A Quasi-Bayes Approach to Nonparametric Demand Estimation with Economic Constraints

James Brand* Adam N. Smith†

January 17, 2025

Abstract

This paper presents a quasi-Bayes approach to estimating nonparametric demand systems for differentiated products. We transform the GMM objective function developed by Compiani (2022) into a quasi-likelihood, specify priors that penalize violations of micro-founded economic constraints, and develop novel Bayesian inference procedures. We use simulations and retail scanner data from 12 consumer packaged goods categories to show that our quasi-Bayes approach improves both the accuracy of estimated elasticities and the validity of estimated demand functions. Together, our results demonstrate the value of (i) disciplining flexible nonparametric estimators with judicious economic constraints, and (ii) Bayesian methods for accommodating such constraints. Finally, we introduce a new Julia package (`NPDemand.jl`) that implements both GMM and quasi-Bayes approaches to estimation.

Keywords: differentiated products, price elasticities, shape constraints, Bernstein polynomials, nonparametric instrumental variables, Sequential Monte Carlo, counterfactual analysis

*Office of the Chief Economist, Microsoft
†UCL School of Management

1 Introduction

Flexible estimation of consumer demand is a cornerstone of empirical industrial organization (IO) and quantitative marketing. In empirical work, researchers face a trade-off between the flexibility of the specified model, the enforcement of economic constraints, and feasibility of estimation. The norm in these fields has been to estimate mixed logit and probit models which, as most commonly specified, satisfy many economic constraints by construction, but do so at the cost of strong parametric assumptions that restrict the flexibility of the estimated demand curves. The development of alternative models which are more flexible, comply with economic theory, and are easy to implement continues to be an open area of research.

In this paper, we offer a new approach for estimating nonparametric demand (NPD) systems for differentiated products under economic constraints. We build on the work of Compiani (2022), who develops a generalized method of moments (GMM) estimator of inverse demand functions which are approximated via Bernstein polynomials. We target the same class of nonparametric estimands—which are flexible enough to accommodate both weak substitutes and complements—but construct a new "quasi-Bayesian" estimator (Chernozhukov and Hong, 2003). Specifically, we define a quasi-likelihood using a transformation of the GMM objective function in Compiani (2022), specify priors over model parameters, and then apply Bayesian inference procedures. We make the following contributions.

First, we show how priors can be used to tractably enforce a variety of desirable economic constraints. Imposing constraints on the NPD estimation problem is challenging because the target estimands are inverse demand functions. As a result, most economic objects of interest are complex functions of model parameters. One contribution of Compiani (2022) is to translate constraints on the demand function to constraints on the inverse demand function, which can be enforced though linear restrictions on the Bernstein coefficients during estimation. However, while the derived restrictions are *necessary* for the desired constraints to hold everywhere, they are not sufficient. That is, when taken to the data, there is no guarantee that the estimated demand functions satisfy all shape constraints (even in-sample) which can limit the practical value of nonparametrics for policy counterfactuals.

Our approach closes this gap. We impose constraints in our priors through two complementary steps. First, we apply a reparameterization to the Bernstein coefficients to enforce any necessary linear restrictions from Compiani (2022). For example, if there are two parameters (θ_1, θ_2) and a constraint requires that $\theta_1 < \theta_2$, then we reparameterize our model using (θ_1^*, θ_2^*), where $(\theta_1, \theta_2) = (\theta_1^*, \theta_1^* + e^{\theta_2^*})$. Such reparameterizations, when feasible, are common ways of enforcing constraints in Bayesian modeling (Gelman, 2004). Second, we specify dogmatic priors that place zero mass on regions of the parameter space which violate constraints. This step allows us to (i) ensure that constraints on the demand functions are satisfied at all points in the data, and (ii) enforce a wider class of constraints, including those which are nonlinear in model parameters. For example, we extend the linear cross-good monotonicity constraints developed by Compiani (2022) to settings with multiple product groups to allow for within-group substitutes and across-group complements.

Our second contribution is to develop a tractable algorithm for sampling from the quasi-posterior induced by these new priors. Modern probabilistic programming languages offer powerful off-the-shelf samplers for Bayesian models using workhorse Markov chain Monte Carlo (MCMC) algorithms. However, our dogmatic priors that place zero mass on potentially disparate regions of the parameter space pose practical challenges for sampling. If the domain of the target posterior is small relative to the domain of the sampler's proposal distribution, then the sampler will struggle to mix and converge. We overcome this problem by adopting the Sequentially Constrained Monte Carlo (SCMC) algorithm of Golchi and Campbell (2016), which is a novel variant of Sequential Monte Carlo (Del Moral et al., 2006). The key idea of SCMC is to define a sequence of softened priors such that the penalty for violating constraints increases monotonically and converges to the prior placing zero mass outside of the feasible set. SCMC samples from the induced sequence of posteriors through a series of importance sampling, resampling, and jittering steps. In doing so, SCMC replaces a difficult (hard constraint) sampling problem with a sequence of easier (soft constraint) sampling problems that converges to the desired hard constraint problem.

