A Conditional Gradient Approach for Nonparametric Estimation of Mixing Distributions

Srikanth Jagabathula
Stern School of Business, New York University, New York, NY 10012,sjagabat@stern.nyu.edu
Lakshminarayanan Subramanian, Ashwin Venkataraman
Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, {lakshmi,ashwin}@cs.nyu.edu

Mixture models are versatile tools that are used extensively in many fields, including operations, marketing, and econometrics. The main challenge in estimating mixture models is that the mixing distribution is often unknown and imposing apriori parametric assumptions can lead to model misspecification issues. In this paper, we propose a new methodology for nonparametric estimation of the mixing distribution of a mixture of logit models. We formulate the likelihood-based estimation problem as a constrained convex program and apply the conditional gradient (a.k.a. Frank-Wolfe) algorithm to solve this convex program. We show that our method iteratively generates the support of the mixing distribution and the mixing proportions. Theoretically, we establish sublinear convergence rate of our estimator and characterize the structure of the recovered mixing distribution. Empirically, we test our approach on real-world datasets. We show that it outperforms the standard expectation-maximization (EM) benchmark on speed (16× faster), in-sample fit (up to 24% reduction in the log-likelihood loss), and predictive (average 27% reduction in standard error metrics) and decision accuracies (extracts around 23% more revenue). On synthetic data, we show that our estimator is robust to different ground-truth mixing distributions and can also account for endogeneity.

Key words: nonparametric estimation, mixture of logit, convex optimization, consideration sets

1. Introduction

Mixture models are used for modeling a wide-range of phenomena in many fields. Within operations, they have been used to model customer demand, which changes in response to the changes a firm makes to its product offerings. Predicting these changes allows firms to optimize their product and price offerings, such as discontinuing low demand products or enforcing price changes to shift demand to specific products. Demand predictions also serve as key inputs to inventory control and price optimization models that are used in retail operations and revenue management (RM) systems. A typical prediction problem involves fitting a mixture model to historical sales transactions and inventory data. The most popular model that is fit is the mixture of multinomial logit (MNL) models, also known simply as the mixture of logit models. This model has received considerable attention in the literature and has also been successfully applied in practice. In addition, it has been shown to approximate a wide class of mixtures (McFadden and Train 2000).

Because of its significance for demand modeling, we focus on the problem of estimating the mixing distribution of a mixture of logit models, from sales transaction and inventory data. The main

challenge in this problem is that the structure of the mixing distribution is not known in practice. A common work-around is to assume that the mixing distribution comes from a pre-specified *parametric* family, such as the normal or the log-normal distribution, and then estimate the parameters via maximum likelihood estimation (Train 2009). This approach is reasonable when there is some prior knowledge about the structure of the underlying mixing distribution. But when no such knowledge exists, as often happens in practice, the ground-truth mixing distribution may very well not conform to the imposed parametric form. This leads to model misspecification, which can result in biased parameter estimates (Train 2008) or low goodness-of-fit measures (Fox et al. 2011).

To avoid model misspecification, we take a nonparametric approach, in which we search for the best fitting mixing distribution from the class of all possible mixing distributions. The challenge with this approach is a computational one. The class of all possible mixing distributions lacks sufficient structure to allow for tractable estimation methods. One approach in the literature has been to approximate the class of all possible mixing distributions with another (large) class, and then search for the best fitting mixing distribution in that approximate space. For instance, Train (2008) takes this approach, in which the space of all mixing distributions is approximated with the class of finite mixtures of normal distributions or the class of discrete distributions with a large support size. Such approximations allow the application of standard optimization techniques, such as the expectation-maximization (EM) framework. But the resulting optimization problems are still non-convex and become difficult to solve, running into numerical issues, as the number of parameters increases.

Our main contribution in this paper is to reformulate the nonparametric mixture estimation problem as a constrained convex program, *without* resorting to any approximations to the space of all possible mixing distributions. We pose the mixture estimation problem as the problem of searching for the distribution that minimizes a loss function from among the class of all possible mixing distributions. The standard log-likelihood loss (which results in the maximum likelihood estimator) and squared loss are two example loss functions. Then, we use the insight that the mixing distribution affects the objective function only through the choice probabilities it predicts for the observed choices in the data; we call the vector of these choice probabilities, the *data mixture likelihood vector*. Now, instead of optimizing over the space of mixing distributions, the mixture estimation problem can be solved by directly optimizing over the space of all possible data mixture likelihood vectors. The constraints ensure that the mixture likelihood vector is indeed consistent with a valid mixing distribution. We show that for the standard loss functions used in the literature, the objective function is convex in the mixture likelihood vector. Further, although not apriori clear, we show that the constraint set is also convex. Together, these two properties result in a constrained convex program formulation for the mixture estimation problem. We emphasize here that although we obtain a convex program, the constraint space lacks an efficient description. Therefore, the resulting program, though convex, may

be theoretically hard to solve. Nevertheless, there is vast literature on solving such convex programs, which we leverage to obtain scalable and numerically stable methods that are efficient for special cases and result in good approximations more generally.

A more immediate concern is that simply solving the above program is not sufficient because the optimal solution will be expressed as the mixture likelihood vector and not as the mixing distribution. Backing out the underlying mixing distribution from the mixture likelihood vector may again be a computationally intensive exercise. To counter this issue, we apply the conditional gradient (a.k.a. Frank-Wolfe) algorithm to solve the above convex program. We show that the special structure of the conditional gradient (CG) algorithm allows it to simultaneously perform both the tasks of optimizing over the predicted choice probabilities *and* recovering the best fitting mixing distribution. The CG algorithm is an iterative first-order method for constrained convex optimization. We show that when applied to our method, each iteration of the CG algorithm yields a single mixture component. The CG algorithm has seen an impressive revival in the machine learning literature recently because of its favorable properties compared to standard projected/proximal gradient methods, such as efficient handling of complex constraint sets. The vast literature on the CG algorithm in the machine learning area confers two key advantages to our estimation technique: (a) availability of precise convergence guarantees (Lacoste-Julien and Jaggi 2015) and (b) scalability to large-scale and high-dimensional settings (Wang et al. 2016).

Summary of key results. Our work makes the following contributions:

1. *Novel mixture estimation methodology.* Our estimator is (a) *general-purpose*: can be applied with little to no customization for a broad class of loss functions; (b) *fast*: order of magnitude faster than the benchmark expectation-maximization (EM) algorithm; and (c) *nonparametric*: makes no assumption on the mixing distribution and estimates customer types in the population in a data-driven fashion.
2. *Analytical results.* We obtain two key theoretical results:

(i) We provide a sublinear convergence guarantee, i.e. $O(1/k)$ after k iterations, for our CG-based estimator, for both the log-likelihood and squared loss functions. Refer to Theorem 1 and Section 5.1 for the details.

(ii) We characterize the structure of the mixing distribution recovered by our estimator. Our method recovers two types of mixture components: what we call, (a) non-boundary and (b) boundary types. A non-boundary type is described by a standard logit model with a parameter vector $\boldsymbol{\omega}$. The boundary types, on the other hand, are limiting logit models that result from unbounded solutions in which the parameter vector $\boldsymbol{\omega}$ is pushed to infinity. We show that each boundary type can be described by two parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$. The parameter vector $\boldsymbol{\theta}$ induces a (weak) preference order over the set of products and determines a consideration set the customer forms, when given an offer-set.

The parameter vector $\boldsymbol{\omega}_0$ then determines the logit choice probabilities from within the consideration set. Refer to Section 5.2 for the details.

For the case of a single offer-set (such as market share data), we also identify conditions under which our estimator recovers boundary types, and characterize the corresponding consideration sets of the recovered types. Our conditions depend on the geometry of the observed product features, viz. the (convex) polytope formed by the convex hull of the product feature vectors; see Section 5.3.1 for the details. In addition, when some features are binary, we show that our estimator recovers boundary types in *each* iteration, with the consideration sets reflecting strong non-compensatory preferences in the population, refer to Section 5.3.2 for more details.

3. *Empirical results.* We conducted three numerical studies to validate our methodology:

(a) Using synthetic data, we show that our estimator is robust to several complex ground-truth mixing distributions and consistently recovers a good approximation to the underlying distribution. Note that this is despite the fact that our estimator has no knowledge of the true mixing distribution. In particular, its performance is significantly better than a standard benchmark method imposing a parametric assumption on the mixing distribution, and highlights the potential impact of model misspecification in practice.

(b) On the SUSHI Preference Dataset (Kamishima et al. 2005), where customers rank different sushi varieties according to their preference, we show that our method achieves superior in-sample fit compared to fitting a latent class MNL (LC-MNL) model using the EM algorithm (Bhat 1997), for both the log-likelihood (24% better) and squared loss (58% better), with 16× speedup in the estimation time. The CG algorithm iteratively adds customer types that explain the observed choice data to the mixing distribution, which results in a much better fit as compared to the EM algorithm that updates all customer types together in each iteration. Our approach also achieves better predictive accuracy than EM, with an average 27% and 16% reduction in the RMSE (root mean square error) and MAPE (mean absolute percentage error) metrics for predicting market shares on new assortments. In solving the assortment decision, we show that our method can extract upto 23% more revenue from the population than the EM benchmark.

(c) On real-world sales transaction data from the IRI Academic Dataset (Bronnenberg et al. 2008), we show that our method achieves upto 8% (resp. 7%) and 7% (resp. 5%) reduction, respectively, in the in-sample and out-of-sample log-likelihood loss (resp. squared loss), compared to the EM benchmark. In particular, we outperformed EM-based estimation in all 5 product categories that we considered.

2. Relevant literature

Our work has connections to two broad areas:

Nonparametric maximum likelihood estimation (NPMLE). Our estimation approach generalizes the NPMLE techniques—our method is applicable for any convex loss function including the standard log-likelihood loss—which have a long and rich history in classical statistics (Robbins 1950, Kiefer and Wolfowitz 1956). These techniques search for a distribution that maximizes the likelihood function from a large class of mixing distributions. In the context of studying properties of the maximum likelihood estimator (such as existence, uniqueness, support size, etc.) for the mixing distribution via the geometric structure of the constraint set, Lindsay (1983) shows that when the mixing distribution is unrestricted, the NPMLE can be formulated as a convex program. However, such a formulation is computationally difficult to solve when the underlying parameter space is high dimensional. To address this issue, existing work has taken two approaches. The first approach reduces the search space to a large (but finite) number of mixture components, and uses the EM algorithm for estimation (Laird 1978). Though now the estimation problem is finite-dimensional, convexity is lost and standard issues related to non-convexity and finite mixture models become a significant obstacle (McLachlan and Peel 2000). The second approach retains convexity but gains tractability through a finite-dimensional convex approximation where the support of the mixing distribution is assumed to be finite and pre-specified (such as a uniform grid) and only the mixing weights need to be estimated. Fox et al. (2011) specialize this approach to estimating a mixture of logit models. However, it is unclear how to choose the support. When the dimensionality of the parameter space is small, Fox et al. demonstrate that a uniform grid is sufficient to reasonably capture the underlying distribution, but this approach quickly becomes intractable for even moderately large parameter dimensions.[1] Consequently, existing techniques have usually focused on simple models with univariate or low-dimensional (bi- and tri-variate) mixing distributions (Bohning et al. 1992, Jiang and Zhang 2009, Feng and Dicker 2018) to retain tractability.

In the context of the above, we avoid the issues resulting from the non-convex formulation by retaining convexity, but at the same time we do *not* need access to a pre-specified support. We leverage the conditional gradient algorithm to directly solve the seemingly intractable convex program, which iteratively generates the support of the mixing distribution by searching over the underlying parameter space. This allows our method to scale to higher-dimensional settings, 5 in our SUSHI case study and 11 in the IRI case study; see Sections 7.1 and 7.2.

Conditional gradient algorithms. The conditional gradient algorithm is one of the earliest methods (Frank and Wolfe 1956) for constrained convex optimization, and has recently seen an impressive revival for solving large-scale problems with structured constraint sets (see Clarkson 2010 and Jaggi 2011 for excellent overviews). The algorithm has been used in diverse domains including

[1] Indeed, their numerical experiments consider only bivariate mixing distributions.

computer vision (Joulin et al. 2014), submodular function optimization (Bach 2013), collaborative filtering (Jaggi and Sulovsk 2010), as well as inference in graphical models (Krishnan et al. 2015). In addition, numerous related variants of the algorithm have been proposed such as solving non-linear subproblems to increase sparsity (Zhang 2003) and incorporating regularization to improve predictive performance (Harchaoui et al. 2015). In terms of theoretical performance, Jaggi (2013) gave a convergence analysis that guarantees an error of at most $O(1/t)$ (sublinear convergence) after t iterations for any compact convex constraint set. Recently, Lacoste-Julien and Jaggi (2015) proved that many versions of the classical Frank-Wolfe algorithm enjoy global linear convergence for any strongly convex function optimized over a polytope domain.

Our main contribution is leveraging the conditional gradient algorithm for estimating the mixing distribution, which also allows us to provide convergence guarantees for our estimator. For the squared loss function, the sublinear convergence rate of $O(1/k)$ after k iterations follows from existing results. But, for the log-likelihood loss, existing results don't apply because the gradient blows up at the boundary of the constraint region. We address this issue by showing that the iterates produced by the fully corrective variant of the CG algorithm (the one that we implement) are strictly bounded away from the boundary. We then adapt and extend existing arguments to establish the same $O(1/k)$ sublinear convergence guarantee, as for the squared loss. We also show that, under appropriate structure in the observed product features, our estimator converges to the optimal solution in a *finite* number of iterations (see Section 5.3).

3. Problem Setup and Formulation

We consider a universe $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ of n products, which customers interact with over $T \geq 1$ discrete time periods.[2] We assume access to aggregate sales data for these n products in each time period.[3] In each time period $t \in [T]$, the firm offers a subset $S_t \subseteq [n]$ of products to the customers and collects sales counts for each of the products. We let N_{jt} denote the number of times product j was purchased in period t, $N_t \stackrel{\text{def}}{=} \sum_{j \in S_t} N_{jt}$ denote the total number sales in period t, and $N \stackrel{\text{def}}{=} \sum_{t \in [T]} N_t$ denote the total number of sales over all the time periods. We suppose that we observe at least one sale in each period t, so that $N_t > 0$ for all periods $t \in [T]$; if there was no observed sale in a time period, then we assume that it was already dropped from the observation periods. Let $\mathsf{Data} \stackrel{\text{def}}{=} \{(N_{jt} \colon j \in S_t) \mid t \in [T]\}$ denote all the observations collected over the T discrete time periods. We assume that product $j \in S_t$ is represented by a D-dimensional feature vector $\mathbf{z}_{jt}$ in some feature space $\mathcal{Z} \subseteq \mathbb{R}^D$. Example features include price, brand, and color. Product features could vary over

[2] We use the notation $[m] \stackrel{\text{def}}{=} \{1, 2, \ldots, m\}$ for any positive integer m in the rest of the paper.

[3] Our method requires some modifications in order to be applied to individual-level panel data. We discuss these modifications in Appendix E.

time; for instance, product prices may vary because of promotions, discounts, etc. In practice, these data are often available to firms in the form of purchase transactions, which provide sales information, and inventory data, which provide offer-set information.

We assume that each customer makes choices according to an MNL (aka logit) model, which specifies that a customer purchases product j from offer-set S with probability

$$f_{j,S}(\boldsymbol{\omega}) = \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{jS}\right)}{\sum_{\ell \in S} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{\ell S}\right)}, \tag{1}$$

where $\mathbf{z}_{\ell S}$ is the feature vector of product ℓ when offered as part of offer-set S and $\boldsymbol{\omega}$ is the parameter or "taste" vector. This taste vector specifies the "value" that a customer places on each of the product features in deciding which product to purchase. Customers often have heterogeneous preferences over product features. To capture this heterogeneity, we assume that the population of customers is described by a mixture of MNL models, where in each choice instance, a customer samples a vector $\boldsymbol{\omega}$ according to some distribution Q (over the parameter space $\mathbb{R}^D$) and then makes choices according to the MNL model with parameter vector $\boldsymbol{\omega}$.

Our goal is to estimate the best fitting mixing distribution to the collection Data of sales observations, from the class of all possible mixing distributions $\mathcal{Q} \stackrel{\text{def}}{=} \{Q\colon Q \text{ is a distribution over } \mathbb{R}^D\}$. The fit to the data is measured via a *loss function* that quantifies the mismatch between the observed sales fractions in Data and those predicted by the mixture of logit model. In order to state the problem formally, we need to introduce additional notation. For each (product, offer-set) pair, define the mapping $\mathrm{g}_{jt}\colon \mathcal{Q} \to [0,1]$ as

$$\mathrm{g}_{jt}(Q) = \int f_{jt}(\boldsymbol{\omega})\, \mathrm{d}Q(\boldsymbol{\omega}),$$

where for brevity of notation, we let $f_{jt}(\boldsymbol{\omega})$ denote $f_{j,S_t}(\boldsymbol{\omega})$. In other words, $\mathrm{g}_{jt}(Q)$ is the probability of choosing product j from offer-set S_t under the mixing distribution Q. Let $M \stackrel{\text{def}}{=} |S_1| + \cdots + |S_T|$ and $\mathbf{g}\colon \mathcal{Q} \to [0,1]^M$ denote the vector-valued mapping, defined as $\mathbf{g}(Q) = (\mathrm{g}_{jt}(Q)\colon t \in [T], j \in S_t)$. We call $\mathbf{g}(Q)$ the *data mixture likelihood vector* or simply the *mixture likelihood vector*, under mixing distribution Q. We let $\mathcal{G} = \{\mathbf{g}(Q)\colon Q \in \mathcal{Q}\}$ denote the set of all mixture likelihood vectors.

Given the above, our goal is to solve the following problem:

$$\min_{Q \in \mathcal{Q}} \ \mathsf{loss}(\mathbf{g}(Q); \mathsf{Data}), \tag{Mixture Estimation}$$

where $\mathsf{loss}(\cdot;\, \mathsf{Data})\colon \mathcal{G} \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is a non-negative convex function. We make the standard assumption that $\mathsf{loss}(\cdot)$ is continuously differentiable on the relative interior of $\mathcal{G}$. Two example functions include:

- **Negative log-likelihood (NLL) Loss**: This loss function is by far the most widely used in practice (Train 2008):

$$\mathsf{NLL}(\mathbf{g}(Q); \mathsf{Data}) = -\frac{1}{N} \sum_{t=1}^{T} \sum_{j \in S_t} N_{jt} \log\left(\mathrm{g}_{jt}(Q)\right).$$

- **Squared (SQ) Loss**: This loss function was employed by Fox et al. (2011):

$$\mathsf{SQ}(\mathbf{g}(Q);\mathsf{Data}) = \frac{1}{2\cdot N}\sum_{t=1}^{T} N_t \cdot \sum_{j\in S_t} \left(\mathrm{g}_{jt}(Q) - y_{jt}\right)^2.$$

where $y_{jt} \stackrel{\text{def}}{=} N_{jt}/N_t$ denotes the fraction of sales for product j in offer-set S_t.

We first describe traditional approaches to solving the MIXTURE ESTIMATION problem, and their limitations, to motivate the need for our approach.

3.1. Traditional approaches to mixture estimation

Directly solving the MIXTURE ESTIMATION problem is challenging due to the complexity of searching over all possible mixing distributions. Consequently, traditional approaches assume that the mixing distribution belongs to a family $\mathcal{Q}(\Theta)$ of distributions parametrized via parameter space Θ, where $\mathcal{Q}(\Theta) \stackrel{\text{def}}{=} \{Q_{\boldsymbol{\theta}} : \boldsymbol{\theta}\in\Theta\}$ and $Q_{\boldsymbol{\theta}}$ is the mixing distribution corresponding to the parameter vector $\boldsymbol{\theta}\in\Theta$. The best fitting distribution is then obtained by solving the following likelihood problem:[4]

$$\min_{\boldsymbol{\theta}\in\Theta} \; -\frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t} N_{jt}\log\left(\int f_{jt}(\boldsymbol{\omega})\, dQ_{\boldsymbol{\theta}}(\boldsymbol{\omega})\right). \tag{2}$$

Different assumptions for the family $\mathcal{Q}(\Theta)$ result in different estimation techniques.

The most common 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 variance-covariance matrix of the distribution. The resulting model is referred to as the random parameters logit (RPL) model (Train 2009), and the corresponding likelihood problem is given by

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} -\frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t} N_{jt}\log\left(\int f_{jt}(\boldsymbol{\omega})\cdot\frac{1}{\sqrt{(2\pi)^D\,|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{\omega}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{\omega}-\boldsymbol{\mu})\right)d\boldsymbol{\omega}\right). \tag{3}$$

The integral in the above problem is often approximated through a Monte Carlo simulation.

The other common assumption is that the mixing distribution has a finite support of size K. The distribution is then parametrized by $\boldsymbol{\theta}=(\alpha_1,\ldots,\alpha_K,\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_K)$, where $(\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_K)$ denotes the support of the distribution and $(\alpha_1,\ldots,\alpha_K)$ denote the corresponding mixture proportions with $\sum_{k\in[K]}\alpha_k=1$ 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), and the corresponding likelihood problem is given by

$$\min_{\substack{\alpha_1,\alpha_2,\ldots,\alpha_K\\ \boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_K}} -\frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t} N_{jt}\log\left(\sum_{k=1}^{K}\alpha_k f_{jt}(\boldsymbol{\omega}_k)\right) \text{ subject to } \sum_{k\in[K]}\alpha_k=1, \alpha_k\geq 0\;\forall\; k\in[K]. \tag{4}$$

Although commonly used, these traditional approaches suffer from two key limitations:

[4] The negative log-likelihood loss is the typical choice in existing techniques.

- *Model misspecification*: The most significant issue with traditional approaches is model misspecification, which occurs when the ground-truth mixing distribution is not contained in the search space $\mathcal{Q}(\Theta)$. In practice, such misspecification is common because the selection of the search space $\mathcal{Q}(\Theta)$ is often driven by tractability considerations as opposed to knowledge of the structure of the ground-truth mixing distribution. Model misspecification can result in biased parameter estimates (Train 2008) and low goodness-of-fit measures (Fox et al. 2011).
- *Computational issues*: Another practical issue is that, even if the model is not misspecified, the resulting likelihood problems are non-convex and therefore hard to solve in general.

3.2. Our approach: mixture estimation by solving a convex program

Our approach is designed to address the challenges described above. We avoid the model misspecification issue as we directly search over all possible mixing distributions, instead of restricting our search to specific parametric families.[5] However, this can introduce computational concerns, given the complexity of the search space $\mathcal{Q}$. To address the computational issue, we formulate the MIXTURE ESTIMATION problem as a constrained convex program, as described next. This formulation allows us to tap into the vast existing literature on solving convex programs efficiently.

We now describe the steps in our formulation. We first observe that the objective function only depends on Q through the corresponding mixture likelihood vector $\mathbf{g}(Q)$. Therefore, instead of searching over the mixing distributions, we directly search over the mixture likelihood vectors, obtaining the following equivalence:

$$\min_{Q \in \mathcal{Q}} \mathsf{loss}(\mathbf{g}(Q)) \quad \equiv \quad \min_{\mathbf{g} \in \mathcal{G}} \mathsf{loss}(\mathbf{g}), \tag{5}$$

where we have dropped the explicit dependence of loss on Data for simplicity of notation. Recall that $\mathcal{G} = \{\mathbf{g}(Q) \colon Q \in \mathcal{Q}\}$ is the set of all possible mixture likelihood vectors.

With the above equivalence, our ability to solve the MIXTURE ESTIMATION problem depends on our ability to describe the constraint set $\mathcal{G}$. We show next that this constraint set can indeed be expressed as a convex set. For that, analogous to the mixture likelihood vector earlier, define the *atomic likelihood vector* $\boldsymbol{f}(\boldsymbol{\omega}) \stackrel{\text{def}}{=} (f_{jt}(\boldsymbol{\omega}) \colon t \in [T], j \in S_t)$. We let $\mathcal{P} = \{\boldsymbol{f}(\boldsymbol{\omega}) \colon \boldsymbol{\omega} \in \mathbb{R}^D\}$ denote the set of all possible atomic likelihood vectors, and $\overline{\mathcal{P}}$ denote its closure.[6] Now, it is clear that if $Q \in \mathcal{Q}$ is a discrete distribution with finite support $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_K$ and corresponding mixing weights

[5] We note that searching over the space of all possible mixing distributions can lead to potential overfit issues, which we discuss in Section 4.1.

[6] For technical reasons, we need to consider the closure of the set $\mathcal{P}$, which also contains all limit points of convergent sequences in the set $\mathcal{P}$.

$\alpha_1, \alpha_2, \ldots, \alpha_K$, then $\mathbf{g}(Q) = \sum_{k=1}^{K} \alpha_k \boldsymbol{f}(\boldsymbol{\omega}_k)$, and so $\mathbf{g}(Q)$ belongs to the convex hull, $\text{conv}(\overline{\mathcal{P}})$, of vectors in $\overline{\mathcal{P}}$, defined as

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

It can be verified that $\text{conv}(\overline{\mathcal{P}})$ is a convex set in $\mathbb{R}^M$. In other words, for any discrete mixing distribution Q with finite support, we can express $\mathbf{g}(Q)$ as a convex combination of atomic likelihood vectors $\{\boldsymbol{f}\}_{\boldsymbol{f} \in \mathcal{F}}$ for some finite subset $\mathcal{F} \subset \overline{\mathcal{P}}$. More generally, it can be shown (Lindsay 1983) that $\mathcal{G} = \text{conv}(\overline{\mathcal{P}})$, i.e. the set of all possible mixture likelihood vectors coincides with the convex hull of all atomic likelihood vectors. This fact, combined with the equivalence in (5), implies that instead of solving Mixture Estimation, we can equivalently solve the following problem:

$$\min_{\mathbf{g} \in \text{conv}(\overline{\mathcal{P}})} \mathsf{loss}(\mathbf{g}) \qquad \text{(Convex Mixture)}$$

We can show that the above is a constrained convex program (the proof is given in Appendix A.1):

Lemma 1. *For any convex function* $\mathsf{loss}(\cdot)$*,* Convex Mixture *is a convex program with a compact constraint set in the Euclidean space.*

So, our task now is to solve the Convex Mixture problem. However, solving it alone does not provide the mixing distribution—it only provides the optimal mixture likelihood vector. We show next that the conditional gradient algorithm is the ideal candidate to not only obtain the optimal mixture likelihood vector, but also the optimal mixing distribution.

4. Conditional gradient algorithm for estimating the mixing distribution

We now apply the conditional gradient (hereafter CG) algorithm to solve the Convex Mixture problem. The CG algorithm (Frank and Wolfe 1956, Jaggi 2013) is an iterative method for solving constrained convex programs. It has seen an impressive revival in the machine learning literature recently because of its favorable properties compared to standard projected/proximal gradient methods, such as efficient handling of complex constraint sets. Appendix C provides an overview of the general CG algorithm. Here, we describe how it applies to solving $\min_{\mathbf{g} \in \text{conv}(\overline{\mathcal{P}})} \mathsf{loss}(\mathbf{g})$.

The CG algorithm is an iterative first-order method that starts from an initial feasible solution, say $\mathbf{g}^{(0)} \in \text{conv}(\overline{\mathcal{P}})$, and generates a sequence of feasible solutions $\mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \ldots$ that converge to the optimal solution. Letting $\nabla \mathsf{loss}(\cdot)$ denote the gradient of the loss function and $\langle \cdot, \cdot \rangle$ the standard inner product in the Euclidean space, the algorithm computes a *descent direction* $\mathbf{d}$ such that $\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{d} \rangle < 0$ in iteration $k \geq 1$ and takes a suitable step in that direction (see for instance, Nocedal and Wright 2006). The main distinction of the CG algorithm is that it always chooses *feasible* descent steps, where by a feasible step we mean a step from the current solution towards the next solution such that the

next solution remains feasible as long as the current is feasible. By contrast, other classical algorithms may take infeasible steps, which are then projected back onto the feasible region after each step; such projection steps are usually computationally expensive. To find a feasible step, the CG algorithm first obtains a descent direction by optimizing a linear approximation of the convex loss function at the current iterate $\mathbf{g}^{(k-1)}$:

$$\min_{\mathbf{v}\in\text{conv}(\overline{\mathcal{P}})} \mathsf{loss}(\mathbf{g}^{(k-1)}) + \left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{v} - \mathbf{g}^{(k-1)} \right\rangle, \tag{6}$$

where the objective function in the above subproblem describes a supporting hyperplane to the convex loss function $\mathsf{loss}(\cdot)$ at the current iterate $\mathbf{g}^{(k-1)}$. The optimal solution, say, $\mathbf{v}^*$, provides the optimal direction $\mathbf{d}^* = \mathbf{v}^* - \mathbf{g}^{(k-1)}$. It can be shown that $\mathbf{d}^*$ is a descent direction if $\mathbf{g}^{(k-1)}$ is not already an optimal solution to the Convex Mixture problem. The next solution $\mathbf{g}^{(k)}$ is obtained by taking a step $\alpha \in [0,1]$ in the direction of $\mathbf{d}^*$, so that $\mathbf{g}^{(k)} = \mathbf{g}^{(k-1)} + \alpha\mathbf{d}^* = \alpha\mathbf{v}^* + (1-\alpha)\mathbf{g}^{(k-1)}$. Since $\mathbf{v}^*$ and $\mathbf{g}^{(k-1)}$ both belong to $\text{conv}(\overline{\mathcal{P}})$ and $\text{conv}(\overline{\mathcal{P}})$ is convex, it follows that $\mathbf{g}^{(k)} \in \text{conv}(\overline{\mathcal{P}})$ for any $\alpha \in [0,1]$.

Solving the above subproblem is the most computationally challenging component in each iteration of the CG algorithm. In our context, we have additional structure that we can exploit to solve this subproblem. Specifically, the objective function is linear in the decision variable $\mathbf{v}$. And, linear functions always achieve optimal solutions at extreme points when optimized over a convex set. Our constraint set $\text{conv}(\overline{\mathcal{P}})$ is the convex hull of all the atomic likelihood vectors in $\overline{\mathcal{P}}$. Therefore, the set of extreme points of the constraint set is a subset of $\overline{\mathcal{P}}$. It thus follows that it is sufficient to search over the set of all atomic likelihood vectors in $\overline{\mathcal{P}}$, resulting in the following optimization problem:

$$\min_{\mathbf{v}\in\overline{\mathcal{P}}} \left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{v} - \mathbf{g}^{(k-1)} \right\rangle. \tag{7}$$

Our ability to solve (7) efficiently depends on the structure of the set $\overline{\mathcal{P}}$. We discuss this aspect in more detail in Section 4.1. For now, we suppose that we have access to an oracle that returns an optimal solution, say, $\boldsymbol{f}^{(k)}$, to (7) in each iteration k.

