Chapter 8
Nonparametric Estimation of Choice Models

Srikanth Jagabathula and Ashwin Venkataraman

8.1 Introduction

Firms rely on demand predictions to make critical operational decisions. For example, firms need to know how customers respond to price changes in order to optimize the prices it charges. Traditionally, operational decision models relied on what is known as the "independent" demand model. As its name implies, an independent demand model assumes that the demand observed for a product is *independent* of the availability or characteristics, such as price, of other products. That is, the model ignores any cross-product cannibalization effects. Ignoring cross-product effects is hard to justify when products are close substitutes of each other; for example, products belonging to the same product category (e.g., different brands of toothpaste), different fare classes of an airline itinerary, different transportation modes (e.g., car, bus, train, etc.) are all close substitutes of each other. In such cases, ignoring the cross-product effects lead to biased demand estimates, especially when product prices and availability change over time. To deal with such cross-product effects, choice-based demand models have gained in popularity over the last couple of decades.

In the most general form, a choice model specifies the probability that a customer purchases a product from a given subset, or *offer set*, of products. If there are N products, then the model specifies choice probabilities for each of the 2^N subsets. Because the model is intractable in such a general form, existing literature

S. Jagabathula
Stern School of Business, New York University, New York, NY, USA
e-mail: sjagabat@stern.nyu.edu

A. Venkataraman (✉)
Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA
e-mail: ashwin.venkataraman@utdallas.edu

X. Chen et al. (eds.), *The Elements of Joint Learning and Optimization in Operations Management*, Springer Series in Supply Chain Management 18,
https://doi.org/10.1007/978-3-031-01926-5_8

has studied various model sub-classes with varying degrees of tractability. The most studied sub-class, by far, is the random utility maximization (RUM) class of models (McFadden, 1981). These models specify a joint distribution over product utilities and assume that each customer samples a utility vector from the underlying joint distribution and purchases the available product with the maximum utility. A special case of the RUM class that has received the most attention in the literature is the multinomial logit (MNL) model (see, e.g., Ben-Akiva et al., 1985; Train, 2009). Other special cases include the nested logit model, the d-level nested logit or the tree logit model (Li et al., 2015), the generalized extreme value (GEV) model, the mixed logit model, etc. These special cases differ in the assumptions they impose on the structure of the joint utility distribution. We refer the reader to Train's book (Train, 2009) and the overview by Gallego and Topaloglu (2019) for a detailed introduction to these and related choice models studied in the operations literature.

In this book chapter, we discuss recent developments in the literature on estimating the RUM class of models from observed sales transaction data. Sales transaction data provide historical choice observations: the product chosen and the other products on offer when the choice was made. These data are regularly collected by firms through their point-of-sale (POS) and inventory systems. Our focus will be specifically on nonparametric estimation techniques, which differ from the traditional, and more prevalent, parametric model estimation techniques. In the context of choice models, parametric models restrict the joint utility distribution to belong to a parametrized class of distributions. This additional structure lends them tractability, and the model parameters are typically estimated using standard model fitting techniques, such as the maximum likelihood estimation (MLE) technique. While parametric restrictions lend tractability, they typically also result in *model misspecification*, which occurs when the imposed restrictions exclude the ground-truth data generating distribution. Model misspecification leads to biased parameter estimates and inaccurate demand predictions. To alleviate this issue, nonparametric techniques do not restrict the joint utility distribution and allow it to be described by any member of the RUM model class. They then use sophisticated mathematical techniques to search for the model that has the best fit to the observed data. Nonparametric techniques generally work best when the volume of data is "large," which has increasingly been the case in the recent past because of firms' ability to collect highly fine-grained data.

We focus our discussion on broadly two types of techniques. The first technique deals with the so-called rank-based choice model (Mahajan and Van Ryzin, 2001). In this model, each product is treated as a bundle of features (e.g., color, weight, size, price, etc.), which remain fixed and do not vary. The model is fit on transaction data in which only the offer sets vary, and the trained model is used to predict demand for a heretofore unseen offer set. The model can accommodate varying product features by treating each product variant (e.g., the same product but with different prices) as a separate product. This modeling approach is ideal when product feature representations are not readily available (e.g., when purchases are driven by hedonistic features, such as taste, feel, etc.; see, for instance, Hoyer and

Ridgway, 1984; Kahn and Lehmann, 1991) and firms want to predict demand for various offer sets of their *existing* products, for which sufficient observed data exist. For example, airlines have existing transactions on customer bookings, which contain information on the purchase fare class and the corresponding offered fare classes for a set of customers. The airline wants to use this data to predict the expected demand for each combination of offered fare classes in order to determine the optimal collection of fare classes to open. Similarly, retailers (both online and offline) want to optimize the offered assortments of existing products to customers. One limitation of the rank-based model is that it cannot extrapolate demand to *new* products or new variants of existing products.

The second technique we discuss addresses the inability of the rank-based choice model to extrapolate demand. It deals with what we call the *nonparametric mixture of closed logit model* (NPMXCL model), which was considered in Jagabathula et al. (2020b).[1] This model assumes that all products have consistent feature representations and specifies a flexible functional form relating product features to choice probabilities. When trained on existing transaction data with "sufficient" variation in features, the model can extrapolate demand to heretofore unseen products or product variants. This model subsumes the rank-based model as a special case and is ideally suited for estimating price elasticities, optimizing discount levels and promotion mix, and determining the cannibalization effects of introducing new products.

Both techniques above formulate the estimation problem as a large-scale constrained convex optimization problem and build on recent developments within the machine learning (ML) literature to propose efficient algorithms for model estimation. We also discuss some of the theoretical guarantees that can be established for these methods.

The rest of the chapter is organized as follows. We first present an overview of the setup, notation, and the data model. We then discuss the model assumptions and the details of the corresponding estimation techniques for the rank-based model and the NPMXCL model. We then briefly review other nonparametric choice models proposed in the literature and conclude with some thoughts on future directions in nonparametric choice modeling.

Notation We first summarize notation that is common to the rest of the chapter. For any positive integer m, we let $[m]$ denote the set $\{1, 2, \ldots, m\}$, $\mathbf{0}_m$ denote the all-zeros vector in $\mathbb{R}^m$, and Δ_m denote the unit m-simplex in $\mathbb{R}^{m+1}$. Vectors are denoted by lower-case bold letters such as $\boldsymbol{x}$, $\boldsymbol{g}$, etc. For any multivariate function $h(\cdot)$ on the Euclidean space, $\nabla h(\cdot)$ denotes the gradient of $h(\cdot)$, i.e., the vector of partial derivatives with respect to each of the input variables. We let $\|\boldsymbol{x}\|$ denote the L^2-norm of any vector $\boldsymbol{x}$ in the Euclidean space. When we write $\boldsymbol{x}_1 > \boldsymbol{x}_2$ for vectors $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, we mean that each element of $\boldsymbol{x}_1$ is greater than the corresponding element in $\boldsymbol{x}_2$. For any set A, $|A|$ denotes its cardinality. Finally, $\langle \cdot, \cdot \rangle$ denotes the standard inner product in the Euclidean space.

[1] However, they did not introduce this nomenclature.

8.2 General Setup

We consider the setting of a firm whose offerings belong to a universe $[N] = \{1, 2, \ldots, N\}$ of N products. The firm collects choice data over a collection $\mathcal{M}$ of offer sets, where each offer set is a subset of the product universe $[N]$ offered to the customers. For each subset $S \in \mathcal{M}$, we let $y_{i,S} \in [0, 1]$ denote the observed fraction of customers who purchased product i when S was offered. Typically, the customer can leave without making a purchase, which is represented by a special product called the no-purchase or outside option. In our development below, the no-purchase option can be treated as any other product and consequently, we suppose that the product universe $[N]$ and the offer sets already include the no-purchase option. Note that we are implicitly assuming here that the firm can keep track of customers who visited with an intent to purchase but did not make a purchase. This can be done to a certain extent in the online e-commerce settings, but in the offline settings, the no-purchase observations are typically censored. We do not explicitly deal with this issue in this book chapter and suppose that the demand has been uncensored using other means.[2] We represent the observed data as the vector $\boldsymbol{y}_{\mathcal{M}} = (y_{i,S} : i \in S,\ S \in \mathcal{M})$. Let $M \stackrel{\text{def}}{=} \sum_{S \in \mathcal{M}} |S|$ denote the total number of choice observations, so that $\boldsymbol{y}_{\mathcal{M}} \in [0, 1]^M$.

As mentioned earlier, a choice model specifies the probability that a customer purchases a product from a given offer set. For the collection $\mathcal{M}$, we represent the collection of choice probabilities under a given model as the vector $\boldsymbol{g}_{\mathcal{M}} = (g_{i,S} : i \in S,\ S \in \mathcal{M})$ where $g_{i,S} \in [0, 1]$ is the probability of choosing product i from offer set S specified by the choice model. Estimating a model typically involves finding the model parameters that best fit the observed data, where the model misfit is measured using a *loss function*. More specifically, we measure the degree of model misfit using a non-negative loss function $\boldsymbol{g}_{\mathcal{M}} \mapsto \mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}})$ that measures the "distance" between the observed choice fractions $\boldsymbol{y}_{\mathcal{M}}$ and model predicted choice probabilities $\boldsymbol{g}_{\mathcal{M}}$. We consider loss functions $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \cdot)$ that are (strictly) convex in the second argument and have the property that $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = 0$ if and only if $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$. Letting $\mathcal{G}$ denote the set of all choice probability vectors (for the observed offer set collection) that can be generated by the choice model family of interest, we solve the following estimation problem:

$$\min_{\boldsymbol{g}_{\mathcal{M}} \in \mathcal{G}} \mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}). \qquad \text{(GENERAL ESTIMATION PROBLEM)}$$

[2] There are numerous papers that explicitly account for the demand censoring issue while estimating the choice model; see, for instance, Haensel and Koole (2011), Newman et al. (2014), and Abdallah and Vulcano (2020).

The following are two commonly used loss functions:

Example 1 (Log-Likelihood/Kullback–Leibler (KL) Divergence Loss Function) This loss function is defined as follows:

$$\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = -\sum_{S \in \mathcal{M}} M_S \sum_{i \in S} y_{i,S} \log\left(g_{i,S}/y_{i,S}\right),$$

where the weight $M_S > 0$ associated with offer set $S \in \mathcal{M}$ is equal to the number of customers who were offered the assortment S. Note that if $y_{i,S} = 0$ for some (i, S) pair, then the corresponding term is dropped from the loss objective. It can be verified that this loss function is non-negative since it is a weighted sum (with non-negative weights) of individual KL-divergence terms $-\sum_{i \in S} y_{i,S} \log(g_{i,S}/y_{i,S})$ between the distributions $\left(y_{i,S} : i \in S\right)$ and $\left(g_{i,S} : i \in S\right)$ for each $S \in \mathcal{M}$, which are always non-negative. For the same reason, we also have that $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = 0$ if and only if $y_{i,S} = g_{i,S}$ for all $i \in S$ and $S \in \mathcal{M}$. The loss function is strictly convex in the second argument, provided that the observed fractions for all choice observations are strictly positive, i.e., $\boldsymbol{y}_{\mathcal{M}} > \mathbf{0}_M$. This follows from the strict concavity of the logarithm function. Because the terms involving the observed choice fractions $(y_{i,S} : i \in S, S \in \mathcal{M})$ are constants for the optimization problem, it can be shown that minimizing the KL-divergence loss function is equivalent to maximizing the log-likelihood. Therefore, employing this loss function in the GENERAL ESTIMATION PROBLEM results in the maximum likelihood estimate (MLE).

Example 2 (Squared Norm Loss Function) This loss function is defined as

$$\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}_{\mathcal{M}}) = \left\|\boldsymbol{y}_{\mathcal{M}} - \boldsymbol{g}_{\mathcal{M}}\right\|^2.$$

It is easy to see that the squared norm loss function is non-negative and takes a value of 0 if and only if $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$. Further, it is strictly convex in $\boldsymbol{g}_{\mathcal{M}}$ for any fixed $\boldsymbol{y}_{\mathcal{M}}$.

Having introduced the general setup for the estimation problem, we now discuss in more detail two choice model families, the rank-based model and the NPMXCL model.

8.3 Estimating the Rank-Based Model

The rank-based choice model is the most general representation of the RUM class of models. Recall that a RUM model assumes that customers sample a utility for each of the products and choose the available product with the highest utility value. For finitely many products, it is clear that as far as the customer's choice is concerned, the actual utility values do not matter—only the preference ordering induced by the sampled utilities matter; see, for instance, Block and Marschak

(1960) and Mas-Colell et al. (1995). The rank-based choice model recognizes this and models the preferences of each customer as a ranking or preference ordering of the products. The preferences of a population of customers are, as a result, modeled as a probability distribution over rankings.

