \{\{i,j\} : i \neq j\}$. Without loss of generality, we prove the result for the vertex $\{1,2\}$: there are edges from $\{1,2\}$ to $\{1,\ell\}$ and $\{2,\ell\}$, for all $\ell = 3,4,\ldots,n$.

The proof is by contradiction, so suppose that $\{1,2\}$ is **not** connected to $\{1,3\}$ by an edge. Let $\mathcal{A} = \{\{1,2\}\}$, $\mathcal{B} = \{\{1,3\}\}$, and $\mathcal{C} = \mathsf{Pairs} \setminus \{\{1,2\},\{1,3\}\}$. Because $\{1,2\}$ is not connected to $\{1,3\}$ by an edge, it follows that $\mathcal{C}$ separates $\mathcal{A}$ from $\mathcal{B}$ in the graph G. Since G is a choice graph of Pairs, we must then have $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$ i.e., if $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$ and $\mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$, then $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$.

We arrive at a contradiction by construction choices $z_{\mathcal{A}}$, $z_{\mathcal{B}}$, and $z_{\mathcal{C}}$ that violate the above implication. For that, let π be an arbitrary ordering among products $4,5,6,\ldots,n$. Consider an ordering σ that has product 1 as the most preferred product, followed by product 2 as the second ranked product, followed by product 3 as the third ranked, and then follows the ordering of π. So, under σ, the top three products are $1,2,3$ in that order. Let $z_{\mathcal{A}} = 1$ and let $z_{\mathcal{C}} = (\arg\min_{i \in C} \sigma(i) : C \in \mathcal{C})$ be the vector consisting of the most preferred product for each set $C \in \mathcal{C}$ under σ. By our construction, $\sigma \in \mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$.

Now, consider the ordering τ that has product 2 as the most preferred product, followed by product 3 as the second ranked product, followed by product 1 as the third ranked, and then follows the ordering of π. Under the ordering τ, the top three products are $2,3,1$ in that order. Let $z_{\mathcal{B}} = 3$. Note that for every set $C \in \mathcal{C}$, the most preferred product in C under τ is *exactly the same* as the most preferred product in C under σ; that is, $\arg\min_{i \in C} \sigma(i) = \arg\min_{i \in C} \tau(i)$. This follows because each set C in $\mathcal{C}$ belongs to one of three types: (1) $C = \{a,b\}$ where $a \notin \{1,2,3\}$ and $b \notin \{1,2,3\}$, (2) $C = \{1,\ell\}$, $\{2,\ell\}$, or $\{3,\ell\}$ for some $\ell \notin \{1,2,3\}$, or (3) $C = \{2,3\}$. If C is the first type, then both σ and τ yield the same top ranked product by our construction. Similarly, if C is the second type, then we still get the same top ranked product because under both σ and

τ, products $1,2,3$ are always preferred over ℓ for $\ell \notin \{1,2,3\}$. Finally, if $C = \{2,3\}$, then the top ranked product under both σ and τ is product 2 by our construction. Therefore, we have that $\tau \in \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}})$.

If G is indeed a choice graph, it must be that $\mathscr{D}_{\mathcal{A}}(z_{\mathcal{A}}) \cap \mathscr{D}_{\mathcal{B}}(z_{\mathcal{B}}) \cap \mathscr{D}_{\mathcal{C}}(z_{\mathcal{C}}) \neq \varnothing$. Note that $\{2,3\} \in \mathcal{C}$. This means that there exists an ordering such that $z_{\mathcal{A}} = z_{\{1,2\}} = 1$, $z_{\{2,3\}} = 2$, and $z_{\mathcal{B}} = z_{\{1,3\}} = 3$ simultaneously. However, this violates the transitivity of preference. Contradiction! Therefore, it must be the case that there is an edge between $\{1,2\}$ and $\{1,3\}$.

Similar arguments show that there are edges from $\{1,2\}$ to $\{1,\ell\}$ and $\{2,\ell\}$, for all $\ell \notin \{1,2\}$. Therefore, the degree of the vertex $\{1,2\}$ is at least $2(n-2)$. This completes the proof. ■

Lemma D.2 *For every choice graph* G *and every tree decomposition* T_{G}*, there is a tree decomposition* T'_{G} *with a collection of bags* $\{\mathcal{X}_b : b \in \mathsf{B}\}$ *such that:*

- $\mathsf{cd}(\mathsf{G}, \mathsf{T}'_{\mathsf{G}}) = \mathsf{cd}(\mathsf{G}, \mathsf{T}_{\mathsf{G}})$
- *For every pair of adjacent bags* $\{\mathcal{X}_a, \mathcal{X}_b\}$ *in* T'_{G}*,* $\mathcal{X}_a \nsubseteq \mathcal{X}_b$ *and* $\mathcal{X}_b \nsubseteq \mathcal{X}_a$*.*

Proof: The proof follows the arguments from the proof of Lemma 2 in Bodlaender and Koster (2011). Specifically, suppose T_{G} is a tree decomposition of G such that for some pair of adjacent bags $\mathcal{X}_a$ and $\mathcal{X}_b$, we have that $\mathcal{X}_a \subseteq \mathcal{X}_b$. Then, we contract the edge between the bags $\mathcal{X}_a$ and $\mathcal{X}_b$: remove the bag $\mathcal{X}_a$ and connect all its neighbors to the bag $\mathcal{X}_b$. Let $\tilde{\mathsf{T}}_{\mathsf{G}}$ denote the resulting tree. It can be seen that $\tilde{\mathsf{T}}_{\mathsf{G}}$ is a tree decomposition of G and is defined over the collection of bags $\left\{\mathcal{X}_{b'} : b' \in \tilde{\mathsf{B}}\right\}$, where $\tilde{\mathsf{B}} = \mathsf{B} \setminus \{a\}$.