In summary, in each iteration, the CG algorithm finds a new customer type (or atomic likelihood vector) $\boldsymbol{f}^{(k)} \in \overline{\mathcal{P}}$ and obtains the new solution $\mathbf{g}^{(k)} = \alpha\boldsymbol{f}^{(k)} + (1-\alpha)\mathbf{g}^{(k-1)}$ by putting a probability mass α on the new customer type $\boldsymbol{f}^{(k)}$ and the remaining probability mass $1-\alpha$ on the previous solution $\mathbf{g}^{(k-1)}$, for some $\alpha \in [0,1]$. In other words, the CG algorithm is iteratively adding customer types $\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots$ to the support of the mixing distribution. This aspect of the CG algorithm makes it most attractive for estimating mixing distributions. In particular, it has two implications: (a) by maintaining the individual customer types and the step sizes, we can maintain the entire mixing distribution along with the current solution $\mathbf{g}^{(k)}$, in each iteration k (see below for details); and (b) since each iteration adds (at most) one new customer type to the support, terminating the program at

iteration K results in a distribution with at most K mixture components. We use the latter property to control the complexity, as measured in terms of the number of mixture components, of the recovered mixing distribution (see the discussion below on "Stopping conditions").

We now discuss the choice of the step size α. The standard variant of the CG algorithm does a line-search to compute the optimal step size that results in the maximum improvement in the objective value. Instead, we use the "fully corrective" Frank-Wolfe (FCFW) variant (Shalev-Shwartz et al. 2010) which after finding $\boldsymbol{f}^{(k)}$ at iteration k, re-optimizes the loss function $\mathsf{loss}(\mathbf{g})$ over the convex hull of the initial solution $\mathbf{g}^{(0)}$ and the atomic likelihood vectors $\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots, \boldsymbol{f}^{(k)}$ found so far. More precisely, the algorithm computes weights $\boldsymbol{\alpha}^{(k)}$ from the $(k+1)$-dimensional simplex Δ_k that minimize the loss function and obtains the next iterate $\mathbf{g}^{(k)} := \alpha_0^{(k)}\mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)} \boldsymbol{f}^{(s)}$. The weights $\boldsymbol{\alpha}^{(k)} = (\alpha_0^{(k)}, \alpha_1^{(k)}, \alpha_2^{(k)}, \ldots, \alpha_k^{(k)})$ represent the proportions of each of the mixture components. This variant of the CG algorithm makes more progress in each iteration and is therefore most suited when the subproblems in (7) are hard to solve. It also promotes sparser solutions (Jaggi 2013) containing fewer mixture components. Algorithm 1 summarizes the entire procedure.

Algorithm 1 CG algorithm for estimating the mixing distribution

1: **Initialize:** $k = 0$; $\mathbf{g}^{(0)} \in \overline{\mathcal{P}}$ such that both $\mathsf{loss}(\mathbf{g}^{(0)}), \nabla\mathsf{loss}(\mathbf{g}^{(0)})$ are bounded, and $\boldsymbol{\alpha}^{(0)} = (1)$
2: **while** stopping condition is not met **do**
3: $\quad k \leftarrow k + 1$
4: $\quad$ Compute $\boldsymbol{f}^{(k)} \in \arg\min_{\mathbf{v} \in \overline{\mathcal{P}}} \left\langle \nabla\mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{v} - \mathbf{g}^{(k-1)} \right\rangle$ (support finding step)
5: $\quad$ Compute $\boldsymbol{\alpha}^{(k)} \in \arg\min_{\boldsymbol{\alpha} \in \Delta_k} \mathsf{loss}\left(\alpha_0 \mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s \boldsymbol{f}^{(s)}\right)$ (proportions update step)
6: $\quad$ Update $\mathbf{g}^{(k)} := \alpha_0^{(k)}\mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)} \boldsymbol{f}^{(s)}$
7: **end while**
8: **Output:** mixture proportions $\alpha_0^{(k)}, \alpha_1^{(k)}, \alpha_2^{(k)}, \ldots, \alpha_k^{(k)}$ and customer types $\mathbf{g}^{(0)}, \boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots, \boldsymbol{f}^{(k)}$

We discuss a few key features of the algorithm. First, the algorithm outputs both the support and the mixture proportions of the mixing distribution, as desired. Second, the proportions update step is also a constrained convex optimization problem, but over a much smaller domain compared to $\mathrm{conv}(\overline{\mathcal{P}})$. We show below that this step can be solved efficiently. Third, Algorithm 1 is agnostic to the choice of the loss function loss (so long as it is convex and differentiable) and readily applies to both the NLL and SQ loss functions. Finally, although not the focus in this work, standard errors for the parameters of the recovered mixing distribution (or relevant summary statistics such as the price elasticity) can be computed via bootstrapping, as is commonly done in the literature; see for instance, Train (2008). Furthermore, as discussed in Section 2, for the negative log-likelihood loss, our method reduces to classical nonparametric maximum likelihood estimation (NPMLE) of the mixing distribution, and therefore inherits its statistical properties. For a detailed discussion on NPMLE, we refer the reader to Lindsay (1995) and references therein.

4.1. Implementation Details

Here, we discuss a few key implementation details for Algorithm 1.

Solving the support finding step. For each loss function introduced in Section 3, the support finding step in iteration k can be written as (by plugging in the gradients and dropping constant terms):

$$\begin{aligned} \mathsf{NLL}: \quad & \min_{\boldsymbol{\omega}\in\mathbb{R}^D} -\frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t}\left(\frac{N_{jt}}{\mathrm{g}_{jt}^{(k-1)}}\right)\cdot\frac{\exp\left(\boldsymbol{\omega}^\top\mathbf{z}_{jt}\right)}{\sum_{\ell\in S_t}\exp\left(\boldsymbol{\omega}^\top\mathbf{z}_{\ell t}\right)} \\ \mathsf{SQ}: \quad & \min_{\boldsymbol{\omega}\in\mathbb{R}^D} \frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t}\left(N_t\cdot\mathrm{g}_{jt}^{(k-1)}-N_{jt}\right)\cdot\frac{\exp\left(\boldsymbol{\omega}^\top\mathbf{z}_{jt}\right)}{\sum_{\ell\in S_t}\exp\left(\boldsymbol{\omega}^\top\mathbf{z}_{\ell t}\right)} \end{aligned} \tag{8}$$

The optimal solutions to the problems above may be unbounded. These unbounded solutions correspond to the atomic likelihood vectors in $\overline{\mathcal{P}}\setminus\mathcal{P}$, as shown in Section 5.2. We solve the problems in (8) approximately using a general-purpose non-linear program solver (Nocedal and Wright 2006). Solving these optimization problems exactly is computationally hard because they are non-convex, as shown in Appendix D. However, we only need to generate an improving solution, that is, find a feasible descent direction (see description above equation 6), to ensure convergence of the algorithm. In our numerical experiments, we found that the standard Broyden-Fletcher-Goldfarb-Shanno (BFGS) method was sufficient to obtain improving solutions.

Solving the proportions update step. This step is itself a constrained convex program, so we use the CG algorithm to solve it. Instead of the variant described above, we adopt the approach of Krishnan et al. (2015) who recently proposed a modified Frank-Wolfe algorithm to approximately solve the proportions update step. In contrast to the standard CG algorithm described above, this variant performs two kinds of steps to update the support of the mixing distribution in each iteration: a support finding step that finds a customer type to be added to the mixture and an "away" step (Guélat and Marcotte 1986) that reduces probability mass (possibly to zero) from a customer type in the existing mixing distribution. Moreover, the support finding step can be solved exactly by searching over the $k+1$ extreme points of the $(k+1)$-dimensional simplex Δ_k. The next iterate is then computed based on which step—support finding step or away step—results in higher improvement in the objective value (see Alg. 3 in Appendix B of Krishnan et al. 2015 for the details). The presence of away steps means that we can (sometimes) 'drop' existing customer types from the mixing distribution, thereby resulting in solutions with fewer number of mixture components.

Initialization. We can start with any $\mathbf{g}^{(0)}\in\overline{\mathcal{P}}$ as the initial solution such that the starting objective $\mathsf{loss}(\mathbf{g}^{(0)})$ and gradient $\nabla\mathsf{loss}(\mathbf{g}^{(0)})$ are bounded. Since we are fitting a mixture of logit models, a natural choice is to fit an LC-MNL model with a "small" number of classes (or even a single class MNL model) to the data and use that as the initialization. In particular, the MNL log-likelihood objective is globally concave in the parameter $\boldsymbol{\omega}$ and there exists efficient algorithms (Hunter 2004)

for its estimation that converge quickly in practice. In our empirical case studies, we initialize by fitting a 2-class LC-MNL model to the data.

Stopping conditions. We can use many stopping conditions to terminate the algorithm: (1) Jaggi (2013) showed that if the subproblem can be solved optimally in each iteration, then we can compute an upper bound on the "optimality gap" of the current solution $\mathbf{g}^{(k)}$, i.e. $\mathsf{loss}(\mathbf{g}^{(k)}) - \mathsf{loss}(\mathbf{g}^*)$ where $\mathbf{g}^*$ denotes the optimal solution to the CONVEX MIXTURE problem. In this case, we can choose an arbitrarily small $\delta > 0$ and choose to terminate the algorithm when $\mathsf{loss}(\mathbf{g}^{(k)}) - \mathsf{loss}(\mathbf{g}^*) \leq \delta$. However, this might result in overfitting—because of the presence of a large number of mixture components—and consequently, perform poorly in out-of-sample predictions. (2) We can utilize standard information-theoretic measures proposed in the mixture modeling literature (McLachlan and Peel 2000) such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) etc. that capture model complexity as a function of the number of mixture components and prevent overfitting. (3) Finally, a simple way to control for model complexity is to just limit the number of iterations of the algorithm[7] at some $k = K_{\max}$ according to the maximum number of customer types that we may be interested in finding. This ensures that the estimated mixture is composed of at most $K_{\max}$ types and we use this stopping condition in our empirical case studies.

5. Theoretical analysis of the estimator

In this section, we derive the convergence rate of our estimator and also theoretically characterize the customer types recovered by our method.

5.1. Convergence rate of the estimator

To state our result on the convergence rate, we need the following notation. For each offer-set S_t, let $\boldsymbol{y}_t \stackrel{\text{def}}{=} (y_{jt})_{j \in S_t}$ denote the vector of sales fractions for offer-set S_t. Let $H(\boldsymbol{y}_t) \stackrel{\text{def}}{=} -\sum_{j \in S_t} y_{jt} \log y_{jt}$ denote the entropy of the vector $\boldsymbol{y}_t$. As $0 \leq y_{jt} \leq 1$, $H(\boldsymbol{y}_t) \geq 0$ for all $t \in [T]$.[8] Moreover, let $D_{KL}(p\|q) \stackrel{\text{def}}{=} p\log(p/q) + (1-p)\log((1-p)/(1-q))$ denote the relative entropy (aka KL-divergence) between p and q for any $0 \leq p, q \leq 1$. It is a known fact that $D_{KL}(p\|q) \geq 0$ for all $0 \leq p, q \leq 1$ and $D_{KL}(p\|q) = 0$ if and only if $p = q$. Finally, let $y_{t,\min} \stackrel{\text{def}}{=} \min\{y_{jt} \mid j \in S_t \text{ s.t. } y_{jt} > 0\}$ and $y_{\min} \stackrel{\text{def}}{=} \min\{y_{t,\min} \mid t \in [T]\}$. Then, we can establish the following convergence guarantee:

THEOREM 1 **(Sublinear convergence)**. *Let* $\mathbf{g}^*$ *denote the optimal solution to the* CONVEX MIXTURE *problem, and* $\mathbf{g}^{(k)}$ *denote the* k*th iterate generated by Algorithm 1. Then, for the loss functions defined in Section 3, it follows*

$$\mathsf{SQ}(\mathbf{g}^{(k)}) - \mathsf{SQ}(\mathbf{g}^*) \leq \frac{4}{k+2} \qquad \text{for all } k \geq 1,$$

$$\mathsf{NLL}(\mathbf{g}^{(k)}) - \mathsf{NLL}(\mathbf{g}^*) \leq \frac{4}{\xi_{\min}^2 \cdot (k+\kappa)} \qquad \text{for all } k \geq \bar{K} \text{ for some constant } \kappa \text{ and index } \bar{K},$$

[7] Following the *early stopping* rule in machine learning literature, see for instance Yao et al. (2007) and Prechelt (2012).

[8] We use the standard convention that $0 \cdot \log 0 = 0$ when computing the entropy.

where $\xi_{\min}$ *is the smallest* ξ *such that* $0 < \xi \le y_{\min}$ *and*

$$\min_{1\le t\le T} N_t \cdot D_{KL}(y_{t,\min}\|\xi) \le N \cdot \mathsf{NLL}(\mathbf{g}^{(0)}) - \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}).$$

Such a $\xi_{\min}$ *always exists.*

The result above establishes an $O(1/k)$ convergence guarantee for our estimator for both loss functions, assuming that the support finding step in Algorithm 1 can be solved optimally. The detailed proof is given in Appendix A.2; here, we provide a proof sketch. Our proof builds on existing techniques developed for establishing convergence rates of the CG algorithm. This is an active area of research with different rates having been derived for different variants of the CG algorithm, under different assumptions for the structures of the objective function and the constraint set (Jaggi 2013, Garber and Hazan 2015, Lacoste-Julien and Jaggi 2015). The convergence guarantee for the SQ loss function, in fact, follows directly from the existing result in Jaggi (2013), which shows that the CG algorithm converges at an $O(1/k)$ rate if the so-called *curvature constant* is bounded from above. If the domain is bounded and the hessian of the objective function is bounded from above, then the curvature constant is known to be bounded from above. In our case, the domain $\text{conv}(\overline{\mathcal{P}})$ is bounded (since any vector $\boldsymbol{f} \in \text{conv}(\overline{\mathcal{P}})$ has entries between 0 and 1). The hessian of the SQ loss is a diagonal matrix, where each entry is bounded above by 1. Therefore, it follows that the curvature constant is bounded from above, thus allowing us to establish the $O(1/k)$ guarantee by directly invoking existing results.

The hessian of the NLL loss function, on the other hand, is not bounded from above. For simplicity, suppose that $N_{jt} > 0$ for all $t \in [T]$ and any $j \in S_t$; our result also holds when some of the sales counts are 0, see the discussion at the beginning of Appendix A.2.2. Then, it is easy to see that the hessian is a diagonal matrix with the entry corresponding to (product, offer-set) pair (j, S_t) equal to $N_{jt}/(N \cdot \mathrm{g}_{jt}^2)$. Since g_{jt} can be arbitrarily close to 0 in the domain $\text{conv}(\overline{\mathcal{P}})$, the diagonal entries are not bounded from above, and thus, existing results don't directly apply. To address this issue, suppose that we can establish a non-trivial lower bound, say, $\xi^* > 0$, for the optimal solution $\mathbf{g}^*$ so that $\mathrm{g}^*_{jt} \ge \xi^* > 0$ for all $t \in [T]$ and all $j \in S_t$. It then follows that the hessian of the NLL loss is bounded from above when the domain is restricted to $\tilde{\mathcal{D}} \stackrel{\text{def}}{=} \left\{\mathbf{g} \in \text{conv}(\overline{\mathcal{P}}) :\ \mathrm{g}_{jt} \ge \xi^*\ \forall\ t \in [T]\ \forall\ j \in S_t\right\}$. And, if we solve Convex Mixture over the restricted domain $\tilde{\mathcal{D}}$,[9] we immediately obtain the $O(1/k)$ convergence rate.

While solving Convex Mixture over the restricted domain $\tilde{\mathcal{D}}$ is feasible in principle, it is difficult to implement in practice because computing a good lower bound ξ^* may not be straightforward. Instead, we show that running the fully corrective variant of the CG algorithm (the variant implemented in Algorithm 1), while being agnostic to a lower bound, still converges at $O(1/k)$ rate. For that, we first show that each iterate $\mathbf{g}^{(k)}$ generated by Algorithm 1 is bounded from below by $\xi_{\min}$, where $\xi_{\min}$ is as

[9] It can be verified that the constraint set $\tilde{\mathcal{D}}$ is still compact and convex.

defined in Theorem 1. Then, we exploit this property to establish the $O(1/k)$ convergence rate with the constant scaling in $1/\xi_{\min}^2$.

To get the best convergence rate, we need to use the tightest lower bound $\xi_{\min}$. Our bound is derived for general cases, and in this generality, the bound is tight. To see that, consider the setting when the observations consist of only market shares, so that $T=1$, $S_1=[n]$, and the sales fractions $\boldsymbol{y}_1$ comprise the observed market shares. In this case, it can be shown that the optimal solution $\mathbf{g}^*=\boldsymbol{y}_1$.[10] When Algorithm 1 is initialized at $\mathbf{g}^{(0)}=\boldsymbol{y}_1$, it follows from the definition that $\xi_{\min}=y_{\min}=y_{1,\min}$, which is the tightest bound possible.

We can also derive a simple-to-compute (lower) bound for $\xi_{\min}$, as stated in the following proposition:

PROPOSITION 1. *Let* $N_{\min}=\min\{N_{jt}\mid t\in[T]; j\in S_t \text{ s.t. } N_{jt}>0\}$. *Then, it follows that*

$$y_{\min}\geq\xi_{\min}\geq y_{\min}\cdot\exp\left(-1-\frac{N\cdot\mathsf{NLL}(\mathbf{g}^{(0)})-\sum_{t=1}^{T}N_t\cdot H(\boldsymbol{y}_t)}{N_{\min}}\right).$$

When $T=1$, $S_1=[n]$, and $\mathbf{g}^{(0)}=\boldsymbol{y}_1$, it follows from the above proposition that $y_{\min}\geq\xi_{\min}\geq y_{\min}/e$. Therefore, the simple-to-compute (lower) bound loses a factor of e in this case.

Remark. Theorem 1 assumes that the support finding step in Algorithm 1 can be solved optimally.[11] In cases where the optimal solution cannot be found, a weaker convergence guarantee can be established as long as the iterates are (sufficiently) improving, i.e., $\mathsf{loss}(\mathbf{g}^{(k)})<\mathsf{loss}(\mathbf{g}^{(k-1)})$, for each iteration k. In this case, it follows from existing results (see for instance Zangwill 1969) that the sequence of iterates converges to a stationary point, which in the case of a convex program is an optimal solution.

5.2. Characterization of the recovered mixture types

We now focus on the support finding step and characterize the structure of the optimal solution. These solutions comprise the support of the resulting mixing distribution. In each iteration k, the support finding step is equivalent to solving the following problem (by dropping constant terms):

$$\min_{\boldsymbol{f}\in\overline{\mathcal{P}}}\sum_{t=1}^{T}\sum_{j\in S_t}c_{jt}^{(k)}f_{jt},$$

where $c_{jt}^{(k)}=\left(\nabla\mathsf{loss}(\mathbf{g}^{(k-1)})\right)_{jt}$. The optimal solution $\boldsymbol{f}^{(k)}$ to the above problem lies either in $\mathcal{P}$ or $\overline{\mathcal{P}}\setminus\mathcal{P}$. If it lies in $\mathcal{P}$, then (by definition) there exists a parameter vector $\boldsymbol{\omega}_k\in\mathbb{R}^D$ such that $\boldsymbol{f}(\boldsymbol{\omega}_k)=\boldsymbol{f}^{(k)}$, so that any such $\boldsymbol{\omega}_k$ may be used to describe the customer type and make the choice probability prediction $e^{\boldsymbol{\omega}_k^\top\mathbf{z}_{jS}}/\left(\sum_{\ell\in S}e^{\boldsymbol{\omega}_k^\top\mathbf{z}_{\ell S}}\right)$ for the probability of choosing product j from some offer-set S. However, if the optimal solution $\boldsymbol{f}^{(k)}$ lies in the boundary, i.e. $\overline{\mathcal{P}}\setminus\mathcal{P}$, then there is no straightforward way to characterize the customer type or make out-of-sample predictions. To deal with this challenge, we provide a compact characterization of what we call the *boundary types*, defined as follows:

[10] Provided the product features satisfy certain structural conditions; see Theorem 4.

[11] Actually, Jaggi (2013) showed that solving it approximately with some fixed additive error is also sufficient to ensure the $O(1/k)$ convergence rate.

Definition 1 (Boundary and Non-boundary types). A customer type $\boldsymbol{f}$ is called a boundary type if $\boldsymbol{f} \in \overline{\mathcal{P}} \setminus \mathcal{P}$, and a non-boundary type, otherwise.

We show below that each boundary type is characterized by two parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$:

Theorem 2 **(Characterization of boundary types)**. *Given a boundary type $\boldsymbol{f}$ in $\overline{\mathcal{P}} \setminus \mathcal{P}$, there exist parameters $\boldsymbol{\omega}_0, \boldsymbol{\theta} \in \mathbb{R}^D$ such that, for each $1 \le t \le T$ and $j \in S_t$, we have*

$$f_{jt} = \lim_{r \to \infty} \frac{\exp\left((\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top \mathbf{z}_{jt}\right)}{\sum_{\ell \in S_t} \exp\left((\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top \mathbf{z}_{\ell t}\right)}.$$

Furthermore, $f_{jt} = 0$ for at least one (product, offer-set) pair (j, S_t).

The proof in Appendix A.3 shows how to compute the parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ given any boundary type $\boldsymbol{f} \in \overline{\mathcal{P}} \setminus \mathcal{P}$. Here, we focus on understanding the implications of the above characterization.

First, since for any boundary type $\boldsymbol{f}$, $f_{jt} = 0$ for at least one (product, offer-set) pair (j, S_t), there exists *no* logit parameter vector $\boldsymbol{\omega}$ such that $f_{jt} = e^{\boldsymbol{\omega}^\top \mathbf{z}_{jt}} / \left(\sum_{\ell \in S_t} e^{\boldsymbol{\omega}^\top \mathbf{z}_{\ell t}}\right)$ for all j, S_t. Second, boundary types arise as a result of limiting logit models, obtained as the parameter vector $\boldsymbol{\omega}$ is pushed to infinity. In particular, Theorem 2 states that for any boundary type $\boldsymbol{f}$, there exists parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ such that the choice probabilities for observed (product, offer-set) pairs under $\boldsymbol{f}$ are equal to those under the limiting type $\lim_{r \to \infty} \boldsymbol{f}(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})$, where recall that $\boldsymbol{f}(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})$ corresponds to the customer type with logit parameter $\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta}$. Below, we discuss this characterization in more detail.

The key aspect of our characterization is a preference ordering over the products defined by the parameter vector $\boldsymbol{\theta}$. This preference order determines the choice of the products from a given offer-set. For ease of exposition, we describe the preference ordering for the case when product features don't change with the offer-set, so we write $\mathbf{z}_j$ instead of $\mathbf{z}_{jt}$ for the feature vector of product j. The discussion below extends immediately to the more general case by associating a separate product to each feature vector of interest. To describe the preference order, define product utility $u_j \stackrel{\text{def}}{=} \boldsymbol{\theta}^\top \mathbf{z}_j$ for product j. These utility values can be visualized as projections of the product feature vectors $\mathbf{z}_j$'s onto the vector $\boldsymbol{\theta}$. They define a preference order $\succeq$ among the products such that $j \succeq j'$, read as "product j is weakly preferred over product j'," if and only if $u_j \ge u_{j'}$. The relation $\succeq$ is in general a weak ordering and *not* a strict ordering because product utilities may be equal. In order to explicitly capture indifferences, we write $j \succ j'$ if $u_j > u_{j'}$ and $j \sim j'$ if $u_j = u_{j'}$.

Now, when offered a set S, customers of this type purchase only the most preferred products as determined according to the preference order $\succeq$. 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 $u^* \stackrel{\text{def}}{=} \max\{u_j : j \in S\}$ denote the maximum utility among the products in S. We have that $u^* = u_j$

for all $j \in C(S)$ and $u^* > u_j$ for all $j \in S \setminus C(S)$. Given this and multiplying the numerator and denominator of the choice probabilities defined in Theorem 2 by $e^{-r \cdot u^*}$, we can write for any $j \in S$,

$$\frac{\exp\left((\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top \mathbf{z}_j\right)}{\sum_{\ell \in S} \exp\left((\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top \mathbf{z}_\ell\right)} = \frac{e^{-r \cdot (u^* - u_j)} \cdot \exp(\boldsymbol{\omega}_0^\top \mathbf{z}_j)}{\sum_{\ell \in C(S)} \exp(\boldsymbol{\omega}_0^\top \mathbf{z}_\ell) + \sum_{\ell \in S \setminus C(S)} e^{-r \cdot (u^* - u_\ell)} \cdot \exp(\boldsymbol{\omega}_0^\top \mathbf{z}_\ell)}. \tag{9}$$

When we take the limit as $r \to \infty$, each of the terms $e^{-r \cdot (u_\ell - u^*)}$, $\ell \in S \setminus C(S)$, goes to zero, so the denominator converges to $\sum_{\ell \in C(S)} \exp(\boldsymbol{\omega}_0^\top \mathbf{z}_\ell)$. The numerator converges to $\exp(\boldsymbol{\omega}_0^\top \mathbf{z}_j)$ if $j \in C(S)$ and 0 if $j \in S \setminus C(S)$. Therefore, we obtain the following choice probability prediction $f_{j,S}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ for any product j and offer-set S from Theorem 2:

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

From the discussion above, we note the contrasting roles of the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}_0$. The parameter vector $\boldsymbol{\theta}$ (through the preference ordering $\succeq$ it induces) determines the *consideration set* $C(S)$, whereas the parameter vector $\boldsymbol{\omega}_0$ determines the logit choice probabilities from within the consideration set. The parameter vector $\boldsymbol{\theta}$ dictates how a product's features impact its inclusion into the consideration set. For instance, suppose u^* is the maximum utility in offer-set S and product j with utility $u_j < u^*$ is not in consideration now. Further, suppose one of the features is price and the corresponding coefficient is $\theta_p < 0$. Then, product j will enter into consideration only if its price is sufficiently dropped to make its utility greater than or equal to u^*. In other words, the price should be dropped by at least $(u^* - u_j)/|\theta_p|$ to ensure consideration of product j.

The choice behavior we identify for the boundary types is consistent with existing literature, which establishes that customers often consider a subset of the products on offer before making the choice (Jagabathula and Rusmevichientong 2016). In fact, consideration sets of the kind we identify are a special case of the linear compensatory decision rule that has been used as a heuristic for forming consideration sets in existing literature (Hauser 2014). The rule computes the utility for each product as a weighted sum of the feature values and chooses all products that have a utility greater than a cutoff to be part of the consideration set. Finally, multiple distinct tuples of parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ can result in the same limiting choice probabilities $\boldsymbol{f}$ for the observed data. Since the data do not provide any further guidance, we arbitrarily select one of them. Studying the impact of different selection rules on the prediction accuracy is a promising avenue for future work.

We conclude this subsection with the following systematic procedure that summarizes our discussion for making out-of-sample choice predictions for a boundary type:

Algorithm 2 Predicting choice probabilities for boundary type $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$

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

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

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

5.3. Analysis of recovered distribution for two special cases

We now analyze scenarios under which the optimal solution to the support finding step is indeed a boundary type. This helps in providing further insights into the structure of the recovered mixing distribution. Solving the support finding step in the general case is a hard problem and therefore, to keep the analysis tractable, we focus on the setting in which the data consist of sales counts in a single time-period when all products are offered (such as market shares data). For this case, the notation can be simplified.

Since there is only a single offer-set $S_1 = [n]$, we represent the features as $\mathbf{z}_i$ for each product $i \in [n]$. Further, the sales counts can be represented using a single vector $\boldsymbol{y} := (y_1, y_2, \ldots, y_n) \in [0,1]^n$ such that $\sum_{i=1}^n y_i = 1$ where $y_i \geq 0$ is the fraction of sales for product i. The choice probabilities $\boldsymbol{f} \in \overline{\mathcal{P}}$ are of the form $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)$, also satisfying $\sum_{i=1}^n f_i = 1$. Similarly, the estimates produced by Algorithm 1 at any iteration k are of the form $\mathbf{g}^{(k)} = (\mathrm{g}_1^{(k)}, \mathrm{g}_2^{(k)}, \ldots, \mathrm{g}_n^{(k)})$, where again $\sum_{i=1}^n \mathrm{g}_i^{(k)} = 1$. With this notation, the loss functions defined in Section 3 can be written as:

$$\mathsf{NLL}(\mathbf{g}) = -\sum_{i=1}^n y_i \log(\mathrm{g}_i) \ \ ; \ \ \mathsf{SQ}(\mathbf{g}) = \frac{1}{2}\sum_{i=1}^n (y_i - \mathrm{g}_i)^2, \tag{10}$$

while the support finding step is of the form, with $c_i \stackrel{\text{def}}{=} -\left(\nabla \mathsf{loss}(\mathbf{g}^{(k-1)})\right)_i$ for each $i \in [n]$ (we switch to maximization to aid the analysis below):

$$\max_{\boldsymbol{f} \in \overline{\mathcal{P}}} \sum_{i=1}^n c_i \cdot f_i, \tag{11}$$

where we drop the explicit dependence of the coefficient c_i on the iteration number k for simplicity of notation. We analyze the optimal solution to the above subproblem under two cases: (1) all product features are continuous, and (2) some product features are binary.[12]

[12] This subsumes the setting of categorical features since a categorical feature is usually transformed into a set of binary features using an encoding scheme like dummy coding or one-hot coding.