The rank-based choice model has origins in the classical preference and utility theory in economics and psychology (Block and Marschak, 1960; Manski, 1977; Falmagne, 1978; Barberá and Pattanaik, 1986). Most of the work in this area has focused on establishing theoretical properties of the model. For instance, Falmagne (1978) shows that a system of choice probabilities defined over all possible offer sets is consistent with a rank-based model if and only if all the Block–Marschak polynomials are non-negative; see also Barberá and Pattanaik (1986). McFadden (2005) provides additional necessary and sufficient conditions in the form of systems of linear inequalities, and shows how the different conditions relate to one another. For much of its history, the rank-based choice model has mostly served as a theoretical construct because estimating it from choice data is a significant computational challenge. Therefore, the literature on choice modeling has largely focused on specific parametric models, which impose additional structure on the utility distributions to trade off the restrictiveness of the models with the computational tractability of estimating them. Farias et al. (2013) was one of the first papers within the operations literature to tackle the computational challenge of estimating the rank-based model from choice data. They used ideas in linear programming to propose tractable techniques to predict revenues for new offer sets. Subsequent work (van Ryzin and Vulcano, 2015, 2017; Jagabathula and Rusmevichientong, 2017) further built on this paper to make the rank-based model operationally tractable, some of which has focused on estimating the model and solving the subsequent operational decision, such as the assortment or the pricing decision.

Before discussing the estimation of the rank-based model, we formally define the model. Let $\mathscr{P}$ denote the set of all permutations (or linear preference orders) of the N products, so that $|\mathscr{P}| = N!$ (N factorial). Each element $\sigma \in \mathscr{P}$ is a ranking of the N products, and for all $i \in [N]$, we let $\sigma(i)$ denote the *rank* of product i. We assume that if $\sigma(i) < \sigma(j)$, then product i is preferred over product j in the ranking σ. Given any offer set S, a customer chooses the product that is most preferred under her ranking σ. Let $\mathbb{1}[\sigma, i, S]$ denote 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 choice behavior of the customer population is then modeled as a probability distribution $\lambda : \mathscr{P} \to [0, 1]$ over the permutations with $\lambda(\sigma)$ denoting the probability that a customer uses the ranking σ when making a purchase. Because λ is a probability distribution, we have that $\lambda(\sigma) \geq 0$ for all $\sigma \in \mathscr{P}$ and $\sum_{\sigma \in \mathscr{P}} \lambda(\sigma) = 1$.

Given any distribution over rankings λ, the vector of choice probabilities for the offer set collection $\mathcal{M}$ under the rank-based model is given by:

$$\boldsymbol{g}_{\mathcal{M}}(\lambda) = (g_{i,S}(\lambda) \colon i \in S,\ S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(\lambda) = \sum_{\sigma \in \mathscr{P}} \mathbb{1}[\sigma, i, S] \cdot \lambda(\sigma). \tag{8.1}$$

The set of all such probability vectors consistent with a rank-based model is denoted by $\mathcal{G}(\mathscr{P})$:

$$\mathcal{G}(\mathscr{P}) = \left\{ \boldsymbol{g}_M(\lambda) \mid \lambda : \mathscr{P} \to [0,1], \sum_{\sigma \in \mathscr{P}} \lambda(\sigma) = 1 \right\}. \tag{8.2}$$

The estimation problem for the rank-based model can then be formulated by plugging in $\mathcal{G} = \mathcal{G}(\mathscr{P})$ in the GENERAL ESTIMATION PROBLEM. However, solving the problem in this form poses some difficulties. This is because the loss function depends on the distribution λ only through the predicted choice probability vector $\boldsymbol{g}_M(\lambda)$, and, therefore, the underlying distribution is not directly identifiable in general. In fact, Sher et al. (2011) showed that if $N \geq 4$, there are multiple distributions over rankings that are consistent with *any* given collection of choice probabilities. The idea is that the choice probabilities impose $O(2^N)$ degrees of freedom (corresponding to all the subsets of $[N]$) whereas the space of distributions has $O(N!) = O(2^{N \log N})$ degrees of freedom.

To see this fact more explicitly, we consider an alternate representation of $\mathcal{G}(\mathscr{P})$. For each $\sigma \in \mathscr{P}$, let $\boldsymbol{f}(\sigma) \in \{0,1\}^M$ be the vector of indicators that determine whether product i is chosen from offer set S under ranking σ:

$$\boldsymbol{f}(\sigma) = (\mathbb{1}[\sigma, i, S] \colon \; S \in \mathcal{M}, i \in S), \tag{8.3}$$

and let $\mathcal{F}(\mathscr{P}) \stackrel{\text{def}}{=} \{\boldsymbol{f}(\sigma) \colon \sigma \in \mathscr{P}\}$ denote the set of all such indicator vectors. Now, consider the convex hull of the set $\mathcal{F}(\mathscr{P})$, which we denote as $\text{conv}(\mathcal{F}(\mathscr{P}))$, defined as:

$$\text{conv}\,(\mathcal{F}(\mathscr{P})) = \left\{ \sum_{\boldsymbol{f} \in \mathcal{F}(\mathscr{P})} \alpha_{\boldsymbol{f}} \boldsymbol{f} \colon \alpha_{\boldsymbol{f}} \geq 0 \; \forall \boldsymbol{f} \in \mathcal{F}(\mathscr{P}), \sum_{\boldsymbol{f} \in \mathcal{F}(\mathscr{P})} \alpha_{\boldsymbol{f}} = 1 \right\}.$$

Then, using the above equations it can be verified that $\mathcal{G}(\mathscr{P}) = \text{conv}(\mathcal{F}(\mathscr{P}))$. This shows that $\mathcal{G}(\mathscr{P})$ is a convex polytope in $\mathbb{R}^M$. While $\mathcal{G}(\mathscr{P})$ as defined in (8.2) appears to have a dependence on $N!$ variables, in practice the number of extreme points of $\mathcal{G}(\mathscr{P}) = \text{conv}(\mathcal{F}(\mathscr{P}))$ can be (significantly) smaller than $N!$ (N factorial). This is because two different rankings $\sigma \neq \sigma'$ may result in the same vector of indicators $\boldsymbol{f}(\sigma) = \boldsymbol{f}(\sigma')$ as in the following example:

Example 3 (Complexity of $\mathcal{G}(\mathscr{P})$ Under Market Shares Data) Suppose that the firm collects only market shares data, so that the offer set collection $\mathcal{M} = \{[N]\}$. In this case $M = N$ and it follows that each $\boldsymbol{f}(\sigma) \in \{0,1\}^N$ with $\boldsymbol{f}(\sigma_1) = \boldsymbol{f}(\sigma_2)$ for any two rankings σ_1, σ_2 in which the top-ranked product is the same. Consequently, $|\mathcal{F}(\mathscr{P})| = N \ll N!$ (N factorial). Moreover, it can be verified that the number of extreme points of $\text{conv}(\mathcal{F}(\mathscr{P}))$ is, in fact, N.

More generally, the number of extreme points of $\text{conv}(\mathcal{F}(\mathscr{P}))$, which is at most $|\mathcal{F}(\mathscr{P})|$, depends on the variation amongst offer sets in $\mathcal{M}$. Therefore, $\text{conv}(\mathcal{F}(\mathscr{P}))$ is a more succinct representation of $\mathcal{G}(\mathscr{P})$.

With the above development, the GENERAL ESTIMATION PROBLEM for the rank-based model takes the form:

$$\min_{\boldsymbol{g} \in \text{conv}(\mathcal{F}(\mathscr{P}))} \mathsf{loss}(\boldsymbol{g}), \quad \text{(RANK-BASED MODEL ESTIMATION PROBLEM)}$$

where we drop the explicit dependence of the set collection $\mathcal{M}$ on the predicted choice probabilities, and the observed choice fractions $\boldsymbol{y}_{\mathcal{M}}$ on the loss function for notational convenience. Since the constraint set is a convex polytope and the objective function is convex, the RANK-BASED MODEL ESTIMATION PROBLEM is a constrained convex program. In theory, it can be solved using standard methods for convex optimization. The challenge, however, is two-fold: (a) the constraint polytope may not have an efficient description and (b) decomposing a candidate solution $\boldsymbol{g}$ into the corresponding proportions $\boldsymbol{\alpha}$ (and, therefore, the underlying distribution λ) is itself a hard problem. Note that the distribution is required so that out-of-sample choice predictions can be made. To address these issues, Jagabathula and Rusmevichientong (2019) (henceforth JR) used the conditional gradient algorithm, which, as we will see shortly, transforms the convex optimization problem into solving a series of linear optimization problems. But first, we show that the RANK-BASED MODEL ESTIMATION PROBLEM has a unique optimal solution:

Theorem 1 (Unique Optimal Solution) *For any strictly convex loss function* $\mathsf{loss}(\cdot)$ *over the domain* $\text{conv}(\mathcal{F}(\mathscr{P}))$*, the* RANK-BASED MODEL ESTIMATION PROBLEM *has a unique optimal solution.*

Proof We prove this result by contradiction. Suppose, if possible, there exist two optimal solutions $\boldsymbol{g}_1^* \neq \boldsymbol{g}_2^*$ and let $\mathsf{loss}^* = \mathsf{loss}(\boldsymbol{g}_1^*) = \mathsf{loss}(\boldsymbol{g}_2^*)$. By strict convexity of $\mathsf{loss}(\cdot)$, it follows that for any $\delta \in (0, 1)$:

$$\begin{aligned} \mathsf{loss}(\delta \boldsymbol{g}_1^* + (1-\delta)\boldsymbol{g}_2^*) &< \delta \cdot \mathsf{loss}(\boldsymbol{g}_1^*) + (1-\delta) \cdot \mathsf{loss}(\boldsymbol{g}_2^*) \\ &= \delta \cdot \mathsf{loss}^* + (1-\delta) \cdot \mathsf{loss}^* = \mathsf{loss}^*. \end{aligned}$$

Since, by definition, $\text{conv}(\mathcal{F}(\mathscr{P}))$ is convex, it follows that $\delta \boldsymbol{g}_1^* + (1-\delta)\boldsymbol{g}_2^* \in \text{conv}(\mathcal{F}(\mathscr{P}))$ is a feasible solution to the RANK-BASED MODEL ESTIMATION PROBLEM. But this contradicts the assumption that loss^* is the optimal objective and, therefore, the optimal solution must be unique. □

8.3.1 Estimation via the Conditional Gradient Algorithm

As mentioned above, JR proposed to solve the RANK-BASED MODEL ESTIMATION PROBLEM using the conditional gradient algorithm. We begin with some background on the algorithm and then discuss its application for estimating the rank-based choice model.

Background The conditional gradient (hereafter CG) algorithm (aka Frank–Wolfe) algorithm (Clarkson, 2010; Jaggi, 2013) is an iterative method for solving optimization problems of the form

$$\min_{\boldsymbol{x}\in\mathcal{D}} h(\boldsymbol{x}), \tag{8.4}$$

where $h(\cdot)$ is a differentiable convex function and $\mathcal{D}$ is a compact convex region in the Euclidean space. It is in fact a generalization of the original algorithm proposed by Frank and Wolfe (1956), who considered solving quadratic programming problems with linear constraints. Starting from an arbitrary feasible point $\boldsymbol{x}^{(0)} \in \mathcal{D}$, in each iteration $k \geq 1$, the algorithm finds a *descent direction* $\boldsymbol{d}^{(k)}$ such that $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{d}^{(k)} \right\rangle < 0$ and takes a suitable step in that direction. The algorithm computes such a descent direction by optimizing the linear approximation of $h(\cdot)$ at the current iterate $\boldsymbol{x}^{(k-1)}$ over the feasible domain $\mathcal{D}$. That is, it solves the following problem:

$$\boldsymbol{v}^{(k)} \in \arg\min_{\boldsymbol{v}\in\mathcal{D}} \; h(\boldsymbol{x}^{(k-1)}) + \left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v} - \boldsymbol{x}^{(k-1)} \right\rangle. \quad \text{(FRANK–WOLFE STEP)}$$

Because the objective function in the FRANK–WOLFE STEP is linear in $\boldsymbol{v}$, the optimal solution $\boldsymbol{v}^{(k)}$ is an extreme point of $\mathcal{D}$. Having found the extreme point $\boldsymbol{v}^{(k)}$, the algorithm updates the solution by taking a step along the direction $\boldsymbol{d}^{(k)} \stackrel{\text{def}}{=} \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)}$obtaining $\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} + \gamma^{(k)} \cdot \boldsymbol{d}^{(k)}$ for some step size $\gamma^{(k)} \in [0, 1]$. Since $\boldsymbol{d}^{(k)}$ is a descent direction, it can be shown that for a suitable choice of $\gamma^{(k)}$, we have $h(\boldsymbol{x}^{(k)}) < h(\boldsymbol{x}^{(k-1)})$ so that moving in the direction of $\boldsymbol{v}^{(k)}$ ensures an improving solution; see, e.g., Nocedal and Wright (2006).[3] In the classical Frank–Wolfe algorithm, the step size was fixed to $\gamma^{(k)} = 2/(k+2)$. A standard alternative is to do a line-search for the optimal step size in each iteration to obtain

$$\gamma^{(k)} \in \arg\min_{\gamma\in[0,1]} \; h\left(\boldsymbol{x}^{(k-1)} + \gamma \cdot \boldsymbol{d}^{(k)}\right).$$

[3] This is true as long as $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)} \right\rangle < 0$. If $\left\langle \nabla h(\boldsymbol{x}^{(k-1)}), \boldsymbol{v}^{(k)} - \boldsymbol{x}^{(k-1)} \right\rangle \geq 0$, then the convexity of $h(\cdot)$ implies that $h(\boldsymbol{x}) \geq h(\boldsymbol{x}^{(k-1)})$ for all $\boldsymbol{x} \in \mathcal{D}$ and consequently, $\boldsymbol{x}^{(k-1)}$ is an optimal solution.