We claim that this process does not change the choice depth of the tree i.e., $\mathsf{cd}(\mathsf{G}, \mathsf{T}_{\mathsf{G}}) = \mathsf{cd}(\mathsf{G}, \tilde{\mathsf{T}}_{\mathsf{G}})$. To see this, first note that $\left|\bigcup_{S \in \mathcal{X}_a} S\right| \leq \left|\bigcup_{S \in \mathcal{X}_b} S\right|$ and $\min_{A \in \mathcal{X}_a} |A| \geq \min_{A \in \mathcal{X}_b} |A|$ because $\mathcal{X}_a \subseteq \mathcal{X}_b$. Therefore,

$$\left|\bigcup_{S \in \mathcal{X}_a} S\right| - \min_{A \in \mathcal{X}_a} |A| \leq \left|\bigcup_{S \in \mathcal{X}_b} S\right| - \min_{A \in \mathcal{X}_b} |A|.$$

It now follows that

$$\mathsf{cd}(\mathsf{G}, \mathsf{T}_{\mathsf{G}}) = \max_{b' \in \mathsf{B}} \left\{ \left|\bigcup_{S \in \mathcal{X}_{b'}} S\right| - \min_{A \in \mathcal{X}_{b'}} |A| \right\} = \max_{b' \in \mathsf{B} \setminus \{\mathcal{X}_a\}} \left\{ \left|\bigcup_{S \in \mathcal{X}_{b'}} S\right| - \min_{A \in \mathcal{X}_{b'}} |A| \right\} = \mathsf{cd}(\mathsf{G}, \tilde{\mathsf{T}}_{\mathsf{G}}),$$

where the last equality follows because $\tilde{\mathsf{T}}_{\mathsf{G}}$ is defined over the collection of bags $\{\mathcal{X}_{b'} : b' \in \mathsf{B}\} \setminus \{\mathcal{X}_a\}$. This establishes the claim.

Repeating the above process iteratively, we obtain the tree decomposition T'_{G} with the desired properties. ■

We now provide the proof of Proposition 5.11.

Proof of Proposition 5.11: Let $\mathsf{G} = (\mathsf{Pairs}, \mathsf{E})$ be an arbitrary choice graph associated with the collection of all pairs of products $\mathsf{Pairs} = \{\{i,j\} : i \neq j\}$. It is a standard result that the tree width of any graph G is lower bounded by the minimum degree of its vertices (Lemma 4 in Bodlaender and Koster 2011). Then, it follows from Lemma D.1 that $\mathsf{tw}(\mathsf{G}) \geq 2(n-2)$, which is the desired result.

We will now establish a lower bound on the choice depth. Consider an arbitrary tree decomposition T_G with a collection of bags $\{\mathcal{X}_b : b \in \mathsf{B}\}$. By Lemma D.2, we can assume without loss of generality that under T_G, every pair of adjacent bags do not contain each other. There are two cases to consider: $|\mathsf{B}| = 1$ and $|\mathsf{B}| \geq 2$.

If T_G has exactly one bag, then the bag must contain every set in the collection Pairs, and by definition

$$\mathsf{cd}(\mathsf{G}, \mathsf{T}_\mathsf{G}) = \max_{b \in \mathsf{B}} \left\{ \left| \bigcup_{S \in \mathcal{X}_b} S \right| - \min_{A \in \mathcal{X}_b} |A| \right\} = \left| \bigcup_{S \in \mathsf{Pairs}} S \right| - \min_{A \in \mathsf{Pairs}} |A| = n - 2,$$

where the last equality follows because $\bigcup_{S \in \mathsf{Pairs}} S = N$ and every set in the collection Pairs has exactly two elements.

Suppose T_G has at least two bags. Following the argument in Lemma 4 in Bodlaender and Koster (2011), take a bag $\mathcal{X}_a$ that is a leaf of T_G, and let $\mathcal{X}_b$ be its (unique) neighbor. Take a set $\{i,j\} \in \mathcal{X}_a \setminus \mathcal{X}_b$. By Lemma D.2, such a set $\{i,j\}$ exists. It must be the case that $\{i,j\}$ does **not** belong to any other bag except $\mathcal{X}_a$, because otherwise it would belong to $\mathcal{X}_b$ by the running intersection property of the tree decomposition. Therefore, all neighbors of $\{i,j\}$ must belong to $\mathcal{X}_a$. By Lemma D.1, this implies that

$$\mathsf{cd}(\mathsf{G}, \mathsf{T}_\mathsf{G}) = \max_{b' \in \mathsf{B}} \left\{ \left| \bigcup_{S \in \mathcal{X}_{b'}} S \right| - \min_{A \in \mathcal{X}_{b'}} |A| \right\} \geq \left| \bigcup_{S \in \mathcal{X}_a} S \right| - \min_{A \in \mathcal{X}_a} |A| = n - 2 \ ,$$

where the equality follows because there is an edge between $\{i,j\}$ and $\{i,\ell\}$, and an edge between $\{i,j\}$ and $\{\ell,j\}$, for all $\ell \notin \{i,j\}$, so $\bigcup_{S \in \mathcal{X}_a} S = N$. Therefore, $\mathsf{cd}(\mathsf{G}) \geq n-2$. ■

D.7 Extension to Allow for Incorporation of Additional Structure on Rankings

In this section, we extend our theoretical framework of rational separation to the case when our focus is on distributions over a *constrained* set of rankings, denoted by $\mathscr{P}_n^{\mathsf{con}} \subseteq \mathscr{P}_n$. As mentioned in the main text in Section 5.3, a constrained set of rankings may be used to capture product price/display promotions by constraining each ranking to prefer the promoted copy of the product over its non-promoted copy. In fact, they may be used to capture even multi-level price promotions, as detailed in the following example.

Example D.3 (Modeling Promotions) Suppose each product i has L different levels of promotions, indexed by $1,\ldots,L$. These levels may correspond to different levels of price discounts in the context of price promotions or different levels of display prominence in the context of display promotions. We assume that the levels are indexed such that for each product, level-L promotion is preferred to level $L-1$, which is preferred to level $L-2$, and so on. To capture this setting, we enlarge the universe to $n(L+1)$ items, where each item is now represented as a tuple $(i,\ell)\in N\times\{0,1,\ldots,L\}$, where i denotes the product index and ℓ denote the promotion level, with $\ell=0$ denoting the non-promoted copy. In this case, the constrained set of rankings is defined over $n(L+1)$ items, with

$$\mathscr{P}^{\mathsf{con}}_{n(L+1)}=\left\{\sigma\in\mathscr{P}_{n(L+1)}\ :\ \forall\ i\in N,\ \sigma(i,L)<\sigma(i,L-1)<\cdots<\sigma(i,1)<\sigma(i,0)\right\}\ ,$$

where the constraint $\sigma(i,L)<\sigma(i,L-1)<\cdots<\sigma(i,1)<\sigma(i,0)$ captures the requirement that level-L promotion is the most preferred, followed by level $L-1$, and the non-promoted version is the least preferred. The above constraint only applies to each individual product, and we do not impose any constraints in the preference orderings of different products.