5.3.1. All product features are continuous. When all features are continuous, the optimal solution to subproblem (11) depends on the geometric structure of the observed product features. Specifically, we consider the (convex) polytope formed by the convex hull of the product features $\mathbf{z}_1, \ldots, \mathbf{z}_n$, denoted as $\mathcal{Z}_n \stackrel{\text{def}}{=} \text{conv}(\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\})$. For this polytope, we define an extreme point as:

DEFINITION 2 (EXTREME POINTS). $\mathbf{z}_j$ is called an *extreme point* of the convex polytope $\mathcal{Z}_n$ if $\mathbf{z}_j \notin \text{conv}(\{\mathbf{z}_i : i \neq j, 1 \leq i \leq n\})$. Equivalently, extreme points correspond to vertices of $\mathcal{Z}_n$.

With this definition, we can establish conditions under which a boundary type is an optimal solution to the support finding step (11). In particular, we have the following result:

THEOREM 3 **(Recovery of boundary types).** *Suppose we observe sales data for only the offer-set $[n]$. Let $j_{\max} = \arg\max_{j \in [n]} c_j$. If $\mathbf{z}_{j_{\max}}$ is an extreme point of the polytope $\mathcal{Z}_n$, then the boundary type $\boldsymbol{f}(\mathbf{0}, \boldsymbol{\theta}_{j_{\max}})$ is an optimal solution to* support finding step (11)*, where $\boldsymbol{\theta}_{j_{\max}}$ is such that $\boldsymbol{\theta}_{j_{\max}}^\top \mathbf{z}_{j_{\max}} > \boldsymbol{\theta}_{j_{\max}}^\top \mathbf{z}_j$ for all $j \neq j_{\max}$. In particular, $\boldsymbol{f}(\mathbf{0}, \boldsymbol{\theta}_{j_{\max}})$ is of the form:*

$$f_j(\mathbf{0}, \boldsymbol{\theta}_{j_{\max}}) = \begin{cases} 1 & \text{if } j = j_{\max} \\ 0 & \text{otherwise,} \end{cases}$$

The proof in Appendix A.4 shows the existence of such a $\boldsymbol{\theta}_{j_{\max}}$. The above result shows that our estimation method recovers boundary types that consider only a single product amongst the offered products. The result also leads to the following corollary:

COROLLARY 1 **(All extreme points $\implies$ Boundary types always optimal).** *Suppose we observe sales data for only the offer-set $[n]$. If all feature vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$ are extreme points of the polytope $\mathcal{Z}_n$, then boundary types are always optimal solutions for the* support finding step (11)*.*

The above result implies that when all product features are extreme points of the polytope $\mathcal{Z}_n$, the support finding step (11) can be solved to optimality in each iteration, where the optimal solution corresponds to a boundary type that chooses a single product with probability 1 from amongst all offered products. Consequently, our estimation method decomposes the population into such boundary types to explain the observed choice data. In fact, in this scenario, we can also establish the following convergence guarantee for the iterates generated by Algorithm 1:

THEOREM 4 **(Convergence in finite number of iterations).** *Suppose we observe sales data for only the offer-set $[n]$. Further, suppose that $\mathbf{z}_j$ is an extreme point of the polytope $\mathcal{Z}_n$ for all $j \in [n]$. For both the* NLL *and* SQ *loss functions defined in* (10)*, the estimates $\mathbf{g}^{(k)}$ produced by Algorithm 1 converge to the optimal solution $\mathbf{g}^*$ in at most n iterations. In particular, the optimal solution $\mathbf{g}^* = \boldsymbol{y}$ and consequently, the CG algorithm is able to perfectly match the observed sales fractions.*

Due to the complexity of the resulting optimization problems, there are few convergence guarantees for the estimation of logit models that exist in the literature. For instance, Hunter (2004) presents

necessary and sufficient conditions for an iterative minorization-maximization (MM) algorithm to converge to the maximum likelihood estimate for a *single class* MNL model. Recently, James (2017) proposed an MM algorithm for estimation of mixed logit models with a multivariate normal mixing distribution, but did not provide any conditions for convergence. To the best of our knowledge, our result is one of the first to provide a convergence guarantee for general mixtures of logit models.

5.3.2. Some (or all) product features are binary. Next, we consider the case when some of the features are binary. To state the result, we need to introduce additional notation. For each product $\ell \in [n]$, let $\mathbf{z}_\ell \in \mathbb{R}^{D_1}$ and $\mathbf{b}_\ell \in \{0,1\}^{D_2}$ represent a set of continuous and binary features respectively, where $D_1 + D_2 = D$ and $D_2 > 0$. Define the binary relation $\sim$ on $[n]$ as: $i \sim j \iff \mathbf{b}_i = \mathbf{b}_j$. It is easy to see that $\sim$ is an equivalence relation on $[n]$, and therefore let $\mathcal{E}$ represent the equivalence classes, i.e. $[n] = \bigcup_{e\in\mathcal{E}} S_e$ and $S_{e_1} \cap S_{e_2} = \emptyset$ for all $e_1, e_2 \in \mathcal{E}$ such that $e_1 \neq e_2$.

With the above notation, we can show that the optimal solution to the support finding step always corresponds to a boundary type, having the following structure:

THEOREM 5 **(Binary feature $\implies$ Boundary types are always optimal)**. *Suppose we observe sales data for only the offer-set $[n]$. Then, there exists $e^* \in \mathcal{E}$ such that the optimal solution to* support finding step (11) *is a boundary type $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \mathbb{R}^D$ satisfying*

$$\boldsymbol{\theta}^\top(\mathbf{z}_j \circ \mathbf{b}_j) > \boldsymbol{\theta}^\top(\mathbf{z}_i \circ \mathbf{b}_i) \;\; \forall\, j \in S_{e^*};\; \forall\, i \in [n] \setminus S_{e^*},$$

where $\circ$ denotes vector concatenation. In particular, $f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = 0$ for all $i \in [n] \setminus S_{e^*}$ so that the boundary type only considers products within subset $S_{e^*} \subset [n]$.

Theorem 5 establishes that if products have certain binary features (in addition to continuous features), then the support finding step (11) always has a boundary type as the optimal solution. The consideration sets of the resulting types follow a conjunctive decision rule (Hauser 2014), where customers screen products with a set of "must have" or "must not have" aspects—corresponding to each binary attribute—reflecting (strong) non-compensatory preferences. We can interpret the above result in the context of our sushi case study (see Section 7.1), where the products represent two different kinds of sushi varieties—*maki* and *non-maki*. The above result says that we recover boundary types in each iteration, each of which only consider one kind of sushi variety: either maki or non-maki. Note that the mixing distribution can contain more than one boundary type with the same consideration set, as the types will be differentiated in their choice behavior according to the parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$. In particular, based on the value of the parameter $\boldsymbol{\theta}$, even some products within subset S_{e^*} may *not* be considered by the boundary type. We analyze the structure of the recovered mixing distribution in more detail in the case study.

5.4. Heuristic approaches based on above theoretical results for solving the support finding step in the general case

Characterizing the optimal solution to the support finding step in the general case is hard since the structure of the optimal solution is governed by the particular values of the coefficients $\left\{c_{jt}^{(k)}\right\}_{j,t}$ at any iteration k—which themselves are dependent on the initial solution $\mathbf{g}^{(0)}$—and the product feature variations within each offer-set. Nevertheless, our theoretical results for the characterization of boundary types (Section 5.2) as well as the single offer-set case (Section 5.3) inform the design of some heuristics for solving the support finding step in more general scenarios:

- *Single offer-set with all continuous features.* If the condition in Theorem 3 is satisfied, we know that the optimal solution to the support finding step is a boundary type. Otherwise, the optimal solution can be a boundary or non-boundary type. If we define $\mathcal{Z}_{n,\text{ext}} \stackrel{\text{def}}{=} \{j \in [n] \mid \mathbf{z}_j \text{ is an extreme point of } \mathcal{Z}_n\}$, then it follows that $\boldsymbol{f}(\mathbf{0}, \boldsymbol{\theta}_j) \in \overline{\mathcal{P}}$ for all $j \in \mathcal{Z}_{n,\text{ext}}$, where $\boldsymbol{\theta}_j$ is as defined in Theorem 3. Consequently, the set of such boundary types, $\mathcal{B} = \{\boldsymbol{f}(\mathbf{0}, \boldsymbol{\theta}_j) \mid j \in \mathcal{Z}_{n,\text{ext}}\}$, provides a feasible search set for subproblem (11). We can also determine a non-boundary type, say $\boldsymbol{f}(\boldsymbol{\omega}^{(k)})$ as an approximate solution, based on the discussion under "Solving the support finding step" in Section 4.1. Then, we can output the type which achieves the best objective as an approximate solution to (11), i.e. we output: $\arg\max_{\boldsymbol{f} \in \mathcal{B} \cup \left\{\boldsymbol{f}(\boldsymbol{\omega}^{(k)})\right\}} \sum_{i=1}^{n} c_i \cdot f_i$. We employ a heuristic based on this approach to solve the support finding step in our sushi case study in Section 7.1.

- *Multiple offer-sets.* Here again, the optimal solution can either be a boundary type or a non-boundary type. Similar to the above scenario, we can determine a non-boundary type as an approximate solution to the support finding step. In our numerical experiments, we observed, in some cases, that the (non-boundary) type output by the BFGS method was assigning very "small" probabilities to some (product, offer-set) pairs. This could be an indication that the optimal solution is actually a boundary type. Motivated by this, we can design a procedure that performs a post-hoc analysis on the type output by the BFGS method, to determine a boundary type as a candidate solution to the support finding step. We then output the customer type which achieves the better objective amongst the two as an approximate solution to the support finding step, refer to Appendix B.3 for an illustration of this procedure for our case study on the IRI Academic dataset (Section 7.2).

6. Robustness to different ground-truth mixing distributions

In this section, we use a simulation study to showcase the ability of our nonparametric estimator to obtain good approximations to various underlying mixing distributions. Our method uses only the transaction data and has no prior knowledge of the structure of the ground-truth distribution. We compare the mixing distribution estimated by our method to the one estimated by a standard random parameters logit (RPL) benchmark which makes the static assumption that the underlying mixing

distribution is multivariate normal. Our results demonstrate the cost of model misspecification—the parametric RPL benchmark yields significantly poor approximations to the ground-truth mixing distributions. On the other hand, our nonparametric method is able to automatically learn from the transaction data to construct a good approximation. This specific property of our estimator makes it very appealing in practice, where one has little knowledge of the ground-truth mixing distribution.

Setup. So that we can readily compare our method to existing methods, we borrow the experimental setup from Fox et al. (2011) for our simulation study. Fox et al. propose a nonparametric linear regression-based estimator for recovering the mixing distribution. The key distinction from our method is that they require knowledge of the support of the mixing distribution, but our method does not. We discuss the implications of this difference towards the end of this section.

The universe consists of $n = 11$ products, one of which is the no-purchase or the outside option. The firm offers all the products in the universe to the customers, but customizes the product features, offering product j with feature vector $\mathbf{z}_{jt} = (z_{jt1}, z_{jt2})$ in period t. We assume that the outside option is represented by the all zeros feature vector $(0, 0)$. Customers make choices according to a mixture of logit model with ground-truth mixing distribution Q. In each time period t, a customer makes a single choice by first sampling a MNL parameter vector $\boldsymbol{\omega}^{(t)} = (\omega_1^{(t)}, \omega_2^{(t)})$ and then choosing product j with probability

$$f_{jt}(\boldsymbol{\omega}^{(t)}) = \frac{\exp(\omega_1^{(t)} \cdot z_{jt1} + \omega_2^{(t)} \cdot z_{jt2})}{\sum_{\ell \in [n]} \exp(\omega_1^{(t)} \cdot z_{\ell t1} + \omega_2^{(t)} \cdot z_{\ell t2})}.$$

We consider three underlying ground-truth distributions Q:

1. Mixture of 2 bivariate Gaussians: $Q^{(2)} = 0.4 \cdot \mathcal{N}([3, -1], \boldsymbol{\Sigma}_1) + 0.6 \cdot \mathcal{N}([-1, 1], \boldsymbol{\Sigma}_2)$.
2. Mixture of 4 bivariate Gaussians:

$$Q^{(4)} = 0.2 \cdot \mathcal{N}([3, 0], \boldsymbol{\Sigma}_1) + 0.4 \cdot \mathcal{N}([0, 3], \boldsymbol{\Sigma}_1) + 0.3 \cdot \mathcal{N}([1, -1], \boldsymbol{\Sigma}_2) + 0.1 \cdot \mathcal{N}([-1, 1], \boldsymbol{\Sigma}_2)$$

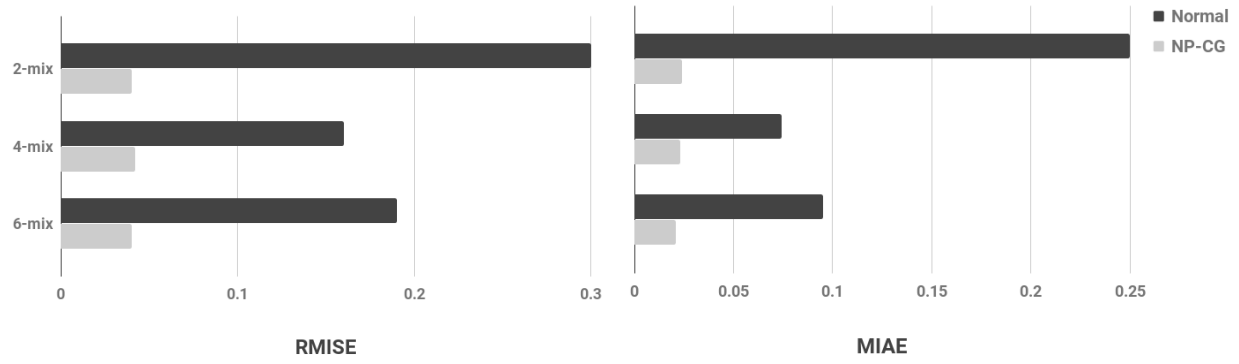
3. Mixture of 6 bivariate Gaussians:

$$Q^{(6)} = 0.1 \cdot \mathcal{N}([3, 0], \boldsymbol{\Sigma}_1) + 0.2 \cdot \mathcal{N}([0, 3], \boldsymbol{\Sigma}_1) + 0.2 \cdot \mathcal{N}([1, -1], \boldsymbol{\Sigma}_1) + 0.1 \cdot \mathcal{N}([-1, 1], \boldsymbol{\Sigma}_2) + 0.3 \cdot \mathcal{N}([2, 1], \boldsymbol{\Sigma}_2) + 0.1 \cdot \mathcal{N}([1, 2], \boldsymbol{\Sigma}_2)$$

where $\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}$ and $\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{bmatrix}$ denote the variance-covariance matrices of the component Gaussian distributions.

We generated nine instances by varying the ground-truth mixing distribution Q over the set $\{Q^{(2)}, Q^{(4)}, Q^{(6)}\}$ and the number of time periods T over the set $\{2000, 5000, 10000\}$. For each combination of Q and T and time period $t \in [T]$, we generate choice data as follows: (a) we sample product features z_{jtd} according to the distribution $\mathcal{N}(0, 1.5^2)$ independently for all products $j \in [n]$, except the no-purchase option, and for all features $d \in \{1, 2\}$; (b) we sample a logit parameter vector $\boldsymbol{\omega}^{(t)}$ from the ground-truth mixing distribution Q, and then (c) we generate a single choice $j \in [n]$ with probability $f_{jt}(\boldsymbol{\omega}^{(t)})$. Note that there is a single choice observation $N_t = 1$ in each time period

Figure 1 Error metrics for the different ground-truth mixing distributions (T = 10,000 periods)



Note. "Normal" and "NP-CG" refer to the RPL model with a bivariate normal mixing distribution and our nonparametric CG-based estimator, respectively. The labels 2-mix, 4-mix and 6-mix refer, respectively, to the ground-truth mixing distributions $Q^{(2)}$, $Q^{(4)}$ and $Q^{(6)}$, described in the main text. Lower values for the error metrics are preferred.

Table 1 Error metrics for the different ground-truth mixing distributions as a function of the number of periods T

	RMISE						MIAE					
	2-mix		4-mix		6-mix		2-mix		4-mix		6-mix	
T	Normal*	NP-CG	Normal	NP-CG	Normal	NP-CG	Normal	NP-CG	Normal	NP-CG	Normal	NP-CG
2,000	0.29	0.067	0.15	0.067	0.18	0.066	0.24	0.039	0.082	0.037	0.094	0.035
5,000	0.3	0.053	0.15	0.051	0.18	0.053	0.25	0.03	0.078	0.0289	0.094	0.028
10,000	0.3	0.04	0.16	0.042	0.19	0.04	0.25	0.024	0.074	0.023	0.095	0.021

*: The metrics for the RPL model with a bivariate normal mixing distribution (referred to as "Normal") are taken from Table 3 in Fox et al. (2011); we obtained similar numbers in our implementations. The labels 2-mix, 4-mix and 6-mix refer, respectively, to the ground-truth mixing distributions $Q^{(2)}$, $Q^{(4)}$ and $Q^{(6)}$, described in the main text. Lower values for the error metrics are preferred.

$t \in [T]$. We replicate the above process $R = 50$ times. For each replication $r \in [R]$, we obtain mixture cumulative distribution functions (CDFs) $\hat{F}_r^{RPL}$ and $\hat{F}_r^{CG}$ by fitting the standard RPL model with a bivariate normal (having non-zero correlation) mixing distribution[13] and optimizing the NLL loss using our CG algorithm, respectively. To assess the goodness of fit, we use the following two metrics proposed by Fox et al. (2011): the root mean integrated squared error (RMISE) and the mean integrated absolute error (MIAE), defined as

$$\text{RMISE} = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[\frac{1}{V}\sum_{v=1}^{V}\left(\hat{F}_r(\boldsymbol{\beta}_v) - F_0(\boldsymbol{\beta}_v)\right)^2\right]} \quad \text{and} \quad \text{MIAE} = \frac{1}{V \cdot R}\sum_{r=1}^{R}\sum_{v=1}^{V}\left|\hat{F}_r(\boldsymbol{\beta}_v) - F_0(\boldsymbol{\beta}_v)\right|,$$

where $\hat{F}_r \in \left\{\hat{F}_r^{RPL}, \hat{F}_r^{CG}\right\}$, $\boldsymbol{\beta}_v$'s represent $V = 10^4$ uniformly spaced points in the rectangle $[-6, 6] \times [-6, 6]$ where the CDF is evaluated[14] and F_0 is the CDF of the ground-truth mixing distribution Q.

[13] We solved problem (3) using Python SciPy library's `minimize` interface with the 'L-BFGS-B' method—`https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html`.

[14] The true mixing distribution's support lies in this region with probability close to 1.

Results. Figure 1 and Table 1 summarize the results we obtained when we ran our estimator for $K_{\max} = 81$ iterations.[15] Figure 1 shows a bar graph comparing our method to the RPL model on the RMISE and MIAE metrics. These metrics are compared for the three ground-truth mixing distributions, for the case with $T = 10,000$ periods. Table 1 shows a more complete comparison, including for the cases with $T = 2,000$ and $T = 5,000$ periods. We make the following observations:

1. Our nonparametric method is able to automatically construct a good approximation of the ground-truth mixing distribution Q from the transaction data, without any prior knowledge of the structure of Q. The benchmark RPL model, on the other hand, performs significantly worse because of model misspecification.

2. Table 1 shows that our estimator becomes better as the number of periods (and correspondingly, the samples) T increases. This improvement, which is characteristic of nonparametric estimators, shows that our method is able to extract more information as more data is made available. The RPL model, by contrast, does not exhibit any such consistent pattern.

3. Although not shown in Table 1, we note that the errors metrics reported by Fox et al. for their method (Fox et al. 2011, Tables 1 & 2) are comparable (or slightly worse) to those obtained under our method. Their method, however, needs the support of the mixing distribution as input. For their experiments under the simulation setup above, they use a uniform discrete grid as the support of the mixing distribution. This approach, however, does not scale to high-dimensional settings with larger D values. Our estimator does not suffer from this limitation—we show that it scales to the feature dimensions in real-world case studies, with $D = 5$ (Section 7.1) and $D = 11$ (Section 7.2).

7. Predictive performance of the estimator

We perform two numerical studies on real-world data to showcase the predictive accuracy of our method. The first case study uses market share data, while the second study applies our estimation technique on sales transaction data from multiple stores with varying offer-sets and product prices.

7.1. Case Study 1: SUSHI Preference Dataset

In this study, we compare our CG method with the expectation-maximization (EM) benchmark on in-sample fit and predictive and decision accuracies. We use the popular SUSHI Preference dataset (Kamishima et al. 2005) for our study. This dataset has been used extensively in prior work on learning customer preferences. It consists of the preferences of 5,000 customers over 100 varieties of sushi. Each customer provides a rank ordering of the top-10 of her most preferred sushi varieties from among all the 100 varieties. Each sushi variety is described by a set of features like price, oiliness in taste, frequency with which the variety is sold in the shop, etc. Table EC.5 in Appendix B.2 describes

[15] Fox et al. consider support sizes of k^2 with $k = 1, 2, \ldots, 9$ in their experiments. We refer the reader to Appendix B.1 for the complete table of results for our estimator.

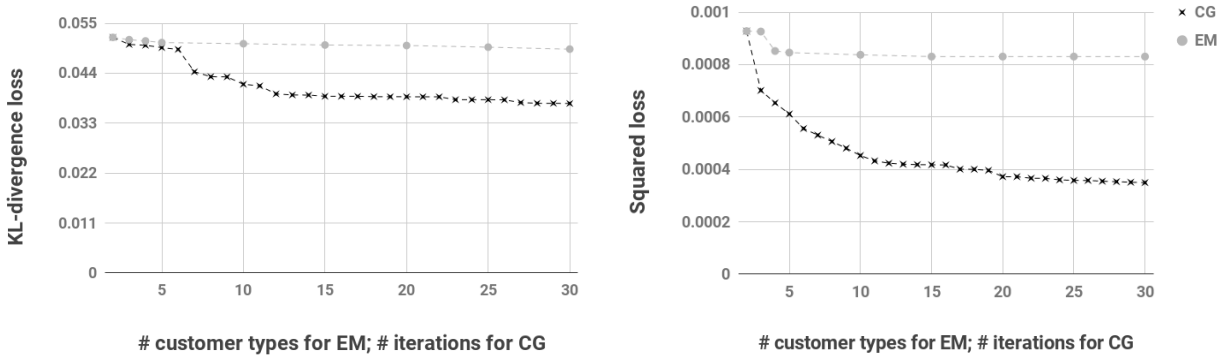
the subset of $D = 5$ features that we used in our experiments. One of the features, style, is binary valued and the rest are continuous-valued.

Setup. We processed the data to obtain aggregate market share information as follows. We assume that customers can choose from any of the 100 varieties of sushi and they choose their most preferred variety. Therefore, the market share y_j of sushi variety j is equal to the fraction of customers who ranked sushi variety j at the top. Only 93 sushi varieties had non-zero market shares, so we restrict our analysis to these varieties; therefore $n = 93$. We represent the data as the *empirical market shares* vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$. We then fit a mixture of logit models to this market share data using our CG estimator and the EM benchmark. For the EM method, we vary the number K of latent classes over the set $\{2, 3, 4, 5, 10, 15, 20, 25, 30\}$ to estimate K-class LC-MNL models. We initialized the CG estimator with the output of a 2-class LC-MNL model fit using the EM algorithm. To solve the optimization problem in the support finding step, we use a heuristic algorithm (see discussion in Section 5.4 and Appendix B.2) that is based on our theoretical development, and obtains an approximate solution by exploiting the fact that the optimal solution to the support finding step must be a boundary type because one of the features is binary-valued (see Theorem 5). We run the CG algorithm for $K_{\max} = 30$ iterations so that the maximum number of types found is at most 30.

7.1.1. In-sample fit and structure of recovered mixing distribution. We first discuss the in-sample performance achieved by both methods. For the NLL loss, we measure the performance in terms of the KL-divergence loss, defined as $D_{KL}^{\mathsf{algo}} \stackrel{\text{def}}{=} \mathsf{NLL}^{\mathsf{algo}} - H(\boldsymbol{y})$, where $H(\boldsymbol{y}) = -\sum_{j=1}^{n} y_j \log y_j$ is the entropy of the empirical market shares vector and represents the lowest achievable in-sample NLL loss (by any method), and $\mathsf{NLL}^{\mathsf{algo}}$ denotes the NLL loss achieved by $\mathsf{algo} \in \{\text{EM}, \text{CG}\}$. Figure 2 plots the in-sample KL-divergence loss and squared loss as a function of the number of customer types for the EM benchmark and the number of iterations for our CG estimator. Note that the number of iterations of the CG method is an upper bound on the number of customer types it recovers. Therefore, in the comparison, the CG method is allowed to use the same number of customer types as—or even fewer than—the EM benchmark.

We make the following observations. First, the CG method consistently achieves a better in-sample fit than the EM benchmark, even when using far fewer customer types. In particular, at the end of 30 iterations, CG achieves $D_{KL}^{\text{CG}} = 0.0372$ with $K = 29$ types as opposed to $D_{KL}^{\text{EM}} = 0.049$ with $K = 30$ types—a 24% reduction. For the squared loss, CG found $K = 23$ types with an in-sample loss of 3.49×10^{-4} as opposed to 8.31×10^{-4} achieved by EM with $K = 30$ types—a 58% reduction. The CG algorithm iteratively adds customer types that explain the observed choice data to the mixing distribution, which results in a much better fit as compared to the EM algorithm that updates all customer types together in each iteration. In particular, the EM algorithm is directly solving a

Figure 2 In-sample performance on the SUSHI dataset



Note. The horizontal axis represents both the number of customer types estimated by the EM benchmark, as well as the number of iterations in the CG algorithm; since the number of iterations is an upper bound on the number of types the CG estimator recovers, the plots represent a fair comparison between the methods.

(non-convex) optimization problem over $K-1+K\cdot D$ parameters, for a K-class LC-MNL model, which makes it challenging to locate the optimal solution. Our method, on the other hand, iteratively searches for the next customer type by solving the support finding step, which although a non-convex problem, can be solved optimally in certain scenarios and possesses a lot more structure. Second, the improvement in SQ loss is significantly higher than the improvement in NLL loss. The reason is that the M-step in the EM benchmark is non-convex when optimizing the SQ loss; consequently, it can only be solved approximately, resulting in slow convergence and worse performance for the EM benchmark. The CG algorithm, on the other hand, required very little customization[16] showing its plug-and-play nature when dealing with different loss functions.

We next analyze the structures of the customer types recovered by our method. For this analysis we focus on the NLL loss. At the end of 30 iterations, the CG method recovered 29 customer types. Except for the two customer types which were part of the initial solution, each of the remaining 27 types found by the CG method is a boundary type. These boundary types fall into two classes: those that consider only the maki (a.k.a rolled sushi) variety and those that consider only the non-maki variety. It follows from Theorem 5 that these are the only two possible boundary types because there is only one binary-valued feature, representing whether the sushi is maki or non-maki. Of the 93 varieties of sushi, 13 varieties are maki and the remaining 80 are non-maki. We find that 5 customer types—comprising 2.5% of the probability mass—only consider the 13 maki varieties, so if one of the maki varieties is stocked-out, they substitute to one of the remaining maki varieties. The remaining 22 customer types, comprising 46.1% of the probability mass, only consider the 80 non-maki varieties. The types recovered by our method exhibit strong preferences over the sushi varieties. In fact of the 10

[16] In fact, we only had to modify the objective and gradient computations.

customer types with the largest proportions, 6 types consider only a single sushi variety. By contrast, the EM algorithm recovers customer types who consider all the sushi varieties and is therefore unable to fully capture the underlying heterogeneity in the population with the same number of customer types. See Figure EC.1 in Appendix B.2 for a visual representation of the distinction. Our theoretical characterization of the choice behavior of boundary types in Section 5.2 further allows managers to determine changes in sushi characteristics (such as the price) to induce maki customer types to consider non-maki varieties and vice-versa. Finally, the presence of customer types that only consider a single sushi variety is consistent with prior work where customers were observed to (consider and) purchase only a single brand of cars (Lapersonne et al. 1995).

7.1.2. Predictive accuracy on new assortments. To test the predictive performance of the recovered mixture on previously unseen assortments, we consider the following tasks:

1. Predict market shares when one/two existing sushi varieties are *dropped* from the assortment.
2. Predict market shares when one new sushi variety is *added* to the assortment.
3. Predict market shares when one existing sushi variety is *replaced* by a new variety.

The above prediction tasks are motivated by real-world situations in which products may be discontinued because of low demand and/or be unavailable due to stockouts, or new products are introduced into the market. Being able to predict how the population reacts to such changes can be very useful for a firm. We measured predictive accuracies in terms of two popular metrics, mean absolute percentage error (MAPE) and root mean-square error (RMSE), which are defined as follows: for each $\mathsf{algo} \in \{\text{EM}, \text{CG}\}$ and given any test offer-set S_{test}, we compute

$$\text{MAPE}^{\mathsf{algo}} = 100 \times \left(\frac{1}{|S_{\text{test}}|} \sum_{i \in S_{\text{test}}} \frac{\left| \hat{y}_i - \hat{y}_i^{\mathsf{algo}} \right|}{\hat{y}_i} \right) \text{ and } \text{RMSE}^{\mathsf{algo}} = \sqrt{\frac{1}{|S_{\text{test}}|} \sum_{i \in S_{\text{test}}} \left(\hat{y}_i - \hat{y}_i^{\mathsf{algo}} \right)^2},$$

where $\hat{y}_i^{\mathsf{algo}}$ is the *predicted* market share for sushi variety $i \in S_{\text{test}}$ under the mixture of logit models[17] estimated using algo and $\hat{y}_i$ is the *true* market share computed from the test data. We report the *average error* across all possible test assortments. For the first scenario when one sushi variety is dropped, there are 93 test assortments resulting from dropping each sushi variety in turn. Similarly for the case when two sushi varieties are dropped, resulting in $\binom{93}{2}$ test assortments. When one new variety is added, the training data consists of the market shares when 92 sushi varieties are offered to the population—we consider all 93 training assortments—and in each case, the test data consists of only a single assortment, containing all 93 varieties. We report the average error on this test assortment across each of the training assortments. Similarly, when one existing variety is replaced by a new variety, the training data consists of market shares when 92 sushi varieties are offered

[17] We use the mixing distribution estimated by optimizing the NLL loss.