Note that the new iterate $\boldsymbol{x}^{(k)}$ remains feasible; this follows because $\boldsymbol{x}^{(k)}$ is a convex combination of $\boldsymbol{x}^{(k-1)}$ and $\boldsymbol{v}^{(k)}$ and $\mathcal{D}$ is convex. Such feasibility of new iterates is the main benefit of the CG algorithm compared to other classical algorithms such as gradient descent, which may take infeasible steps that are then projected back onto the feasible region; such projection steps are usually computationally expensive. Another feature of the algorithm is that the solution at any iteration k is a convex combination of the initial solution $\boldsymbol{x}^{(0)}$ and the extreme points $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(k)}$.

The CG algorithm is particularly attractive when solving the FRANK–WOLFE STEP is "easy"—for instance, if $\mathcal{D}$ is a polyhedron, it reduces to an LP. The CG algorithm has generated tremendous interest in the ML community for solving large-scale convex optimization problems in the recent past because of its "projection-free" property and ability to deal with structured constraint sets. The interested reader is referred to Jaggi's excellent thesis (Jaggi, 2011) for a more thorough development of the algorithm along with example applications.

We now apply the CG algorithm to solve the RANK-BASED MODEL ESTIMATION PROBLEM. This problem is exactly in the form (8.4) above with $h(\cdot) = \mathsf{loss}(\cdot)$ and $\mathcal{D} = \text{conv}(\mathcal{F}(\mathscr{P}))$. We initialize the algorithm by selecting an initial set of rankings $\mathscr{P}^{(0)} \subseteq \mathscr{P}$ and proportions $\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathscr{P}^{(0)}|-1}$, and setting $\boldsymbol{g}^{(0)} = \sum_{\sigma \in \mathscr{P}^{(0)}} \alpha_\sigma^{(0)} \boldsymbol{f}(\sigma)$, which by definition belongs to $\text{conv}(\mathcal{F}(\mathscr{P}))$.[4] However, we need to ensure that the initial loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are both bounded; this aspect is discussed in more detail in Sect. 8.3.1.3 below. Then, in each iteration $k \geq 1$, the FRANK–WOLFE STEP is of the form:

$$\min_{\boldsymbol{v} \in \text{conv}(\mathcal{F}(\mathscr{P}))} \mathsf{loss}(\boldsymbol{g}^{(k-1)}) + \left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle. \tag{8.5}$$

As mentioned earlier, the optimal solution to the above subproblem occurs at an extreme point of the feasible set $\text{conv}(\mathcal{F}(\mathscr{P}))$. Because this set is the convex hull of the vectors in $\mathcal{F}(\mathscr{P})$, the set of extreme points must be a subset of $\mathcal{F}(\mathscr{P})$. Consequently, problem (8.5) is equivalent to the following:

$$\min_{\boldsymbol{v} \in \mathcal{F}(\mathscr{P})} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle \equiv \min_{\sigma \in \mathscr{P}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\sigma) - \boldsymbol{g}^{(k-1)} \right\rangle, \tag{8.6}$$

where the equivalence follows from the definition of $\mathcal{F}(\mathscr{P})$. Let $\sigma^{(k)} \in \mathscr{P}$ denote an optimal solution to (8.6); we discuss how to solve it in more detail in Sect. 8.3.1.1 below. This means that the CG algorithm is iteratively adding rankings $\sigma^{(1)}, \sigma^{(2)}, \ldots$ to the support of the distribution. Consequently, we term subproblem (8.6) as the SUPPORT FINDING STEP.

As mentioned above, the standard variant of the CG algorithm does a line-search to compute the optimal step size, which results in maximum improvement in the objective value. An alternative is the "fully corrective" Frank–Wolfe (FCFW)

[4] We abuse notation and denote $\alpha_{\boldsymbol{f}(\sigma)}$ as α_σ for any $\sigma \in \mathscr{P}$ in the remainder of this section.

Algorithm 1 CG algorithm for solving the RANK-BASED MODEL ESTIMATION PROBLEM

Initialize: $k \leftarrow 0$; $\mathscr{P}^{(0)} \subseteq \mathscr{P}$; $\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathscr{P}^{(0)}|-1}$; $\boldsymbol{g}^{(0)} = \sum_{\sigma \in \mathscr{P}^{(0)}} \alpha_\sigma^{(0)} \boldsymbol{f}(\sigma)$ s.t. $\mathsf{loss}(\boldsymbol{g}^{(0)})$, $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded
while stopping condition is not met **do**
  $k \leftarrow k + 1$
  Compute $\sigma^{(k)} \in \arg\min_{\sigma \in \mathscr{P}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\sigma) - \boldsymbol{g}^{(k-1)} \right\rangle$ (SUPPORT FINDING STEP)
  Update support of rankings $\mathscr{P}^{(k)} \leftarrow \mathscr{P}^{(k-1)} \cup \left\{\sigma^{(k)}\right\}$
  Compute $\boldsymbol{\alpha}^{(k)} \in \arg\min_{\boldsymbol{\alpha} \in \Delta_{|\mathscr{P}^{(k)}|-1}} \mathsf{loss}\left(\sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma \boldsymbol{f}(\sigma)\right)$ (PROPORTIONS UPDATE STEP)
  Update support of rankings $\mathscr{P}^{(k)} \leftarrow \left\{\sigma \in \mathscr{P}^{(k)} : \alpha_\sigma^{(k)} > 0\right\}$
  Update $\boldsymbol{g}^{(k)} \leftarrow \sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma^{(k)} \boldsymbol{f}(\sigma)$
end while
Output: rankings $\mathscr{P}^{(k)}$ and proportions $(\alpha_\sigma^{(k)} : \sigma \in \mathscr{P}^{(k)})$

variant (Shalev-Shwartz et al., 2010), which after finding the extreme point $\boldsymbol{v}^{(k)}$ in the FRANK–WOLFE STEP, re-optimizes the objective function over the convex hull $\mathrm{conv}(\{\boldsymbol{x}^{(0)}, \boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}\})$ of the initial solution and extreme points found so far. When applied to our context, the algorithm computes weights $\boldsymbol{\alpha}^{(k)} = (\alpha_\sigma^{(k)} : \sigma \in \mathscr{P}^{(k)})$ that minimize the loss function $\mathsf{loss}(\cdot)$ over the set $\mathrm{conv}\left(\left\{\boldsymbol{f}(\sigma) : \sigma \in \mathscr{P}^{(k)}\right\}\right)$, where $\mathscr{P}^{(k)}$ is the set of rankings recovered up to iteration k (see the notation in Algorithm 1). It then obtains the next iterate as $\boldsymbol{g}^{(k)} := \sum_{\sigma \in \mathscr{P}^{(k)}} \alpha_\sigma^{(k)} \boldsymbol{f}(\sigma)$. The weights $\boldsymbol{\alpha}^{(k)}$ represent the proportions of each ranking and consequently, we call this the PROPORTIONS UPDATE STEP. The fully corrective variant of the CG algorithm makes more progress (in terms of the improvement in the objective value) in each iteration than the line-search variant and is, therefore, most suited when the FRANK–WOLFE STEP is hard to solve. The entire procedure is summarized in Algorithm 1.

Remark We note that van Ryzin and Vulcano (2015) proposed a *market discovery algorithm* for obtaining the MLE of the rank-based choice model using a column generation procedure. Though the authors derived their algorithm using duality arguments, it can be verified that their procedure is identical to the one obtained from solving the RANK-BASED MODEL ESTIMATION PROBLEM with the KL-divergence loss function using the CG algorithm.

Next, we discuss the details of how to solve each of the SUPPORT FINDING and PROPORTIONS UPDATE STEPS.

8.3.1.1 Solving the SUPPORT FINDING STEP

Noting that $\boldsymbol{f}(\sigma) = (\mathbb{1}[\sigma, i, S] : S \in \mathcal{M}, i \in S)$, the SUPPORT FINDING STEP can be written as follows:

$$\min_{\sigma \in \mathscr{P}} \sum_{S \in \mathcal{M}} \sum_{i \in S} \left(\nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)}) \right)_{i,S} \cdot \mathbb{1}[\sigma, i, S]. \tag{8.7}$$

This problem requires us to find a ranking with the minimum "cost," which is referred to as the rank aggregation problem in the ranking literature (Dwork et al., 2001) and is known to be NP-hard; see, for instance, van Ryzin and Vulcano (2015) and Jagabathula and Rusmevichientong (2019). In practice, subproblem (8.7) does not need to be solved to optimality and any feasible solution that generates a descent direction is sufficient to ensure an improving solution in Algorithm 1. Below we discuss a few different approaches that can be used to obtain an approximate solution.

Mixed Integer Program (MIP) Formulation van Ryzin and Vulcano (2015, Section 4.3.2) formulated a special case of the rank aggregation subproblem (8.7), which they referred to as the "Type Discovery Subproblem," as an MIP. In particular, they considered the case of individual purchase transactions where a single transaction is observed for each offer set. The same formulation extends to the aggregated data setting, which we present below.[5]

To simplify the formulation, we let $\mu_{i,S} = \left(\nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)})\right)_{i,S}$. We encode the ranking σ using binary decision variables $b_{ij} \in \{0, 1\}$ for all $i, j \in [N], i \neq j$, defined so that $b_{ij} = 1$ if and only if product i is preferred to product j, i.e., $\sigma(i) < \sigma(j)$. Further, we let $w_{i,S} = \mathbb{1}[\sigma, i, S]$ and denote the collection of decision variables as $\boldsymbol{b} = (b_{ij} : i, j \in [N], i \neq j)$, and $\boldsymbol{w} = (w_{i,S} : S \in \mathcal{M}, i \in S)$. Then, subproblem (8.7) is equivalent to the following MIP:

$$\min_{\boldsymbol{b}, \boldsymbol{w}} \quad \sum_{S \in \mathcal{M}} \sum_{i \in S} \mu_{i,S} \cdot w_{i,S} \tag{8.8a}$$

$$\text{s.t. } b_{ij} + b_{ji} = 1 \quad \forall\, i, j \in [N], i < j \tag{8.8b}$$

$$b_{ij} + b_{jl} + b_{li} \leq 2 \quad \forall\, i, j, l \in [N], i \neq j \neq l \tag{8.8c}$$

$$w_{j,S} \leq b_{ji} \quad \forall S \in \mathcal{M}, \forall i, j \in S, i \neq j \tag{8.8d}$$

$$\sum_{j \in S} w_{j,S} = 1 \quad \forall S \in \mathcal{M} \tag{8.8e}$$

$$b_{ij} \in \{0, 1\} \forall\, i, j \in [N], i \neq j \tag{8.8f}$$

$$w_{i,S} \in \{0, 1\} \forall\, S \in \mathcal{M}, i \in S. \tag{8.8g}$$

The constraint (8.8b) ensures that either product i is preferred to product j or j is preferred to i in the ranking. The second constraint (8.8c) enforces transitivity amongst any three products in the ranking: if product i is preferred to j and j is

[5] Mišić (2016) also proposed a similar formulation for estimating the rank-based choice model with an L^1-norm loss function using a column generation approach.

preferred to l, then i must be preferred to l. The third constraint (8.8d) encodes the consistency of the indicator variables $\mathbb{1}[\sigma, i, S]$; in particular, if $w_{j,S} = 1$, then it means that product j is the most preferred product from offer set S. This implies that we must have $b_{ji} = 1$ for all $i \in S \setminus \{j\}$, i.e., j is preferred over all other products in S. The fourth constraint (8.8e) ensures that only one of the indicator variables $\mathbb{1}[\sigma, i, S]$ is non-zero for each offer set S. The objective function (8.8a) is exactly the objective in (8.7). The formulation has $O(N^2 + M)$ binary variables, and $O(N^3 + N^2\,|\mathcal{M}|)$ constraints. Again, note that MIP (8.8) does not need to be solved to optimality, all we need is a feasible solution that generates a descent direction. Given any feasible solution $(\boldsymbol{b}, \boldsymbol{w})$, the corresponding ranking σ can be computed by setting $\sigma(i) = 1 + \sum_{j \neq i} b_{ji}$ for all $i \in [N]$.