All the definitions in Sections 2 and 3 now readily extend to the constrained set of rankings $\mathscr{P}^{\mathsf{con}}_n$. Specifically, for any set S and $z_S\in S$, let $\mathscr{D}^{\mathsf{con}}_S(z_S)$ denote the set of constrained rankings in $\mathscr{P}^{\mathsf{con}}_n$ for which z_S is the most preferred product in S; that is,

$$\mathscr{D}^{\mathsf{con}}_S(z_S)=\{\sigma\in\mathscr{P}^{\mathsf{con}}_n\ :\ \sigma(z_S)<\sigma(i)\ \forall\ i\in S,i\neq z_S\}\ .$$

Similarly, for any collection of subsets $\mathcal{A}$, let $z_{\mathcal{A}}=(z_S\in S:S\in\mathcal{A})$ and $\mathscr{D}^{\mathsf{con}}_{\mathcal{A}}(z_{\mathcal{A}})=\bigcap_{S\in\mathcal{A}}\mathscr{D}^{\mathsf{con}}_S(z_S)$. Then, we can extend our notion of rational separation in a straightforward way. Given three subsets A, B, and C, we say that A and B are *rationally separable* given C, written as $A\perp\!\!\!\perp B\mid C$, if for all $z_A\in A$, $z_B\in B$, and $z_C\in z_C$, whenever $\mathscr{D}^{\mathsf{con}}_A(z_A)\cap\mathscr{D}^{\mathsf{con}}_C(z_C)\neq\varnothing$ and $\mathscr{D}^{\mathsf{con}}_B(z_B)\cap\mathscr{D}^{\mathsf{con}}_C(z_C)\neq\varnothing$, we also have that

$$\mathscr{D}^{\mathsf{con}}_A(z_A)\cap\mathscr{D}^{\mathsf{con}}_B(z_B)\cap\mathscr{D}^{\mathsf{con}}_C(z_C)\neq\varnothing\ .$$

The same extension applies to any collections of subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, and correspondingly, a choice graph is defined just as before.

In order to describe the results on computational complexity, we assume that for every $z_A\in A$ and $z_B\in B$, we can determine whether or not $\mathscr{D}^{\mathsf{con}}_A(z_A)\cap\mathscr{D}^{\mathsf{con}}_B(z_B)\neq\varnothing$ by querying an oracle that takes $O(1)$ operations. Then, it can be verified that all the results and proofs of Sections 4 and 5 continue to hold for the case of constrained rankings by replacing $\mathscr{D}_A(z_A)$ with $\mathscr{D}^{\mathsf{con}}_A(z_A)$ for all sets A and products $z_A\in A$.

	Category Shorthand	Expanded Name	# trans.	# of offer sets	Avg. offer set size
1	beer	Beer	380,932	69	5.28
2	blades	Blades	92,404	75	4.96
3	carbbev	Carbonated Beverages	721,506	31	6.61
4	cigets	Cigarettes	249,668	112	6.14
5	coffee	Coffee	372,536	55	6.96
6	coldcer	Cold Cereal	577,236	23	7.35
7	deod	Deodorant	271,286	63	6.48
8	diapers	Diapers	143,055	19	3.32
9	factiss	Facial Tissue	73,806	64	4.77
10	fzdinent	Frozen Dinners/Entrees	979,936	30	7.37
11	fzpizza	Frozen Pizza	292,878	79	6.30
12	hhclean	Household Cleaners	282,981	19	7.89
13	hotdog	Hotdogs	101,624	102	6.41
14	laundet	Laundry Detergent	238,163	84	6.79
15	margbutr	Margarine/Butter	140,969	29	7.52
16	mayo	Mayonnaise	97,282	81	5.62
17	milk	Milk	240,691	57	6.28
18	mustketc	Mustard	134,800	48	7.12
19	paptowl	Paper Towels	82,636	65	6.25
20	peanbutr	Peanut Butter	108,770	82	6.01
21	saltsnck	Salt Snacks	736,148	39	7.08
22	shamp	Shampoo	290,429	73	6.40
23	soup	Soup	905,541	24	7.83
24	spagsauc	Spaghetti/Italian Sauce	276,860	52	6.81
25	sugarsub	Sugar Substitutes	53,834	104	5.92
26	toitisu	Toilet Tissue	112,788	37	6.32
27	toothbr	Toothbrushes	197,676	135	6.16
28	toothpa	Toothpaste	238,271	68	6.40
29	yogurt	Yogurt	499,203	66	5.92

Table EC.1 Summary statistics of the data used in the case study. For each category, we reported the total number of purchase transactions, the number of offer sets, and the average of the offer set sizes (weighted by the corresponding number of sales), observed in the first two weeks of the year 2007 across all the stores in the dataset.

Appendix E: Additional Implementation Details and Discussion of Numerical Results

In this section, we present the omitted details for case study in Section 6.

E.1 Summary statistics of the data

Table EC.1 reports the number of transactions, number of offer sets, and the average offer set size, for each product category.

E.2 Details for computing the total losses for the MNL and LCMNL models

For the MNL model, we obtained the total loss for a category by solving the following optimization problem:

$$\begin{array}{l}\text{Total loss} \\ \text{under MNL}\end{array} = \min_{\boldsymbol{\beta},\gamma} -\frac{1}{|\mathcal{T}|} \sum_{(S,P)\in\mathcal{M}} M_{S,P} \sum_{j\in S} f_{j,S,P} \log\left(\frac{1}{f_{j,S,P}} \cdot \frac{e^{\beta_j+\gamma\cdot\mathbb{1}[j\in P]}}{\sum_{i\in S} e^{\beta_i+\gamma\cdot\mathbb{1}[j\in P]}}\right),$$

where we used the following utility specification: for each product $j \in \{1,\dots,n\}$, $\text{Utility}_j = \beta_j + \gamma \cdot \mathbb{1}[j \text{ is on promotion}] + \varepsilon_j$, where $(\beta_1,\dots,\beta_n)$ are the alternative specific utility parameters, γ

is the additional utility from the product being on promotion, and ε_j are i.i.d. standard Gumbel random variables. The above optimization problem can be shown to be a convex program (Train 2009), and, hence, we solved it by using off-the-shelf convex optimization techniques.