Table 2 Mixture estimation times and error in market share predictions on test assortments

	Estimation	Drop 1		Drop 2		Add 1		Replace 1	
Estimator	time (secs)	RMSE*	MAPE	RMSE*	MAPE	RMSE*	MAPE	RMSE*	MAPE
EM	914	4.6	83.67	4.6	83.17	4.8	90.23	4.8	89.71
CG	59	3.3	69.75	3.4	69.52	3.4	75.06	3.5	75.07
Improvement (%)	93.5	28.3	16.6	26.1	16.4	29.2	16.8	27.1	16.3

*: $\times 10^{-3}$. "Drop 1" and "Drop 2" refer respectively to the cases when the test assortment is formed by dropping one and two existing sushi varieties from the assortment. "Add 1" and "Replace 1" refer respectively to the cases when the test assortment is formed by adding a new sushi variety to the assortment, and replacing an existing sushi variety with a new variety. We obtain an average of 28% improvement in the RMSE and 16% in the MAPE metrics. The experiments were conducted on a computer with a 2.1GHz AMD Opteron(TM) 6272 processor, 32GB RAM and Ubuntu 14.04 OS—our approach is almost 16× faster than EM.

to the population, and for each training assortment, there are 92 test assortments—obtained by replacing each existing sushi variety in turn with a new variety. We first compute the average test error for each training assortment, and finally report the mean of these average test errors across the training assortments.[18]

Table 2 reports the errors for each prediction task. For the EM algorithm, we choose the best performing model amongst all estimated K-class LC-MNL models. It is evident that our mixture estimation method significantly outperforms the EM benchmark across both metrics and all prediction tasks. In particular, we notice an average of 28% reduction in RMSE and 16% reduction in MAPE. Finally, we also observe from Table 2 that the CG method is almost 16× faster than EM-based estimation, showing that it can scale better to datasets containing large number of choice observations.

7.1.3. Decision accuracy. We now focus on the decision accuracies of the methods. We consider the *assortment optimization* decision, which involves determining the subset of products to offer to the population to maximize expected revenue.

Setup. In order to compute the optimal assortment and ground-truth revenues, we pre-processed the data as follows: We assume that the 93 sushi varieties with non-zero market shares in the dataset comprise the entire sushi market. We focus on maximizing the revenue from the sale of the top-49 sushi varieties by market share. The remaining 44 varieties form the outside option. Treating the outside option as one "product," we obtain a total of $n = 50$ products. Without loss of generality, we suppose that the outside option is indexed by $j = 50$. For each sushi variety $j \in [n]$, we let y_j denote its market share; for the outside option, we obtain its market share by summing the market shares of all the 44 sushi varieties it comprises. We let r_j denote the price (present as the normalized price feature in the dataset) of product j. We set the price of the outside option to 0. We suppose that the

[18] The improvements were similar when considering the minimum and maximum of the average test errors.

Table 3 Optimal assortment sizes and ground-truth revenue generated

Estimator	# customer types recovered	Optimal assortment size	Revenue
EM	20	5	9.9×10^3
	25	5	9.9×10^3
	30	5	9.9×10^3
CG	20	17	11.9×10^3
	24	20	12.1×10^3
	28	22	12.2×10^3

Note that our estimation method is able to extract around 23% more revenue than that generated by the EM benchmark. Here, revenue is measured in units of the normalized price feature (see Table EC.5) of each sushi variety.

outside option is always offered. Then, our goal is to find the subset of the remaining products to maximize the expected revenue; that is, our goal is to solve

$$\max_{S \in [n-1]} \sum_{j \in S} r_j \cdot (\text{Probability that } j \text{ is chosen from } S \cup \{n\}) .$$

We fit mixtures of logit models by optimizing the NLL loss using the CG and EM methods and then solve the above optimization problem under both the models. To solve the optimization problem, we used the mixed-integer linear program (MILP) described in Méndez-Díaz et al. (2014). This MILP takes as input the proportions of each mixture component, product utilities under each mixture component and the product prices, and outputs the optimal assortment. We solved the MILPs using Gurobi Optimizer version 6.5.1. The MILPs ran to optimality, so the recovered assortments were optimal for the given models.

Results and Discussion. We fit a K-class LC-MNL model using the EM method and run the CG algorithm for K iterations to estimate a mixture of logit models, where $K \in \{20, 25, 30\}$. Table 3 reports the optimal assortment sizes and the ground-truth revenues extracted from the population. We compute the ground-truth revenue by assuming that each of the $5,000$ customers in the dataset purchases the most preferred of the offered products, as determined from her top-10 ranking; if none of the offered products appears in the customer's top-10 ranking, then we assume that the customer chose the outside option.

We note that the EM method offers only 5 sushi varieties as part of its optimal assortment. The reason is that the customer types recovered by the EM method are not sufficiently diverse (refer to the discussion in Section 7.1.1) because of which the MILP concludes that a small offering suffices to extract the most revenue from the population. In fact, the MILP ends up offering the 5 sushi varieties with the highest prices. By contrast, our method finds customer types with strong preferences who have sufficiently different tastes so that the MILP concludes that a larger variety (around 20), consisting of both high-priced and low-priced sushi varieties, is needed in the optimal offering. The consequence is that we are able to extract upto 23% more revenues from the population.

7.2. Case Study 2: IRI Academic Dataset

We now illustrate how our method applies to a typical operations setting in which both the offer-sets and product prices vary over time. Offer-sets vary because of stock-out events (in retail settings) and deliberate scarcity (in revenue management settings). Prices vary because of promotion activity or dynamic pricing policies. We use real-world sales transaction data from the IRI Academic Dataset (Bronnenberg et al. 2008) which contains purchase transactions of consumer packaged goods (CPG) for chains of grocery and drug stores. The dataset consists of weekly sales transactions aggregated over all customers. Each transaction contains information such as the week and store of purchase, the universal product code (UPC) of the purchased item, price of the item, etc. For our analysis, we consider transactions for five product categories in the first two weeks of the year 2011: shampoo, yogurt, toothbrush (toothbr), household cleaner (hhclean), and coffee. Table EC.6 in Appendix B.3 describes the summary statistics of the dataset.

Setup. We consider a setup similar to that of Jagabathula and Rusmevichientong (2016), who used the IRI dataset to test the predictive power of their pricing method. We pre-process the raw transactions (separately for each product category), as follows. We aggregate the purchased items by vendors[19] to deal with the sparsity of the data. Then, we further aggregate the vendors into $n = 10$ "products"—one product each for the top 9 vendors with the largest market shares and a single product for all remaining vendors. This aggregation ensures that there is sufficient coverage of products in the training and test offer-sets. Next, each combination of store and week corresponds to a discrete time period t. The offer-set S_t is chosen as the union of all products purchased during the particular store-week combination. Then, for each product and offer-set pair (j, S_t), the number of sales N_{jt} is computed using the observed sales for product j in the store-week combination corresponding to S_t. The price p_{jt} of product j in offer-set S_t is set as the sales-weighted average of the prices of the different UPCs that comprise the product. The number of offer-sets obtained for each product category after this pre-processing step are also listed in Table EC.6.

We assume that when offered subset S_t and prices $(p_{jt} : j \in S_t)$, each arriving customer samples the MNL parameter vector $(\boldsymbol{\mu}, \beta)$ according to some mixing distribution Q and chooses product $j \in S_t$ with probability:

$$f_{jt}(\boldsymbol{\mu}, \beta) = \frac{\exp(\mu_j - \beta \cdot p_{jt})}{\sum_{\ell \in S_t} \exp(\mu_\ell - \beta \cdot p_{\ell t})}.$$

Here, $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n) \in \mathbb{R}^n$ are the alternative specific coefficients, $\beta \in \mathbb{R}$ is the price coefficient. The taste vector $\boldsymbol{\omega} = (\boldsymbol{\mu}, \beta) \in \mathbb{R}^D$ with $D = n + 1 = 11$ in this context.

We fit a mixture of logit models to the processed transaction data using both the CG and EM methods. As for the sushi case study, we initialize the CG algorithm with the output of a 2-class

[19] Each purchased item in the dataset is identified by its collapsed universal product code (UPC)—a 13-digit-long code with digits 4 to 8 denoting the vendor.

Table 4 Percentage improvements in average train/test loss over EM benchmark

Product category	SQ loss		NLL loss	
	Train	Test	Train	Test
Shampoo	6.4	5.1	3.4	2.3
Toothbrush	5.3	4.3	2.4	1.3
Household Cleaner	5.3	4.1	1.9	1.2
Yogurt	7.0	5.1	8.3	7.1
Coffee	4.3	2.6	3.7	2.4
Average	5.7	4.2	3.9	2.9

LC-MNL model fit using the EM algorithm and solve the optimization problem in the support finding step using the heuristic method described in Appendix B.3 (also refer to the discussion in Section 5.4). We run the CG method for $K_{\max} = 10$ iterations, which results in a mixture with at most 10 customer types. We also fit a 10-class LC-MNL model using the EM algorithm.

Results and Discussion. Similar to Jagabathula and Rusmevichientong (2016), we conduct a 2-fold cross-validation. We randomly partition the offer-sets into two parts of roughly equal sizes, fit a mixture of logit model to one part (the training set), and then evaluate its predictions on the other part (the test set). We repeat this process with the train and test sets interchanged. We report performance on both the train and test datasets—all quantities referred to below are computed by taking an average across the two folds. For the NLL loss, we measure the performance using the metric $\Delta_{\mathsf{algo}}^{\mathsf{dataset}} = \mathsf{NLL}_{\mathsf{algo}}^{\mathsf{dataset}} - H^{\mathsf{dataset}}$ where H^{dataset} is the sales-weighted entropy of the observed sales, defined as $H^{\mathsf{dataset}} = -\frac{1}{N}\sum_{t=1}^{T}\sum_{j\in S_t} N_{jt}\log y_{jt}$, for $\mathsf{dataset} \in \{\text{train}, \text{test}\}$, $\mathsf{algo} \in \{\text{EM}, \text{CG}\}$, and $\mathsf{NLL}_{\text{EM}}^{\mathsf{dataset}}, \mathsf{NLL}_{\text{CG}}^{\mathsf{dataset}}$ denote the NLL loss achieved by the EM and CG methods, respectively. In Table 4, we report the percentage improvement $100 \times (\Delta_{\text{EM}}^{\mathsf{dataset}} - \Delta_{\text{CG}}^{\mathsf{dataset}})/\Delta_{\text{EM}}^{\mathsf{dataset}}$. Similarly, for the SQ loss, we report $100 \times (\mathsf{SQ}_{\text{EM}}^{\mathsf{dataset}} - \mathsf{SQ}_{\text{CG}}^{\mathsf{dataset}})/\mathsf{SQ}_{\text{EM}}^{\mathsf{dataset}}$, where $\mathsf{SQ}_{\mathsf{algo}}^{\mathsf{dataset}}$ denotes the SQ loss achieved by $\mathsf{algo} \in \{\text{EM}, \text{CG}\}$ on $\mathsf{dataset} \in \{\text{train}, \text{test}\}$.

Our estimator achieves better in-sample loss across all product categories for both loss functions—an average of 5.7% reduction for SQ loss and 3.9% for the NLL loss. The in-sample improvement is largest for the yogurt category—we obtain 7.0% reduction for SQ loss and 8.3% for the NLL loss. The superior in-sample fit translates to better test performance as well, with 5.1% reduction in SQ loss for the yogurt and shampoo categories, and 7.1% reduction in NLL loss for the yogurt category.

We also analyze the structure of the recovered mixing distribution, including the presence of boundary types—see Appendix B.3 for a detailed discussion.

8. Extension: accounting for endogeneity in product features

In many applications of discrete choice modeling, a product feature may be correlated with features not included in the model. The omitted features tend to be those that are unobserved. If such correlations

are ignored during estimation, then the coefficient estimated for the included feature could be biased. This phenomenon is referred to broadly as *endogeneity*. The classical example is that product prices are often correlated with unobservables, such as product quality, and ignoring such unobservables may lead one to conclude that higher prices lead to higher demands, when in fact, the higher demand was caused by higher quality. Petrin and Train (2010) offer other examples of endogeneity.

Several techniques have been proposed in existing literature to deal with the issue of endogeneity in discrete choice models. In this section, we show how one such technique can be incorporated into our method. We use the *control function* method proposed by Petrin and Train (2010), which generalizes the demand shocks approach proposed in Berry et al. (1995). We illustrate its use in our method using the following modification of the simulation setup from Section 6:

Utility model. We follow the setup of Section 6. We fix a choice of the ground-truth mixing distribution Q and number of time periods T. We then generate the choice data as follows. In each period $t \in [T]$, a customer arrives and is offered all the $n = 11$ products, including the no-purchase option. Instead of sampling a two-dimensional parameter vector as before, the customer now samples a three-dimensional parameter vector $(\omega_1^{(t)}, \omega_2^{(t)}, \omega_3^{(t)})$ according to Q and assigns the following utility to product j: $U_{jt} = \omega_1^{(t)} \cdot x_{jt} + \omega_2^{(t)} \cdot z_{jt} + \omega_3^{(t)} \cdot \mu_{jt} + \varepsilon_{jt}$,

where $(x_{jt}, z_{jt}, \mu_{jt})$ is the feature vector of product j in period t and $(\varepsilon_{jt} : j \in [n])$ are independent and identically distributed standard Gumbel random variables. The feature vector of the no-purchase option is set to $(0, 0, 0)$. Customers choose the product with the highest utility, resulting in the standard MNL choice probability. The key difference in the utility model from the setup above is that while x_{jt} and z_{jt} are observed, μ_{jt} is unobserved and is correlated with x_{jt}. As is standard in the literature, we assume that the endogenous feature is impacted by a set of instruments $\mathbf{w}$ and the exogenous feature: $x_{jt} = \gamma_1 \cdot \mathrm{w}_{jt,1} + \gamma_2 \cdot \mathrm{w}_{jt,2} + \gamma_3 \cdot z_{jt} + \mu_{jt}$.

Control function correction. To deal with endogeneity, the control function (CF) approach obtains a proxy for the term μ_{jt} by regressing the endogenous feature x_{jt} on the instruments $(\mathrm{w}_{jt,1}, \mathrm{w}_{jt,2})$ and the exogenous feature z_{jt} and then plugs in the residual $\hat{\mu}_{jt} = x_{jt} - \hat{\boldsymbol{\gamma}}^\top (\mathrm{w}_{jt,1}, \mathrm{w}_{jt,2}, z_{jt})$, where $\hat{\boldsymbol{\gamma}}$ represents the estimated regression parameters. In other words, the method estimates the coefficients using the following utility model: $\hat{U}_{jt} = \omega_1^{(t)} \cdot x_{jt} + \omega_2^{(t)} \cdot z_{jt} + \omega_3^{(t)} \cdot \hat{\mu}_{jt} + \varepsilon_{jt}$.

Once we plug in the residual, the estimators are run as before. They now estimate a mixing distribution over $D = 3$ parameters, where the additional random parameter is for the unobservable μ_{jt}.

Setup. For our experiments, we sample $(\omega_1^{(t)}, \omega_2^{(t)})$ according to the distribution $Q^{(2)}$, which is a mixture of two bivariate Gaussians, as defined in Section 6. We sample $\omega_3^{(t)}$ according to $\mathcal{N}(-1, 0.3^2)$, independently of $\omega_1^{(t)}$ and $\omega_2^{(t)}$. For each time period t and product j (except the no-purchase option), we sample the exogenous feature z_{jt} according to $\mathcal{N}(0, 1.5^2)$, the instruments $\mathrm{w}_{jt1}, \mathrm{w}_{jt2}$ according to $\mathcal{N}(0, 1)$, and the unobservable μ_{jt} according to $\mathcal{N}(0, 1)$, all independently of each other. We choose

Table 5 Recovery metrics with endogenous product features

	RMISE		MIAE	
Estimator	Without CF	With CF	Without CF	With CF
Normal	0.121	0.095	0.057	0.046
NP-CG	0.074	0.059	0.039	0.038

All differences are statistically significant at 1% level according to a paired samples t-test. "Without CF" refers to the case when endogeneity is ignored and "With CF" refers to the case when control function (CF) correction is applied.

$\boldsymbol{\gamma} = (0.54, 0.54, 0.54)$ to ensure that the marginal distribution of x_{jt} matches the marginal distribution of the features for the case without endogeneity in Section 6. Then, we generate choices for $T = 15,000$ periods.

Results. Table 5 compares our CG method to the standard RPL model with a diagonal variance-covariance matrix on the same RMISE and MIAE metrics, both when endogeneity is ignored and when endogeneity is corrected using the CF approach. We compute the error metrics only for the distribution of (ω_1, ω_2), and not ω_3. We make the following observations:

1. Ignoring endogeneity can worsen the recovery of the underlying mixing distribution, as is evident in the noticeably larger RMISE value for the benchmark RPL model.

2. Misspecification in the mixing distribution can impact recovery more adversely than ignoring endogeneity. Our method without the CF correction has lower error metrics than the benchmark *with* the CF correction. This shows that having the freedom of choosing the mixing distribution can help mitigate the effects of endogeneity bias.

3. Our estimator is compatible with the CF approach, allowing one to correct for endogeneity and obtain a better approximation to the underlying mixing distribution.

9. Conclusions

This paper proposes a novel nonparametric method for estimating the mixing distribution of a mixture of logit models given sales transactions and product availability data. Unlike traditional methods that impose a parametric assumption on the mixing distribution, our approach finds the best fitting distribution to the data, where the fit to the data is measured through a loss function such as the standard log-likelihood loss, from the class of all possible mixing distributions. We formulate the estimation problem as a constrained convex program by using the insight that instead of optimizing over the mixing distribution, the estimation problem can be solved by directly optimizing over the predicted choice probabilities for the observed choices in the data—subject to the constraint that they are consistent with some underlying mixing distribution. We then apply the conditional gradient algorithm to solve this convex program, which simultaneously performs both tasks of optimizing over the predicted choice probabilities and recovering the underlying mixing distribution consistent with

those probabilities. Our theoretical results establish sublinear convergence rate of our estimator and characterize the structure of the mixing distribution recovered by our method. Specifically, in addition to standard logit types, we show that our method naturally recovers customer types with consideration sets, and our theoretical analysis studies the consideration set structure of such types. Through a numerical study on synthetic data, we show that our estimator can obtain good approximations to various complex ground-truth mixing distributions, despite having no knowledge of their underlying structure. We also show that our approach outperforms the standard EM benchmark in terms of in-sample fit, predictive and decision accuracies, while being an order of magnitude faster, in two case studies on real data.

There are numerous avenues for future work. We specifically focused on the MNL model in this paper because of its widespread use, but our approach is general and directly applicable for other choice model families (with the caveat that the subproblems can be solved reasonably efficiently). Applying our mixture estimation technique in the context of choice models like the nested logit or Mallows model is an interesting direction for future work. Moreover, it can be shown that our framework can be used to learn distributions over preference orderings, resulting in a fully nonparametric approach. The subproblem in each iteration in that context corresponds to finding a single preference ordering (or ranking) which has connections with the learning to rank (Liu 2009) and rank aggregation (Dwork et al. 2001) literatures. In fact, Jagabathula and Rusmevichientong (2018) recently applied some of these ideas to estimate the best fitting distribution over preference orderings, but they did not take into account any product features. Extending their approach to also account for product features is a promising future direction. Finally, our current estimation method cannot account for fixed parameters (across customer types) in the utility specification. Incorporating fixed effects into the estimation framework will also be an important next step.

Acknowledgments

The authors would like to thank the Department Editor, Associate Editor, and the anonymous referees whose comments and feedback helped improve the manuscript greatly. They would also like to thank the conference participants at MSOM and INFORMS, the workshop participants at the IMA Data-Driven Supply Chain Management workshop, and the seminar participants at NYU Stern, UT Dallas Jindal, and Northwestern Kellogg for their constructive comments that have helped improve the paper.

References

Bach F (2013) Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning* 6(2-3):145–373.

Berry S, Levinsohn J, Pakes A (1995) Automobile prices in market equilibrium. *Econometrica* 63(4):841–890.

Bhat CR (1997) An endogenous segmentation mode choice model with an application to intercity travel. *Transportation science* 31(1):34–48.

Bohning D, Schlattmann P, Lindsay B (1992) Computer-assisted analysis of mixtures (c.a.man): statistical algorithms. *Biometrics* 48(1):283–303.

Bronnenberg BJ, Kruger MW, Mela CF (2008) Database paper-the iri marketing data set. *Marketing Science* 27(4):745–748.

Clarkson KL (2010) Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)* 6(4):63.

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

Feng L, Dicker LH (2018) Approximate nonparametric maximum likelihood for mixture models: A convex optimization approach to fitting arbitrary multivariate mixing distributions. *Computational Statistics & Data Analysis* 122:80–91.

Fox JT, il Kim K, Ryan SP, 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.

Garber D, Hazan E (2015) Faster rates for the frank-wolfe method over strongly-convex sets. *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 541–549.

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

Harchaoui Z, Juditsky A, Nemirovski A (2015) Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming* 152(1-2):75–112.

Hauser JR (2014) Consideration-set heuristics. *Journal of Business Research* 67(8):1688–1699.

Hunter DR (2004) MM algorithms for generalized bradley-terry models. *Annals of Statistics* 32(1):384–406.

Jagabathula S, Rusmevichientong P (2016) A nonparametric joint assortment and price choice model. *Management Science* 63(9):3128–3145.

Jagabathula S, Rusmevichientong P (2018) The limit of rationality in choice modeling: Formulation, computation, and implications. *Management Science* (Articles in Advance).

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. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 427–435.

Jaggi M, Sulovsk M (2010) A simple algorithm for nuclear norm regularized problems. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 471–478.

James J (2017) MM algorithm for general mixed multinomial logit models. *Journal of Applied Econometrics* 32(4):841–857.

Jiang W, Zhang CH (2009) General maximum likelihood empirical bayes estimation of normal means. *The Annals of Statistics* 37(4):1647–1684.

Joulin A, Tang K, Fei-Fei L (2014) Efficient image and video co-localization with frank-wolfe algorithm. *Computer Vision–ECCV 2014*, 253–268 (Springer).

Kamishima T, Kazawa H, Akaho S (2005) Supervised ordering - an empirical survey. *Fifth IEEE International Conference on Data Mining*, 673–676.

Kiefer J, Wolfowitz J (1956) Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics* 27(4):887–906.

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

Lacoste-Julien S, Jaggi M (2015) On the global linear convergence of frank-wolfe optimization variants. *Advances in Neural Information Processing Systems 28*, 496–504.

Laird N (1978) Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association* 73(364):805–811.

Lapersonne E, Laurent G, Le Goff JJ (1995) Consideration sets of size one: An empirical investigation of automobile purchases. *International Journal of Research in Marketing* 12(1):55–66.

Lindsay BG (1983) The geometry of mixture likelihoods: A general theory. *The Annals of Statistics* 11(1):86–94.

Lindsay BG (1995) Mixture models: theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics* 5:i–163.

Liu TY (2009) Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3(3):225–331.

McFadden D, Train K (2000) Mixed mnl models for discrete response. *Journal of Applied Econometrics* 15(5):447–470.

McLachlan G, Peel D (2000) *Finite mixture models* (John Wiley & Sons).

Méndez-Díaz I, Miranda-Bront JJ, Vulcano G, Zabala P (2014) A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics* 164(1):246–263.

Nocedal J, Wright SJ (2006) *Numerical Optimization* (Springer), second edition.

Petrin A, Train K (2010) A control function approach to endogeneity in consumer choice models. *Journal of marketing research* 47(1):3–13.

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

Robbins H (1950) A generalization of the method of maximum likelihood-estimating a mixing distribution. *Annals of Mathematical Statistics*, 21(2):314–315.

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.

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

Train KE (2009) *Discrete choice methods with simulation* (Cambridge university press), second edition.

Wang YX, Sadhanala V, Dai W, Neiswanger W, Sra S, Xing E (2016) Parallel and distributed block-coordinate frank-wolfe algorithms. *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, 1548–1557.

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

Zangwill WI (1969) *Nonlinear programming: a unified approach* (Prentice-Hall Englewood Cliffs, NJ).

Zhang T (2003) Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory* 49(3):682–691.

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

Online Appendix

A Conditional Gradient Approach for Nonparametric Estimation of Mixing Distributions

Appendix A: Proofs of theoretical results

A.1. Proof of Lemma 1: CONVEX MIXTURE is convex program

The objective function is by definition convex. Therefore, it is sufficient to show that the constraint set $\text{conv}(\overline{\mathcal{P}})$ is convex. For that, we note that since all entries of $\boldsymbol{f}(\boldsymbol{\omega})$ are between 0 and 1 for any $\boldsymbol{\omega} \in \mathbb{R}^D$, all limit points of the set $\left\{\boldsymbol{f}(\boldsymbol{\omega})\colon\ \boldsymbol{\omega} \in \mathbb{R}^D\right\}$ are also bounded. Therefore, $\overline{\mathcal{P}}$ is a bounded set in $\mathbb{R}^M$. As it is closed by definition, it follows from the Heine-Borel theorem that $\overline{\mathcal{P}}$ is compact. Further, since the convex hull of a compact subset of the Euclidean space is compact, it follows that $\text{conv}(\overline{\mathcal{P}})$ is compact, and by the definition of a convex hull it is convex. The claim then follows. □

A.2. Proof of Theorem 1: Sublinear convergence

We first prove the rate for the squared loss, and then consider the negative log-likelihood loss.

A.2.1. Squared loss. For SQ loss, the convergence rate follows directly from existing results. For instance, Jaggi (2013) showed, for the optimization problem $\min_{\mathbf{x}\in\mathcal{D}} h(\mathbf{x})$ where $h(\cdot)$ is a differentiable convex function and $\mathcal{D}$ is a compact convex set, that the iterates of the fully corrective Frank-Wolfe variant (which is the one we consider) satisfy:

$$h(\mathbf{x}^{(k)}) - h(\mathbf{x}^*) \leq \frac{2 \cdot C_h}{k+2} \tag{EC.1}$$

for all $k \geq 1$. Here C_h is the *curvature constant*—a measure of the "non-linearity"—of the function $h(\cdot)$ over the domain $\mathcal{D}$, defined as:

$$C_h := \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}\\ \gamma\in[0,1]\\ \mathbf{r}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})}} \frac{2}{\gamma^2} \cdot \Big(h(\mathbf{r}) - h(\mathbf{x}) - \langle \nabla h(\mathbf{x}), \mathbf{r}-\mathbf{x} \rangle \Big).$$

Since $h(\cdot)$ is convex, the curvature constant $C_h \geq 0$. In addition, if the function $h(\cdot)$ is twice differentiable, then it can be shown (Jaggi 2011, Equation 2.12) that

$$C_h \leq \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}\\ \mathbf{r}\in[\mathbf{x},\mathbf{s}]\subseteq\mathcal{D}}} (\mathbf{s}-\mathbf{x})^\top \nabla^2 h(\mathbf{r})(\mathbf{s}-\mathbf{x}),$$

where $[\mathbf{x},\mathbf{s}]$ is the line-segment joining $\mathbf{x}$ and $\mathbf{s}$—since $\mathcal{D}$ is convex, it lies within $\mathcal{D}$.

In our case, the convex objective $h = \mathsf{SQ}$ and the domain $\mathcal{D} = \text{conv}(\overline{\mathcal{P}})$. Further, the hessian $\nabla^2\mathsf{SQ}(\cdot)$ is a diagonal matrix with entry corresponding to product j in offer-set S_t as $\left(\nabla^2\mathsf{SQ}(\mathbf{r})\right)_{jt} = N_t/N$. Then, consider the following:

$$\begin{aligned} C_{\mathsf{SQ}} &\leq \sup_{\substack{\mathbf{x},\mathbf{s}\in\mathcal{D}\\ \mathbf{r}\in[\mathbf{x},\mathbf{s}]\subseteq\mathcal{D}}} (\mathbf{s}-\mathbf{x})^\top \nabla^2\mathsf{SQ}(\mathbf{r})(\mathbf{s}-\mathbf{x}) \\ &= \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D}} \frac{1}{N} \cdot \sum_{t=1}^{T} \sum_{j\in S_t} (s_{jt} - x_{jt})^2 \cdot N_t \end{aligned}$$

$$
\begin{aligned}
&= \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D}} \frac{1}{N}\cdot\sum_{t=1}^{T} N_t \cdot \sum_{j\in S_t}(s_{jt}-x_{jt})^2 \\
&\le \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D}} \frac{1}{N}\cdot\sum_{t=1}^{T} N_t \cdot \sum_{j\in S_t}|x_{jt}-s_{jt}| \quad \text{(since } |x_{jt}-s_{jt}|\le 1\text{)} \\
&\le \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D}} \frac{1}{N}\cdot\sum_{t=1}^{T} N_t \cdot \sum_{j\in S_t}(|x_{jt}|+|s_{jt}|) \quad \text{(using triangle inequality)} \\
&= \sup_{\mathbf{x},\mathbf{s}\in\mathcal{D}} \frac{1}{N}\cdot\sum_{t=1}^{T} N_t \cdot (1+1) \quad \text{(since choice probs. sum to 1 in each offer-set)} \\
&= \frac{1}{N}\cdot 2\cdot N = 2
\end{aligned}
$$

The result then follows by plugging in $C_{\mathsf{SQ}} \le 2$ in equation (EC.1) above.

A.2.2. Negative log-likelihood loss. We refer to the domain $\text{conv}(\overline{\mathcal{P}})$ as $\mathcal{D}$ below for succinct notation. We establish the convergence rate assuming that $y_{jt} > 0$ for all $j \in S_t$ and all offer-sets S_t, but the arguments below can be extended in a straight-forward manner to the case when some of the y_{jt}'s are zero by dropping terms for those (product, offer-set) pairs when defining the NLL loss objective (since they do not contribute anyway to the loss objective), and only maintaining the choice probabilities for the the remaining (product, offer-set) pairs.

The proof proceeds in the following steps:

1. We show that there exists $\eta > 0$ such that the iterates $\mathbf{g}^{(k)}$ in Algorithm 1 satisfy $\mathrm{g}_{jt}^{(k)} \ge \eta$ for all $j \in S_t$ and all $1 \le t \le T$, and all $k \ge 0$. The idea is that if any of the iterates get too close to 0, then the objective value $\mathsf{NLL}(\mathbf{g}^{(k)})$ will exceed the starting objective value $\mathsf{NLL}(\mathbf{g}^{(0)})$ which is a contradiction since the fully corrective variant of the Frank-Wolfe algorithm produces a decreasing objective value in each iteration.