Our implementation proceeds in two steps. In the first step, we use an off-the-shelf MCMC algorithm (e.g., Hamiltonian Monte Carlo) to generate samples from our reparameterized model, which only enforces the necessary linear restrictions. In the second step, we run the SCMC algorithm which iteratively filters and resamples draws from this initial posterior to produce a new set of draws from the fully constrained target posterior. After sampling, conducting inference about model parameters—or any function of model parameters, including price elasticities—is automatic using Monte Carlo methods.

Our third contribution is to provide generalizable evidence that constraints are valuable for NPD estimation. We first conduct Monte Carlo simulations to show that our method is significantly better at constraining the estimated demand function to satisfy economic constraints. These simulations illustrate that linear restrictions alone frequently fail to enforce the desired constraints. Even in our simplest simulations we find that our QB method reliably reduces the fraction of markets with violations of *any* of our constraints to nearly zero, whereas GMM continues to generate violations in more than 15% of markets. In all of the data-generating processes we study, our approach also increases the accuracy of the estimated own- and cross-price elasticities.

We also apply our methods to 12 separate product categories in supermarket scanner data which together span 66 products and more than 2 million market-level price elasticities. In seven of these categories, more than half of all markets violate at least one constraint even when we impose the linear restrictions from Compiani (2022). In our largest categories, *all markets* violate at least one constraint, and at least one own-price elasticity is positive in nearly 75% of markets. Enforcing constraints completely yields estimated price elasticities that adhere to sign constraints. We also demonstrate, using the Butter, Margarine, and Spreads category as an example, that our counterfactual predictions are similarly improved relative to GMM, and that in some cases GMM estimates lie outside of the credible intervals on our counterfactuals. We offer many additional results outlining the distributional differences between GMM and quasi-Bayes estimates, all of

which suggest that both our Bayesian priors and the constraints we impose provide useful and economically important regularization on the estimated demand functions.

Our final contribution is to introduce a new Julia package, `NPDemand.jl`, which implements both the original GMM estimation procedure proposed in Compiani (2022) as well as our quasi-Bayesian estimator with a single function call. Our package relies on `Turing.jl` (Ge et al., 2018), a cutting-edge toolkit for Bayesian estimation in Julia, to map our econometric problem into Julia and to automate the sampling in our first stage. The package also allows users to choose custom samplers and to modify estimation options like burn-in and thinning. After estimation, users can calculate price elasticities of the resulting demand function and counterfactual market shares at alternative prices via additional simple functions. The goal of the package is to distance the user from the underlying complexity of NPD—including the construction and reparameterization of the underlying sieves and Bayesian estimation, as well as the transformation of the resulting parameter estimates into interpretable results—as much as possible. We show some brief examples of the use of this package in Section 4 and Appendix A, and more complete documentation can be found here.

Our results contribute to a number of related strands of the literature. The first is the long-standing microeconometrics literature on testing and imposing economic constraints on flexible functional forms (Deaton and Muellbauer, 1980; Matzkin, 1994; Lewbel, 1995; Haag et al., 2009; Hoderlein and Lewbel, 2012; Blundell et al., 2012, 2017; Mehta, 2015; Romeo, 2016; Farrell et al., 2021; Compiani, 2022; Monardo, 2024; Kong et al., 2024). The second is the development of flexible parametric (Monardo, 2021; Wang, 2023; Fosgerau et al., 2024) and nonparametric (Compiani, 2022; Singh et al., 2023; Wang and Huang, 2024) approaches for estimating consumer substitution patterns using aggregate data. We add to both strands not only through the extensions to Compiani (2022) discussed above, but also through the extensive empirical evidence showing the value of constraints in differentiated product markets. Our analysis also adopts a more prescriptive rather than testing focus, which we believe can make nonparametrics more attractive for counterfactual analysis. Lastly, we contribute to the literature on quasi-Bayesian methods (Kim, 2002; Chernozhukov and Hong, 2003) which have proved useful in estimating structural demand models (Sun and Ishihara, 2019; Hong et al., 2021), latent variable models (Kankanala, 2023), and weakly identified GMM models (Andrews and Mikusheva, 2022). Similar to Gallant et al. (2022), we show how quasi-Bayesian methods can help enforce constraints—albeit in a different modeling setting and with a different operationalization of priors. We also believe our paper to be among the first to apply quasi-Bayes methods at scale and to provide statistical software to support applied work.