In a similar manner, we computed the total loss for the LCMNL model by solving the following optimization problem for each category:

$$\text{Total loss under LCMNL} = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \quad -\frac{1}{|\mathcal{T}|}\sum_{(S,P)\in\mathcal{M}} M_{S,P}\sum_{j\in S} f_{j,S,P}\log\left(\frac{1}{f_{j,S,P}}\cdot\sum_{k=1}^{15}\alpha^k\frac{e^{\beta_j^k+\gamma^k\cdot\mathbb{1}[j\in P]}}{\sum_{i\in S}e^{\beta_i^k+\gamma^k\cdot\mathbb{1}[j\in P]}}\right)$$
$$\text{s.t.} \quad \sum_{k=1}^{15}\alpha^k = 1, \quad \alpha^k \geq 0 \quad \forall\ k$$

where we assumed that the population is described by a mixture of MNL models consisting of 15 classes, with α^k denoting the fraction of customers in class k; $\boldsymbol{\beta}^k = (\beta_1^k,\ldots,\beta_n^k)$, the parameters of the corresponding MNL model of customers in class k; and γ^k, the additional utility to class k customers from the production promotion. The above optimization problem is non-convex in the model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$. However, we obtained an approximate solution by using the popular expectation-maximization (EM) technique; see Train (2009) for details.

The parametric losses under the MNL and LCMNL models were then obtained by subtracting the rationality loss from their respective total losses.

E.3 Breakdown of parametric losses

Figure EC.1 presents the breakdown of the parametric losses by the number K of latent classes. The length of the first (leftmost) bar represents the parametric loss under the 15-class LCMNL model, which is the most complex parametric model fitted in our experiment. The subsequent bars show *incremental* losses incurred from fitting progressively "simpler" parametric models. So, for instance, the length of the second bar represents the difference in the parametric losses under the 10-class and 15-class LCMNL models; consequently, the sum of the lengths of the first two bars represents the parametric loss under the 10-class LCMNL model. The product categories are arranged in the descending order of their respective parametric losses under the MNL model.

We make the following observations from Figure EC.1. First, it is immediately apparent that conditioned on restricting oneself to the RUM class, the loss incurred from assuming that the utilities have independent Gumbel distributions (the "MNL assumption") varies widely across product categories, with the highest loss incurred for milk and the lowest loss incurred for sugar substitute. In other words, the "value" of making more complex distributional assumptions varies by categories. Second, we note that the rightmost bar is generally longer than the second and third bars. This observation implies that as we add more complexity in the form of additional latent classes, we observe the greatest reductions in parametric losses when we go from $K = 1$ (the MNL model) to $K = 5$, with the reductions tapering off for $K = 10$ and $K = 15$. This finding is intuitive, indicating that the marginal benefit from additional latent classes goes down. Finally, for yogurt and cigarette categories, the parametric losses under the 15-class LCMNL model (represented by the length of the first bar) are significantly higher than those for the other categories, indicating a greater degree of heterogeneity, as measured by the number of latent classes, in the purchase behavior for these categories.

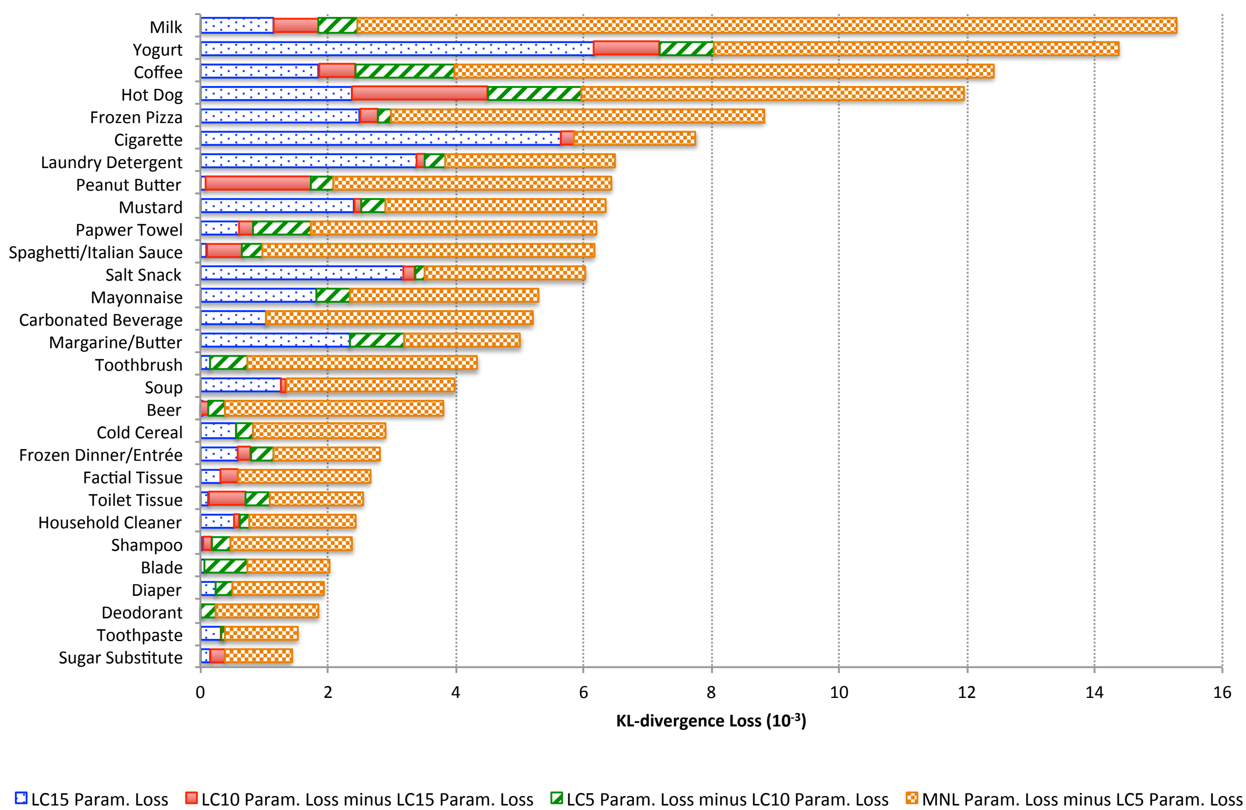


Figure EC.1 **Breakdown of parametric losses:** For each product category, the first (leftmost) bar shows the parametric loss under the 15-class LCMNL model (dotted blue bars), followed by the difference between the parametric losses under the 10-class and 15-class LCMNL models (solid red bars). The difference represents the incremental loss from using a 10-class model instead of a 15-class model. The figure also shows the difference between the losses under the 5-class and 10-class LCMNL models (patterned green bars), and the rightmost bar shows the difference between the losses under the 5-class LCMNL and simple MNL models (checkered orange bars).