2. Utilizing the lower bound computed in Step 1, we adapt existing arguments from Guélat and Marcotte (1986) to show that Algorithm 1 achieves $O(1/k)$ convergence to the optimal solution.

We first prove Step 2 assuming the existence of a lower bound η, and then compute a tight value for η.

Step 2: convergence rate. In particular, we establish the following lemma:

LEMMA EC.1. *Suppose there exists $\eta > 0$ such that the iterates $\mathbf{g}^{(k)}$ in Algorithm 1 satisfy $\mathrm{g}_{jt}^{(k)} \ge \eta$ for all $j \in S_t$ and all $1 \le t \le T$, and all $k \ge 0$. Then, there exists index $\bar{K}$ and constant κ such that*

$$\mathsf{NLL}(\mathbf{g}^{(k)}) - \mathsf{NLL}(\mathbf{g}^*) \le \frac{4}{\eta^2\cdot(k+\kappa)} \quad \forall\, k \ge \bar{K}.$$

Proof. Define $\widetilde{\mathcal{D}} = \left\{\mathbf{g}\in\mathcal{D} \mid \mathrm{g}_{jt} \ge \frac{\eta}{\sqrt{2}} \ \forall\, j\in S_t \ \forall 1\le t\le T\right\}$. At any iteration $k \ge 1$, let $\mathbf{d}^{(k)} := \boldsymbol{f}^{(k)} - \mathbf{g}^{(k-1)}$ where recall that $\boldsymbol{f}^{(k)} \in \arg\min_{\mathbf{v}\in\overline{\mathcal{P}}} \left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{v} - \mathbf{g}^{(k-1)}\right\rangle$. Now, observe that for any $k \ge 1$:

$$
\begin{aligned}
\mathsf{NLL}(\mathbf{g}^*) - \mathsf{NLL}(\mathbf{g}^{(k-1)}) &\ge \left\langle \nabla\mathsf{NLL}(\mathbf{g}^{(k-1)}), \mathbf{g}^* - \mathbf{g}^{(k-1)}\right\rangle \quad \text{(since } \mathsf{NLL}(\cdot) \text{ is convex)} \\
&\ge \left\langle \nabla\mathsf{NLL}(\mathbf{g}^{(k-1)}), \boldsymbol{f}^{(k)} - \mathbf{g}^{(k-1)}\right\rangle \quad \text{(using definition of } \boldsymbol{f}^{(k)}\text{)} \qquad \text{(EC.2)} \\
&= \left\langle \nabla\mathsf{NLL}(\mathbf{g}^{(k-1)}), \mathbf{d}^{(k)}\right\rangle
\end{aligned}
$$

Next, consider a step size $\gamma \in [0,1]$ such that $\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)} \in \widetilde{\mathcal{D}}$. Using second-order Taylor series approximation of $\mathsf{NLL}(\cdot)$ around the point $\mathbf{g}^{(k-1)}$, we have:

$$\mathsf{NLL}(\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)}) = \mathsf{NLL}(\mathbf{g}^{(k-1)}) + \gamma \left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \mathbf{d}^{(k)} \right\rangle + \frac{\gamma^2}{2} {\mathbf{d}^{(k)}}^\top \nabla^2 \mathsf{NLL}(\mathbf{r}_k) \mathbf{d}^{(k)},$$

where $\mathbf{r}_k$ lies on the line segment $[\mathbf{g}^{(k-1)}, \mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)}]$. Since $\mathbf{g}^{(k-1)} \in \widetilde{\mathcal{D}}$ and $\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)} \in \widetilde{\mathcal{D}}$, it implies $\mathbf{r}_k \in \widetilde{\mathcal{D}}$ (since $\widetilde{\mathcal{D}}$ is convex), and consequently $r_{k,jt} \geq \frac{\eta}{\sqrt{2}}$ for all $j \in S_t$ and all $1 \leq t \leq T$. Then, consider the following:

$$\begin{aligned}
\mathsf{NLL}(\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)}) &= \mathsf{NLL}(\mathbf{g}^{(k-1)}) + \gamma \left\langle \nabla \mathsf{NLL}(\mathbf{g}^{(k-1)}), \mathbf{d}^{(k)} \right\rangle + \frac{\gamma^2}{2} {\mathbf{d}^{(k)}}^\top \nabla^2 \mathsf{NLL}(\mathbf{r}_k) \mathbf{d}^{(k)} \\
&= \mathsf{NLL}(\mathbf{g}^{(k-1)}) + \gamma \left\langle \nabla \mathsf{NLL}(\mathbf{g}^{(k-1)}), \mathbf{d}^{(k)} \right\rangle + \frac{\gamma^2}{2 \cdot N} \sum_{t=1}^{T} \sum_{j \in S_t} \frac{N_{jt} \cdot {\mathrm{d}_{jt}^{(k)}}^2}{r_{k,jt}^2} \\
&\leq \mathsf{NLL}(\mathbf{g}^{(k-1)}) + \gamma \left\langle \nabla \mathsf{NLL}(\mathbf{g}^{(k-1)}), \mathbf{d}^{(k)} \right\rangle + \frac{\gamma^2}{\eta^2 \cdot N} \sum_{t=1}^{T} \sum_{j \in S_t} N_{jt} \\
&\quad \left(\text{since } \left|\mathrm{d}_{jt}^{(k)}\right| \leq 1 \text{ and } r_{k,jt} \geq \frac{\eta}{\sqrt{2}} \;\; \forall j \in S_t \;\; \forall 1 \leq t \leq T\right) \\
&\leq \mathsf{NLL}(\mathbf{g}^{(k-1)}) - \gamma \cdot \left(\mathsf{NLL}(\mathbf{g}^{(k-1)}) - \mathsf{NLL}(\mathbf{g}^*)\right) + \frac{\gamma^2}{\eta^2} \;\; \{\text{using equation (EC.2)}\}
\end{aligned}$$

Denoting $\mathsf{gap}(\mathbf{g}) = \mathsf{NLL}(\mathbf{g}) - \mathsf{NLL}(\mathbf{g}^*)$ as the optimality gap, we get

$$\mathsf{NLL}(\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)}) \leq \mathsf{NLL}(\mathbf{g}^{(k-1)}) - \gamma \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) + \frac{\gamma^2}{\eta^2} \tag{EC.3}$$

Now, choose $\gamma^* = \frac{\eta^2 \cdot \mathsf{gap}(\mathbf{g}^{(k-1)})}{2}$ and observe that γ^* minimizes the RHS of equation (EC.3).

Claim 1: $\mathbf{g}^{(k-1)} + \gamma^* \cdot \mathbf{d}^{(k)} \in \widetilde{\mathcal{D}}$.

This means that for any $1 \leq t \leq T$ and any $j \in S_t$, we need to show:

$$\begin{aligned}
&\mathrm{g}_{jt}^{(k-1)} + \frac{\mathrm{d}_{jt}^{(k)} \cdot \eta^2}{2} \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) \geq \frac{\eta}{\sqrt{2}} \\
&\iff -\mathrm{d}_{jt}^{(k)} \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) \leq 2 \cdot \left(\frac{\mathrm{g}_{jt}^{(k-1)} - \frac{\eta}{\sqrt{2}}}{\eta^2} \right) \\
&\iff (\mathrm{g}_{jt}^{(k-1)} - f_{jt}^{(k)}) \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) \leq 2 \cdot \left(\frac{\mathrm{g}_{jt}^{(k-1)} - \frac{\eta}{\sqrt{2}}}{\eta^2} \right) \quad (\text{using defn. of } \mathbf{d}^{(k)}) \\
&\impliedby \mathrm{g}_{jt}^{(k-1)} \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) \leq 2 \cdot \left(\frac{\mathrm{g}_{jt}^{(k-1)} - \frac{\eta}{\sqrt{2}}}{\eta^2} \right) \quad (\text{since } \mathsf{gap}(\mathbf{g}^{(k-1)}) \geq 0) \\
&\iff \frac{1}{N} \sum_{t=1}^{T} \sum_{\ell \in S_t} N_{\ell t} \log \frac{\mathrm{g}_{\ell t}^*}{\mathrm{g}_{\ell t}^{(k-1)}} \leq \frac{2}{\eta^2} - \frac{\sqrt{2}}{\eta \cdot \mathrm{g}_{jt}^{(k-1)}} \quad (\text{using defn. of } \mathsf{gap}(\cdot)) \\
&\impliedby \frac{1}{N} \sum_{t=1}^{T} \sum_{\ell \in S_t} N_{\ell t} \cdot \left(\frac{\mathrm{g}_{\ell t}^*}{\mathrm{g}_{\ell t}^{(k-1)}} - 1 \right) \leq \frac{2}{\eta^2} - \frac{\sqrt{2}}{\eta \cdot \mathrm{g}_{jt}^{(k-1)}} \;\; (\text{since } \log z \leq z - 1 \;\; \forall \, z > 0) \\
&\iff \frac{1}{N} \sum_{t=1}^{T} \sum_{\ell \in S_t} N_{\ell t} \cdot \left(\frac{\mathrm{g}_{\ell t}^*}{\mathrm{g}_{\ell t}^{(k-1)}} \right) \leq 1 + \frac{2}{\eta^2} - \frac{\sqrt{2}}{\eta \cdot \mathrm{g}_{jt}^{(k-1)}} \\
&\impliedby \frac{1}{N} \sum_{t=1}^{T} \sum_{\ell \in S_t} N_{\ell t} \cdot \left(\frac{1}{\eta} \right) \leq 1 + \frac{2}{\eta^2} - \frac{\sqrt{2}}{\eta \cdot \mathrm{g}_{jt}^{(k-1)}} \quad (\text{since } 0 \leq \mathrm{g}_{\ell t}^* \leq 1 \text{ and } \mathrm{g}_{\ell t}^{(k-1)} \geq \eta \;\; \forall \ell \in S_t \, \forall \, 1 \leq t \leq T)
\end{aligned}$$

$$\begin{aligned}
&\impliedby \frac{1}{\eta} \le 1 + \frac{2}{\eta^2} - \frac{\sqrt{2}}{\eta \cdot \eta} \quad (\text{since } \mathrm{g}_{jt}^{(k-1)} \ge \eta) \\
&\iff 0 \le \eta^2 - \eta + 2 - \sqrt{2}
\end{aligned}$$

The final statement is true for any $\eta > 0$ and therefore the claim follows. In addition, it is easy to see that $\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)} \in \widetilde{\mathcal{D}}$ for all $\gamma \in [0, m_k]$ where $m_k \stackrel{\text{def}}{=} \min\left(1, \frac{\eta^2 \cdot \mathsf{gap}(\mathbf{g}^{(k-1)})}{2}\right)$.

Claim 2:

$$\mathsf{gap}(\mathbf{g}^{(k)}) \le \mathsf{gap}(\mathbf{g}^{(k-1)}) \cdot \left(1 - \frac{m_k}{2}\right) \quad \forall\, k \ge 1. \tag{EC.4}$$

Consider the following:

$$\begin{aligned}
&\underbrace{\mathsf{NLL}(\mathbf{g}^{(k)}) \le \min_{\gamma \in [0,1]} \mathsf{NLL}(\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)})}_{\text{since FCFW guarantees as much progress as line-search FW}} \quad \le \min_{\gamma \in [0, m_k]} \mathsf{NLL}(\mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)}) \\
&\le \min_{\gamma \in [0, m_k]} \mathsf{NLL}(\mathbf{g}^{(k-1)}) - \gamma \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) + \frac{\gamma^2}{\eta^2} \\
&\left(\text{since } \mathbf{g}^{(k-1)} + \gamma \cdot \mathbf{d}^{(k)} \in \widetilde{\mathcal{D}}\ \forall\, \gamma \in [0, m_k] \text{ and using equation (EC.3)}\right) \\
&= \mathsf{NLL}(\mathbf{g}^{(k-1)}) - m_k \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) + \frac{m_k^2}{\eta^2} \\
&\text{(by choice of } m_k) \\
&\le \mathsf{NLL}(\mathbf{g}^{(k-1)}) - m_k \cdot \mathsf{gap}(\mathbf{g}^{(k-1)}) + \frac{m_k \cdot \eta^2 \cdot \mathsf{gap}(\mathbf{g}^{(k-1)})}{2 \cdot \eta^2} \\
&\left(\text{since } m_k \le \frac{\eta^2 \cdot \mathsf{gap}(\mathbf{g}^{(k-1)})}{2}\right) \\
&= \mathsf{NLL}(\mathbf{g}^{(k-1)}) - \frac{m_k}{2} \cdot \mathsf{gap}(\mathbf{g}^{(k-1)})
\end{aligned}$$

Then subtracting $\mathsf{NLL}(\mathbf{g}^*)$ from both sides, the claim follows.

Claim 3: $\lim_{k \to \infty} m_k = 0$.

Since both sequences $\left\{\mathsf{gap}(\mathbf{g}^{(k)})\right\}_k$ and $\{m_k\}_k$ are non-increasing and bounded below by zero, it follows that (taking limits on both sides of equation (EC.4)):

$$\begin{aligned}
&\lim_{k \to \infty} \mathsf{gap}(\mathbf{g}^{(k)}) \le \lim_{k \to \infty} \mathsf{gap}(\mathbf{g}^{(k-1)}) \cdot \left(1 - \frac{1}{2} \lim_{k \to \infty} m_k\right) \\
&\implies \lim_{k \to \infty} \mathsf{gap}(\mathbf{g}^{(k)}) \cdot \lim_{k \to \infty} m_k \le 0 \\
&\implies \lim_{k \to \infty} \mathsf{gap}(\mathbf{g}^{(k)}) = 0 \ \text{ or } \ \lim_{k \to \infty} m_k = 0 \quad \left(\text{since } m_k \ge 0 \text{ and } \mathsf{gap}(\mathbf{g}^{(k)}) \ge 0\ \forall k\right) \\
&\implies \lim_{k \to \infty} m_k = 0 \quad (\text{using defn of } m_k)
\end{aligned}$$

Given the above claims, let $\bar{K}$ be the smallest iteration number such that $m_{\bar{K}} \le 1$. Then from equation (EC.4) it follows that

$$\begin{aligned}
\mathsf{gap}(\mathbf{g}^{(k)}) \le \mathsf{gap}(\mathbf{g}^{(k-1)}) \cdot \left(1 - \frac{m_k}{2}\right)\ \forall\, k \ge \bar{K} &\iff \frac{m_{k+1}}{2} \le \frac{m_k}{2} \cdot \left(1 - \frac{m_k}{2}\right)\ \forall\, k \ge \bar{K} \\
&\iff \frac{2}{m_{k+1}} \ge \frac{2}{m_k} \cdot \frac{2}{2 - m_k}\ \forall\, k \ge \bar{K} \\
&\iff \frac{2}{m_{k+1}} \ge \frac{2}{m_k} \cdot \left(1 + \frac{m_k}{2 - m_k}\right)\ \forall\, k \ge \bar{K} \\
&\implies \frac{2}{m_{k+1}} \ge \frac{2}{m_k} + 1\ \forall\, k \ge \bar{K} \quad \left(\text{since } m_k \ge 0\ \forall\ k \ge \bar{K}\right)
\end{aligned}$$

$$\begin{aligned}
&\implies \frac{2}{m_k} \geq k - \bar{K} + \frac{2}{m_{\bar{K}}} \;\; \forall\, k \geq \bar{K} \\
&\iff m_k \leq \frac{2}{k+\tilde{\kappa}} \;\; \forall\, k \geq \bar{K} \;\; \left(\text{where } \tilde{\kappa} = \frac{2}{m_{\bar{K}}} - \bar{K}\right) \\
&\iff \mathsf{gap}(\mathbf{g}^{(k-1)}) \leq \frac{4}{\eta^2 \cdot (k+\tilde{\kappa})} \;\; \forall\, k \geq \bar{K} \\
&\iff \mathsf{gap}(\mathbf{g}^{(k)}) \leq \frac{4}{\eta^2 \cdot (k+\kappa)} \;\; \forall\, k \geq \bar{K} \,(\text{where } \kappa = 1+\tilde{\kappa})
\end{aligned}$$

□

Step 1: lower bound for iterates. Lemma EC.1 establishes $1/k$ convergence rate of Algorithm 1 given *any* lower bound $\eta > 0$ for the iterates $\mathbf{g}^{(k)}$. We now come up with a tight lower bound— $\xi_{\min}$ in the main text, based on the initialization, i.e. we show that

$$\mathrm{g}_{jt}^{(k)} \geq \xi_{\min} \;\; \forall\, j \in S_t \;\; \forall\, 1 \leq t \leq T \text{ and } \forall\, k \geq 0$$

For any vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, denote $x_{\min} \stackrel{\text{def}}{=} \min_{i=1,2,\ldots,m} x_i$. Then, for any $\xi \in (0,1]$, consider the following optimization problem:

$$G(\xi) \equiv \min_{\mathbf{x} \in \mathbb{R}^M} \mathsf{NLL}(\mathbf{x}) \;\; \text{s.t. } x_{jt} \geq 0 \;\forall\, j \in S_t;\; \sum_{j \in S_t} x_{jt} = 1; \;\; \forall 1 \leq t \leq T \text{ and } x_{\min} \leq \xi \tag{EC.5}$$

We first come up with a closed-form expression for $G(\xi)$. For each $1 \leq t \leq T$ and each $i \in S_t$, define the following optimization problem:

$$G_{i,t}(\xi) \equiv \min_{\mathbf{x} \in \mathbb{R}^M} \mathsf{NLL}(\mathbf{x}) \;\; \text{s.t. } x_{jt'} \geq 0 \;\forall\, j \in S_{t'};\; \sum_{j \in S_{t'}} x_{jt'} = 1; \;\; \forall 1 \leq t' \leq T \text{ and } x_{it} \leq \xi \tag{EC.6}$$

Claim 1:

$$G(\xi) = \min_{1 \leq t \leq T} \min_{i \in S_t} G_{i,t}(\xi) \text{ for all } \xi \in (0,1]$$

It is easy to see that $\min_{1 \leq i \leq t} \min_{i \in S_t} G_{i,t}(\xi) \leq G(\xi)$ for any $\xi \in (0,1]$ since the optimal solution for problem (EC.5) is feasible for some (product, offer-set) pair (i,t).

For the other direction, suppose $\min_{1 \leq t \leq T} \min_{i \in S_t} G_{i,t}(\xi) = G_{j^*,t^*}(\xi)$ for some $j^* \in S_{t^*}$. Let $\mathbf{x}^*$ denote the optimal solution for $G_{j^*,t^*}(\xi)$, so that $x^*_{j^*,t^*} \leq \xi$. This also means that $x^*_{\min} \leq \xi$ and consequently $\mathbf{x}^*$ is a feasible solution for problem (EC.5). Therefore, $G(\xi) \leq G_{j^*,t^*}(\xi) = \min_{1 \leq t \leq T} \min_{i \in S_t} G_{i,t}(\xi)$ and the claim then follows.

Claim 2:

$$G_{i,t}(\xi) = \frac{1}{N} \cdot \left(N_t \cdot D_{KL}(y_{it} \| \xi) + \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right) \;\; \forall\, \xi \in (0, y_{it}].$$

This follows because the optimal solution, say $\mathbf{x}^*$, to problem (EC.6) is of the form: (1) $x^*_{jt'} = y_{jt'}$ for all $j \in S_{t'}$, $\forall t' \neq t$ (2) $x^*_{jt} = \frac{y_{jt}}{1-y_{it}} \cdot (1-\xi)$ for all $j \in S_t \setminus \{i\}$ and (3) $x^*_{it} = \xi$.

This can be verified by solving the KKT conditions for problem (EC.6). In particular, letting $\lambda_{t'}$ denote the dual variable for the constraint $\sum_{j \in S_{t'}} x_{jt'} = 1$, and μ denote the dual variable for the constraint $x_{it} \leq \xi$, the KKT conditions are given by:

$$\frac{N_{jt'}}{x_{jt'} \cdot N} = \lambda_{t'} \;\; \forall j \in S_{t'} \text{ and } \forall t' \neq t; \;\; \frac{N_{jt}}{x_{jt} \cdot N} = \lambda_t \;\; \forall j \in S_t \setminus \{i\}; \;\; \frac{N_{it}}{x_{it} \cdot N} = \lambda_t + \mu \;\; \text{(Stationarity)}$$

$$\mu \cdot (x_{it} - \xi) = 0 \quad \text{(Complementary slackness)}$$

$$\mu \geq 0 \quad \text{(Dual feasibility)}$$

$$\sum_{j \in S_{t'}} x_{jt'} = 1;\ x_{jt'} \geq 0\ \forall j \in S_{t'}\ \forall\ 1 \leq t \leq T \quad \text{(Primal feasibility)}$$

Solving these equations gives the optimal solution mentioned above. Plugging the optimal solution $\mathbf{x}^*$ into the $\mathsf{NLL}(\cdot)$ loss objective, we obtain

$$\begin{aligned}
G_{i,t}(\xi) &= \frac{1}{N} \cdot \left(-N_{it} \log \xi - \sum_{j \in S_t \setminus \{i\}} N_{jt} \log \frac{y_{jt} \cdot (1-\xi)}{1 - y_{it}} - \sum_{t' \neq t} \sum_{j \in S_{t'}} N_{jt'} \log (y_{jt'}) \right) \\
&= \frac{1}{N} \cdot \left(-N_{it} \log \xi - \sum_{j \in S_t \setminus \{i\}} N_{jt} \log \frac{(1-\xi)}{1 - y_{it}} - \sum_{j \in S_t \setminus \{i\}} N_{jt} \log y_{jt} + \sum_{t' \neq t} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right) \\
&= \frac{1}{N} \cdot \left(-N_{it} \log \xi - (N_t - N_{it}) \log \frac{1-\xi}{1 - y_{it}} + N_{it} \log y_{it} + N_t \cdot H(\boldsymbol{y}_t) + \sum_{t' \neq t} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right) \\
&= \frac{N_t}{N} \cdot \left(y_{it} \log \frac{y_{it}}{\xi} + (1 - y_{it}) \log \frac{1 - y_{it}}{1 - \xi} \right) + \frac{1}{N} \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \\
&= \frac{1}{N} \cdot \left(N_t \cdot D_{KL}(y_{it} \| \xi) + \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right)
\end{aligned}$$

where $D_{KL}(y_{it} \| \xi)$ is the relative entropy between y_{it} and ξ, and $H(\boldsymbol{y}_{t'})$ is the entropy of vector $\boldsymbol{y}_{t'}$.

Claim 3: For each $1 \leq t \leq T$ and any $\xi \in (0, y_{t,\min}]$, it follows that

$$\min_{i \in S_t} G_{i,t}(\xi) = \frac{1}{N} \cdot \left(N_t \cdot D_{KL}(y_{t,\min} \| \xi) + \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right).$$

This follows since $\frac{\partial D_{KL}(y \| \xi)}{\partial y} > 0$ for any $y > \xi$ and therefore $D_{KL}(y_{it} \| \xi) \geq D_{KL}(y_{t,\min} \| \xi)$ for all $i \in S_t$ and any $\xi \in (0, y_{t,\min}]$.

Now using Claims 1, 2 and 3, it follows that

$$G(\xi) = \min_{1 \leq t \leq T} \frac{1}{N} \cdot \left(N_t \cdot D_{KL}(y_{t,\min} \| \xi) + \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \right) \quad \text{for any } 0 < \xi \leq y_{\min}.$$

where recall that $y_{\min} = \min_{1 \leq t \leq T} y_{t,\min}$ Given this, it can be verified that:

1. $G(\xi)$ is non-increasing as ξ increases—since we are optimizing over a larger domain.
2. $G(y_{\min}) = \frac{1}{N} \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}) \leq F_0 \stackrel{\text{def}}{=} \mathsf{NLL}(\mathbf{g}^{(0)})$ for any initialization $\mathbf{g}^{(0)}$.
3. $G(\xi) \to +\infty$ as $\xi \to 0$.

The above three facts imply that there exists $0 < \xi_{\min} \leq y_{\min}$ such that

$$G(\xi_{\min}) \leq F_0 \ \text{ and } \ G(\xi) > F_0 \text{ for all } 0 < \xi < \xi_{\min} \tag{EC.7}$$

This establishes the definition of $\xi_{\min}$ provided in the main text.

Given the above, we claim that for each iterate of the CG algorithm $\mathbf{g}^{(k)}$, $\mathrm{g}^{(k)}_{\min} \geq \xi_{\min}$. Suppose this is not the case, i.e. $\mathrm{g}^{(k)}_{\min} < \xi_{\min}$ for some iterate k. Then, consider the following:

$$\begin{aligned}
\mathrm{g}^{(k)}_{\min} < \xi_{\min} &\implies G(\mathrm{g}^{(k)}_{\min}) > F_0 \quad \{\text{from equation (EC.7) above}\} \\
&\implies \mathsf{NLL}(\mathbf{g}^{(k)}) > F_0 = \mathsf{NLL}(\mathbf{g}^{(0)})
\end{aligned}$$

where the last implication follows since $\mathbf{g}^{(k)}$ is feasible for the opt. problem (EC.5) with $\xi = \mathrm{g}_{\min}^{(k)}$ and consequently, $\mathsf{NLL}(\mathbf{g}^{(k)}) \geq G(\mathrm{g}_{\min}^{(k)})$. However, this results in a contradiction since the FCFW variant improves the objective value in each iteration.

Finally, the convergence result follows from choosing $\eta = \xi_{\min}$ in Lemma EC.1.

A.2.3. Proof of Proposition 1. From equation (EC.7), it follows that

$$G(\xi_{\min}) \leq F_0 \implies \min_{1\leq t\leq T} N_t \cdot D_{KL}(y_{t,\min}\|\xi_{\min}) \leq N \cdot F_0 - \sum_{t'=1}^{T} N_{t'} \cdot H(\boldsymbol{y}_{t'}).$$

Now, suppose that $\min_{1\leq t\leq T} N_t \cdot D_{KL}(y_{t,\min}\|\xi_{\min}) = N_{t^*} \cdot D_{KL}(y_{t^*,\min}\|\xi_{\min})$ for some $1 \leq t^* \leq T$. Then consider the following:

$$\begin{aligned}
& N_{t^*} \cdot D_{KL}(y_{t^*,\min}\|\xi_{\min}) \leq N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t) \\
\iff & D_{KL}(y_{t^*,\min}\|\xi_{\min}) \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{t^*}} \\
\iff & y_{t^*,\min} \log \frac{y_{t^*,\min}}{\xi_{\min}} + (1 - y_{t^*,\min}) \log \frac{1 - y_{t^*,\min}}{1 - \xi_{\min}} \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{t^*}} \\
\iff & y_{t^*,\min} \log \frac{y_{t^*,\min}}{\xi_{\min}} \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{t^*}} + (1 - y_{t^*,\min}) \log \frac{1 - \xi_{\min}}{1 - y_{t^*,\min}} \\
\implies & y_{t^*,\min} \log \frac{y_{t^*,\min}}{\xi_{\min}} \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{t^*}} + (1 - y_{t^*,\min}) \cdot \left(\frac{1 - \xi_{\min}}{1 - y_{t^*,\min}} - 1 \right) \\
& \quad (\text{since } \log z \leq z - 1 \ \forall\, z > 0) \\
\implies & y_{t^*,\min} \log \frac{y_{t^*,\min}}{\xi_{\min}} \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{t^*}} + y_{t^*,\min} \\
\iff & \log \frac{y_{t^*,\min}}{\xi_{\min}} \leq \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{y_{t^*,\min} \cdot N_{t^*}} + 1 \\
\iff & \frac{y_{t^*,\min}}{\xi_{\min}} \leq \exp\left(\frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{y_{t^*,\min} \cdot N_{t^*}} + 1 \right) \\
\iff & \xi_{\min} \geq y_{t^*,\min} \cdot \exp\left(-1 - \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{y_{t^*,\min} \cdot N_{t^*}} \right) \\
\implies & \xi_{\min} \geq y_{\min} \cdot \exp\left(-1 - \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{y_{t^*,\min} \cdot N_{t^*}} \right) \quad (\text{since } y_{t^*,\min} \geq y_{\min}) \\
\implies & \xi_{\min} \geq y_{\min} \cdot \exp\left(-1 - \frac{N \cdot F_0 - \sum_{t=1}^{T} N_t \cdot H(\boldsymbol{y}_t)}{N_{\min}} \right) \quad (\text{since } y_{t^*,\min} \cdot N_{t^*} \geq N_{\min})
\end{aligned}$$

A.3. Proof of Theorem 2: Characterization of boundary types

In the following, let $\|\cdot\|$ denote the standard ℓ_2 norm on the Euclidean space. We first establish a few key lemmas that will be used in the proof.