Leverage Structure in Observed Offer Set Collection Though the rank aggregation subproblem (8.7) is NP-hard in general, JR showed that if the observed offer set collection $\mathcal{M}$ possesses certain structures, it can be solved efficiently. The structure is captured via a *choice graph* over the observed offer sets: each offer set is a vertex and the edges capture relationships amongst the most preferred products (under any ranking) in the different offer sets. They show that subproblem (8.7), which they refer to as the RANK AGGREGATION LP, can be formulated as a DP or LP over the choice graph with linear or polynomial complexity (in N and $|\mathcal{M}|$) for offer set collections that commonly arise in retail and revenue management settings. See Section 3 in JR for more details.

Local Search Heuristic A simple method to find an approximate solution to (8.7) is the local search heuristic that was proposed in Mišić (2016) and Jagabathula and Rusmevichientong (2017). This heuristic starts with a randomly chosen ranking and then tries to find a better solution by evaluating all "neighboring" rankings obtained by swapping the positions of any two products. The procedure is repeated until no neighboring ranking yields a smaller objective value for (8.7), resulting in a locally optimal solution $\hat{\sigma}$. If $\hat{\sigma}$ does not produce an improving solution in Algorithm 1, which can be verified by checking if $\boldsymbol{f}(\hat{\sigma}) - \boldsymbol{g}^{(k-1)}$ is a descent direction, i.e., $\left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{f}(\hat{\sigma}) - \boldsymbol{g}^{(k-1)} \right\rangle < 0$, then we redo the search starting from a different ranking, until we exhaust a limit on the number of tries.

8.3.1.2 Solving the PROPORTIONS UPDATE STEP

When compared to the SUPPORT FINDING STEP, THE PROPORTIONS UPDATE STEP is easier to solve because the corresponding subproblem is itself a convex program over the unit simplex $\Delta_{|\mathscr{P}^{(k)}|-1}$. It can be solved via the "away steps" variant of the CG algorithm described in Sect. 8.4.1.2, which promotes recovery of sparse distributions. Note that in line 7 in Algorithm 1, we drop the rankings with zero probability mass from the support, decreasing the support size and resulting in a sparser distribution. Another approach to solving the PROPORTIONS UPDATE STEP is to use the expectation-maximization (EM) algorithm proposed by van Ryzin and Vulcano (2017), which was utilized by the same authors in their market discovery

algorithm (van Ryzin and Vulcano, 2015) for estimating the rank-based choice model. An appealing feature of this approach is that the M-step involves closed-form updates for the proportions $\boldsymbol{\alpha}$ and, therefore, is simple to implement.

8.3.1.3 Initialization and Stopping Criterion

Line 1 in Algorithm 1 specifies that the initial collection of rankings $\mathscr{P}^{(0)}$ should be chosen such that the loss function and its gradient are bounded. In particular, for the KL-divergence loss function, choosing $\mathscr{P}^{(0)} = \left\{\sigma^{(0)}\right\}$ (and $\boldsymbol{\alpha}^{(0)} = (1)$) is not possible since this results in $g_{i,S}^{(0)} = 0$ for any (i, S) where $\mathbb{1}[\sigma^{(0)}, i, S] = 0$, making the initial loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ unbounded. van Ryzin and Vulcano (2015) initialized their method with N rankings, with each product $i \in [N]$ being the top-ranked product in exactly one ranking.[6] This ensures that $\boldsymbol{g}^{(0)} > \mathbf{0}_M$ so that both $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and the gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded. Jagabathula and Rusmevichientong (2017) considered an alternative approach where they start with a 'sales ranking' in which products are ranked according to their aggregate sales (across all offer sets), with higher sales products being more preferred in the ranking. Then, they obtain N rankings by modifying the sales ranking: ranking i is obtained by moving product i to the top-rank while the remaining products are shifted down in the ranking. Again, this initialization ensures that $\boldsymbol{g}^{(0)} > \mathbf{0}_M$.

Depending on the end goal, different stopping conditions may be used to terminate the algorithm. If the objective is to get the best possible fit to the data, then ideally we would like to run the algorithm until we are "close" to the optimal solution $\boldsymbol{g}^*$ of the RANK-BASED MODEL ESTIMATION PROBLEM. If the SUPPORT FINDING STEP can be solved optimally in each iteration, then its solution can be used to construct an upper bound on the *optimality gap* of the current solution $\boldsymbol{g}^{(k)}$, defined as $\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*)$; see Jaggi (2011) for details. Consequently, we can choose to terminate the algorithm when $\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*) \leq \varepsilon$ for some small $\varepsilon > 0$. An alternative approach is to stop when the absolute (or relative) change in the loss function objective is smaller than some pre-defined threshold. On the other hand, if the objective is to achieve good predictive performance out-of-sample, then the above approach may not work well as the final support may have a large number of rankings and thus overfit to the observed choice data. In such cases, standard information-theoretic measures proposed in the mixture modeling literature (McLachlan and Peel, 2004) such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), etc. that penalize overly complex mixture models or ML techniques such as cross-validation can be used for model selection. The approach used in Jagabathula et al. (2020b) was to limit the number of iterations of the algorithm based on an upper bound on the support size of rankings that one

[6] The remaining products in each ranking can be chosen arbitrarily.

is interested in finding. This idea is inspired by the *early stopping* rule in the ML literature (Yao et al., 2007; Prechelt, 2012).

8.3.2 Convergence Guarantee for the Estimation Algorithm

We can establish a convergence rate guarantee for the iterates $\boldsymbol{g}^{(k)}$ generated by Algorithm 1. Since the guarantee is identical to the case of the NPMXCL model discussed below, we do not repeat it here and refer the reader to Sect. 8.4.2 for the formal result. However, an interesting question is whether the special structure of the polytope $\text{conv}(\mathcal{F}(\mathscr{P}))$ can be leveraged to come up with stronger convergence rates. Of course, the result does not address recovery of the underlying distribution over rankings since it is not identifiable in general, as discussed earlier. To identify the distribution, additional constraints need to be imposed. One such approach was taken by Farias et al. (2013) where the goal was to find a distribution over rankings compatible with observed transaction data that produces the worst-case revenue for a given fixed assortment. The authors showed that such a distribution is approximately the *sparsest* rank-based choice model that explains the observed data.

8.4 Estimating the Nonparametric Mixture of Closed Logit (NPMXCL) Model

The rank-based model does not have the ability to extrapolate demand to new products or newer variants of existing products. One approach to address this issue was considered in Jagabathula and Rusmevichientong (2017). These authors extended the rank-based model to accommodate products with varying prices by adding a random *consideration set* layer on top of the rank-based model. Their consideration set model assumes that customers sample a threshold parameter and consider for purchase only those products who prices are less than the sampled threshold. From among the considered products, the customers then choose according to a rank-based choice model. In this model, the consideration set layer captures the impact of price changes on preferences and the rank-based model layer captures the impact of assortment changes on the preferences. The authors proposed an EM method to estimate the parameters of this generalized model and also showed how to use this model to jointly optimize product prices and the assortment offered to customers. However, this approach does not directly extend to capturing the variation in other product features.

For that, Jagabathula et al. (2020b) (henceforth JSV) generalize the rankings in the rank-based model to have a more flexible functional form that can incorporate product features. More formally, suppose that each product is represented by a

D-dimensional feature vector in some feature space $\mathcal{Z} \subseteq \mathbb{R}^D$. Example features include the price, brand, color, size, weight, etc. We let z_{iS} denote the feature vector of product i in offer set S, allowing product features (such as prices) to change over time/location with each offer set. If one of the products is the no-purchase option, then its feature vector is set to $\mathbf{0}_D$ in all offer sets.[7] For any offer set S, let $\boldsymbol{Z}_S = (z_{iS} : i \in S)$. Then we denote the collection of all observed feature vectors as $\boldsymbol{Z}_{\mathcal{M}} = (\boldsymbol{Z}_S : S \in \mathcal{M})$.

The population preferences are modeled as a distribution over *customer types*, defined as follows. We first consider the standard multinomial logit (MNL) types, whose choice behavior is governed by the MNL model. In particular, given a parameter (or "taste") vector $\boldsymbol{\beta} \in \mathbb{R}^D$, the MNL model specifies that a customer purchases product i from offer set S with probability

$$f_{i,S}(\boldsymbol{\beta}; \boldsymbol{Z}_S) = \frac{\exp(\boldsymbol{\beta}^\top z_{iS})}{\sum_{j \in S} \exp(\boldsymbol{\beta}^\top z_{jS})}, \quad \text{(MNL CHOICE PROBABILITY FUNCTION)}$$

where we have made the dependence on the set of feature vectors $\boldsymbol{Z}_S$ explicit. The taste vector $\boldsymbol{\beta}$ captures the "value" that a customer places on each of the product features in deciding which product to purchase. Each standard logit type is represented using the vector $\boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) \in (0, 1)^M$ which specifies the choice probabilities for the observed offer set collection:

$$\boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) = (f_{i,S}(\boldsymbol{\beta}; \boldsymbol{Z}_S) : i \in S, S \in \mathcal{M}). \tag{8.9}$$

Denote the set of all standard logit types as $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}}) \stackrel{\text{def}}{=} \left\{ \boldsymbol{f}(\boldsymbol{\beta}; \boldsymbol{Z}_{\mathcal{M}}) : \boldsymbol{\beta} \in \mathbb{R}^D \right\}$. A key limitation of standard logit types is that they always assign a non-zero purchase probability to every product in every offer set. As a result, they cannot capture rank-based preferences, which allow for zero probabilities of purchase. To address this limitation, JSV allow the customer types to be also described by what they call *boundary* logit types, which include types on the "boundary" of the set $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})$. Formally, these types arise when the parameter vector $\boldsymbol{\beta}$ becomes unbounded, as we see below. Including the boundary types results in a model that is a distribution over the *closed logit types* $\overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})}$, which is the closure of the set $\mathcal{F}_{\text{MNL}}(\boldsymbol{Z}_{\mathcal{M}})$ in $\mathbb{R}^M$; we consider closure with respect to the standard Euclidean topology on $\mathbb{R}^M$.

The following lemma establishes that the closed logit types contain rankings as special cases, showcasing that the rank-based choice model is subsumed by this model.

Lemma 1 *For any offer set collection $\mathcal{M}$, there exists a feature specification $\boldsymbol{Z}_{\mathcal{M}}$ such that $\mathcal{F}(\mathscr{P}) \subset \overline{\mathcal{F}_{MNL}(\boldsymbol{Z}_{\mathcal{M}})}$.*

[7] In this case, the feature vector for other products would typically include a constant feature 1 to allow for general no-purchase market shares.

Proof Recall that $\mathcal{F}(\mathscr{P}) = \{\boldsymbol{f}(\sigma)\colon \sigma \in \mathscr{P}\}$, where $\boldsymbol{f}(\sigma)$ is defined in (8.3). Suppose that the feature representation of each product $j \in [N]$ is set to the one-hot encoded vector, so that, $z_{jS} = \boldsymbol{e}_j$ for all offer sets S, where $\boldsymbol{e}_j \in \mathbb{R}^N$ is a vector of all zeros except 1 at the j^{th} position. Note that the number of features $D = N$ in this case. Letting $\boldsymbol{E}_S = (\boldsymbol{e}_j\colon j \in S)$ and $\boldsymbol{E}_{\mathcal{M}} = (\boldsymbol{E}_S\colon S \in \mathcal{M})$, we will show that $\boldsymbol{f}(\sigma) \in \overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{E}_{\mathcal{M}})}$ for all $\sigma \in \mathscr{P}$.