The rest of the paper proceeds as follows. In Section 2 we introduce our nonparametric model of consumer demand, the estimator introduced by Compiani (2022), and the practical challenges thereof. In Section 3 we outline our proposed quasi-Bayes approach to estimation, and we introduce our new Julia package in Section 4. Section 5 presents a set of simulation results, and Section 6 presents empirical evidence of the value of our approach in real-world retail demand data—both utilizing our new Julia package. Finally, we conclude in Section 7 with a discussion of our findings and suggestions for future work.

2 Nonparametric Demand

We start by outlining a generic market-level demand model for a fixed assortment of J goods. For each product $j = 1, \ldots, J$ and market $t = 1, \ldots, T$, we observe a market share s_{jt}, a price p_{jt}, and a P-vector of product characteristics $\mathbf{x}_{jt}$. We also assume there is an unobserved product characteristic ξ_{jt} that may be correlated with any of the observable characteristics. We define the demand equation for good j in market t as:

$$s_{jt} = \sigma_j\big(\delta_1(\mathbf{x}_{1t}, p_{1t}, \xi_{1t}), \ldots, \delta_J(\mathbf{x}_{Jt}, p_{Jt}, \xi_{Jt})\big), \tag{1}$$

where each $\delta_j(\cdot)$ is a mapping from the characteristics space into a scalar product index and σ_j is a mapping from the J product indices into shares.

The system of demand in (1) is generic in that we have not yet assumed functional forms for the indices or demand functions. We have, however, made two structural assumptions. The first is that the J-vector of unobservables enters the demand for each product. Such structure allows demand to shift in response to variation in *any* competing product characteristics—a feature that is absent from typical "product-space" demand specifications like log-log or AIDS. The presence of J error terms on the right-hand side of each demand equation is also one of the key econometric challenges when estimating structural market-level demand systems (Berry and Haile, 2021). The second assumption is that all product characteristics, including price, enter demand through the index function $\delta(\cdot)$.

Although not strictly required for identification nor for σ_j to be invertible in δ_j, for the rest of the paper we further assume that δ_j is additively separable in the unobservable demand shifters.

$$\delta_j(\mathbf{x}_{jt}, p_{jt}, \xi_{jt}) = \bar{\delta}_j(\mathbf{x}_{jt}, p_{jt}) + \xi_{jt} \tag{2}$$

Notably, this assumption differs from some common models in the industrial organization literature, including the canonical model from Berry et al. (1995) in which observable product characteristics, but not the unobservable demand shifter ξ, are multiplied by consumer-specific utility coefficients.[1] Under this structure, and an assumption that $s_{jt} > 0$ for all j and t, the system in (1) is invertible (Berry et al., 2013; Berry and Haile, 2014) and we have the following system of *inverse demand* equations.

$$\bar{\delta}_j(\mathbf{x}_{jt}, p_{jt}) = \sigma_j^{-1}(\boldsymbol{s}_t) - \xi_{jt} \tag{3}$$

Note that the inversion lays the foundation for solving the econometric problem of J right-hand side unobservables noted above. Each equation in the inverse demand system in (3) now only contains one error term and estimation can follow via nonparametric IV methods (Newey and Powell, 2003).

[1] While the assumptions of an additive index and characteristics affecting demand only through the index come with significant bite (Berry and Haile, 2021; Compiani, 2022), we believe this is a reasonable starting point for our analysis. If the estimated demand functions are sufficiently noisy and violate constraints *despite* the added structure, then the problem will only get worse as the assumptions are relaxed, which only further motivates our regularizing quasi-Bayes approach.

2.1 Shape Constraints

One of the frequent challenges of nonparametric estimation of economic functions is that the same flexibility which motivates the use of nonparametric estimators also gives rise to noisy estimates which may violate economic theory. Even in simpler parametric models of demand, it is common for researchers to rely on constraints to avoid such violations. For example, Brand (2021) uses constraints to simultaneously enforce downward-sloping demand and nonnegative variances of random coefficients in a mixed logit model. `PyBLP`, the dominant Python package for estimating mixed-logit models with market-level data, allows for general box contraints on model parameters for similar purposes (Conlon and Gortmaker, 2020). Other constraints are imposed directly by the model itself: the logit functional form ensures that cross-price elasticities are positive (as long as the price coefficient is negative) and that own-price effects dominate cross-price effects (hereafter referred to as "diagonal dominance"). Moving to nonparametric estimators makes these constraints both more likely to be violated and more challenging to enforce due to the dimensionality of estimated parameter vectors and the complexity of feasible sets corresponding to the desired constraints.