E.4 Going beyond the RUM class

As noted in Section 6.3.3, our approach allows us to identify product categories for which it is necessary to go beyond the RUM class to obtain acceptable performance. To obtain a model class outside the RUM class, we extended the generalized attraction model (GAM) class. This model class was proposed by Gallego et al. (2014) as a generalization of the basic attraction model (BAM), of which the MNL model is a special case, to address the issue of overestimation of demand recapture in revenue management (RM) applications. With each product j, it associates a "shadow" attraction $w_j \geq 0$, in addition to a basic attraction $v_j \geq 0$, such that the probability $\mathbb{P}(j|S)$ that j will be chosen from offer set S is given by

$$\mathbb{P}(j|S) = \frac{v_j}{V_0(S) + \sum_{i \in S} v_i}, \text{ where } V_0(S) = v_0 + \sum_{i \notin S} w_i,$$

v_0 is the basic attraction of the no-purchase option, indexed by 0, and S is implicitly assumed to include the no-purchase option. The key difference from a BAM model is the presence of shadow weights for products that continue to influence the choice behavior of a customer through $V_0(S)$ even when not offered. Setting $w_j = 0$ for all j results in the BAM model and $w_j = v_j$ for all j results in the independent demand model (IDM), in which the demand for a product is independent of the offered set.

We note that the GAM model is outside the RUM class whenever $w_j > v_j$ for some product j. To see this, note that for two products $i, j \in S$,

$$\frac{v_i}{\mathbb{P}(i|S)} - \frac{v_i}{\mathbb{P}(i|S \setminus \{j\})} = V_0(S) + v_j - V_0(S \setminus \{j\}) = v_j - w_j < 0 \implies \mathbb{P}(i|S) > \mathbb{P}(i|S \setminus \{j\}), \tag{8}$$

which violates rationality because it stipulates that choice probabilities can only *decrease* when a product is dropped from the offering. We make use of this property below.

For our purposes, we extend the GAM model into a latent-class GAM (LC-GAM) model as follows. We assume that the population is composed of K market segments with the choice behavior of each segment described by a different GAM model. With segment k and product j, we associate the following set of parameters: basic and shadow attractions $v_{jk} = e^{\beta_{jk}}$ and $w_{jk} = e^{\gamma_{jk}}$ when j is not promoted and $v'_{jk} = e^{\beta'_{jk}}$ and $w'_{jk} = e^{\gamma'_{jk}}$ when j is promoted, respectively. Because our dataset does not contain a no-purchase option, we designate one of the n products as the 'default' product d_k for each segment k. We let $\alpha_k \geq 0$, such that $\sum_{k=1}^{K} \alpha_k = 1$, denote the weight of segment k. Then, the probability that product j is chosen when the offering is (S, P) is given by

$$\mathbb{P}(j|S,P) = \sum_{k=1}^{K} \alpha_k \cdot \frac{e^{\beta_{jk} + (\beta'_{jk} - \beta_{jk}) \cdot \mathbb{1}[j \in P]}}{V_k(S,P) + \sum_{i \in S \setminus P} e^{\beta_{ik}} + \sum_{i \in P \setminus S} e^{\beta'_{ik}}}, \tag{9}$$

where

$$V_k(S,P) = \begin{cases} 1 + \sum_{i \notin S \setminus P} e^{\gamma_{ik}} + \sum_{i \notin P} e^{\gamma'_{ik}}, & \text{if } d_k \in S \\ 0, & \text{o.w.} \end{cases}$$

denotes the weight of the default product, which includes the shadow weights of the products not offered.

Under the above LC-GAM model, we computed the total loss by solving the following optimization problem:

$$\text{Total loss under LC-GAM} = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\beta}',\boldsymbol{\gamma},\boldsymbol{\gamma}'} -\frac{1}{|\mathcal{T}|} \sum_{(S,P)\in\mathcal{M}} M_{S,P} \sum_{j\in S} f_{j,S,P} \log\left(\frac{1}{f_{j,S,P}} \cdot \mathbb{P}(j|S,P,\boldsymbol{\beta},\boldsymbol{\beta}',\boldsymbol{\gamma},\boldsymbol{\gamma}')\right) \quad (10)$$

$$\text{s.t.} \sum_{k=1}^{K} \alpha_k = 1, \alpha_k \geq 0 \;\forall\, k$$

with $K = 5$, where we abuse notation to let $\mathbb{P}(j|S,P,\boldsymbol{\beta},\boldsymbol{\beta}',\boldsymbol{\gamma},\boldsymbol{\gamma}')$ denote the choice probability expression in (9) with the dependence on the model parameters made explicit. As for the LC-MNL model, the above optimization problem is non-convex in the parameters, therefore, we used the EM algorithm to solve it.

For each category, we verified that the LC-GAM model obtained from solving (10) is indeed outside the RUM class by ensuring that the fitted model parameters satisfy at least one of the following three conditions for some mixture component k and some product j: (a) $w_{jk} > v_{jk}$, (b) $w'_{jk} > v'_{jk}$, or (c) $v_{jk} > v'_{jk}$. Each of the first two conditions implies that the resulting GAM for segment k is outside the RUM class because of the observation in (8) above. The last condition also implies that the resulting GAM is outside the RUM class because rationality stipulates that promotion never decreases the attractiveness of a product.

Figure 4 (in Section 6.3.3 on p. 32) overlays the loss from fitting 5-class LC-GAM model onto Figure 3 (in Section 6.3.2 on p. 31), plotting the rationality loss against the total loss under the single-class MNL model. The plot shows how fitting the LC-GAM model allows us to breach the LoR for most of the 29 product categories. In fact, we observe that 5-class LC-GAM is able to obtain acceptable performance (loss < 0.03) for the Margarine/Butter category, which belongs to 'Region 3' and hence it is necessary to go beyond the RUM class to obtain acceptable performance.