Given any ranking σ, define $\boldsymbol{\beta}_\sigma \stackrel{\text{def}}{=} (-\sigma(1), -\sigma(2), \cdots, -\sigma(N))$ and consider the sequence of standard logit types $\boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}})$ for each $r \in \mathbb{N}$. Using the MNL CHOICE PROBABILITY FUNCTION, it follows that for any $S \in \mathcal{M}$ and any $i \in S$:

$$\begin{aligned}
\lim_{r\to\infty} f_{i,S}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_S) &= \lim_{r\to\infty} \frac{\exp\left(r \cdot (\boldsymbol{\beta}_\sigma^\top \boldsymbol{e}_i)\right)}{\sum_{j\in S} \exp\left(r \cdot (\boldsymbol{\beta}_\sigma^\top \boldsymbol{e}_j)\right)} \\
&= \lim_{r\to\infty} \frac{\exp\left(-r \cdot \sigma(i)\right)}{\sum_{j\in S} \exp\left(-r \cdot \sigma(j)\right)} \\
&= \lim_{r\to\infty} \frac{1}{1 + \sum_{j\in S\setminus\{i\}} \exp\left(r \cdot (\sigma(i) - \sigma(j))\right)} \\
&= \mathbb{1}[\sigma, i, S],
\end{aligned}$$

where the last equality follows from the definition of $\mathbb{1}[\sigma, i, S]$. Letting $\lim_{r\to\infty} \boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}}) \stackrel{\text{def}}{=} \left(\lim_{r\to\infty} f_{i,S}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_S)\colon i \in S, S \in \mathcal{M}\right)$, it follows that $\lim_{r\to\infty} \boldsymbol{f}(r \cdot \boldsymbol{\beta}_\sigma; \boldsymbol{E}_{\mathcal{M}}) = \boldsymbol{f}(\sigma)$. Since the closure of a set contains all limit points, $\boldsymbol{f}(\sigma) \in \overline{\mathcal{F}_{\text{MNL}}(\boldsymbol{E}_{\mathcal{M}})}$ and the claim follows. □

In the remainder of the section, we leave the dependence of the closed logit types on the observed feature vectors implicit and use $f_{i,S}(\boldsymbol{\beta})$ and $\boldsymbol{f}(\boldsymbol{\beta})$, respectively, to denote the choice probability under an MNL model and a standard logit type, and $\overline{\mathcal{F}_{\text{MNL}}}$ to denote the set of closed logit types. We also use $\mathcal{B}_{\text{MNL}} \stackrel{\text{def}}{=} \overline{\mathcal{F}_{\text{MNL}}} \setminus \mathcal{F}_{\text{MNL}}$ to denote the set of boundary logit types. Further, because the parameter vector $\boldsymbol{\beta}$ for a boundary logit type is not well-defined, we refer to a general customer type in $\overline{\mathcal{F}_{\text{MNL}}}$ simply as $\boldsymbol{f} = (f_{i,S}\colon i \in S, S \in \mathcal{M})$.

Now, as mentioned above, the population is described by a distribution over the customer types $\overline{\mathcal{F}_{\text{MNL}}}$. Let $\mathcal{Q} \stackrel{\text{def}}{=} \left\{Q\colon Q \text{ is a distribution over } \overline{\mathcal{F}_{\text{MNL}}}\right\}$ denote the space of all distributions over $\overline{\mathcal{F}_{\text{MNL}}}$.[8] Given any distribution $Q \in \mathcal{Q}$, the vector of choice probabilities for the offer set collection $\mathcal{M}$ is given by:

$$\boldsymbol{g}_{\mathcal{M}}(Q) = (g_{i,S}(Q)\colon i \in S, S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(Q) = \int_{\overline{\mathcal{F}_{\text{MNL}}}} f_{i,S}\, \mathrm{d}Q(\boldsymbol{f}). \tag{8.10}$$

[8] Our development here is closely related to that in JSV but with slight differences.

Then, defining $\mathcal{G}(\mathcal{Q}) \stackrel{\text{def}}{=} \{\boldsymbol{g}_{\mathcal{M}}(Q) : Q \in \mathcal{Q}\}$, the goal is to solve the GENERAL ESTIMATION PROBLEM with $\mathcal{G} = \mathcal{G}(\mathcal{Q})$. Unlike the rank-based model, however, where the distribution λ was over the finite set of permutations $\mathscr{P}$, the distribution Q is now defined over an infinite set of customer types $\overline{\mathcal{F}_{\text{MNL}}}$, and consequently it is more challenging to describe the constraint set $\mathcal{G}(\mathcal{Q})$. Despite this, JSV showed that $\mathcal{G}(\mathcal{Q})$ does permit an alternative representation that is easier to handle. For instance, suppose that Q is a discrete distribution with finite support. Then, it is easy to see that $\boldsymbol{g}_{\mathcal{M}}(Q)$ must belong to the convex hull of the set $\overline{\mathcal{F}_{\text{MNL}}}$, defined as:

$$\text{conv}(\overline{\mathcal{F}_{\text{MNL}}}) = \left\{ \sum_{\boldsymbol{f} \in F} \alpha_{\boldsymbol{f}} \boldsymbol{f} \ : \ F \subset \overline{\mathcal{F}_{\text{MNL}}} \text{ is finite and } \sum_{\boldsymbol{f} \in F} \alpha_{\boldsymbol{f}} = 1, \alpha_{\boldsymbol{f}} \geq 0 \ \ \forall \boldsymbol{f} \in F \right\}.$$

More generally, since $\overline{\mathcal{F}_{\text{MNL}}}$ is a compact subset of $\mathbb{R}^M$ (it is closed by definition and bounded since each $\boldsymbol{f} \in [0,1]^M$), it follows from existing results (see, e.g., Lindsay, 1983) that the set $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ contains vectors $\boldsymbol{g}_{\mathcal{M}}(Q)$ generated by *any* distribution Q over $\overline{\mathcal{F}_{\text{MNL}}}$, so in fact $\mathcal{G}(\mathcal{Q}) = \text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$. This is the reason we term this model the *nonparametric* mixture of closed logit (NPMXCL) model, since it does not impose any parametric assumptions on the mixing distribution Q.

With the above development, the GENERAL ESTIMATION PROBLEM for the NPMXCL model takes the form:

$$\min_{\boldsymbol{g} \in \text{conv}(\overline{\mathcal{F}_{\text{MNL}}})} \ \mathsf{loss}(\boldsymbol{g}), \qquad \text{(NPMXCL MODEL ESTIMATION PROBLEM)}$$

where again we drop the explicit dependence of the predicted choice probability vector $\boldsymbol{g}_{\mathcal{M}}$ on the offer set collection $\mathcal{M}$, and of the loss function on $\boldsymbol{y}_{\mathcal{M}}$. It can be verified that the NPMXCL MODEL ESTIMATION PROBLEM is a convex program with a compact constraint set; see Lemma 1 in JSV. Moreover, the strict convexity of the loss function again ensures that the NPMXCL MODEL ESTIMATION PROBLEM has a unique optimal solution (the proof is identical to that of Theorem 1 earlier).

Relation to the Mixed Logit Models The mixture of logit or mixed logit model (Hensher and Greene, 2003; Train, 2009) assumes that customer preferences are modeled as a distribution over standard logit types, that is, as a distribution over $\mathcal{F}_{\text{MNL}}$.[9] This model is designed to capture heterogeneity in customer preferences and also to overcome the restrictive independence of irrelevant alternatives (IIA) property of the MNL model (Luce, 1959) to allow for complex substitution patterns. In fact, McFadden and Train (2000) showed that any model in the RUM class can

[9] Technically, the distribution is modeled over the parameter vector $\boldsymbol{\beta}$ as opposed to its "type" representation $\boldsymbol{f}(\boldsymbol{\beta})$.

be approximated to arbitrary degree of accuracy by a mixed logit model with an appropriate specification for the product features and the mixing distribution.

While the mixed logit model is stated in this general form, it is rarely estimated as such. Traditionally, for purposes of tractability, the mixing distribution is restricted to belong to some parametric family $\mathcal{Q}(\Theta)$ of distributions defined over parameter space Θ such that $\mathcal{Q}(\Theta) \stackrel{\text{def}}{=} \{Q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ and $Q_{\boldsymbol{\theta}}$ is the mixing distribution over the MNL taste vector $\boldsymbol{\beta}$ corresponding to the parameter vector $\boldsymbol{\theta} \in \Theta$. Analogous to (8.10), the predicted choice probability vector $\boldsymbol{g}_{\mathcal{M}}(Q_{\boldsymbol{\theta}})$ corresponding to mixing distribution $Q_{\boldsymbol{\theta}}$ is given by:

$$\boldsymbol{g}_{\mathcal{M}}(Q_{\boldsymbol{\theta}}) = (g_{i,S}(Q_{\boldsymbol{\theta}}) : i \in S, S \in \mathcal{M}) \quad \text{where} \quad g_{i,S}(Q_{\boldsymbol{\theta}}) = \int_{\mathbb{R}^D} f_{i,S}(\boldsymbol{\beta}) \, \mathrm{d} \, Q_{\boldsymbol{\theta}}(\boldsymbol{\beta}). \tag{8.11}$$

The best fitting distribution from the family $\mathcal{Q}(\Theta)$ is then obtained by solving the following MLE problem:[10]

$$\max_{\boldsymbol{\theta} \in \Theta} \sum_{S \in \mathcal{M}} M_S \sum_{i \in S} y_{i,S} \log \left(g_{i,S}(Q_{\boldsymbol{\theta}}) \right). \tag{8.12}$$

Different assumptions for the family $\mathcal{Q}(\Theta)$ lead to different mixed logit models.

The most standard assumption is that the mixing distribution follows a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, parametrized by $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix of the distribution. The resulting model is referred to as the random parameters logit (RPL) model (Train, 2009). Under the RPL model, computing the choice probabilities in (8.11) requires the evaluation of an integral, which is often approximated through a Monte Carlo simulation. This results in a maximum simulated likelihood estimator (MSLE). Since the log-likelihood objective is typically non-convex in the parameters $\boldsymbol{\theta}$, gradient-based optimization routines are used to reach a local optimal solution. Often, additional structure is imposed on the covariance matrix (such as a diagonal matrix) to reduce the dimensionality of the parameter space. The interested reader is referred to Chapters 8 and 9 in Train (2009) for an overview of such estimation procedures.

The other common assumption is that the mixing distribution has a finite support of size K. The distribution is then parametrized by $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_K, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$, where $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K)$ denotes the support of the distribution and $(\alpha_1, \ldots, \alpha_K)$ denotes the corresponding mixture proportions, so that $\sum_{k \in [K]} \alpha_k = 1$ and $\alpha_k \geq 0$ for all $k \in [K]$. The resulting model is referred to as the latent class MNL (LC-MNL) model (Bhat, 1997; Boxall and Adamowicz, 2002; Greene and Hensher, 2003). In this case, the predicted choice probabilities in (8.11) simplify to $g_{i,S}(Q_{\boldsymbol{\theta}}) = \sum_{k=1}^{K} \alpha_k f_{i,S}(\boldsymbol{\beta}_k)$. However, direct optimization of the log-likelihood objective is challenging since it is non-convex in the parameters $\boldsymbol{\theta}$ and further, the

[10] This is equivalent to minimizing the KL-divergence loss function and is the standard choice when estimating the mixed logit model.

number of parameters scales with the number of mixture components: for a K class LC-MNL model, we need to estimate $K \cdot D + K - 1$ parameters. Consequently, the EM algorithm is employed to solve the MLE problem, which reduces the original problem into iteratively fitting K MNL models on weighted transformations of the observed sales fractions $\boldsymbol{y}_{\mathcal{M}}$. We refer the reader to Chapter 14 in Train (2009) for a detailed description of the EM algorithm for estimating LC-MNL models.

The NPMXCL model differs from the traditional mixed logit model in two key ways: it allows (a) individual customer types to be boundary logit types, as opposed to only standard logit types, and (b) the mixing distribution to be an arbitrary distribution. By allowing for boundary logit types, it subsumes the rank-based model (as shown in Lemma 1 above). In addition, by allowing for arbitrary mixing distributions, it mitigates the *model misspecification* issue. Both the RPL and the LC-MNL models are susceptible to model misspecification, which occurs when the ground-truth mixing distribution is not contained in the search space $Q(\Theta)$. Model misspecification can result in biased estimates for the parameters (Train, 2008) as well as poor goodness-of-fit (Fox et al., 2011). These issues are mitigated by the NPMXCL model.

8.4.1 Estimation via the Conditional Gradient Algorithm

We now discuss how to estimate the model parameters from observed choice data. The development of this section closely follows that of the rank-based model above. Since the NPMXCL MODEL ESTIMATION PROBLEM is a constrained convex program, in theory, we can use any standard method for convex optimization to solve it. However, similar to estimating the rank-based model earlier, there are two challenges: (a) the constraint region $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ lacks an efficient description; and (b) decomposing any candidate solution $\boldsymbol{g}$ into the underlying mixing distribution Q, which is needed so that out-of-sample predictions can be made, is a hard problem. In particular, note that $\text{conv}(\overline{\mathcal{F}_{\text{MNL}}})$ may *not* be a convex polytope as it could have infinitely many extreme points. JSV showed that the conditional gradient (CG) algorithm is again the ideal candidate to address both of these challenges.