LEMMA EC.2. *Let $\mathbf{A} \in \mathbb{R}^{m\times D}$ and $\mathbf{u} \in \mathbb{R}^m$ be given, for some $m \geq 1$. Suppose there exists sequence $\{\boldsymbol{\omega}_r\}_{r\in\mathbb{N}}$ such that $\lim_{r\to\infty} \|\mathbf{A}\cdot\boldsymbol{\omega}_r - \mathbf{u}\|^2 = 0$. Then, $\mathbf{u} \in \mathrm{Range}(\mathbf{A})$, i.e. $\mathbf{u}$ lies in the subspace spanned by the columns of the matrix $\mathbf{A}$.*

Proof. Suppose that $\mathbf{u} \notin \text{Range}(\mathbf{A})$ and let $\mathbf{u} := \mathbf{u}_{\parallel} + \mathbf{u}_{\perp}$ where $\mathbf{u}_{\parallel}$ denote the orthogonal projection of the vector $\mathbf{u}$ onto the subspace Range($\mathbf{A}$), so that $\mathbf{u}_{\perp}$ is orthogonal to Range($\mathbf{A}$). Since $\mathbf{u} \notin \text{Range}(\mathbf{A})$, we have $\mathbf{u}_{\perp} \neq \mathbf{0}$. Then, for any $r \in \mathbb{N}$ it follows that

$$\|\mathbf{A} \cdot \boldsymbol{\omega}_r - \mathbf{u}\|^2 = \left\|\left(\mathbf{A} \cdot \boldsymbol{\omega}_r - \mathbf{u}_{\parallel}\right) - \mathbf{u}_{\perp}\right\|^2 = \left\|\mathbf{A} \cdot \boldsymbol{\omega}_r - \mathbf{u}_{\parallel}\right\|^2 + \|\mathbf{u}_{\perp}\|^2 \geq \|\mathbf{u}_{\perp}\|^2 > 0.$$

But this contradicts the hypothesis of the lemma, and therefore $\mathbf{u} \in \text{Range}(\mathbf{A})$. □

LEMMA EC.3. *Let $\mathbf{A} \in \mathbb{R}^{m \times D}$ and $\mathbf{u} \in \mathbb{R}^m$ be given, for some $m \geq 1$. Suppose that there exists $\boldsymbol{\omega} \in \mathbb{R}^D$ such that $\|\mathbf{A} \cdot \boldsymbol{\omega} - \mathbf{u}\| < \varepsilon$ for some $\varepsilon > 0$. Let $\Pi_{\mathbf{A}}(\boldsymbol{\omega})$ denote the orthogonal projection of the vector $\boldsymbol{\omega}$ onto the subspace spanned by the* rows *of the matrix $\mathbf{A}$. Then, it follows that $\|\Pi_{\mathbf{A}}(\boldsymbol{\omega})\| \leq \frac{\varepsilon + \|\mathbf{u}\|}{\sigma_{\min}(\mathbf{A})}$ where $\sigma_{\min}(\mathbf{A}) > 0$ is the smallest non-zero singular value of the matrix $\mathbf{A}$.*

Proof. Let $D' \leq \min(m, D)$ denote the rank of matrix $\mathbf{A}$. Then, using the singular value decomposition (SVD) of $\mathbf{A}$, we get

$$\mathbf{A} = \mathbf{C}\Sigma\mathbf{R}^{\top},$$

where $\mathbf{C} \in \mathbb{R}^{m \times D'}$ is such that its columns represent an orthonormal basis for the column space of $\mathbf{A}$, $\mathbf{R} \in \mathbb{R}^{D \times D'}$ is such that its columns represent an orthonormal basis for the row space of $\mathbf{A}$ and $\Sigma \in \mathbb{R}^{D' \times D'}$ is a diagonal matrix containing the (non-zero) singular values $\sigma_1, \sigma_2, \ldots, \sigma_{D'}$ of the matrix $\mathbf{A}$. Next, represent the vector $\boldsymbol{\omega}$ as $\boldsymbol{\omega} = \boldsymbol{\theta} + \Pi_{\mathbf{A}}(\boldsymbol{\omega})$ where $\boldsymbol{\theta}$ represents the component that is orthogonal to the row space. Since $\Pi_{\mathbf{A}}(\boldsymbol{\omega})$, by definition, lies in the row space, we can write it as $\Pi_{\mathbf{A}}(\boldsymbol{\omega}) = \mathbf{R} \cdot \boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \in \mathbb{R}^{D'}$. Then, it follows that

$$\begin{aligned}
\|\mathbf{A} \cdot \boldsymbol{\omega} - \mathbf{u}\| < \varepsilon &\implies \|\mathbf{A} \cdot \boldsymbol{\omega}\| < \varepsilon + \|\mathbf{u}\| \quad \{\text{by (reverse) triangle inequality}\} \\
&\iff \|\mathbf{A} \cdot \Pi_{\mathbf{A}}(\boldsymbol{\omega})\| < \varepsilon + \|\mathbf{u}\| \\
&\qquad (\text{since } \boldsymbol{\omega} = \boldsymbol{\theta} + \Pi_{\mathbf{A}}(\boldsymbol{\omega}) \text{ and } \mathbf{A} \cdot \boldsymbol{\theta} = \mathbf{0})
\end{aligned}$$

Next, consider the following:

$$\begin{aligned}
\|\mathbf{A} \cdot \Pi_{\mathbf{A}}(\boldsymbol{\omega})\| &= \|\mathbf{A} \cdot (\mathbf{R} \cdot \boldsymbol{\alpha})\| \\
&= \left\|\mathbf{C}\Sigma\mathbf{R}^{\top} \cdot (\mathbf{R} \cdot \boldsymbol{\alpha})\right\| \quad (\text{using SVD of } \mathbf{A}) \\
&= \|(\mathbf{C}\Sigma) \cdot \boldsymbol{\alpha}\| \quad (\text{since columns of } \mathbf{R} \text{ are orthonormal}) \\
&= \|\Sigma \cdot \boldsymbol{\alpha}\| \quad (\text{since } \|\mathbf{C} \cdot \mathbf{x}\| = \|\mathbf{x}\| \text{ for any } \mathbf{x} \text{ as } \mathbf{C} \text{ is unitary}) \\
&= \sqrt{\sum_{d=1}^{D'} \sigma_d^2 \alpha_d^2} \\
&\geq \sigma_{\min}(\mathbf{A}) \cdot \|\boldsymbol{\alpha}\| \\
&= \sigma_{\min}(\mathbf{A}) \cdot \|\mathbf{R} \cdot \boldsymbol{\alpha}\| \quad (\text{since } \mathbf{R} \text{ is unitary}) \\
&= \sigma_{\min}(\mathbf{A}) \cdot \|\Pi_{\mathbf{A}}(\boldsymbol{\omega})\|
\end{aligned}$$

The claim then follows. □

Next, w.l.o.g suppose that the product features remain fixed across the offer-sets, i.e. $\mathbf{z}_{jt} = \mathbf{z}_{jt'}$ for all $j \in [n]$ and all $t \neq t'$—if features are varying across offer-sets, we can just expand the product universe $[n]$. We represent the features as $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$ in the remainder.

Let $\boldsymbol{f} \in \overline{\mathcal{P}} \setminus \mathcal{P}$ be any boundary type. Since $\mathbb{R}^M$ is a metric space, $\overline{\mathcal{P}}$ is precisely the set of the limits of all convergent sequences in $\mathcal{P}$. Therefore, there exists a sequence $\{\boldsymbol{\omega}_r\}_{r\in\mathbb{N}} \subset \mathbb{R}^D$ such that $\lim_{r\to\infty} \boldsymbol{f}(\boldsymbol{\omega}_r) = \boldsymbol{f}$. In addition, it follows that there exists a permutation $\pi : [n] \to [n]$ such that there is a subsequence $\{\boldsymbol{\omega}_{r_\ell}\}_{\ell\in\mathbb{N}}$ satisfying $\boldsymbol{\omega}_{r_\ell}^\top \mathbf{z}_{\pi(1)} \geq \boldsymbol{\omega}_{r_\ell}^\top \mathbf{z}_{\pi(2)} \geq \ldots \geq \boldsymbol{\omega}_{r_\ell}^\top \mathbf{z}_{\pi(n)}$ for all $\ell \in \mathbb{N}$ (this is because the set of permutations of n elements is finite). Since every subsequence must converge to the same limit, we must have

$$\lim_{\ell\to\infty} \boldsymbol{f}(\boldsymbol{\omega}_{r_\ell}) = \boldsymbol{f}.$$

For brevity of notation, we refer to the sequence $\{\boldsymbol{\omega}_{r_\ell}\}_{\ell\in\mathbb{N}}$ as the sequence $\{\boldsymbol{\omega}_r\}_{r\in\mathbb{N}}$ in the remainder, and w.l.o.g assume that the products are indexed such that $\boldsymbol{\omega}_r^\top \mathbf{z}_1 \geq \boldsymbol{\omega}_r^\top \mathbf{z}_2 \geq \ldots \geq \boldsymbol{\omega}_r^\top \mathbf{z}_n$.

We then establish the following key lemma:

LEMMA EC.4. *Consider any offer-set S_t. Let $i_t = \arg\min_{j\in S_t} j$, i.e. i_t is the product with the minimum index in S_t. For any $j \in S_t$, it follows that*

1. *If $f_{jt} = 0$, then $\lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_{i_t} - \mathbf{z}_j) = +\infty$.*
2. *If $f_{jt} > 0$, then $\lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_{i_t} - \mathbf{z}_j) = u_{i_t j} \geq 0$ for some finite $u_{i_t j}$.*

Proof. Note that since $\boldsymbol{\omega}_r^\top \mathbf{z}_{i_t} \geq \boldsymbol{\omega}_r^\top \mathbf{z}_j$ for all $r \in \mathbb{N}$, it follows that $f_{i_t t} \geq f_{jt}$ for all $j \in S_t$. Further, note that $f_{i_t t} > 0$ otherwise $f_{jt} = 0$ for all $j \in S_t$ which is a contradiction since choice probabilities within each offer-set must sum to 1. Now for any $j \in S_t$, consider the following:

$$\begin{aligned} &\exp\left(\boldsymbol{\omega}_r^\top (\mathbf{z}_j - \mathbf{z}_{i_t})\right) = \frac{f_{jt}(\boldsymbol{\omega}_r)}{f_{i_t t}(\boldsymbol{\omega}_r)} \\ \implies &\lim_{r\to\infty} \exp\left(\boldsymbol{\omega}_r^\top (\mathbf{z}_j - \mathbf{z}_{i_t})\right) = \lim_{r\to\infty} \frac{f_{jt}(\boldsymbol{\omega}_r)}{f_{i_t t}(\boldsymbol{\omega}_r)} = \frac{\lim_{r\to\infty} f_{jt}(\boldsymbol{\omega}_r)}{\lim_{r\to\infty} f_{i_t t}(\boldsymbol{\omega}_r)} = \frac{f_{jt}}{f_{i_t t}}. \end{aligned}$$

From the above, it follows that if $f_{jt} = 0$, then $\lim_{r\to\infty} \exp\left(\boldsymbol{\omega}_r^\top (\mathbf{z}_j - \mathbf{z}_{i_t})\right) = 0$ or equivalently, $\lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_j - \mathbf{z}_{i_t}) = -\infty$. When $f_{jt} > 0$, then since $\log(\cdot)$ is continuous, it follows that $\lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_j - \mathbf{z}_{i_t}) = \log \frac{f_{jt}}{f_{i_t t}} \leq 0$ because $f_{jt} \leq f_{i_t t}$ for all $j \in S_t$. The claim then follows.

Finally, note that the same pair of products (i, j) could appear in two different offer-sets, but the uniqueness of limits ensures that the sequence $\boldsymbol{\omega}_r^\top (\mathbf{z}_i - \mathbf{z}_j)$ will converge to the same quantity in both cases. □

We are now ready to prove the result. We first need some additional notation. Let $\text{Pairs}_t = \left\{(i_t, j) \mid j \in S_t \setminus \{i_t\} \text{ and } \lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_{i_t} - \mathbf{z}_j) < \infty\right\}$, note that Pairs_t could be empty for any $1 \leq t \leq T$. Denote $\text{Pairs} = \cup_{t=1}^T \text{Pairs}_t$. Similarly, define the set $\overline{\text{Pairs}}_t = \left\{(i_t, j) \mid j \in S_t \setminus \{i_t\} \text{ and } \lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_{i_t} - \mathbf{z}_j) = +\infty\right\}$ and denote $\overline{\text{Pairs}} = \cup_{t=1}^T \overline{\text{Pairs}}_t$. Note that $\overline{\text{Pairs}}_t$ could also be empty for some $t \in [T]$, but we make the following claim:

Claim 1:

$$\exists t' \in [T] \text{ such that } \overline{\text{Pairs}}_{t'} \neq \emptyset.$$

Suppose this is not true, so that $\overline{\text{Pairs}} = \cup_{t=1}^{T} \overline{\text{Pairs}}_t = \emptyset$. This means that $\text{Pairs} \neq \emptyset$ since, by definition, each product pair must belong to either Pairs or $\overline{\text{Pairs}}$. Then, construct the matrix $\mathbf{A}^{\text{Pairs}} \in \mathbb{R}^{|\text{Pairs}| \times D}$ where row $\mathbf{a}_{ij}^{\text{Pairs}}$ corresponding to pair $(i,j) \in \text{Pairs}$ is given by $\mathbf{a}_{ij}^{\text{Pairs}} = \mathbf{z}_i - \mathbf{z}_j$. Similarly, let $\mathbf{u} \in \mathbb{R}^{|\text{Pairs}|}$ denote the vector of utilities u_{ij} for each pair $(i,j) \in \text{Pairs}$ (refer to Lemma EC.4). Then, it follows from Lemma EC.4 that for any $(i,j) \in \text{Pairs}$:

$$\lim_{r\to\infty} \boldsymbol{\omega}_r^\top (\mathbf{z}_i - \mathbf{z}_j) - u_{ij} = 0 \implies \lim_{r\to\infty} \left(\boldsymbol{\omega}_r^\top (\mathbf{z}_i - \mathbf{z}_j) - u_{ij}\right)^2 = 0 \;\; \text{(since } x^2 \text{ is continuous at } x = 0)$$

which in turn implies that $\lim_{r\to\infty} \left\|\mathbf{A}^{\text{Pairs}} \cdot \boldsymbol{\omega}_r - \mathbf{u}\right\|^2 = 0$.

Now, applying Lemma EC.2 tells us that $\mathbf{u} \in \text{Range}(\mathbf{A}^{\text{Pairs}})$, i.e. there exists $\boldsymbol{\omega}_0 \in \mathbb{R}^D$ such that $\mathbf{A}^{\text{Pairs}} \cdot \boldsymbol{\omega}_0 = \mathbf{u}$. Then, from Lemma EC.4 it follows that for any $j \in S_t$ and any $1 \le t \le T$:

$$\begin{aligned}
f_{jt} &= \exp\left(-u_{i_t j}\right) \cdot f_{i_t t} \;\; \text{(since } f_{jt} > 0) \\
&= \exp\left(\boldsymbol{\omega}_0^\top (\mathbf{z}_j - \mathbf{z}_{i_t})\right) \cdot f_{i_t t} \;\; \left(\text{since } u_{i_t j} = (\mathbf{z}_{i_t} - \mathbf{z}_j)^\top \boldsymbol{\omega}_0 \text{ as shown above}\right) \\
\implies f_{jt} &= \frac{\exp\left(\boldsymbol{\omega}_0^\top (\mathbf{z}_j - \mathbf{z}_{i_t})\right)}{1 + \sum_{\ell \in S_t \setminus \{i_t\}} \exp\left(\boldsymbol{\omega}_0^\top (\mathbf{z}_\ell - \mathbf{z}_{i_t})\right)} \qquad \left(\text{since } \sum_{\ell \in S_t} f_{\ell t} = 1\right) \\
\iff f_{jt} &= \frac{\exp\left(\boldsymbol{\omega}_0^\top \mathbf{z}_j\right)}{\sum_{\ell \in S_t} \exp\left(\boldsymbol{\omega}_0^\top \mathbf{z}_\ell\right)} = f_{jt}(\boldsymbol{\omega}_0)
\end{aligned}$$

That is, $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{\omega}_0)$. But since $\boldsymbol{f}(\boldsymbol{\omega}_0) \in \mathcal{P}$ by definition, this means that $\boldsymbol{f} \in \mathcal{P}$ which contradicts the assumption that $\boldsymbol{f}$ is a boundary type and belongs to $\overline{\mathcal{P}} \setminus \mathcal{P}$. The claim then follows.

In addition, if $\overline{\text{Pairs}}_{t'} \neq \emptyset$, then for any pair $(i_{t'}, j') \in \overline{\text{Pairs}}_{t'}$, it follows from Lemma EC.4 that $f_{j't'} = 0$ establishing the second part of the theorem.

Following Claim 1, there are two cases which we deal with separately:

Case 1: $\text{Pairs} = \emptyset$. In this case, it follows from Lemma EC.4 that $f_{i_t t} = 1$ for all $1 \le t \le T$, which implies that $\boldsymbol{f}$ is a boundary type that chooses only a single product, viz. i_t from each offer-set S_t. Choose any $U > 0$. Based on the definition of $\overline{\text{Pairs}}$, we know that there exists $\tilde{\boldsymbol{\omega}}$ such that $\tilde{\boldsymbol{\omega}}^\top (\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U$ for all $(\bar{i}, \bar{j}) \in \overline{\text{Pairs}}$. Then, it follows that the choice probabilities under the boundary type $\boldsymbol{f}$ are equal to the limiting choice probabilities using $\boldsymbol{\omega}_0 = \mathbf{0}$ (i.e. the all zeros vector) and $\boldsymbol{\theta} = \tilde{\boldsymbol{\omega}}$.

Case 2: $\text{Pairs} \neq \emptyset$. In this case, first we construct the matrix $\mathbf{A}^{\text{Pairs}} \in \mathbb{R}^{|\text{Pairs}| \times D}$ as outlined above in the proof of Claim 1. Given this, we choose the parameters $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ as follows:

Choosing $\boldsymbol{\omega}_0$. As shown in the proof of Claim 1 above, it follows that $\lim_{r\to\infty} \left\|\mathbf{A}^{\text{Pairs}} \cdot \boldsymbol{\omega}_r - \mathbf{u}\right\|^2 = 0$. Then, Lemma EC.2 tells us that $\mathbf{u} \in \text{Range}(\mathbf{A}^{\text{Pairs}})$, i.e. there exists $\boldsymbol{\omega}_0 \in \mathbb{R}^D$ such that $\mathbf{A}^{\text{Pairs}} \cdot \boldsymbol{\omega}_0 = \mathbf{u}$.

Choosing $\boldsymbol{\theta}$. Next, using the definition of Pairs and $\overline{\text{Pairs}}$, it follows that given *any* $\varepsilon > 0$ and $U > 0$, there exists $\tilde{\boldsymbol{\omega}}$ such that:

$$\left|\tilde{\boldsymbol{\omega}}^\top (\mathbf{z}_i - \mathbf{z}_j) - u_{ij}\right| < \varepsilon \;\; \forall\, (i,j) \in \text{Pairs} \quad \text{and} \quad \tilde{\boldsymbol{\omega}}^\top (\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U \;\; \forall\, (\bar{i}, \bar{j}) \in \overline{\text{Pairs}}.$$

Fix some $\varepsilon > 0$ and choose U such that

$$U > \frac{\sqrt{|\text{Pairs}|} \cdot \varepsilon + \|\mathbf{u}\|}{\sigma_{\min}(\mathbf{A}^{\text{Pairs}})} \cdot B,$$

where $B \stackrel{\text{def}}{=} \max_{(\bar{i},\bar{j}) \in \overline{\text{Pairs}}} \left\|\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}\right\|$.

Then, the choice of $\tilde{\boldsymbol{\omega}}$ implies that $\left\|\mathbf{A}^{\text{Pairs}} \cdot \tilde{\boldsymbol{\omega}} - \mathbf{u}\right\| < \sqrt{|\text{Pairs}|} \cdot \varepsilon$ so that we can apply Lemma EC.3 to establish

$$\|\Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})\| < \frac{\sqrt{|\text{Pairs}|} \cdot \varepsilon + \|\mathbf{u}\|}{\sigma_{\min}(\mathbf{A}^{\text{Pairs}})}. \tag{EC.8}$$

Next, choose $\boldsymbol{\theta} = \tilde{\boldsymbol{\omega}} - \Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})$ where $\Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})$ is the projection of $\tilde{\boldsymbol{\omega}}$ onto the subspace spanned by the rows of $\mathbf{A}^{\text{Pairs}}$. We show that $\boldsymbol{\theta}$ satisfies:

$$(1)\ \boldsymbol{\theta}^\top(\mathbf{z}_i - \mathbf{z}_j) = 0 \ \ \forall\, (i,j) \in \text{Pairs} \ \text{ and } \ (2)\ \boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > 0 \ \ \forall\, (\bar{i},\bar{j}) \in \overline{\text{Pairs}}.$$

Part (1) follows since $\boldsymbol{\theta}$ is orthogonal to the subspace spanned by the rows of $\mathbf{A}^{\text{Pairs}}$. For part (2), consider the following, for any $(\bar{i},\bar{j}) \in \overline{\text{Pairs}}$:

$$\begin{aligned}
&\tilde{\boldsymbol{\omega}}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U \\
\iff &\left(\boldsymbol{\theta} + \Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})\right)^\top (\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U \\
\iff &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) + \Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U \\
\implies &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) + \|\Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})\| \cdot \left\|\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}\right\| > U \ \ \text{(using Cauchy-Schwarz inequality)} \\
\iff &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U - \|\Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})\| \cdot \left\|\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}\right\| \\
\implies &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U - \|\Pi_{\mathbf{A}^{\text{Pairs}}}(\tilde{\boldsymbol{\omega}})\| \cdot B \ \ \left(\text{since } \left\|\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}\right\| \le B \text{ by definition of } B\right) \\
\implies &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > U - \frac{\sqrt{|\text{Pairs}|} \cdot \varepsilon + \|\mathbf{u}\|}{\sigma_{\min}(\mathbf{A}^{\text{Pairs}})} \cdot B \ \ \{\text{using equation (EC.8)}\} \\
\implies &\boldsymbol{\theta}^\top(\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) > 0 \ \ \text{(by choice of } U)
\end{aligned}$$

Then, it is easy to see that the choice probabilities under the boundary type $\boldsymbol{f}$ are equal to the limiting probabilities for the choice $(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ computed above.

A.4. Proof of Theorem 3: Recovery of boundary types

Define the function $H : \overline{\mathcal{P}} \to \mathbb{R}$ such that $H(\boldsymbol{f}) = \sum_{i=1}^n c_i f_i$ for each $\boldsymbol{f} \in \overline{\mathcal{P}}$. Consequently, the support finding step (11) can be equivalently written as:

$$\arg\max_{\boldsymbol{f} \in \overline{\mathcal{P}}} \ H(\boldsymbol{f}) \tag{EC.9}$$

Let $C^* = \max_{\boldsymbol{f} \in \overline{\mathcal{P}}} H(\boldsymbol{f})$ denote the optimal objective of the above subproblem, note that this is well-defined since $H(\cdot)$ is continuous and $\overline{\mathcal{P}}$ is compact.

Proof Overview. Without loss of generality, index the products such that $c_1 \ge c_2 \ge \cdots \ge c_n$. Note that $C^* \le c_1$ because the objective value in subproblem (EC.9) is a convex combination of $\{c_1, c_2, \cdots, c_n\}$. Since $\mathbf{z}_1$ is an extreme point, it follows from Lemma EC.6 (proved below) that $\mathbf{e}_1 \in \overline{\mathcal{P}}$, where $\mathbf{e}_1$ is defined as:

$$e_{1i} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then it follows that

$$H(\mathbf{e}_1) = \sum_{i=1}^n c_i e_{1i} = c_1.$$

In addition, Lemma EC.5 below shows the existence of $\boldsymbol{\theta}_1 \in \mathbb{R}^D$ such that $\boldsymbol{\theta}_1^\top \mathbf{z}_1 > \boldsymbol{\theta}_1^\top \mathbf{z}_i$ for all $1 < i \le n$. Then, it is easy to see that $\mathbf{e}_1 = \boldsymbol{f}(\mathbf{0}, \boldsymbol{\theta}_1)$ from which the result follows.

To complete the proof, we now establish the two lemmas referenced above. The first provides a characterization of extreme points of the polytope $\mathcal{Z}_n$:

LEMMA EC.5 **(Characterization of extreme points)**. *$\mathbf{z}_j$ is an extreme point of the polytope $\mathcal{Z}_n$ if and only if there exists $\boldsymbol{\theta} \in \mathbb{R}^D$ such that $\boldsymbol{\theta}^\top \mathbf{z}_j > \boldsymbol{\theta}^\top \mathbf{z}_i$ for all $i \neq j$.*

Proof. **"if" direction**. If possible, suppose that $\mathbf{z}_j$ is not an extreme point, i.e. $\mathbf{z}_j \in \text{conv}(\{\mathbf{z}_i : i \neq j, 1 \leq i \leq n\})$ so that there exists coefficients $\lambda_{ij} \geq 0$ such that

$$\mathbf{z}_j = \sum_{i \neq j} \lambda_{ij} \mathbf{z}_i \,;\; \sum_{i \neq j} \lambda_{ij} = 1 \,;\; \lambda_{ij} \geq 0 \;\; \forall \; i \neq j.$$

But this results in the following contradiction:

$$\begin{aligned}
\mathbf{z}_j = \sum_{i \neq j} \lambda_{ij} \mathbf{z}_i &\implies \boldsymbol{\theta}^\top \mathbf{z}_j = \sum_{i \neq j} \lambda_{ij} \boldsymbol{\theta}^\top \mathbf{z}_i \\
&\implies \boldsymbol{\theta}^\top \mathbf{z}_j < \sum_{i \neq j} \lambda_{ij} \boldsymbol{\theta}^\top \mathbf{z}_j \;\; (\text{since } \lambda_{ij} > 0 \text{ for some } i) \\
&\implies \boldsymbol{\theta}^\top \mathbf{z}_j < \boldsymbol{\theta}^\top \mathbf{z}_j \cdot \left(\sum_{i \neq j} \lambda_{ij} \right) \\
&\implies \boldsymbol{\theta}^\top \mathbf{z}_j < \boldsymbol{\theta}^\top \mathbf{z}_j
\end{aligned}$$

"only if" direction. Denote $\mathcal{M} = \text{conv}(\{\mathbf{z}_i : i \neq j, 1 \leq i \leq n\})$ and observe that $\mathcal{M}$ is a closed, convex and proper subset of $\mathbb{R}^D$. Now since $\mathbf{z}_j \notin \mathcal{M}$, it follows from the (strong) separation theorem in convex analysis that there exists $\boldsymbol{\theta} \in \mathbb{R}^D$ such that

$$\boldsymbol{\theta}^\top \mathbf{z}_j > \boldsymbol{\theta}^\top \mathbf{z}_i \;\; \forall \; i \neq j.$$

□

Next, we show that for each product feature vector that is an extreme point, there exists a boundary type that chooses the product with probability 1 from offer-set $[n]$:

LEMMA EC.6. *Suppose $\mathbf{z}_j$ is an extreme point of the polytope $\mathcal{Z}_n$. Define the vector $\mathbf{e}_j = (e_{j1}, e_{j2}, \ldots, e_{jn})$ as follows:*

$$e_{ji} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathbf{e}_j \in \overline{\mathcal{P}}$, in other words, there exists a boundary type that chooses product j with probability 1 from offer-set $[n]$.

Proof. Since $\mathbf{z}_j$ is an extreme point, it follows from Lemma EC.5 that

$$\exists \boldsymbol{\theta} \in \mathbb{R}^D \text{ s.t. } \boldsymbol{\theta}^\top \mathbf{z}_j > \boldsymbol{\theta}^\top \mathbf{z}_i \;\; \forall i \neq j.$$

Consider the sequence $\{r \cdot \boldsymbol{\theta}\}_{r \in \mathbb{N}} \subseteq \mathbb{R}^D$. For any $i \neq j$, note that:

$$\lim_{r \to \infty} f_i(r \cdot \boldsymbol{\theta}) = \lim_{r \to \infty} \frac{\exp(r \cdot \boldsymbol{\theta}^\top \mathbf{z}_i)}{\sum_{\ell=1}^n \exp(r \cdot \boldsymbol{\theta}^\top \mathbf{z}_\ell)} = \lim_{r \to \infty} \frac{\exp(-r \cdot (\boldsymbol{\theta}^\top \mathbf{z}_j - \boldsymbol{\theta}^\top \mathbf{z}_i))}{1 + \sum_{\ell \neq j} \exp(-r \cdot (\boldsymbol{\theta}^\top \mathbf{z}_j - \boldsymbol{\theta}^\top \mathbf{z}_\ell))} = \frac{0}{1+0} = 0.$$

An analogous argument shows that

$$\lim_{r \to \infty} f_j(r \cdot \boldsymbol{\theta}) = \lim_{r \to \infty} \frac{\exp(r \cdot \boldsymbol{\theta}^\top \mathbf{z}_j)}{\sum_{\ell=1}^n \exp(r \cdot \boldsymbol{\theta}^\top \mathbf{z}_\ell)} = 1.$$

From the above statements it follows that

$$\lim_{r \to \infty} \boldsymbol{f}(r \cdot \boldsymbol{\theta}) = \mathbf{e}_j,$$

and since the closure contains all limit points of convergent sequences, it follows that $\mathbf{e}_j \in \overline{\mathcal{P}}$. □

A.5. Proof of Theorem 4: Convergence in finite number of iterations

Since each $\mathbf{z}_j$ is an extreme point, it follows from Lemma EC.6 that $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n \in \overline{\mathcal{P}}$. Then it follows that $\boldsymbol{y} \in \text{conv}(\overline{\mathcal{P}})$ since $\boldsymbol{y} = \sum_{i=1}^n y_i \cdot \mathbf{e}_i$ and $\sum_{i=1}^n y_i = 1$. Further, it is easy to see that $\mathsf{SQ}(\boldsymbol{y}) = 0$ and $\mathsf{SQ}(\mathbf{g}) > 0$ for all $\mathbf{g} \neq \boldsymbol{y}$. Similarly, we have that $\mathsf{NLL}(\boldsymbol{y}) = -\sum_{i=1}^n y_i \log y_i$ and it can be shown that $\mathsf{NLL}(\mathbf{g}) > \mathsf{NLL}(\boldsymbol{y})$ for all $\mathbf{g} \neq \boldsymbol{y}$ (using the fact that relative entropy or KL-divergence is non-negative). Therefore $\mathbf{g}^* = \boldsymbol{y}$ for both the squared and negative log-likelihood loss functions.

Now, if at some iteration $1 \leq k \leq n$, we have $\left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \boldsymbol{f}^{(k)} - \mathbf{g}^{(k-1)} \right\rangle \geq 0$, then by convexity it follows that $\mathsf{loss}(\mathbf{g}) \geq \mathsf{loss}(\mathbf{g}^{(k-1)})$ for all $\mathbf{g} \in \text{conv}(\overline{\mathcal{P}})$ which means that $\mathbf{g}^{(k-1)} = \mathbf{g}^*$ and the algorithm terminates. So, suppose that $\left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \boldsymbol{f}^{(k)} - \mathbf{g}^{(k-1)} \right\rangle < 0$ for each $1 \leq k \leq n$, which means that $\boldsymbol{f}^{(k)} - \mathbf{g}^{(k-1)}$ is a descent direction and an improving solution can be found. The proportions update step in Algorithm 1 ensures that we have $\boldsymbol{f}^{(k_1)} \neq \boldsymbol{f}^{(k_2)}$ in any two iterations $k_1 \neq k_2$ (since it optimizes over all previously found types). In addition, Theorem 3 shows that we recover only boundary types in each iteration, from which it follows that at the end of n iterations, we have found the types $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n$. Now, clearly $\sum_{i=1}^n y_i \cdot \mathbf{e}_i = \boldsymbol{y} = \mathbf{g}^*$ and since the proportions update step optimizes over the convex hull of all previously found types, the claim follows.