Behaviors that are inconsistent with RUM models: Our analysis above shows that relaxing the RUM assumption can yield better fitting models. This finding is consistent with existing literature that has identified consumer choice behaviors that violate the RUM assumption:

- *Variety-seeking:* van Trijp (1995) defined variety seeking as "the biased behavioral response by some decision making unit to a specific item relative to previous responses within the same behavioral category, due to the utility inherent in variation per se, independent of the instrumental or functional value of the alternatives or items". This phenomenon is well documented in the marketing literature; see, for example, Hoyer and Ridgway (1984), Kahn and Lehmann (1991), Holbrook and Hirschman (1982), van Trijp et al. (1996). The frequent switching of brands from one week to the next due to variety seeking often results in non-transitive preferences for each customer and a violation of the RUM assumption in the aggregate.
- *Product confusion due to choice overload:* In a landmark study, Iyengar and Lepper (2000) showed that people are more likely to purchase gourmet jams or chocolates when offered a limited array of 6 choices rather than a more extensive array of 24 or 30 choices. This means that as we have more choices, the probability of no-purchase can increase due to product confusion; see Chernev et al. (2015) for the most recent comprehensive meta-analysis and references. This increase in the choice probability of an existing item when new items are added is inconsistent with the RUM principle.
- *Synergistic products:* Davis et al. (2011) discussed synergistic product categories in the context of nested logit models, where adding a product to a nest *increases* the probability of purchasing an existing product in the nest. A real-world example is the inclusion of "loss leaders" to drive traffic to a particular category. Another example is that of "halo" products (such as iPhones) that increase the sales of other offered products (such as Mac computers). Synergistic products can be captured using a nested logit model with the nest dissimilarity parameters taking values larger than one. Of course, when the nest dissimilarity parameters are larger than one, then the model instances cease to be a part of the RUM family; see Davis et al. (2011) for details.

The LC-GAM models can capture the choice behaviors specified in the second and third bullets.

Cost of going beyond the RUM family: Despite the benefits described above, given the current state of the literature, leaving the "well traversed" RUM class requires caution due to the following potential downside risks:

1. *Lack of a de facto non-RUM model class:* The existing literature in marketing, econometrics, and operations management does not offer a "go-to" model outside of the RUM model class. In fact, an arbitrary non-RUM model (such as the 5-class LCGAM) can provide a *worse* fit than the RUM class. Therefore, careful consideration is required to assess whether a candidate non-RUM model *will* indeed provide a better fit than the RUM class before adopting it.
2. *Lack of efficient computation schemes to fit the models:* Techniques for estimating the parameters of non-RUM choice models have received little attention from researchers so far. Consequently, data processing techniques and model specifications are not well understood yet.

3. *Lack of efficient computation schemes for decision-making:* General non-RUM models also lack efficient techniques for solving the assortment, pricing, and inventory optimization problems.

Our work has shown the boundaries of the RUM class, and the potential benefits of going beyond the RUM models. Thus, it lays the groundwork for future work to address the above challenges.

E.5 Factors that might influence rationality loss

For each product category, we define its market share concentration as follows:

$$\log|N| - \sum_{j\in N} m_j \log m_j, \qquad \text{where } m_j = \frac{\sum_{(S,P)\in\mathcal{M}} f_{j,S,P}\cdot M_{S,P}}{\sum_{(S,P)\in\mathcal{M}} M_{S,P}},$$

where m_j denotes the fraction of the market captured by product j and the market concentration is the KL-divergence between the uniform distribution and the market share distribution. In addition, for each category, we define a "hedonic" indicator variable that is equal to 1 if the purchases from that category are driven by its hedonic (or sensory) attributes (Baltas et al. 2017). We manually classified each product category to be hedonic or not, by applying the following rule: a category is hedonic if it is associated with food/snack (excl. condiments), drinks, or cigarettes; otherwise, it is not. Of the 29 categories, our rule classified the following 14 product categories as hedonic: beer, carbonated beverages, cigarettes, coffee, cold cereal, frozen dinner, frozen pizza, hot dog, milk, peanut butter, salt snacks, soup, spaghetti sauce, and yogurt. The remaining 15 categories were classified as non-hedonic. The classification we obtained is supported by the existing literature for the following categories: carbonated beverages, paper towel, peanut butter (Crowley et al. 1992), toothpaste, and yogurt (Givon 1984, Kahn et al. 1986, Baltas et al. 2017).

We regressed the rationality loss (loss) against the market concentration (mconc) and the hedonic indicator (hedi) variables. Table EC.2 reports the results from fitting three different models: $\text{loss}_j = \beta_{01} + \beta_{11}\cdot \text{mconc}_j + \varepsilon_j$ (Model 1), $\text{loss}_j = \beta_{02} + \beta_{22}\cdot \text{hedi}_j + \varepsilon_j$ (Model 2), and $\text{loss}_j = \beta_{03} + \beta_{13}\cdot \text{mconc}_j + \beta_{23}\cdot \text{hedi} + \varepsilon_j$ (Model 3). We observe that all the estimated coefficients are statistically significant.

We note that rationality loss is negatively correlated with the market concentration variable (Model 1). As noted in Section 6.3.4, categories with lower market concentration, such as yogurt, have frequent product introductions, and customers often exhibit variety seeking behavior, resulting in higher rationality loss because of the non-transitivity of preferences caused by frequent brand switching. Similarly, for categories with higher market concentration, such as diapers, customers tend to be brand loyal, generally resulting in transitive preferences and lower rationality loss.

Model 2 shows that rationality loss is positively correlated with the hedonic indicator variable. Existing work in marketing has shown that consumers exhibit complex choice behaviors when purchasing hedonic products. For example, consumer self-control issues may lead to a purchase of smaller quantities of hedonic products at a higher per unit price (e.g., chocolate); loss aversion and anticipation utility of hedonic products are higher; consumption of hedonic products can evoke a sense of guilt, and hedonic products are more likely to exhibit preference reversal, which is a phenomenon that violates the standard utility theory; see, for example, Wertenbroch (1998), Dhar and Wertenbroch (2000), O'curry and Strahilevitz (2001), Kivetz and Simonson (2002a,b), Okada (2005) and the references therein. These complex choice behaviors are not sufficiently well-captured by transitive preference orderings, leading to high rationality losses.