As in the case of the rank-based model, we start with a distribution on an initial set of types $\mathcal{F}^{(0)} \subseteq \overline{\mathcal{F}_{\text{MNL}}}$ such that both the loss objective $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\, \mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded (see the discussion in Sect. 8.4.1.3 below). Then, using analogous arguments as in Sect. 8.3.1, the FRANK–WOLFE STEP in iteration $k \geq 1$ can be shown to be of the form:

$$\min_{\boldsymbol{v} \in \overline{\mathcal{F}_{\text{MNL}}}} \left\langle \nabla \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle. \tag{8.13}$$

Let $\boldsymbol{f}^{(k)}$ denote an optimal solution to (8.13); we discuss how to solve it in Sect. 8.4.1.1 below. Again, we observe that the CG algorithm is iteratively adding customer types $\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots$ to the support of the mixing distribution. As before,

we use the FCFW variant that re-optimizes the loss objective over the support of the customer types recovered so far to promote recovery of sparser mixing distributions. Algorithm 2 summarizes the estimation procedure.

Algorithm 2 CG algorithm for solving the NPMXCL MODEL ESTIMATION PROBLEM

1: **Initialize:** $k \leftarrow 0$; $\mathcal{F}^{(0)} \subseteq \overline{\mathcal{F}_{\text{MNL}}}$; $\boldsymbol{\alpha}^{(0)} \in \Delta_{\left|\mathcal{F}^{(0)}\right|-1}$; $\boldsymbol{g}^{(0)} = \sum_{\boldsymbol{f}\in\mathcal{F}^{(0)}} \alpha_{\boldsymbol{f}}^{(0)} \boldsymbol{f}$ s.t. $\mathsf{loss}(\boldsymbol{g}^{(0)})$, $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded
2: **while** stopping condition is not met **do**
3: $k \leftarrow k+1$
4: Compute $\boldsymbol{f}^{(k)} \in \arg\min_{\boldsymbol{v}\in\overline{\mathcal{F}_{\text{MNL}}}} \left\langle \nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle$ (SUPPORT FINDING STEP)
5: Update support of types $\mathcal{F}^{(k)} \leftarrow \mathcal{F}^{(k-1)} \cup \left\{\boldsymbol{f}^{(k)}\right\}$
6: Compute $\boldsymbol{\alpha}^{(k)} \in \arg\min_{\boldsymbol{\alpha}\in\Delta_{\left|\mathcal{F}^{(k)}\right|-1}} \mathsf{loss}\left(\sum_{\boldsymbol{f}\in\mathcal{F}^{(k)}} \alpha_{\boldsymbol{f}} \boldsymbol{f}\right)$ (PROPORTIONS UPDATE STEP)
7: Update support of types $\mathcal{F}^{(k)} \leftarrow \left\{\boldsymbol{f} \in \mathcal{F}^{(k)} : \alpha_{\boldsymbol{f}}^{(k)} > 0\right\}$
8: Update $\boldsymbol{g}^{(k)} \leftarrow \sum_{\boldsymbol{f}\in\mathcal{F}^{(k)}} \alpha_{\boldsymbol{f}}^{(k)} \boldsymbol{f}$
9: **end while**
10: **Output:** customer types $\mathcal{F}^{(k)}$ and proportions $(\alpha_{\boldsymbol{f}}^{(k)} : \boldsymbol{f} \in \mathcal{F}^{(k)})$

Below, we discuss how to solve the SUPPORT FINDING STEP and PROPORTIONS UPDATE STEP in more detail.

8.4.1.1 Solving the SUPPORT FINDING STEP

Recall that $\mathcal{F}_{\text{MNL}} = \left\{\boldsymbol{f}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{R}^D\right\}$ and $\boldsymbol{f}(\boldsymbol{\beta}) = (f_{i,S}(\boldsymbol{\beta}) : S \in \mathcal{M}, i \in S)$. By plugging in the MNL CHOICE PROBABILITY FUNCTION and ignoring constant terms, it follows that:

$$\begin{aligned} &\min_{\boldsymbol{v}\in\mathcal{F}_{\text{MNL}}} \left\langle \nabla\mathsf{loss}(\boldsymbol{g}^{(k-1)}), \boldsymbol{v} - \boldsymbol{g}^{(k-1)} \right\rangle \\ &\equiv \min_{\boldsymbol{\beta}\in\mathbb{R}^D} \sum_{S\in\mathcal{M}} \sum_{i\in S} \left(\nabla\, \mathsf{loss}(\boldsymbol{g}^{(k-1)})\right)_{i,S} \cdot \frac{\exp\left(\boldsymbol{\beta}^\top \boldsymbol{z}_{iS}\right)}{\sum_{j\in S} \exp\left(\boldsymbol{\beta}^\top \boldsymbol{z}_{jS}\right)}. \end{aligned} \tag{8.14}$$

The optimal solution to the above problem may be unbounded. Such unbounded solutions are instances of the boundary logit types $\mathcal{B}_{\text{MNL}} = \overline{\mathcal{F}_{\text{MNL}}} \setminus \mathcal{F}_{\text{MNL}}$, as we show in Sect. 8.4.3 below.

Even if the optimal solution is bounded, finding it may be intractable because the objective in (8.14) is non-convex in the parameter $\boldsymbol{\beta}$ (see Online Appendix D in JSV). However, in practice, we only need to find a feasible descent direction to ensure an improving solution in Algorithm 2 and, therefore, general-purpose nonlinear program solvers can be employed to obtain approximate solutions to (8.14).

JSV reported favorable performance of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Nocedal and Wright, 2006, Section 6.1) in generating improving solutions, although other methods could also be explored.

8.4.1.2 Solving the PROPORTIONS UPDATE STEP

As in the case of the rank-based model, the PROPORTIONS UPDATE STEP is a convex program over the unit simplex $\Delta_{\left|\mathcal{F}^{(k)}\right|-1}$. While in principle any method can be used to solve it, a particular variant of the CG algorithm is ideally suited. This variant (Guélat and Marcotte, 1986) compares two opposing steps to update the estimate in each iteration: the FRANK–WOLFE STEP that finds a descent direction, and an "away" step that reduces probability mass—possibly to zero—from a previously found extreme point (one amongst $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}$) or the initial solution $\boldsymbol{x}^{(0)}$. Observe that the FRANK–WOLFE STEP can be solved exactly for the PROPORTIONS UPDATE STEP by searching over the extreme points of the unit simplex $\Delta_{\left|\mathcal{F}^{(k)}\right|-1}$. The next iterate $\boldsymbol{x}^{(k)}$ is determined by the step (Frank–Wolfe or away) that results in larger improvement in the objective value; see Krishnan et al. (2015, Appendix B.1) for the precise description of this variant. The presence of away steps implies that the algorithm can 'drop' customer types, i.e., assign zero probability mass to, found in previous iterations from the support of the mixing distribution, resulting in sparser solutions. This is implemented in line 7 of Algorithm 2.

8.4.1.3 Initialization and Stopping Criterion

Algorithm 2 can be initialized with any $\boldsymbol{g}^{(0)} \in \overline{\mathcal{F}_{\text{MNL}}}$ such that both the initial loss $\mathsf{loss}(\boldsymbol{g}^{(0)})$ and its gradient $\nabla\mathsf{loss}(\boldsymbol{g}^{(0)})$ are bounded. For instance, we could choose $\mathcal{F}^{(0)} = \left\{\boldsymbol{f}(\boldsymbol{\beta}_{\text{MNL}})\right\}$ and $\boldsymbol{\alpha}^{(0)} = (1)$, resulting in $\boldsymbol{g}^{(0)} = \boldsymbol{f}(\boldsymbol{\beta}_{\text{MNL}})$; where $\boldsymbol{\beta}_{\text{MNL}}$ is the parameter estimate obtained by fitting an MNL model to the data. The MNL log-likelihood objective is globally concave in $\boldsymbol{\beta}$ and there exist efficient algorithms (Hunter, 2004; Jagabathula and Venkataraman, 2020) that exhibit fast convergence in practice. Another option is to fit an LC-MNL model with a "small" number of classes using the EM algorithm.

The same stopping criterion listed in Sect. 8.3.1.3 can also be adopted for Algorithm 2.

8.4.2 Convergence Guarantee for the Estimation Algorithm

JSV established a sublinear convergence guarantee for Algorithm 2. We state here a simplified version of their result (ignoring the derived constants) and the interested reader is referred to Section 5.1 in JSV for the precise guarantee:

Theorem 2 (Sublinear Convergence of the CG Algorithm) *For both loss functions defined in Sect. 8.2, the iterates generated by Algorithm 2 satisfy:*

$$\mathsf{loss}(\boldsymbol{g}^{(k)}) - \mathsf{loss}(\boldsymbol{g}^*) = O\left(\frac{1}{k}\right) \qquad \text{for all } k \geq \bar{K},$$

where $\boldsymbol{g}^*$ *is an optimal solution to the* NPMXCL MODEL ESTIMATION PROBLEM *and* $\bar{K} \geq 1$ *is some index.*

Proof For the detailed proof, please see Online Appendix A.2 in JSV; here we provide a sketch of the proof. Jaggi (2013) showed that the CG algorithm converges at an $O(1/k)$ rate if the (non-negative) *curvature constant* is bounded from above. The curvature constant is bounded if the constraint set is bounded and the hessian of the objective function is bounded from above. For the NPMXCL MODEL ESTIMATION PROBLEM, the domain $\mathrm{conv}(\overline{\mathcal{F}_{\mathrm{MNL}}}) \subseteq [0, 1]^M$ is bounded. For the squared norm loss function (Example 2 in Sect. 8.2), the hessian is also bounded from above and so the convergence guarantee follows from existing results. However, it can be verified that the hessian of the KL-divergence loss function (Example 1 in Sect. 8.2) becomes unbounded close to the boundary of the domain $\mathrm{conv}(\overline{\mathcal{F}_{\mathrm{MNL}}})$, i.e., when $\boldsymbol{g}$ has entries that are close to 0, and thus, the existing guarantee does not apply. JSV showed that each iterate $\boldsymbol{g}^{(k)}$ generated by Algorithm 2 has entries that are bounded from below by a data-dependent constant $\xi_{\min} > 0$. In other words, the iterates do not get too close to the boundary of the domain and they exploit this fact to establish the $O(1/k)$ convergence rate for the KL-divergence loss function as well, with the constant scaling in $1/\xi_{\min}^2$. □

While the above result establishes convergence of $\mathsf{loss}(\boldsymbol{g}^{(k)})$ to the optimal objective $\mathsf{loss}(\boldsymbol{g}^*)$, it does not say anything regarding convergence to the true mixing distribution from which the data was generated. Without additional assumptions, establishing convergence to the ground-truth mixing distribution is challenging since $\boldsymbol{g}^*$ can be decomposed into many underlying distributions in general. JSV showed through a simulation study that Algorithm 2 does recover good approximations to different ground-truth mixing distributions when there is sufficient variation in the observed choice data. Identifying conditions under which the CG algorithm recovers the ground-truth mixing distribution is an interesting direction for future work.

To gain further insights, JSV also analyzed the support of the mixing distribution recovered by the CG algorithm, which is determined by the structure of the optimal solutions to the SUPPORT FINDING STEP. We discuss this next.

8.4.3 Characterizing the Choice Behavior of Closed Logit Types

As alluded to earlier, the optimal solution to the SUPPORT FINDING STEP can either be a standard logit type or a boundary logit type. Standard logit types are

characterized by their corresponding taste parameters $\boldsymbol{\beta}$, which can be used to make out-of-sample predictions on new offer sets. However, it is not immediately clear how to think of boundary logit types, since by definition there exists no parameter $\boldsymbol{\beta}$ that can describe such types. To address this issue, JSV provided the following concise characterization of boundary logit types (see Online Appendix A.3 in JSV for the proof):

Theorem 3 (Characterization of Boundary Logit Types) *Any boundary logit type $\boldsymbol{f} \in \mathcal{B}_{MNL}$ satisfies $f_{i,S} = 0$ for at least one (i, S) pair in the observed offer set collection $\mathcal{M}$. Moreover, we can find parameters $\boldsymbol{\beta}_0, \boldsymbol{\omega} \in \mathbb{R}^D$ such that, for each $S \in \mathcal{M}$ and all $i \in S$ (with $r \in \mathbb{N}$ below):*

$$f_{i,S} = \lim_{r \to \infty} \frac{\exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_{iS}\right)}{\sum_{j \in S} \exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_{jS}\right)}.$$

The result establishes that boundary logit types assign zero probability to at least one data point in the observed offer set collection $\mathcal{M}$, compared to standard logit types that assign non-zero probabilities to all observations. Moreover, boundary logit types arise as a result of limiting MNL models, obtained as the parameter vector $\boldsymbol{\beta}$ is pushed to infinity. In particular, for any boundary logit type $\boldsymbol{f}$, there exist parameters $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$ such that $\boldsymbol{f} = \lim_{r \to \infty} \boldsymbol{f}(\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})$, where recall that $\boldsymbol{f}(\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})$ corresponds to a standard logit type with parameter vector $\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega}$. Thus, unlike standard logit types that are described by a single parameter vector, boundary types are characterized by a pair of parameters. In fact, boundary logit types can be considered as natural generalizations of rankings to capture the impact of changing product features, as we show next.