A.6. Proof of Theorem 5: Boundary types are always optimal

If $D_1 = 0$, i.e. all product features are binary, then it follows that $\mathcal{E} = \cup_{\ell=1}^n \{\ell\}$, i.e. each product ℓ belongs to its own equivalence class. Then, treating the feature vectors $\{\mathbf{b}_\ell\}_{\ell \in [n]}$ as elements of $\mathbb{R}^D$, it follows that each $\mathbf{b}_\ell$ is an extreme point of the polytope $\mathcal{Z}_n = \text{conv}(\{\mathbf{b}_1, \cdots, \mathbf{b}_n\})$. Therefore, we can apply the result of Corollary 1 to establish that in each iteration, the optimal solution to the support finding step is a boundary type that chooses only a single product. Further, choosing $\boldsymbol{\omega}_0 = \mathbf{0}$ and $\boldsymbol{\theta}$ as in Theorem 3, the claim follows.

Now, we consider the scenario when $D_1 > 0$. For ease of exposition, we prove the result for the case when $D_2 = 1$, i.e. there is only a single binary feature but the proof can be easily extended, albeit with additional notation, to the general case. Define the function $G(\cdot) : \mathbb{R}^{D_1+1} \to \mathbb{R}$ for $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$ and $\delta \in \mathbb{R}$ as:

$$G(\boldsymbol{\omega}, \delta) = \frac{\sum_{i=1}^n c_i \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_i + \delta \cdot b_i\right)}{\sum_{j=1}^n \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j + \delta \cdot b_j\right)}.$$

Further, let $G^* = \sup_{\boldsymbol{\omega} \in \mathbb{R}^{D_1}, \delta \in \mathbb{R}} G(\boldsymbol{\omega}, \delta)$; since $G(\cdot)$ is bounded above, G^* is finite. We denote the atomic likelihood vector corresponding to logit parameter $(\boldsymbol{\omega}, \delta)$ as $\boldsymbol{f}_{(\boldsymbol{\omega}, \delta)}$ so that the the set of atomic likelihood vectors is given by $\mathcal{P} = \left\{ \boldsymbol{f}_{(\boldsymbol{\omega}, \delta)} : \boldsymbol{\omega} \in \mathbb{R}^{D_1}, \delta \in \mathbb{R} \right\}$.

Let S_0 be the set of products that have the binary feature absent, i.e. $S_0 = \{i \in [n] : b_i = 0\}$ and let $S_1 = [n] \setminus S_0$. For $e \in \{0, 1\}$, define the sets $\mathcal{P}_e = \left\{ \boldsymbol{f}^{(e)}(\boldsymbol{\omega}) : \boldsymbol{\omega} \in \mathbb{R}^{D_1} \right\} \subseteq \Delta_{|S_e|-1}$, where $\boldsymbol{f}^{(e)}(\boldsymbol{\omega}) = \left(f_i^{(e)}(\boldsymbol{\omega}) \right)_{i \in S_e}$ and

$$f_i^{(e)}(\boldsymbol{\omega}) = \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_i\right)}{\sum_{j \in S_e} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right)} \tag{EC.10}$$

In other words, $\mathcal{P}_0$ (resp. $\mathcal{P}_1$) corresponds to choice probabilities under boundary types that do not consider any product in S_1 (resp. S_0). Let $\boldsymbol{f}^{(0)}, \boldsymbol{f}^{(1)}$ denote arbitrary elements in $\overline{\mathcal{P}}_0$ and $\overline{\mathcal{P}}_1$ where $\overline{\mathcal{P}}_0$ and $\overline{\mathcal{P}}_1$ denote the closures of the sets $\mathcal{P}_0$ and $\mathcal{P}_1$ respectively. Then, consider the following optimization problems for $e \in \{0, 1\}$:

$$(\mathrm{P}_e): \quad \arg\max_{\boldsymbol{f}^{(e)} \in \overline{\mathcal{P}}_e} H_e(\boldsymbol{f}^{(e)}),$$

where $H_e : \overline{\mathcal{P}}_e \to \mathbb{R}$ is such that $H_e(\boldsymbol{f}^{(e)}) = \sum_{i \in S_e} c_i f_i^{(e)}$.

Next, for any $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$, define the following:

$$a_0(\boldsymbol{\omega}) = \sum_{j \in S_0} c_j \exp(\boldsymbol{\omega}^\top \mathbf{z}_j);\ b_0(\boldsymbol{\omega}) = \sum_{j \in S_0} \exp(\boldsymbol{\omega}^\top \mathbf{z}_j);\quad G_0(\boldsymbol{\omega}) = \frac{a_0(\boldsymbol{\omega})}{b_0(\boldsymbol{\omega})}.$$
$$a_1(\boldsymbol{\omega}) = \sum_{j \in S_1} c_j \exp(\boldsymbol{\omega}^\top \mathbf{z}_j);\ b_1(\boldsymbol{\omega}) = \sum_{j \in S_1} \exp(\boldsymbol{\omega}^\top \mathbf{z}_j);\quad G_1(\boldsymbol{\omega}) = \frac{a_1(\boldsymbol{\omega})}{b_1(\boldsymbol{\omega})}.$$

Further, let $C^*_{(0)} = \sup_{\boldsymbol{\omega} \in \mathbb{R}^{D_1}} G_0(\boldsymbol{\omega})$ and similarly, $C^*_{(1)} = \sup_{\boldsymbol{\omega} \in \mathbb{R}^{D_1}} G_1(\boldsymbol{\omega})$. Also, recall from the beginning of Appendix A.4 that $C^* = \max_{\boldsymbol{f} \in \overline{\mathcal{P}}} H(\boldsymbol{f})$.

Claim 1:

$$(1)\ C^* = G^*;\quad (2)\ C^*_{(0)} = \max_{\boldsymbol{f}^{(0)} \in \overline{\mathcal{P}}_0} H_0(\boldsymbol{f}^{(0)}) \text{ and } C^*_{(1)} = \max_{\boldsymbol{f}^{(1)} \in \overline{\mathcal{P}}_1} H_1(\boldsymbol{f}^{(1)}).$$

Consider part (1). First observe that, $G(\boldsymbol{\omega}, \delta) = H\left(\boldsymbol{f}_{(\boldsymbol{\omega}, \delta)}\right) \le C^*$ for all $\boldsymbol{\omega} \in \mathbb{R}^{D_1}, \delta \in \mathbb{R}$. As supremum is the least upper bound, it follows that $G^* \le C^*$. Next, for *any* $\boldsymbol{f} \in \overline{\mathcal{P}}$, there exists a sequence $\{(\boldsymbol{\omega}_r, \delta_r)\}_{r \in \mathbb{N}} \subset \mathbb{R}^{D_1+1}$ such that $\lim_{r \to \infty} \boldsymbol{f}_{(\boldsymbol{\omega}_r, \delta_r)} = \boldsymbol{f}$. Then, since $H(\cdot)$ is continuous, it follows that

$$\boldsymbol{f} = \lim_{r \to \infty} \boldsymbol{f}_{(\boldsymbol{\omega}_r, \delta_r)} \implies H(\boldsymbol{f}) = \lim_{r \to \infty} H\left(\boldsymbol{f}_{(\boldsymbol{\omega}_r, \delta_r)}\right) = \lim_{r \to \infty} G(\boldsymbol{\omega}_r, \delta_r) \le G^*,$$

where the last inequality follows since $G(\boldsymbol{\omega}_r, \delta_r) \le G^*$ for all $r \in \mathbb{N}$. Since $\boldsymbol{f} \in \overline{\mathcal{P}}$ was arbitrary, this means that $H(\boldsymbol{f}) \le G^*$ for all $\boldsymbol{f} \in \overline{\mathcal{P}}$. Finally, since $H(\cdot)$ is continuous and $\overline{\mathcal{P}}$ is compact, there exists $\boldsymbol{f}^* \in \overline{\mathcal{P}}$ such that $H(\boldsymbol{f}^*) = C^*$. This means that $C^* = H(\boldsymbol{f}^*) \le G^*$ and combining with $G^* \le C^*$, the result of part (1) follows.

The above argument can be repeated while restricting to the domains $\overline{\mathcal{P}}_0$ and $\overline{\mathcal{P}}_1$ to establish part (2). The claim then follows.

Claim 2:

$$C^* \le \max(C^*_{(0)}, C^*_{(1)}).$$

Proof. First observe that for any $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$ and $\delta \in \mathbb{R}$:

$$\begin{aligned} G(\boldsymbol{\omega}, \delta) &= \frac{\sum_{i=1}^n c_i \exp(\boldsymbol{\omega}^\top \mathbf{z}_i + \delta \cdot b_i)}{\sum_{j=1}^n \exp(\boldsymbol{\omega}^\top \mathbf{z}_j + \delta \cdot b_j)} \\ &= G_0(\boldsymbol{\omega}) \cdot \frac{b_0(\boldsymbol{\omega})}{b_0(\boldsymbol{\omega}) + e^\delta \cdot b_1(\boldsymbol{\omega})} + G_1(\boldsymbol{\omega}) \cdot \frac{e^\delta \cdot b_1(\boldsymbol{\omega})}{b_0(\boldsymbol{\omega}) + e^\delta \cdot b_1(\boldsymbol{\omega})}. \end{aligned}$$

That is, $G(\boldsymbol{\omega}, \delta)$ is a convex combination of $G_0(\boldsymbol{\omega})$ and $G_1(\boldsymbol{\omega})$ and consequently we have

$$\forall \boldsymbol{\omega} \in \mathbb{R}^{D_1}, \forall \delta \in \mathbb{R}\ \ G(\boldsymbol{\omega}, \delta) \le \max(G_0(\boldsymbol{\omega}), G_1(\boldsymbol{\omega})) \le \max(C^*_{(0)}, C^*_{(1)}),$$

where the last inequality follows from the definition of $C^*_{(0)}$ and $C^*_{(1)}$. The claim then follows from the definition of supremum and the fact that $G^* = C^*$ (from Claim 1 above). □

Given Claims 1 and 2 above, there are two cases to consider: $C^*_{(1)} \ge C^*_{(0)}$ or $C^*_{(1)} < C^*_{(0)}$. We focus on the case when $C^*_{(1)} \ge C^*_{(0)}$, the other case can be dealt with a symmetric argument:

Case 1: $\mathbf{C^*_{(1)} \ge C^*_{(0)}}$. For each $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$, define the vector $\tilde{\boldsymbol{f}}(\boldsymbol{\omega}) \in \Delta_{n-1}$ as follows:

$$\tilde{f}_j(\boldsymbol{\omega}) = \begin{cases} f_j^{(1)}(\boldsymbol{\omega}) & \text{if } j \in S_1 \\ 0 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{f}^{(1)}(\boldsymbol{\omega}) \in \mathcal{P}_1$ is as defined above in equation (EC.10).

Claim 3:

$$\tilde{\boldsymbol{f}}(\boldsymbol{\omega}) \in \overline{\mathcal{P}} \;\; \forall\, \boldsymbol{\omega} \in \mathbb{R}^{D_1}.$$

Given any $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$, consider the sequence $\{(\boldsymbol{\omega}, r)\}_{r\in\mathbb{N}} \subset \mathbb{R}^{D_1+1}$. Now for any $i \in S_0$, it follows that

$$\begin{aligned}\lim_{r\to\infty} f_{(\boldsymbol{\omega},r),i} &= \lim_{r\to\infty} \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_i + r\cdot b_i\right)}{\sum_{\ell=1}^n \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell + r\cdot b_\ell\right)} \\ &= \lim_{r\to\infty} \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_i\right)}{\sum_{j\in S_1} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j + r\right) + \sum_{\ell\in S_0} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell\right)} = 0.\end{aligned}$$

Similarly, for any $j \in S_1$, it follows that

$$\lim_{r\to\infty} f_{(\boldsymbol{\omega},r),j} = \lim_{r\to\infty} \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j + r\cdot b_j\right)}{\sum_{\ell=1}^n \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell + r\cdot b_\ell\right)} = \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right)}{\sum_{\ell\in S_1} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell\right)} = f_j^{(1)}(\boldsymbol{\omega}).$$

From the above statements, it follows that

$$\lim_{r\to\infty} \boldsymbol{f}_{(\boldsymbol{\omega},r)} = \tilde{\boldsymbol{f}}(\boldsymbol{\omega}) \implies \tilde{\boldsymbol{f}}(\boldsymbol{\omega}) \in \overline{\mathcal{P}},$$

where the implication follows since $\overline{\mathcal{P}}$ contains the limit of all convergent sequences in $\mathcal{P}$.

Claim 4:

$$C^* = C^*_{(1)}.$$

Proof. From Claim 3, it follows that $\tilde{\boldsymbol{f}}(\boldsymbol{\omega}) \in \overline{\mathcal{P}}$ for all $\boldsymbol{\omega} \in \mathbb{R}^{D_1}$. Then, consider the following:

$$C^* \geq H\left(\tilde{\boldsymbol{f}}(\boldsymbol{\omega})\right) = \sum_{\ell=1}^n c_\ell \tilde{f}_\ell(\boldsymbol{\omega}) = \frac{\sum_{j\in S_1} c_j \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right)}{\sum_{\ell\in S_1} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell\right)} = G_1(\boldsymbol{\omega}).$$

where the first inequality follows from the definition of C^*. Then, it follows that

$$G_1(\boldsymbol{\omega}) \leq C^* \;\; \forall \boldsymbol{\omega} \in \mathbb{R}^{D_1} \implies C^*_{(1)} \leq C^* \text{ (since } C^*_{(1)} \text{ is the supremum).}$$

Combining with Claim 2, it follows that $C^* = C^*_{(1)}$. □

Next, let $\boldsymbol{f}^{(1,*)} \in \overline{\mathcal{P}}_1$ denote the optimal solution for problem (P_1). Then, using the arguments given in the proof of Theorem 2 above, it follows that there exists $\boldsymbol{\omega}_0^{(1)}, \boldsymbol{\theta}^{(1)} \in \mathbb{R}^{D_1}$ such that $\boldsymbol{f}^{(1,*)} = \boldsymbol{f}^{(1)}(\boldsymbol{\omega}_0^{(1)}, \boldsymbol{\theta}^{(1)})$ where $\boldsymbol{f}^{(1)}(\boldsymbol{\omega}_0^{(1)}, \boldsymbol{\theta}^{(1)})$ are the limiting choice probabilities given by (if $\boldsymbol{f}^{(1,*)} > \mathbf{0}$, then we choose $\boldsymbol{\theta}^{(1)} = \mathbf{0}$):

$$f_j^{(1,*)} = f_j^{(1)}(\boldsymbol{\omega}_0^{(1)}, \boldsymbol{\theta}^{(1)}) = \lim_{r\to\infty} \frac{\exp\left\{(\boldsymbol{\omega}_0^{(1)} + r\cdot\boldsymbol{\theta}^{(1)})^\top \mathbf{z}_j\right\}}{\sum_{\ell\in S_1} \exp\left\{(\boldsymbol{\omega}_0^{(1)} + r\cdot\boldsymbol{\theta}^{(1)})^\top \mathbf{z}_\ell\right\}} \;\; \forall\, j \in S_1. \tag{EC.11}$$

Define $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_0^{(1)} \circ (0)$ where recall that $\circ$ denotes the concatenation operator, and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(1)} \circ (\theta_1)$ with $\theta_1 = \sqrt{5}\cdot\left\|\boldsymbol{\theta}^{(1)}\right\|\cdot Z_{\max}$, where $Z_{\max}$ is a constant that satisfies $Z_{\max} \geq \|\mathbf{z}_\ell\|$ for all $\ell \in [n]$. Then, we can show the following:

Claim 5:

$$\begin{aligned}&(1)\; \boldsymbol{\theta}^\top(\mathbf{z}_j \circ b_j - \mathbf{z}_i \circ b_i) > 0 \;\; \forall\, j \in S_1, \forall\, i \in S_0. \\ &(2)\; f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = 0 \;\; \forall\, i \in S_0. \\ &(3)\; f_j(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = f_j^{(1,*)} \;\; \forall\, j \in S_1.\end{aligned}$$

Proof. We start with part (1). Consider any $j \in S_1$ and any $i \in S_0$, then it follows

$$\begin{aligned}
\boldsymbol{\theta}^\top(\mathbf{z}_j \circ b_j - \mathbf{z}_i \circ b_i) > 0 &\iff \boldsymbol{\theta}^\top(\mathbf{z}_j \circ 1 - \mathbf{z}_i \circ 0) > 0 \\
&\iff {\boldsymbol{\theta}^{(1)}}^\top(\mathbf{z}_j - \mathbf{z}_i) + \theta_1 > 0 \\
&\impliedby -\left\|\boldsymbol{\theta}^{(1)}\right\| \cdot \|\mathbf{z}_j - \mathbf{z}_i\| + \theta_1 > 0 \quad \text{(by Cauchy-Schwarz inequality)} \\
&\iff \theta_1 > \left\|\boldsymbol{\theta}^{(1)}\right\| \cdot \|\mathbf{z}_j - \mathbf{z}_i\| \\
&\impliedby \theta_1 > \left\|\boldsymbol{\theta}^{(1)}\right\| \cdot (\|\mathbf{z}_i\| + \|\mathbf{z}_j\|) \quad \text{(by triangle inequality)} \\
&\impliedby \theta_1 > 2 \cdot \left\|\boldsymbol{\theta}^{(1)}\right\| \cdot \max_{1 \le \ell \le n} \|\mathbf{z}_\ell\| \\
&\impliedby \theta_1 > 2 \cdot \left\|\boldsymbol{\theta}^{(1)}\right\| \cdot Z_{\max}
\end{aligned}$$

and the last inequality is true by the choice of θ_1.

Next, for part (2) consider the following, given any $i \in S_0$:

$$\begin{aligned}
0 \le f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) &= \frac{\exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_i \circ 0)\right\}}{\sum_{\ell=1}^n \exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_\ell \circ b_\ell)\right\}} \\
&\le \frac{\exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_i \circ 0)\right\}}{\sum_{\ell \in S_1} \exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_\ell \circ b_\ell)\right\}} \\
&= \frac{\exp\left({\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_i\right)}{\sum_{\ell \in S_1} \exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_\ell + r \cdot \boldsymbol{\theta}^\top(\mathbf{z}_\ell \circ b_\ell - \mathbf{z}_i \circ 0)\right\}}
\end{aligned}$$

Then using the Squeeze theorem and part (1), it follows that $\lim_{r \to \infty} f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = 0$.

Finally, for part (3), consider the following for any $j \in S_1$:

$$\begin{aligned}
& f_j(\boldsymbol{\omega}_0, \boldsymbol{\theta}) \\
&= \lim_{r \to \infty} \frac{\exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_j \circ 1)\right\}}{\sum_{\ell=1}^n \exp\left\{(\boldsymbol{\omega}_0 + r \cdot \boldsymbol{\theta})^\top(\mathbf{z}_\ell \circ b_\ell)\right\}} \\
&= \lim_{r \to \infty} \frac{\exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_j\right\}}{\sum_{\ell \in S_1} \exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_\ell + r \cdot {\boldsymbol{\theta}^{(1)}}^\top(\mathbf{z}_\ell - \mathbf{z}_j)\right\} + \sum_{i \in S_0} \exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_i + r \cdot \boldsymbol{\theta}^\top(\mathbf{z}_i \circ b_i - \mathbf{z}_j \circ b_j)\right\}} \\
&= \lim_{r \to \infty} \frac{\exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_j\right\}}{\sum_{\ell \in S_1} \exp\left\{{\boldsymbol{\omega}_0^{(1)}}^\top \mathbf{z}_\ell + r \cdot {\boldsymbol{\theta}^{(1)}}^\top(\mathbf{z}_\ell - \mathbf{z}_j)\right\} + \sum_{i \in S_0} 0} \quad \text{(from part (1) of Claim 5)} \\
&= f_j^{(1,*)} \quad \text{(from equation (EC.11))}
\end{aligned}$$

□

Having proved Claim 5, it follows that

$$H\left(\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})\right) = \sum_{j \in S_1} c_j f_j^{(1,*)} = C_{(1)}^* = C^*,$$

where the second last equality follows since $\boldsymbol{f}^{(1,*)}$ is an optimal solution to problem (P_1) and the last follows from Claim 4 above. This shows that $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ is the optimal solution to the support finding step, which establishes the result.

Case 2: $\mathbf{C}_{(0)}^* > \mathbf{C}_{(1)}^*$. A symmetric argument from above shows $C^* = C_{(0)}^*$ in this case. In addition, if $\boldsymbol{f}^{(0,*)} = \boldsymbol{f}^{(0)}(\boldsymbol{\omega}_0^{(0)}, \boldsymbol{\theta}^{(0)})$ denotes the optimal solution to problem (P_0), where $\boldsymbol{\omega}_0^{(0)}, \boldsymbol{\theta}^{(0)} \in \mathbb{R}^{D_1}$ are computed

using the procedure in the proof of Theorem 2, then choosing $\boldsymbol{\omega}_0 = \boldsymbol{\omega}_0^{(0)} \circ (0)$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)} \circ (\theta_0)$ with $\theta_0 = -\sqrt{5} \cdot \left\|\boldsymbol{\theta}^{(0)}\right\| \cdot Z_{\max}$, it follows that

$$(1)\ \boldsymbol{\theta}^\top (\mathbf{z}_i \circ b_i - \mathbf{z}_j \circ b_j) > 0\ \forall\, j \in S_1, \forall\, i \in S_0.$$
$$(2)\ f_j(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = 0\ \forall\, j \in S_1.$$
$$(3)\ f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = f_i^{(0,*)}\ \forall\, i \in S_0.$$

and in addition,

$$H\left(\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})\right) = \sum_{i \in S_0} c_i f_i^{(0,*)} = C_{(0)}^* = C^*,$$

so that the optimal solution is $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ which is a boundary type that only considers products in S_0. □

In the general case, when there is more than one binary feature, the above sequence of arguments shows that $C^* = C_{(e^*)}^*$ for some $e^* \in \mathcal{E}$, so that the optimal solution corresponds to a boundary type that only considers products in the subset S_{e^*}. Further, if $\boldsymbol{f}^{(e^*,*)} = \boldsymbol{f}^{(e^*)}(\boldsymbol{\omega}_0^{(e^*)}, \boldsymbol{\theta}^{(e^*)})$ denotes the optimal solution to problem (P_e), then choosing

$$\boldsymbol{\omega}_0 = \boldsymbol{\omega}_0^{(e^*)} \circ \underbrace{(0, 0, \ldots, 0)}_{D_2 \text{ times}} \text{ and } \boldsymbol{\theta} = \boldsymbol{\theta}^{(e^*)} \circ \boldsymbol{\theta}_e\,; \quad \boldsymbol{\theta}_e = \sqrt{5} \cdot \left\|\boldsymbol{\theta}^{(e^*)}\right\| \cdot Z_{\max} \cdot \left(2 \cdot \mathbf{b}_{e^*} - \underbrace{(1, 1, \ldots, 1)}_{D_2 \text{ times}}\right),$$

where $\mathbf{b}_{e^*} \in \{0,1\}^{D_2}$ is the binary feature vector for products in the equivalence class S_{e^*}, it follows that

$$(1)\ \boldsymbol{\theta}^\top (\mathbf{z}_j \circ b_j - \mathbf{z}_i \circ b_i) > 0\ \forall\, j \in S_{e^*}, \forall\, i \in [n] \setminus S_{e^*}.$$
$$(2)\ f_i(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = 0\ \forall\, i \notin S_{e^*}.$$
$$(3)\ f_j(\boldsymbol{\omega}_0, \boldsymbol{\theta}) = f_j^{(e^*,*)}\ \forall\, j \in S_{e^*},$$

which implies that

$$H\left(\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})\right) = \sum_{j \in S_{e^*}} c_j f_j^{(e^*,*)} = C_{(e^*)}^* = C^*,$$

so that the optimal solution to the support finding step is $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ which is a boundary type that only considers products in S_{e^*}.

Appendix B: Details of Numerical Experiments

B.1. Synthetic datasets

For the experiments in Sections 6 and 8—since the goal is to recover the underlying mixing distribution—we employ the standard BFGS solver (Nocedal and Wright 2006) to compute a candidate solution for the support finding step, which was enough to obtain an improving objective value in each iteration. Since the subproblem in the support finding step is nonconvex, we run the BFGS solver from 20 different (randomly chosen) starting values to ensure that we sufficiently explore the parameter space, and choose the solution which obtains the best objective.

Table EC.1 Recovery metrics as function of number of iterations for ground-truth mixing distribution $Q^{(2)}$

			IAE			No. of mixture components		
T	$K_{\max}$	RMISE	Mean	Minimum	Maximum	Mean	Minimum	Maximum
2,000	9	0.11	0.058	0.0418	0.087	8.4	7	9
2,000	16	0.0757	0.0403	0.0283	0.0565	12.6	10	15
2,000	25	0.0688	0.0386	0.0216	0.0541	18.8	15	22
2,000	36	0.066	0.0379	0.025	0.0546	26.2	22	31
2,000	49	0.0662	0.038	0.0244	0.0539	36	31	41
2,000	64	0.0667	0.0384	0.0242	0.0531	48.1	43	55
2,000	81	0.0674	0.0388	0.0252	0.0535	62	55	70
5,000	9	0.1058	0.0547	0.0442	0.088	8.3	7	9
5,000	16	0.0718	0.0371	0.0255	0.0471	12.4	10	16
5,000	25	0.0616	0.0342	0.0259	0.0493	18.6	16	21
5,000	36	0.0534	0.0304	0.0208	0.05	26.4	22	30
5,000	49	0.0519	0.0298	0.0188	0.0515	35.8	31	41
5,000	64	0.0521	0.03	0.0179	0.0517	48.1	43	53
5,000	81	0.0526	0.0303	0.0189	0.0515	62.3	56	68
10,000	9	0.0927	0.0482	0.0433	0.0852	8.1	7	9
10,000	16	0.0716	0.0359	0.0244	0.042	12.2	10	15
10,000	25	0.0592	0.0315	0.0207	0.041	18.5	16	22
10,000	36	0.0476	0.0266	0.0197	0.0343	25.9	21	30
10,000	49	0.0418	0.0243	0.0178	0.0316	35.9	32	41
10,000	64	0.0403	0.0238	0.0171	0.0303	47.4	40	53
10,000	81	0.04	0.0237	0.0177	0.0305	61.4	54	67

Here, we report a more comprehensive set of results for the experiments in Section 6. Apart from the RMISE metric that was defined in the main text, we also compute the IAE metric for each replication as follows:

$$\text{IAE}_r = \frac{1}{V}\sum_{v=1}^{V}\left|\hat{F}_r(\boldsymbol{\beta}_v) - F_0(\boldsymbol{\beta}_v)\right|$$

Then, in addition to the MIAE (mean IAE), we also compute the minimum: $\min_r \text{IAE}_r$ and maximum: $\max_r \text{IAE}_r$ across the $R = 50$ replications. Since the number of iterations only provides an upper bound on the number of mixture components recovered by the CG-based estimator, we also compute the minimum, mean and maximum number of mixture components in the recovered mixing distribution across the different replications. Tables EC.1, EC.2 and EC.3 reports these metrics as a function of the number of iterations of the CG algorithm—these tables are similar in structure to Tables 1 & 2 in Fox et al. (2011). Comparing with the metrics reported by Fox et al. we see that our method outperforms their estimator as the ground-truth distribution becomes more complex, i.e. for mixing distributions $Q^{(4)}$ and $Q^{(6)}$. For $Q^{(2)}$, our estimator performs worse probably because of overfitting—as the number of periods T increases, our estimates improve and the performance gap decreases .