Finally, Model 3 shows that using the two variables together explains the variations in the rationality loss better, with a substantially higher R^2 value. The coefficients also remain statistically significant, indicating that the rationality loss of a category is driven by both the market concentration and hedonic attributes of the product category.

Our findings indicate that going beyond the RUM class may be particularly fruitful for product categories with low market concentrations (or high variety seeking) and hedonic attributes.

Table EC.2 Regression Results

	Dependent variable:		
	Rationality loss (KL-divergence (10^{-2}))		
	Model 1	Model 2	Model 3
Market concentration	-2.439^{**}		-2.011^{**}
	(0.999)		(0.914)
Hedonic indicator		1.455^{***}	1.274^{**}
		(0.498)	(0.473)
Constant	3.959^{***}	2.046^{***}	3.131^{***}
	(0.559)	(0.346)	(0.590)
Observations	29	29	29
R^2	0.181	0.240	0.360
Adjusted R^2	0.151	0.212	0.310
Residual Std. Error	1.391 (df = 27)	1.340 (df = 27)	1.254 (df = 26)
F Statistic	5.964^{**} (df = 1; 27)	8.539^{***} (df = 1; 27)	7.297^{***} (df = 2; 26)

Note: *p<0.1; **p<0.05; ***p<0.01
The values in the parentheses under the coefficients are the standard errors of the parameter estimates.

E.6 Robustness of our results under different loss metrics

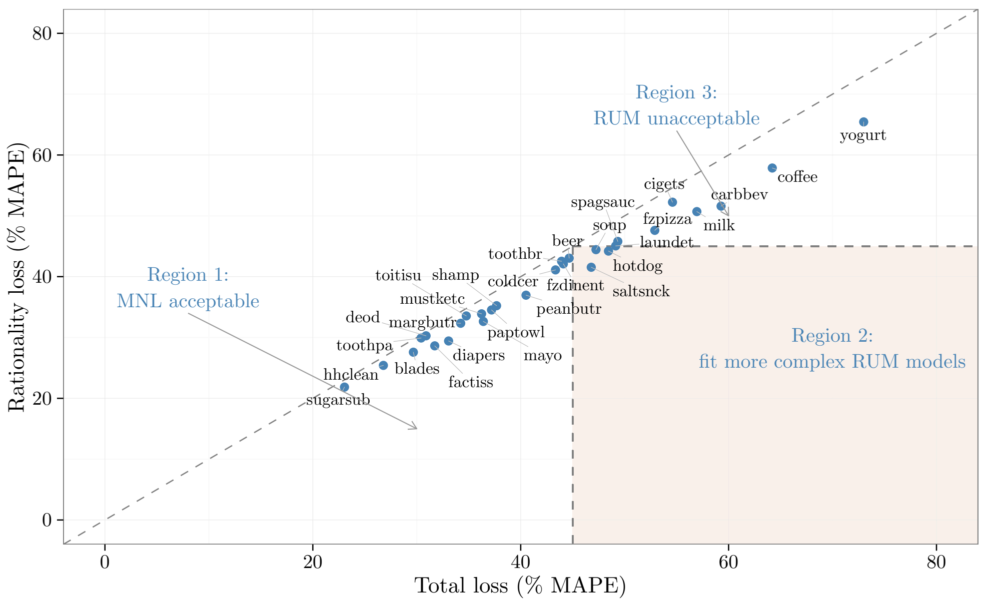


Figure EC.2 **Robustness across different loss metrics:** The figure plots the rationality loss against the total losses under the MNL model, both reported in terms of the mean absolute percentage error (MAPE). The plot region is partitioned assuming a MAPE of $< 45\%$ is acceptable. For categories in Region 1, the MNL model provides acceptable performance. Categories in Region 2 can benefit from fitting more complex RUM models. For categories in Region 3, one must go beyond RUM class to obtain acceptable performance.

Number of Product Categories		Classification under MAPE			Total
		Region 1	Region 2	Region 3	
Classification under KL-divergence	Region 1	12	1	1	14
	Region 2	3	1	1	5
	Region 3	3	1	6	10
Total		18	3	8	

Table EC.3 A contingency table showing the classification of the 29 product categories under the KL-divergence and MAPE loss metric. The table is based on the classification of each product category into three different regions in Figure 3 and Figure EC.2.

To test the robustness of our result, we use another loss metric based on the mean absolute percentage error (MAPE). Figure EC.2 shows a scatter plot of the rationality loss against the total losses under the MNL model under the MAPE metric. We use 45% as the cutoff threshold, and as in Figure 3, it partitions the plot into three regions. It turns out that the classification of products into the three regions is quite similar under both the KL-divergence loss metric and MAPE. Table EC.3 shows the number of product categories that fall into each region under the two metric. As see from Table EC.3, there are 19 = 12 + 1 + 6 product categories whose classification under both KL-divergence and MAPE loss metrics are exactly the same, representing 66% = 19/29 of the categories that we considered in our dataset. This shows that our method is fairly robust under different loss metric.

E.7 Robustness of our results across time

This section shows that the classification of product categories presented in Section 6.3.2 is robust across time. In Section 6.3.2, we classify the product categories into three different regions for the purpose of model selection. For a given threshold of acceptable performance loss, Region 1 comprises categories for which the MNL model is acceptable, Region 2 comprises the categories for which fitting more complex RUM model is required to obtain acceptable performance, and Region 3 comprises categories for which the RUM model is unacceptable. Figure 3 illustrates this classification. The rationality and total losses reported in the figure were computed using sales transactions data from weeks 1 and 2 of the calendar year 2007.

In order to test the robustness of this classification over time, we ran the same analysis using data from weeks 27 and 28 (about six months later) of the same year 2007. Figure EC.3 shows the corresponding classification of product categories. It turns out that the classification of products into the three regions is quite similar for both time periods. Table EC.4 shows the number of product categories that fall into each region for the two time periods. As seen from Table EC.4, there are 23 = 13 + 2 + 8 product categories that are classified into the same regions for both time periods, representing 79% = 23/29 of the categories considered in our dataset. This shows that our classification remains fairly robust over time.