The above characterization reveals a preference ordering over the products induced by the parameter vector $\boldsymbol{\omega}$, that determines which product is chosen from a given offer set. For ease of exposition, suppose that product features do not vary with the offer set, so that we can write $\boldsymbol{z}_j$ instead of $\boldsymbol{z}_{jS}$ for the feature vector of product j in each offer set S. The preference order is determined by product utilities $u_j \stackrel{\text{def}}{=} \boldsymbol{\omega}^\top \boldsymbol{z}_j$ for each product $j \in [N]$. In particular, the utilities induce a preference order $\succsim$ among the products such that $j \succsim j'$, read as "product j is weakly preferred over product j'," if and only if $u_j \geq u_{j'}$. The relation $\succsim$ is in general a weak (or partial) ordering and *not* a strict (or complete) ordering because utilities of two products may be equal. Consequently, we write $j \succ j'$ if $u_j > u_{j'}$ and $j \sim j'$ if $u_j = u_{j'}$. Note that such a preference order differs from a ranking in two ways: (a) it can be a partial ordering, and (b) the ordering depends on the values of the product features.

Similar to rankings, it can be shown that when offered any set $S \subseteq [N]$, boundary logit types choose only amongst the most preferred products in S, determined according to the preference order $\succsim$. To see that, let $C(S)$ denote the set of most preferred products in S, so that for all $j \in C(S)$, we have $j \sim \ell$ if $\ell \in C(S)$ and $j \succ \ell$ if $\ell \in S \setminus C(S)$. Let $u^* \stackrel{\text{def}}{=} \max\left\{u_j : j \in S\right\}$ denote the maximum utility

among the products in S. From the definition of $\succsim$, it follows that $u^* = u_j$ for all $j \in C(S)$ and $u^* > u_j$ for all $j \in S \setminus C(S)$. Note it is possible that $C(S) = S$ in case all the product utilities are equal. Now to determine which products will be chosen from S, we first multiply the numerator and denominator of the choice probabilities defined in Theorem 3 by $e^{-r \cdot u^*}$. Then, it follows that for any $j \in S$:

$$\begin{aligned} &\frac{\exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_j\right)}{\sum_{\ell \in S} \exp\left((\boldsymbol{\beta}_0 + r \cdot \boldsymbol{\omega})^\top \boldsymbol{z}_\ell\right)} \\ &= \frac{e^{-r \cdot (u^* - u_j)} \cdot \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_j)}{\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_\ell) + \sum_{\ell \in S \setminus C(S)} e^{-r \cdot (u^* - u_\ell)} \cdot \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_\ell)}, \end{aligned} \tag{8.15}$$

where we plugged in $\boldsymbol{\omega}^\top \boldsymbol{z}_\ell = u_\ell$ for each $\ell \in S$. Taking the limit $r \to \infty$, it follows that each of the terms $e^{-r \cdot (u^* - u_\ell)}$, $\ell \in S \setminus C(S)$, goes to zero since $u_\ell < u^*$. As a result, the denominator in (8.15) converges to $\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_\ell)$. On the other hand, the numerator converges to $\exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_j)$ if $j \in C(S)$ and 0 if $j \in S \setminus C(S)$. Combining the two, we obtain the following choice probability prediction for any product j in offer set S from Theorem 3:

$$f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) = \begin{cases} \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_j) / \left(\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_\ell)\right), & \text{if } j \in C(S) \text{ and} \\ 0, & \text{if } j \in S \setminus C(S), \end{cases}$$

where we abuse notation and let $f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega})$ denote the probability of choosing product j from offer set S under the boundary logit type described by $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$. This implies that only products that are within $C(S)$ are considered for purchase. Algorithm 3 outlines the above procedure for the general case.

Note the contrasting roles of the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\beta}_0$ in defining the choice probabilities for a boundary logit type. The parameter vector $\boldsymbol{\omega}$ (through the preference ordering $\succsim$ it induces) determines the *consideration set* $C(S)$—the subset of products that the customer considers for purchase—whereas the parameter vector

Algorithm 3 Predicting choice probabilities for boundary logit type described by parameters $(\boldsymbol{\beta}_0, \boldsymbol{\omega})$

1: **Input:** Offer set S with product features $\boldsymbol{z}_{jS} \in \mathbb{R}^D$ for each $j \in S$
2: Compute utilities $u_j = \boldsymbol{\omega}^\top \boldsymbol{z}_{jS}$ for each $j \in S$.
3: Form consideration set $C(S) = \left\{j \in S \mid u_j = \max_{\ell \in S} u_\ell\right\}$
4: For any $j \notin C(S)$, $f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) \leftarrow 0$
5: For any $j \in C(S)$,

$$f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) \leftarrow \frac{\exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_{jS})}{\sum_{\ell \in C(S)} \exp(\boldsymbol{\beta}_0^\top \boldsymbol{z}_{\ell S})}$$

6: **Output:** Choice probabilities $(f_{j,S}(\boldsymbol{\beta}_0, \boldsymbol{\omega}) : j \in S)$

$\boldsymbol{\beta}_0$ determines the choice probabilities from within the consideration set, governed by an MNL model. In particular, the parameter vector $\boldsymbol{\omega}$ dictates how a product's features impact its inclusion into the consideration set. For instance, suppose that product j with utility $u_j < u^*$ is not in consideration currently, where recall that u^* is the maximum utility of a product in offer set S. Further, suppose one of the features is price and the corresponding coefficient in parameter vector $\boldsymbol{\omega}$ is $\omega_p < 0$. Then, product j will enter into consideration only if its price is sufficiently reduced so that its resulting utility is at least u^* (assuming all other features are held constant). In other words, the price should be dropped by at least $\frac{u^* - u_j}{-\omega_p}$ to ensure consideration of product j. Such a dependence cannot be modeled via rankings since they do not capture the impact of changing product features on the choice probabilities. Consequently, boundary logit types can be viewed as generalizations of rankings that account for more nuanced dependence of the choice behavior on the product features.

The choice behavior of boundary logit types is consistent with prior literature, which establishes that customers often consider a subset of the products on offer before making the choice; see, e.g., Hauser (2014), Jagabathula and Rusmevichientong (2017), and Aouad et al. (2020b). For further insights, we refer the reader to Section 5.3 in JSV where the authors analyze the consideration sets of the boundary logit types recovered by the CG algorithm.

8.5 Other Nonparametric Choice Models

There is growing interest in developing nonparametric methods to estimate choice models, and our discussion above has but scratched the surface. In this section, we briefly discuss other nonparametric choice models that have received attention in the operations literature.

Choice Model Trees Aouad et al. (2020a) propose *choice model trees*, a novel choice model which leverages a decision tree to segment the customer population based on observable characteristics like demographics and prior purchase history, and then fits an MNL model for each segment, where the segments correspond to the leaf/terminal nodes in the tree. The tree splits are recursively chosen to maximize the log-likelihood of the observed choice data, which is obtained by summing over the log-likelihoods for each leaf node. Their approach can be viewed as a nonparametric variant of the LC-MNL model introduced in Sect. 8.4, since the decision tree splits can be used to capture flexible mappings from customer characteristics to segments. Moreover, choice model trees assign each customer to exactly one segment, unlike the classical LC-MNL model that outputs a probabilistic assignment over the different segments. The authors show that their proposed model outperforms natural benchmarks in predictive accuracy, while also providing an interpretable segmentation of the population.

Nonparametric Tree Choice Model Paul et al. (2018) propose a general tree choice model where the customer demand is modeled via a rooted (undirected) binary tree in which each node corresponds to a product, and the set of all possible customer types is characterized by the set of all linear paths—paths that move either progressively toward or away from the root node—in the tree. Since each path can be viewed as a preference ordering of the products appearing on the path, their model can be viewed as a special case of the rank-based choice model as it considers only a subset of all possible rankings.[11] Their model generalizes the one proposed in Honhon et al. (2012), which only considered paths that start or end at the root node. To estimate the model, Paul et al. (2018) propose a greedy heuristic that incrementally adds nodes to the existing tree with the goal of maximizing the number of customer types that is consistent with the observed choice data, and prevents overfitting by controlling the depth of the tree. Having estimated the tree and, therefore, the set of customer types, they solve the MLE problem for estimating the distribution λ over these types (recall the notation in Sect. 8.3). Since the log-likelihood objective is concave in the ranking probabilities $\lambda(\sigma)$ and the number of customer types is $O(N^2)$, the MLE problem can be solved efficiently using standard non-linear solvers. They also propose tractable algorithms for several assortment and pricing problems under the proposed choice model.

Mixture of Mallows Model One limitation of the rank-based choice model is that it assigns zero probability to any choice that is not consistent with any of the rankings in its support. This can be problematic since typically sparse models are chosen that have "small" support sizes. One remedy to this is the NPMXCL model of Jagabathula et al. (2020b) that we discussed above. An alternative approach was recently proposed by Désir et al. (2021), who consider a smoothed generalization of (sparse) rank-based models by assuming that the underlying probability distribution over rankings is specified as a mixture of Mallows models, with the number of mixture components equal to the support size of the rank-based model. The Mallows model (Mallows, 1957) assumes that consumer preferences are concentrated around a central ranking τ and the probability of sampling a ranking σ different from τ falls exponentially with the Kendall-Tau distance $d(\sigma, \tau)$, defined as the number of pairwise disagreements between σ and τ. In other words, the Mallows model creates a smoothing property around the central ranking τ. Therefore, the mixture of Mallows model provides a natural generalization of the rank-based choice model, assigning a non-zero probability to every possible choice. Désir et al. (2021) propose an EM algorithm to estimate the mixture of Mallows model, where the M-step involves solving a MIP. Moreover, they propose several practical approaches for solving the assortment optimization problem and show that Mallows-based smoothing can improve both the prediction as well as decision accuracy compared to the rank-based model.

[11] The rank-based model can allow for the number of products in a ranking to be strictly smaller than the size of the product universe, in which case the customer selects the no-purchase option if none of the products in the ranking is part of the offer set.

DAG-Based Choice Model The existing work on choice-based demand models in the operations literature has largely focused on using aggregate sales transaction data for estimation, and this has been the focus of our discussion in this book chapter as well. However, with the increasing availability of individual-level transaction data (also referred to as *panel* data), there is an opportunity to capture and estimate individual preferences. One of the first steps in this direction was taken by Jagabathula and Vulcano (2018) who introduced a nonparametric choice model in which each customer is characterized by a directed acyclic graph (DAG) representing a partial order among products in a category. A directed edge from node i to node j in the DAG indicates that the customer prefers the product corresponding to node i over the product corresponding to node j. The DAG captures the fact that customer preferences are acyclic or *transitive*. Unlike a full preference ordering, a DAG specifies pairwise preferences for only a subset of product pairs; therefore, it represents a partial order. When visiting the store, the customer samples a full preference ordering (ranking) consistent with her DAG according to a pre-specified distribution, forms a consideration set and then purchases the most preferred product (according to the sampled ranking) amongst the ones she considers. The authors provide a procedure to construct the DAG for each customer based on her store visits, and they define several behavioral models to form consideration sets. Then, they estimate the distribution over rankings that best explains the observed purchasing patterns of the customers. Using real-world panel data on grocery store visits, the authors show that their proposed approach provides more accurate and fine-grained predictions for individual purchase behavior compared to state-of-the-art benchmark methods. Recently, Jagabathula et al. (2020a) consider a refinement of this choice model with the objective of designing personalized promotions.