Additional results. We also report the performance of our estimator for two additional ground-truth mixing distributions: (a) a bivariate normal $Q^{(1)} = \mathcal{N}([3,-1], \boldsymbol{\Sigma}_1)$, and (b) LC-MNL model with $K = 6$ classes, say $\mathcal{D}^{(6)}$, having proportions $\alpha_1 = 0.1, \alpha_2 = 0.2, \alpha_3 = 0.2, \alpha_4 = 0.1, \alpha_5 = 0.3, \alpha_6 = 0.1$ and logit parameters $\boldsymbol{\omega}_1 = [3,0], \boldsymbol{\omega}_2 = [0,3], \boldsymbol{\omega}_3 = [1,-1], \boldsymbol{\omega}_4 = [-1,1], \boldsymbol{\omega}_5 = [2,1], \boldsymbol{\omega}_6 = [1,2]$. For $Q^{(1)}$, we compare our performance

Table EC.2 Recovery metrics as function of number of iterations for ground-truth mixing distribution $Q^{(4)}$

			IAE			No. of mixture components		
T	$K_{\max}$	RMISE	Mean	Minimum	Maximum	Mean	Minimum	Maximum
2,000	9	0.1411	0.0716	0.043	0.1134	8.9	8	9
2,000	16	0.0755	0.0412	0.0262	0.0582	14.2	11	16
2,000	25	0.067	0.0365	0.0207	0.0594	20.4	17	24
2,000	36	0.0656	0.0361	0.0224	0.0581	27.7	24	32
2,000	49	0.0662	0.0363	0.0228	0.0589	37.2	32	42
2,000	64	0.067	0.0366	0.0223	0.0585	48.5	41	55
2,000	81	0.0674	0.0369	0.022	0.0585	61.8	54	69
5,000	9	0.118	0.0614	0.0411	0.0877	9	8	9
5,000	16	0.0701	0.0383	0.0301	0.0488	14.4	12	16
5,000	25	0.0568	0.0316	0.022	0.0422	20.3	16	24
5,000	36	0.0523	0.0295	0.0181	0.0461	28	24	34
5,000	49	0.0509	0.0288	0.0161	0.0454	37.6	31	43
5,000	64	0.0508	0.0288	0.0145	0.0457	49.1	44	56
5,000	81	0.051	0.0289	0.0145	0.0444	62.9	58	71
10,000	9	0.1174	0.0621	0.0464	0.0895	9	8	9
10,000	16	0.0687	0.0359	0.0258	0.0522	13.9	12	16
10,000	25	0.0543	0.0283	0.0199	0.0386	20.1	17	23
10,000	36	0.0479	0.0255	0.0194	0.0339	28	24	32
10,000	49	0.0436	0.0237	0.0173	0.0317	38.2	34	44
10,000	64	0.0427	0.0233	0.0173	0.0328	49.6	45	57
10,000	81	0.0422	0.0232	0.0179	0.0333	63.4	58	68

Table EC.3 Recovery metrics as function of number of iterations for ground-truth mixing distribution $Q^{(6)}$

			IAE			No. of mixture components		
T	$K_{\max}$	RMISE	Mean	Minimum	Maximum	Mean	Minimum	Maximum
2,000	9	0.1443	0.07	0.0524	0.109	8.9	8	9
2,000	16	0.0734	0.0389	0.0236	0.0584	14.7	12	16
2,000	25	0.0661	0.035	0.0211	0.0522	20.8	17	23
2,000	36	0.0641	0.0342	0.0207	0.053	28.2	24	32
2,000	49	0.065	0.0347	0.021	0.054	37.8	32	42
2,000	64	0.0657	0.035	0.0218	0.0557	49.9	43	56
2,000	81	0.0662	0.0354	0.022	0.0562	63.7	57	70
5,000	9	0.1402	0.068	0.052	0.0928	9	8	9
5,000	16	0.0704	0.0365	0.0234	0.0455	14.7	12	16
5,000	25	0.0587	0.0299	0.0225	0.0389	21.1	18	24
5,000	36	0.0537	0.0276	0.0187	0.0355	28.9	24	32
5,000	49	0.0529	0.0272	0.0153	0.0368	38.2	33	42
5,000	64	0.053	0.0274	0.015	0.0356	50	44	55
5,000	81	0.0533	0.0276	0.0158	0.0354	63.7	57	70
10,000	9	0.1386	0.0673	0.0517	0.082	9	8	9
10,000	16	0.0691	0.0354	0.025	0.0456	14.3	12	16
10,000	25	0.052	0.0262	0.0187	0.0345	20.6	16	24
10,000	36	0.0434	0.0227	0.0164	0.0305	28.3	23	32
10,000	49	0.0401	0.0213	0.0149	0.027	38.4	31	42
10,000	64	0.0397	0.0211	0.0145	0.0268	50.4	42	54
10,000	81	0.0396	0.0211	0.015	0.0266	64.1	53	70

Table EC.4 Error metrics as a function of the number of periods T

	RMISE				MIAE			
	$Q^{(1)}$		$\mathcal{D}^{(6)}$		$Q^{(1)}$		$\mathcal{D}^{(6)}$	
T	Normal	NP-CG	LC-MNL (EM)	NP-CG	Normal	NP-CG	LC-MNL (EM)	NP-CG
2,000	0.044	0.082	0.064	0.086	0.013	0.033	0.025	0.042
5,000	0.031	0.065	0.063	0.079	0.009	0.025	0.024	0.038
10,000	0.028	0.053	0.062	0.077	0.008	0.020	0.024	0.037

$Q^{(1)}$ and $\mathcal{D}^{(6)}$ refer to the bivariate normal and the LC-MNL ground-truth mixing distributions respectively. "Normal", "NP-CG" and "LC-MNL (EM)" correspond respectively to the RPL model with a bivariate mixing distribution, our nonparametric CG-based estimator, and the LC-MNL model estimated using the EM algorithm.

with the RPL model, whereas for the $\mathcal{D}^{(6)}$ ground-truth, the benchmark is an LC-MNL model with $K=6$ classes fit using the EM algorithm—note that the benchmark is provided the knowledge of the true number of classes.

Table EC.4 reports the RMISE and MIAE metrics (defined in the main text) as a function of the number of periods T. We first discuss the results for the ground-truth $Q^{(1)}$. Since there is no model misspecification, the RPL model performs very well and also recovers better approximations of the ground-truth as the number of samples, T, increases. In particular, for $T=10,000$ periods, the average (across all replications) mean vector obtained by the RPL estimator was $[2.95,-0.89]$, pretty close to the true mean $[3,-1]$. Our method performs worse but is able to improve its estimate of the ground-truth as T increases, and becomes closer to the RPL estimator. Considering the support of the recovered mixing distribution within the rectangle $[-6,6]\times[-6,6]$—over which the RMISE and MIAE metrics are computed—the average mean vector (for $T=10,000$ periods) obtained by our method was $[2.61,-0.86]$.

For the $\mathcal{D}^{(6)}$ ground-truth, the LC-MNL model estimated using the EM algorithm outperforms our proposed estimator. Again, this is expected since the benchmark knows the true number of latent classes in the underlying mixing distribution. Since our method does not have this information a priori, we let the CG algorithm run for $K_{\max}=16$ iterations and use the mixture model so obtained to compute the metrics. The average support size in the mixing distribution recovered by our method was 12.6. Under this scenario, the ground-truth mean vector is $[1.1,1.0]$ and the mean vectors obtained (for $T=10,000$ periods) using the EM and CG algorithms are $[0.99,0.83]$ and $[0.91,0.76]$ respectively. Here again, when computing the mean vector for our method, we only focus on the support within $[-6,6]\times[-6,6]$.

B.2. SUSHI dataset

As described in the main text, there were two kinds of sushi varieties—maki and non-maki, represented using a single binary feature. Let S_{maki} and $S_{\text{non-maki}}$ refer to the two kinds of sushi varieties so that $[n]=S_{\text{maki}}\cup S_{\text{non-maki}}$. Let $\mathbf{z}_i\in\mathbb{R}^4$ denote the remaining (non-binary) features for each sushi variety $i\in[n]$ and let $\mathcal{Z}_{\text{maki}}$ and $\mathcal{Z}_{\text{non-maki}}$ denote the convex polytope w.r.t. to these features for both sushi types respectively (similar to the definition $\mathcal{Z}_n$ in the main text). Finally, let $\mathcal{J}_{\text{maki}}$ and $\mathcal{J}_{\text{non-maki}}$ denote the extreme points of the polytopes $\mathcal{Z}_{\text{maki}}$ and $\mathcal{Z}_{\text{non-maki}}$.

We take inspiration from the results (in particular the proofs) of Theorems 3 and 5 to come up with the heuristic approach in Algorithm 3 for solving the support finding step (refer to equation (11) in the main

Table EC.5 Product features used from the SUSHI dataset.

Feature	Type	Range
Style	Binary	0 (maki) or 1 (otherwise)
Oiliness in taste	Continuous	[0, 4] (0: most oily)
Frequency sold in shop	Continuous	[0, 1] (1: most frequent)
Frequency of consumption	Continuous	[0, 3] (3: most frequent)
Normalized price	Continuous	[1, 5]

text). In particular, the types $\boldsymbol{f}^{\text{maki}}$ and $\boldsymbol{f}^{\text{non-maki}}$ are constructed according to the arguments in the proof of Theorem 5 above.

Algorithm 3 Solving the support finding step for the SUSHI dataset

1: $C_{\text{maki,ext}} \leftarrow \max_{j \in \mathcal{J}_{\text{maki}}} c_j$; $C_{\text{non-maki,ext}} \leftarrow \max_{j \in \mathcal{J}_{\text{non-maki}}} c_j$

2: Let $C_{\text{maki,BFGS}}, \boldsymbol{\omega}_{\text{maki,BFGS}}$ be the best objective and corresponding solution of the following subproblem as returned by the standard BFGS solver

$$\max_{\boldsymbol{\omega} \in \mathbb{R}^4} \sum_{i \in S_{\text{maki}}} c_i \cdot \left(\frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_i\right)}{\sum_{j \in S_{\text{maki}}} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right)} \right)$$

Similarly, compute $C_{\text{non-maki,BFGS}}, \boldsymbol{\omega}_{\text{non-maki,BFGS}}$.

3: $C_{\text{maki}} \leftarrow \max\left(C_{\text{maki,BFGS}}, C_{\text{maki,ext}}\right)$ and $C_{\text{non-maki}} \leftarrow \max\left(C_{\text{non-maki,BFGS}}, C_{\text{non-maki,ext}}\right)$

4: If $C_{\text{maki}} \geq C_{\text{non-maki}}$, then output type $\boldsymbol{f}^{\text{maki}}$ that only considers maki sushi varieties, otherwise output type $\boldsymbol{f}^{\text{non-maki}}$ that only considers non-maki sushi varieties

Figure EC.1 plots a heatmap of the choice probabilities for the sushi varieties under the recovered types for both the EM and CG estimators. As can be seen, the types recovered by EM are very similar to each other. The CG estimator, on the other hand, recovers types that are very distinct in terms of their choice probabilities. In particular, Types 2 and 4 only consider non-maki style sushi varieties, Types 5-8 and 10 consider only a single non-maki variety whereas Type 9 only considers the maki variety with the largest market share (see Figure caption for more details).

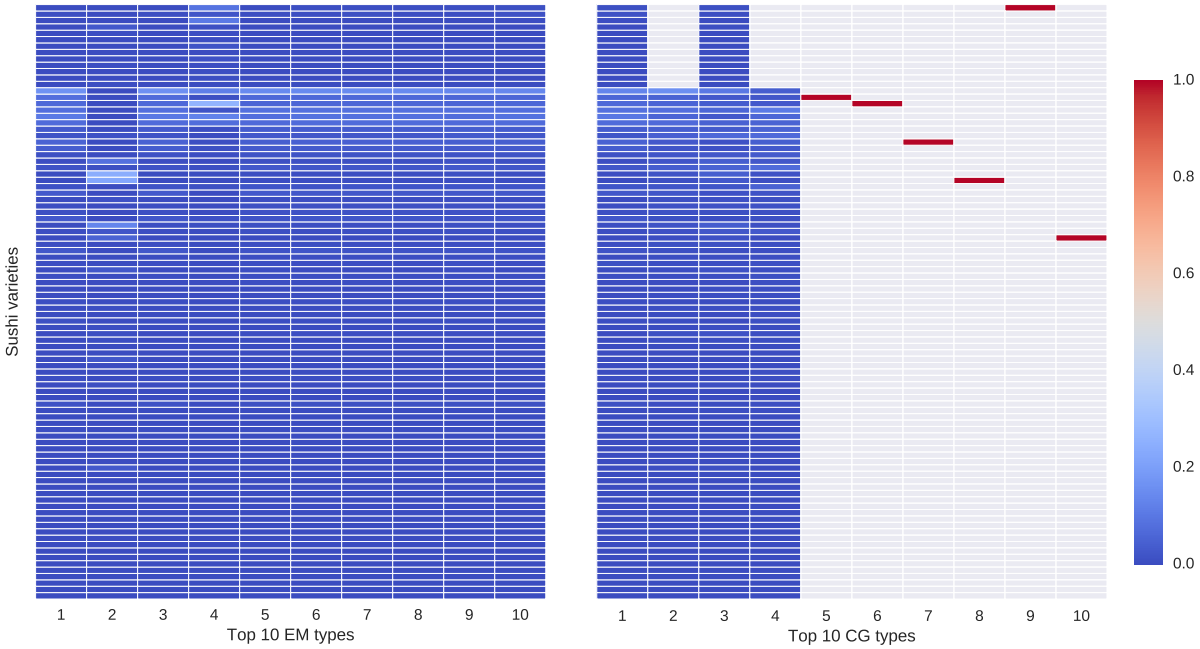
B.3. IRI dataset

Table EC.6 Statistics for IRI Academic Dataset

Product category	# Transactions	# Vendors	# Offer-sets
Shampoo	235K	168	2,464
Toothbrush	163K	122	2,462
Household cleaner	236K	217	2,470
Yogurt	544K	90	2,470
Coffee	374K	290	2,470

We consider transactions in the first two weeks of the year 2011 which had a total of 1,272 stores

Figure EC.1 Heatmap of choice probabilities for each sushi variety under customer types recovered by EM (left) and CG (right)



Note. Each row corresponds to a sushi variety and each column corresponds to a customer type—for both EM and CG we choose the 10 largest types (in terms of proportions). The top 13 rows correspond to maki style sushi varieties (sorted in decreasing order of empirical market shares) and the remaining 80 rows correspond to non-maki style sushi varieties (again sorted in decreasing order of market shares). The cells depict the probability of the corresponding sushi variety being chosen by the corresponding customer type; the cells in grey correspond to sushi varieties that are not part of the consideration set, and therefore are never chosen.

Since we had more than one offer-set (with varying prices), we could not utilize the heuristic approach outlined in Algorithm 3 above for solving the support finding step. So instead, we just used the BFGS solver to determine an approximate solution. The choice probabilities $\boldsymbol{f}^{\text{BFGS}}$ obtained using the BFGS solver contained entries which were very "small" ($< 10^{-5}$), indicating the (possible) presence of boundary types. Motivated by this, we considered the heuristic approach outlined in Algorithm 4 to determine whether a recovered type was a boundary type (which we base on the proof of Theorem 2 above). In particular, let $\mathbf{z}_i = \mathbf{e}_i \circ p_i$ denote the feature vector for product $i \in [n]$, where $e_{ij} = [\![j = i]\!]$ and $[\![\cdot]\!]$ is the indicator function, and p_i is the price of product i. In the algorithm, similar to the proof earlier, we assume that the universe of products is expanded so that the same product with different prices in two offer-sets is indexed as two distinct products.

We used Gurobi Optimizer version 6.5.1 to solve the LP in Step 5 of Algorithm 4. Following the above procedure, we recovered boundary types in the mixing distribution for each of the product categories. The consideration sets for the recovered boundary types were of two kinds: (a) the type never considers a particular product (or a subset of the products) and (b) the type only considers a single product (or subset of the products). Some examples of the $\boldsymbol{\theta}$ parameters (refer to Theorem 2 and the subsequent discussion) of the recovered boundary types include:

Algorithm 4 Solving the support finding step for the IRI dataset

1: $\boldsymbol{f}^{\text{BFGS}}, \boldsymbol{\omega}_{\text{BFGS}} \leftarrow$ choice probabilities and logit parameter vector returned by BFGS solver

2: For each offer-set S_t, let $i_t \leftarrow \arg\max_{j \in S_t} f_{jt}^{\text{BFGS}}$

3: For each offer-set S_t, let $\text{Pairs}_t \leftarrow \left\{(i_t, j) \mid j \in S_t \setminus \{i_t\} \text{ and } \log\left(\frac{f_{i_t t}^{\text{BFGS}}}{f_{jt}^{\text{BFGS}}}\right) < 10^5\right\}$ and similarly $\overline{\text{Pairs}}_t \leftarrow \left\{(i_t, j) \mid j \in S_t \setminus \{i_t\} \text{ and } \log\left(\frac{f_{i_t t}^{\text{BFGS}}}{f_{jt}^{\text{BFGS}}}\right) \geq 10^5\right\}$

4: Let $\text{Pairs} \leftarrow \cup_{t=1}^{T} \text{Pairs}_t$ and $\overline{\text{Pairs}} \leftarrow \cup_{t=1}^{T} \overline{\text{Pairs}}_t$

5: Let $\boldsymbol{\theta}$ (normalized to unit norm) be the solution of the following linear program (LP):

$$\max_{\boldsymbol{\omega} \in \mathbb{R}^{11}} \sum_{(\bar{i},\bar{j}) \in \overline{\text{Pairs}}} \boldsymbol{\omega}^\top (\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}})$$
$$\text{s.t. } \boldsymbol{\omega}^\top (\mathbf{z}_i - \mathbf{z}_j) = 0 \ \forall\, (i,j) \in \text{Pairs}; \text{ and } \boldsymbol{\omega}^\top (\mathbf{z}_{\bar{i}} - \mathbf{z}_{\bar{j}}) \geq 0 \ \forall\, (\bar{i},\bar{j}) \in \overline{\text{Pairs}}$$

6: Let $\boldsymbol{\omega}_0 \leftarrow \boldsymbol{\omega}_{\text{BFGS}} - \|\boldsymbol{\omega}_{\text{BFGS}}\| \cdot \boldsymbol{\theta}$

7: Compute $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$ as the limiting choice probabilities defined in Theorem 2

8: If $\left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta}) - \mathbf{g}^{(k-1)} \right\rangle < \left\langle \nabla \mathsf{loss}(\mathbf{g}^{(k-1)}), \boldsymbol{f}^{\text{BFGS}} - \mathbf{g}^{(k-1)} \right\rangle$, then output boundary type $\boldsymbol{f}(\boldsymbol{\omega}_0, \boldsymbol{\theta})$, otherwise output non-boundary type $\boldsymbol{f}^{\text{BFGS}}$

Table EC.7 Average number of boundary types (out of 10 types) recovered in IRI dataset

Product category	SQ loss	NLL loss
Shampoo	5.0	3.0
Toothbrush	4.0	2.0
Household Cleaner	2.0	2.0
Yogurt	5.0	3.0
Coffee	3.0	2.0

1. $[0., 0., 0., -1., 0., 0., 0., 0., 0., 0., 0.]$, which means that product 4 will never be considered by this type (as long as there is another product in the offer-set).

2. $[0, 0., 0., 0.5, 0.5, 0.5, 0., 0., 0., 0.5, 0.]$, which means that products in the set $\{4, 5, 6, 10\}$ are strictly preferred to all other products, and the type will only choose amongst them (provided at least one of them is in the offer-set).

3. $[0., 0., -0.707, 0., -0.707, 0., 0., 0., 0., 0., 0.]$ which means that product 3 and/or 5 will never be considered by the type (as long as there is some other product in the offer-set).

Table EC.7 reports the number of boundary types recovered via the procedure outlined above for each of the product categories and for both loss functions.

Appendix C: Background on Conditional Gradient (aka Frank-Wolfe) algorithm

The conditional gradient algorithm is used for solving optimization problems of the form $\min_{\mathbf{x} \in \mathcal{D}} h(\mathbf{x})$ where $h(\cdot)$ is a differentiable convex function and $\mathcal{D}$ is a compact convex region in the Euclidean space. Starting

from an arbitrary feasible point $\mathbf{x}^{(0)} \in \mathcal{D}$, the algorithm finds a *descent direction* by solving the following subproblem in each iteration $k \geq 1$:

$$\mathbf{v}^{(k)} \in \arg\min_{\mathbf{v} \in \mathcal{D}} \left\langle \nabla h(\mathbf{x}^{(k-1)}), \mathbf{v} - \mathbf{x}^{(k-1)} \right\rangle, \tag{EC.12}$$

where $\mathbf{x}^{(k-1)}$ is the current solution, $\nabla h(\mathbf{x}^{(k-1)})$ is the gradient of h at the point $\mathbf{x}^{(k-1)}$, and $\langle \cdot, \cdot \rangle$ is the standard inner product in Euclidean space. Then, it updates the solution by taking a convex step in the direction $\mathbf{v}^{(k)} - \mathbf{x}^{(k-1)}$, i.e. $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \gamma^{(k)} \cdot (\mathbf{v}^{(k)} - \mathbf{x}^{(k-1)})$ for some step size $\gamma^{(k)} \in [0, 1]$. The direction $\mathbf{v}^{(k)} - \mathbf{x}^{(k-1)}$ is a descent direction, since it can be shown that for a suitable choice of $\gamma^{(k)}$, we have $h(\mathbf{x}^{(k)}) < h(\mathbf{x}^{(k-1)})$ so that moving in the direction of $\mathbf{v}^{(k)}$ ensures an improving solution.[20] In the classical Frank-Wolfe algorithm, the step size was fixed as $\gamma^{(k)} = 2/(k+2)$. A popular alternative is to do a line-search for the optimal step size in each iteration, i.e. choose

$$\gamma^{(k)} = \arg\min_{\gamma \in [0,1]} h\left(\mathbf{x}^{(k-1)} + \gamma \cdot (\mathbf{v}^{(k)} - \mathbf{x}^{(k-1)})\right)$$

The new iterate remains in the feasible domain (since $\mathcal{D}$ is convex) and therefore expensive projection operators (such as those employed in projected/proximal gradient methods) are not required. The Frank-Wolfe algorithm is particularly attractive when solving the subproblems in (EC.12) is "easy"—for instance, if $\mathcal{D}$ is a polyhedron, then (EC.12) is just an LP. For more details, refer to Jaggi's excellent thesis (Jaggi 2011).

Appendix D: Hardness of solving the support finding step

The support finding step can be hard to solve, even when there is only a single offer-set, $S_1 = [n]$. To observe this, first note that for each iteration k, the subproblem is of the following form:

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^D} \sum_{j=1}^{n} c_j \frac{\exp(\boldsymbol{\omega}^\top \mathbf{z}_j)}{\sum_{\ell=1}^{n} \exp(\boldsymbol{\omega}^\top \mathbf{z}_\ell)}, \tag{EC.13}$$

where we have dropped the explicit dependence of the coefficients $\left\{c_j^{(k)}\right\}_{j \in [n]}$ on the iteration number k for succinct notation. To discuss the hardness of solving (EC.13), we consider the following decision variant of the problem—given some $\lambda \in \mathbb{R}$, is

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^D} \sum_{j=1}^{n} c_j \frac{\exp(\boldsymbol{\omega}^\top \mathbf{z}_j)}{\sum_{\ell=1}^{n} \exp(\boldsymbol{\omega}^\top \mathbf{z}_\ell)} \stackrel{?}{\geq} \lambda. \tag{EC.14}$$

Note that if we can (efficiently) solve the above decision problem, then the optimal solution to (EC.13) can be determined through a binary search for the best value of λ in the interval $[c_{\min}, c_{\max}]$, where $c_{\min} = \min_{j \in [n]} c_j$ and $c_{\max} = \max_{j \in [n]} c_j$. This is because the objective function in (EC.13) is a convex combination of the coefficients $\{c_j\}_{j \in [n]}$, and therefore, the optimal value must belong to the interval $[c_{\min}, c_{\max}]$.

Given this, consider the following:

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^D} \sum_{j=1}^{n} c_j \frac{\exp(\boldsymbol{\omega}^\top \mathbf{z}_j)}{\sum_{\ell=1}^{n} \exp(\boldsymbol{\omega}^\top \mathbf{z}_\ell)} \geq \lambda \iff \sum_{j=1}^{n} c_j \frac{\exp(\boldsymbol{\omega}^\top \mathbf{z}_j)}{\sum_{\ell=1}^{n} \exp(\boldsymbol{\omega}^\top \mathbf{z}_\ell)} \geq \lambda \ \ \forall\, \boldsymbol{\omega} \in \mathbb{R}^D$$

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

$$\begin{aligned}
&\iff \sum_{j=1}^{n} c_j \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right) \geq \lambda \cdot \left(\sum_{\ell=1}^{n} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_\ell\right)\right) \quad \forall\, \boldsymbol{\omega} \in \mathbb{R}^D \\
&\iff \sum_{j=1}^{n} (c_j - \lambda) \cdot \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right) \geq 0 \quad \forall\, \boldsymbol{\omega} \in \mathbb{R}^D \\
&\iff G(\boldsymbol{\omega}) \geq 0 \quad \forall\, \boldsymbol{\omega} \in \mathbb{R}^D
\end{aligned}$$

where $G(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \sum_{j=1}^{n} (c_j - \lambda) \cdot \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_j\right)$. In other words, solving the decision problem (EC.14) is equivalent to certifying the global nonnegativity of the function $G(\boldsymbol{\omega})$. Since not all the coefficients $\{c_j - \lambda\}_{j \in [n]}$ are non-negative for any $\lambda \in (c_{\min}, c_{\max}]$, $G(\boldsymbol{\omega})$ is a *signomial* (Boyd et al. 2007). In particular it is non-convex, and certifying the global nonnegativity of signomials, or equivalently globally minimizing signomials, is computationally intractable in general (Chandrasekaran and Shah 2016).

Nevertheless, Chandrasekaran and Shah (2016) provide efficiently computable (via solving convex programs) certificates for certain classes of globally nonnegative signomials, which can be leveraged in designing a binary search procedure, as discussed above, for solving the support finding step.

Appendix E: Application of our method to panel data

In this section, we show how our CG-based estimator can also be applied to panel data. As much as possible, we reuse notation from the aggregate data setting.

Suppose we have a population of customers, indexed $t = 1, 2, \ldots, T$. For each customer t, we observe $N_t > 0$ choices over the universe of n products. Let $S_{tm} \subseteq [n]$ and $\mathrm{y}_{tm} \in S_{tm}$ denote respectively the subset of products offered to customer t and his/her chosen product, in choice situation $m \in [N_t]$. To allow the dependence of product features on customer characteristics, we denote $\mathbf{z}_{jtm} \in \mathbb{R}^D$ as the feature vector of product $j \in S_{tm}$. We summarize all the choice observations as $\mathsf{Data} \stackrel{\text{def}}{=} \{(\mathrm{y}_{tm} \colon m \in [N_t]) \mid t \in [T]\}$.

We assume that each customer t makes choices according to a logit model of the following form:

$$f_{tj,S}(\boldsymbol{\omega}) = \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{jtS}\right)}{\sum_{\ell \in S} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{\ell tS}\right)}$$

where $\mathbf{z}_{\ell tS}$ is the feature vector of product ℓ when offered to customer t as part of offer-set S, and $\boldsymbol{\omega}$ is the parameter vector. The population of customers is described by a mixture of logit models, where each customer samples a vector $\boldsymbol{\omega}$ according to some distribution Q (over the parameter space $\mathbb{R}^D$) and then makes *all* choices according to the logit model with parameter vector $\boldsymbol{\omega}$.

For each customer t, let $F_t(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \prod_{m=1}^{N_t} f_{t\mathrm{y}_{tm},m}(\boldsymbol{\omega})$ denote the probability of observing the choices $(\mathrm{y}_{t1}, \mathrm{y}_{t2}, \ldots, \mathrm{y}_{tN_t})$ under a logit model with parameter $\boldsymbol{\omega}$, where for brevity of notation, we let $f_{t\mathrm{y}_{tm},m}(\boldsymbol{\omega})$ denote $f_{t\mathrm{y}_{tm},S_{tm}}(\boldsymbol{\omega})$. Then, define the mapping $\mathrm{g}_t \colon \mathcal{Q} \to [0,1]$ as

$$\mathrm{g}_t(Q) = \int F_t(\boldsymbol{\omega})\, \mathrm{d}Q(\boldsymbol{\omega}),$$

and let $\mathbf{g} \colon \mathcal{Q} \to [0,1]^T$ denote the vector-valued mapping, defined as $\mathbf{g}(Q) = (\mathrm{g}_t(Q) \colon t \in [T])$. For the panel data setting, $\mathbf{g}(Q)$ represents the mixture likelihood vector under mixing distribution Q.

With the above notation, the negative log-likelihood of observing the choice data under mixing distribution Q is given by:

$$\mathsf{NLL}(\mathbf{g}(Q)) = -\frac{1}{T} \sum_{t=1}^{T} \log\left(\mathrm{g}_t(Q)\right)$$

and our goal is to find the mixing distribution that minimizes the negative log-likelihood:

$$\min_{Q \in \mathcal{Q}} \ \mathsf{NLL}(\mathbf{g}(Q)) \tag{EC.15}$$

Analogous to the aggregate data setting, we show how problem (EC.15) can be formulated as a constrained convex program. Define $\boldsymbol{F}(\boldsymbol{\omega}) \stackrel{\text{def}}{=} (F_t(\boldsymbol{\omega}) \colon t \in [T])$ as the atomic likelihood vector, and let $\mathcal{P} \stackrel{\text{def}}{=} \left\{ \boldsymbol{F}(\boldsymbol{\omega}) \colon \boldsymbol{\omega} \in \mathbb{R}^D \right\}$ be the set of all possible atomic likelihood vectors. Using similar arguments as in Section 3.2, it can be shown that $\text{conv}(\overline{\mathcal{P}})$ is a compact convex set and $\{\mathbf{g}(Q) \colon Q \in \mathcal{Q}\} = \text{conv}(\overline{\mathcal{P}})$. Then, following the sequence of transformations from Section 3.2, it follows that instead of solving problem (EC.15), we can equivalently solve:

$$\min_{\mathbf{g} \in \text{conv}(\overline{\mathcal{P}})} \mathsf{NLL}(\mathbf{g}),$$

which is a convex program with a compact convex constraint set in $\mathbb{R}^T$. The conditional gradient algorithm can again be used to solve the above program, where now the support finding step at iteration k is of the form:

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^D} -\frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{\mathrm{g}_t^{(k-1)}} \right) \cdot \left(\prod_{m=1}^{N_t} \frac{\exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{jtm}\right)}{\sum_{\ell \in S_{tm}} \exp\left(\boldsymbol{\omega}^\top \mathbf{z}_{\ell tm}\right)} \right)$$

Our theoretical results do not extend to the above subproblem, which can be much harder to solve than in the aggregate data setting. However, one can still use an off-the-shelf solver like BFGS to determine an approximate solution, and check whether it generates an improving solution for the outer convex program.

In the panel data setting, one can think of different ways of defining the squared loss function. One possibility is to use the following definition:

$$\mathsf{SQ}(\mathbf{g}(Q)) = \frac{1}{2 \cdot T} \sum_{t=1}^{T} \left(1 - \mathrm{g}_t(Q)\right)^2,$$

for which the conditional gradient algorithm is directly applicable, since the loss function is convex in the mixture likelihood vector $\mathbf{g}(Q)$. Another definition that has been suggested in the literature (see Section III in Bajari et al. 2007) sums over all possible choice sequences, for each customer. Under this definition, the estimation problem can still be formulated as a convex program and the conditional gradient algorithm can be used to recover the mixing distribution, with the caveat that evaluating the loss function (and its gradient) can be intractable.

References

Bajari P, Fox JT, Ryan SP (2007) Linear regression estimation of discrete choice models with nonparametric distributions of random coefficients. *American Economic Review* 97(2):459–463.

Boyd S, Kim SJ, Vandenberghe L, Hassibi A (2007) A tutorial on geometric programming. *Optimization and engineering* 8(1):67–127.

Chandrasekaran V, Shah P (2016) Relative entropy relaxations for signomial optimization. *SIAM Journal on Optimization* 26(2):1147–1173.

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

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

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. *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 427–435.

Nocedal J, Wright SJ (2006) *Numerical Optimization* (Springer), second edition.