Number of Product Categories		Classification during weeks 1-2			Total
		Region 1	Region 2	Region 3	
Classification during weeks 27-28	Region 1	13	1	0	14
	Region 2	1	2	2	5
	Region 3	1	1	8	10
Total		15	4	10	

Table EC.4 A contingency table showing the classification of the 29 product categories during the time periods weeks 1-2 and weeks 27-28. The table is based on the classification of the product categories into three different regions in Figures 3 and EC.3.

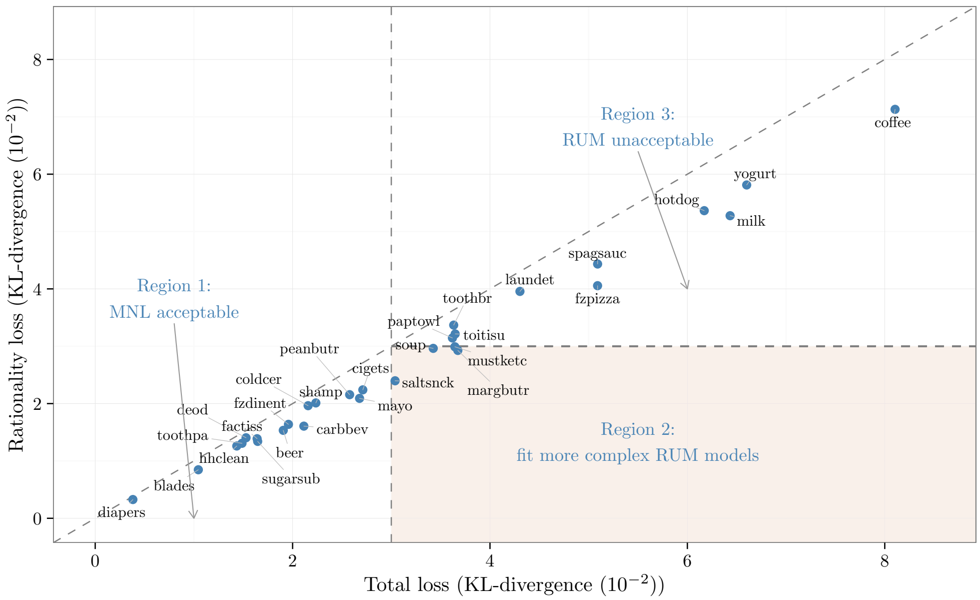


Figure EC.3 **Robustness across time:** Rationality and total losses computed using transactions data from weeks 27-28 of the calendar year 2007. The classification of the product categories into the three regions is similar to one computed using transactions data from weeks 1-2 of the same year, as shown in Figure 3.

References

Baltas, G., F. Kokkinaki, A. Loukopoulou. 2017. Does variety seeking vary between hedonic and utilitarian products? the role of attribute type. *Journal of Consumer Behaviour* .

Bodlaender, H. L., A. M. C. A. Koster. 2011. Treewidth computations II. Lower bounds. *Information and Computation* **209**(7) 1103–1119.

Bodlaender, H. L., T. Wolle, A. M. C. A. Koster. 2006. Contraction and treewidth lower bounds. *Journal of Graph Algorithms and Applications* **10**(1) 5–49.

Chernev, A., U. Böckenholt, J. Goodman. 2015. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology* **25**(2) 333–358.

Crowley, A. E., E. R. Spangenberg, K. R. Hughes. 1992. Measuring the hedonic and utilitarian dimensions of attitudes toward product categories. *Marketing Letters* **3**(3) 239–249.

Davis, J., G. Gallego, H. Topaloglu. 2011. Assortment optimization under variants of the nested logit model. Working Paper, Cornell University.

Dhar, R., K. Wertenbroch. 2000. Consumer choice between hedonic and utilitarian goods. *Journal of marketing research* **37**(1) 60–71.

Gallego, G., R. Ratliff, S. Shebalov. 2014. A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research* **63**(1) 212–232.

Givon, M. 1984. Variety seeking through brand switching. *Marketing Science* **3**(1) 1–22.

Holbrook, M. B., E. C. Hirschman. 1982. The experiential aspects of consumption: Consumer fantasies, feelings, and fun. *Journal of consumer research* **9**(2) 132–140.

Hoyer, W. D., N. M. Ridgway. 1984. Variety seeking as an explanation for exploratory purchase behavior: A theoretical model. Thomas C. Kinnear, ed., *Advances in Consumer Research*, vol. 11. Association for Consumer Research, Provo, UT, 114–119.

Iyengar, S. S., M. R. Lepper. 2000. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology* **79**(6) 995.

Kahn, B. E., M. U. Kalwani, D. G. Morrison. 1986. Measuring variety-seeking and reinforcement behaviors using panel data. *Journal of Marketing Research* **23**(2) 89–100.

Kahn, B. E., D. R. Lehmann. 1991. Modeling choice among assortments. *Journal of retailing* **67**(3) 274–299.

Kivetz, R., I. Simonson. 2002a. Earning the right to indulge: Effort as a determinant of customer preferences toward frequency program rewards. *Journal of Marketing Research* **39**(2) 155–170.

Kivetz, R., I. Simonson. 2002b. Self-control for the righteous: Toward a theory of precommitment to indulgence. *Journal of Consumer Research* **29**(2) 199–217.

O'curry, S., M. Strahilevitz. 2001. Probability and mode of acquisition effects on choices between hedonic and utilitarian options. *Marketing Letters* **12**(1) 37–49.

Okada, E. M. 2005. Justification effects on consumer choice of hedonic and utilitarian goods. *Journal of marketing research* **42**(1) 43–53.

Robertson, N., P. D. Seymour. 1986. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms* **7**(3) 309–322.

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM journal on computing* **1**(2) 146–160.

Train, K. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge University Press.

van Trijp, H. C. M. 1995. Variety-seeking in product choice behavior. theory with applications in the food domain. Ph.D. thesis, Wageningen University.

van Trijp, H. C. M., W. D. Hoyer, J. J. Inman. 1996. Why switch? product category-level explanations for true variety-seeking behavior. *Journal of Marketing Research* **33**(3) 281–292.

Wertenbroch, K. 1998. Consumption self-control by rationing purchase quantities of virtue and vice. *Marketing science* **17**(4) 317–337.