Models Beyond the RUM Class Our discussion has focused primarily on the RUM model class as it has been the de-facto choice model in the operations and revenue management literature for the past two decades. However, the recent work of Jagabathula and Rusmevichientong (2019) on the *limit of stochastic rationality* (LoR) provides evidence for the need to go beyond the RUM class. Recall from Sect. 8.2 that the global minimum of the loss function is achieved when $\boldsymbol{y}_{\mathcal{M}} = \boldsymbol{g}_{\mathcal{M}}$, resulting in zero loss and a perfect fit to the observed choice data. However, this may not be achievable if the observed choice data is inconsistent with the RUM model, so that $\boldsymbol{y}_{\mathcal{M}} \notin \mathcal{G}$. Using a case study on grocery stores sales transaction data, Jagabathula and Rusmevichientong (2019) showed that the *rationality loss*, which they define as the best fit achievable using a model in the RUM class, i.e., $\mathsf{loss}(\boldsymbol{y}_{\mathcal{M}}, \boldsymbol{g}^*)$ where $\boldsymbol{g}^*$ is the optimal solution to the RANK-BASED MODEL ESTIMATION PROBLEM, can be high for many product categories, suggesting the need for more sophisticated choice models. In their paper, the authors show that fitting a latent class generalized attraction model (LC-GAM) (Gallego et al., 2015), a parametric choice model that lies outside the RUM class, can help to breach the LoR for many categories. Since then, there has been significant progress in developing nonparametric models that extend the RUM class: the generalized stochastic preference (GSP) choice model (Berbeglia, 2018), the decision forest

choice model (Chen and Mišić, 2019) and the binary choice forest model (Chen et al., 2019) to name a few. This is an emerging research area and we expect a lot more work in this space.

8.6 Concluding Thoughts

Developing nonparametric methods for estimating choice models is an active area of research, with substantial interest both from academics and practitioners. With the availability of large volumes of increasingly granular data and corresponding access to flexible large-scale computing, nonparametric methods are not only possible but also necessary for attaining a high degree of prediction accuracy. We expect firms to continue to invest in implementing these methods to improve automated decision making.

We note that while the focus of this chapter has been on estimating choice models, there is a parallel stream of literature on using these models to solve operational decision problems of interest to firms; see Strauss et al. (2018) for a recent review. Two decision problems that have received significant attention within the literature are the assortment and the price optimization problems. In these decision problems, the firm wants to find the assortment (or offer set) and prices to offer to its customers, respectively, to maximize expected revenue or profit. Because of the cross-product cannibalization effects, firms must use choice models to solve these decision problems. Finding the optimal assortment or prices is significantly more difficult in nonparametric choice models because of the lack of exploitable structure. Existing literature has taken the approach of proposing efficient algorithms, sometimes using recent developments in solving mixed integer programs (MIPs), to approximate the optimal solution, see, e.g., Rusmevichientong et al. (2014), Jagabathula and Rusmevichientong (2017), Paul et al. (2018), Bertsimas and Mišić (2019), Aouad et al. (2020b), and Désir et al. (2021). We expect this parallel development to continue for the newer (and often, more general) choice models being proposed in the literature.

The design of general methods to effectively estimate large-scale choice models is taking place within the larger context of broader developments in artificial intelligence (AI) and machine learning (ML). The areas of AI/ML and operations research (OR) overlap significantly especially when it comes to model estimation. There is a healthy cross-pollination of ideas across these two communities (for example, the conditional gradient algorithm, which is a classical OR algorithm for solving quadratic programs, has recently gained in popularity in the ML community), and we expect this cross-pollination to push more of the model developments. As an example, consider that the methods discussed in this book chapter focused on generalizing the distributions over individual customer types. Each customer type in the NPMXCL model can be made more complex by allowing product utility values to depend on the features in a non-linear fashion. Linear specification is most common, partly driven by tractability reasons and partly by behavioral reasons (as

model parameters could then be conceived as marginal utilities, see, e.g., Ben-Akiva et al., 1985). Misspecified utility functions result in biased parameter estimates and low predictive accuracy. Popular ML approaches (such as random forests, neural networks, etc.) are well-suited for this purpose as they can learn highly non-linear representations of the utility, without imposing any a priori structures. Recent work has taken this approach in the context of transportation mode choices (see Han et al., 2020; Sifringer et al., 2020 and the references therein), and we expect this to be a fruitful future direction to pursue.

In addition, ML techniques can leverage unstructured data sources such as text, image, and video to construct feature representations, which can then be plugged into the utility specification along with other observed features such as price. Leveraging such sources is especially important in the context of online retail and e-commerce, where signals such as the image quality of the product, the (textual) reviews posted by prior customers, etc. are critical indicators of customer choice; see Liu et al. (2019, 2020) for some recent work using such types of data. We believe this is an exciting direction for the field and look forward to reading papers within this theme.

References

Abdallah, T., & Vulcano, G. (2020). Demand estimation under the multinomial logit model from sales transaction data. *Manufacturing & Service Operations Management, 23*, 1005–1331.

Aouad, A., Elmachtoub, A. N., Ferreira, K. J., & McNellis, R. (2020a). Market segmentation trees. arXiv:1906.01174.

Aouad, A., Farias, V., & Levi, R. (2020b). Assortment optimization under consider-then-choose choice models. *Management Science, 67*, 3321–3984.

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

Ben-Akiva, M. E., Lerman, S. R., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and application to travel demand* (vol. 9). Cambridge: MIT Press.

Berbeglia, G. (2018). The generalized stochastic preference choice model. Available at SSRN 3136227.

Bertsimas, D., & Mišić, V. V. (2019). Exact first-choice product line optimization. *Operations Research, 67*(3), 651–670.

Bhat, C. R. (1997). An endogenous segmentation mode choice model with an application to intercity travel. *Transportation Science, 31*(1), 34–48.

Block, H. D., & Marschak, J. (1960). Random orderings and stochastic theories of responses. *Contributions to Probability and Statistics, 2*, 97–132.

Boxall, P. C., & Adamowicz, W. L. (2002). Understanding heterogeneous preferences in random utility models: A latent class approach. *Environmental and Resource Economics, 23*(4), 421–446.

Chen, N., Gallego, G., & Tang, Z. (2019). The use of binary choice forests to model and estimate discrete choices. Available at SSRN 3430886.

Chen, Y. C., & Mišić, V. (2019). Decision forest: A nonparametric approach to modeling irrational choice. Available at SSRN 3376273.

Clarkson, K. L. (2010). Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms, 6*(4), 63.

Désir, A., Goyal, V., Jagabathula, S., & Segev, D. (2021). Mallows-smoothed distribution over rankings approach for modeling choice. *Operations Research, 69*, 1015–1348.

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

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

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

Fox, J. T., il Kim, K., Ryan, S. P., & Bajari, P. (2011). A simple estimator for the distribution of random coefficients. *Quantitative Economics, 2*(3), 381–418.

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly, 3*(1–2), 95–110.

Gallego, G., Ratliff, R., & Shebalov, S. (2015). A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research, 63*(1), 212–232.

Gallego, G., & Topaloglu, H. (2019). Introduction to choice modeling. In *Revenue management and pricing analytics* (pp. 109–128). Berlin: Springer.

Greene, W. H., & Hensher, D. A. (2003). A latent class model for discrete choice analysis: Contrasts with mixed logit. *Transportation Research Part B: Methodological, 37*(8), 681–698.

Guélat, J., & Marcotte, P. (1986). Some comments on wolfe's 'away step'. *Mathematical Programming, 35*(1), 110–119.

Haensel, A., & Koole, G. (2011). Estimating unconstrained demand rate functions using customer choice sets. *Journal of Revenue and Pricing Management, 10*(5), 438–454.

Han, Y., Zegras, C., Pereira, F. C., & Ben-Akiva, M. (2020). A neural-embedded choice model: Tastenet-mnl modeling taste heterogeneity with flexibility and interpretability. arXiv:200200922.

Hauser, J. R. (2014). Consideration-set heuristics. *Journal of Business Research, 67*(8), 1688–1699.

Hensher, D. A., & Greene, W. H. (2003). The mixed logit model: The state of practice. *Transportation, 30*(2), 133–176.

Honhon, D., Jonnalagedda, S., & Pan, X. A. (2012). Optimal algorithms for assortment selection under ranking-based consumer choice models. *Manufacturing & Service Operations Management, 14*(2), 279–289.

Hoyer, W. D., & Ridgway, N. M. (1984). Variety seeking as an explanation for exploratory purchase behavior: A theoretical model. In T. C. Kinnear (Ed.), *NA - Advances in consumer research* (vol. 11, pp. 114–119). Provo: ACR North American Advances.

Hunter, D. R. (2004). MM algorithms for generalized bradley-terry models. *Annals of Statistics, 32*, 384–406.

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

Jagabathula, S., Mitrofanov, D., & Vulcano, G. (2020a). Personalized retail promotions through a dag-based representation of customer preferences. *Operations Research, 70*, 641–1291.

Jagabathula, S., & Rusmevichientong, P. (2019). The limit of rationality in choice modeling: Formulation, computation, and implications. *Management Science, 65*(5), 2196–2215.

Jagabathula, S., Subramanian, L., & Venkataraman, A. (2020b). A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science, 66*(8), 3635–3656.

Jagabathula, S., & Venkataraman, A. (2020). An MM algorithm for estimating the MNL model with product features. Available at SSRN: https://ssrncom/abstract=3733971

Jagabathula, S., & Vulcano, G. (2018). A partial-order-based model to estimate individual preferences using panel data. *Management Science, 64*(4), 1609–1628.

Jaggi, M. (2011). *Sparse convex optimization methods for machine learning*. Ph.D. Thesis, ETH Zürich.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 427–435).

Kahn, B. E., & Lehmann, D. R. (1991). Modeling choice among assortments. *Journal of Retailing, 67*(3), 274–300.

Krishnan, R. G., Lacoste-Julien, S., & Sontag, D. (2015). Barrier frank-wolfe for marginal inference. In *Advances in Neural Information Processing Systems* (vol. 28, pp. 532–540)

Li, G., Rusmevichientong, P., & Topaloglu, H. (2015). The d-level nested logit model: Assortment and price optimization problems. *Operations Research, 63*(2), 325–342.

Lindsay, B. G. (1983). The geometry of mixture likelihoods: A general theory. *The Annals of Statistics, 11*, 86–94.

Liu, L., Dzyabura, D., & Mizik, N. (2020). Visual listening in: Extracting brand image portrayed on social media. *Marketing Science, 39*(4), 669–686.

Liu, X., Lee, D., & Srinivasan, K. (2019). Large-scale cross-category analysis of consumer review content on sales conversion leveraging deep learning. *Journal of Marketing Research, 56*(6), 918–943.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical analysis*. New York: Wiley.

Mahajan, S., & Van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research, 49*(3), 334–351.

Mallows, C. L. (1957). Non-null ranking models. I. *Biometrika, 44*(1–2), 114–130.

Manski, C. F. (1977). The structure of random utility models. *Theory and Decision, 8*(3), 229–254.

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

McFadden, D. (1981). Econometric models of probabilistic choice. In: *Structural analysis of discrete data with econometric applications* (pp. 198–272). Cambridge: MIT Press.

McFadden, D., & Train, K. (2000). Mixed MNL models for discrete response. *Journal of Applied Econometrics, 15*, 447–470.

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

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. Hoboken: Wiley.

Mišić, V. V. (2016). *Data, models and decisions for large-scale stochastic optimization problems*. Ph. D. Thesis, Massachusetts Institute of Technology, chapter 4: Data-driven Assortment Optimization.

Newman, J. P., Ferguson, M. E., Garrow, L. A., & Jacobs, T. L. (2014). Estimation of choice-based models using sales data from a single firm. *Manufacturing & Service Operations Management, 16*(2), 184–197.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd edn.). Berlin: Springer.

Paul, A., Feldman, J., & Davis, J. M. (2018). Assortment optimization and pricing under a nonparametric tree choice model. *Manufacturing & Service Operations Management, 20*(3), 550–565.

Prechelt, L. (2012). Early stopping—but when? In *Neural networks: Tricks of the trade* (pp. 53–67), Berlin: Springer.

Rusmevichientong, P., Shmoys, D., Tong, C., & Topaloglu, H. (2014). Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management, 23*(11), 2023–2039.

Shalev-Shwartz, S., Srebro, N., & Zhang, T. (2010). Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization, 20*(6), 2807–2832.

Sher, I., Fox, J. T., il Kim, K., & Bajari, P. (2011). Partial identification of heterogeneity in preference orderings over discrete choices. Tech. Rep., National Bureau of Economic Research.

Sifringer, B., Lurkin, V., & Alahi, A. (2020). Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological, 140*, 236–261.

Strauss, A. K., Klein, R., & Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research, 271*(2), 375–387.

Train, K. E. (2008). EM algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling, 1*(1), 40–69.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge: Cambridge University Press.

van Ryzin, G., & Vulcano, G. (2015). A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science, 61*(2), 281–300.

van Ryzin, G., & Vulcano, G. (2017). An expectation-maximization method to estimate a rank-based choice model of demand. *Operations Research, 65*(2), 396–407.

Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation, 26*(2), 289–315.