In our context, some care must be given to the choice of nonparametric estimator $\hat{\sigma}_j^{-1}$ in (3) so that $\hat{\sigma}_j$ satisfies various micro-founded constraints on the Jacobian of demand $\mathbf{J}_p^\sigma$. This motivates a nonparametric framework that can tractably operationalize and propagate constraints from $\hat{\sigma}_j^{-1}$ to $\hat{\sigma}_j$. To this end, we follow Compiani (2022) and use Bernstein polynomials.[2] Specifically, we approximate the inverse demand function using a tensor product of Bernstein basis functions:

$$\hat{\sigma}_j^{-1}(\mathbf{s}_t; \theta_j) = B_d(s_{1t}; \theta_j) \otimes \cdots \otimes B_d(s_{Jt}; \theta_j) \tag{4}$$

where the d-th order basis functions are defined as

$$B_d(s; \theta_j) = \sum_{k=0}^{d} \theta_{jk} b_{kd}(s) \tag{5}$$

and

$$b_{kd}(s) = \binom{d}{k} s^k (1-s)^{d-k}. \tag{6}$$

Bernstein polynomials are attractive because they are amenable to shape constraints (Wang and Ghosh, 2012; Ghosal et al., 2023).[3] For example, if the target function is monotonically increasing, then the Bernstein coefficients θ_{jk} will be ordered over k.

Still, one remaining challenge is that is that we aim to estimate σ_j^{-1}, but the constraints of interest are defined with respect to σ^{-1}. Thus, we must now formulate and enforce constraints on the parameters of $\hat{\sigma}_j^{-1}$ (i.e., θ_{jk}) which, when inverted to calculate (functionals of) σ_j itself, satisfy our constraints. The complex relationship between this inversion and the desired constraints is the fundamental challenge tackled herein.

[2] Although Compiani's approach relies heavily on properties of Bernstein polynomials, the approach we introduce in Section 3 is more general and could be used with essentially any approximation.

[3] Many econometrics results relating to sieves require that d (the order of the polynomial) increases in JT. Throughout this paper, for both estimation and inference, we treat d as fixed.

Table 1: Desired Shape Constraints for Estimation

Constraint	Definition	Linear	Compiani (2022)
1. Own-good monotonicity	$\mathbf{J}_p^\sigma(\theta)_{jj} \leq 0 \; \forall j$	yes	✓
2. Diagonal dominance	$\lvert\mathbf{J}_p^\sigma(\theta)_{jj}\rvert \geq \sum_{k\neq j} \lvert\mathbf{J}_p^\sigma(\theta)_{jk}\rvert$	yes	✓
3. Weak substitutes (all goods)	$\mathbf{J}_p^\sigma(\theta)_{jk} \geq 0 \; \forall j \neq k$	yes	✓
4. Weak substitutes (within groups)	$\mathbf{J}_p^\sigma(\theta)_{jk} \geq 0 \; \forall j \neq k, G_j = G_k$	no	
5. Weak substitutes (across groups)	$\mathbf{J}_p^\sigma(\theta)_{jk} \geq 0 \; \forall j \neq k, G_j \neq G_k$	no	
6. Weak complements (across groups)	$\mathbf{J}_p^\sigma(\theta)_{jk} \leq 0 \; \forall j \neq k, G_j \neq G_k$	no	

Table 1 provides a list of the shape constraints we cover in this paper. The first three—own-good monotonicity, diagonal dominance, and all substitutes—have representations as linear restrictions on the Bernstein coefficients θ_{jk} and are implemented by Compiani (2022). The second three allow for some combination of substitutes and complements within/across groups (e.g., chips and soda), cannot be enforced via linear restrictions, and are novel relative to Compiani (2022).

2.2 Challenges with GMM and Non-Convex Optimization

One key contribution of Compiani (2022) is to map constraints on the Jacobian of σ to constraints on the Jacobian of σ^{-1} and, in turn, to constraints on the Bernstein coefficients themselves. Own and cross-good monotonicity, for example, imply convex constraints on θ that can be easily incorporated into a convex optimization program. Let $\mathcal{C}(\Theta)$ denote the set of parameter values satisfying all desired shape constraints, and let $A\theta \leq 0$ denote the feasible set implied by the linear restrictions developed by Compiani (2022). Let $\mathbf{R}(\theta)$ denote the $J \times T$ matrix of residuals with elements $r_{jt}(\theta) = \bar{\delta}_j(\mathbf{x}_{jt}, p_{jt}) - \hat{\sigma}_j^{-1}(\boldsymbol{s}_t; \theta_j)$ and let $\mathbf{W}$ denote the weighting matrix. Then the GMM problem introduced by Compiani (2022) solves

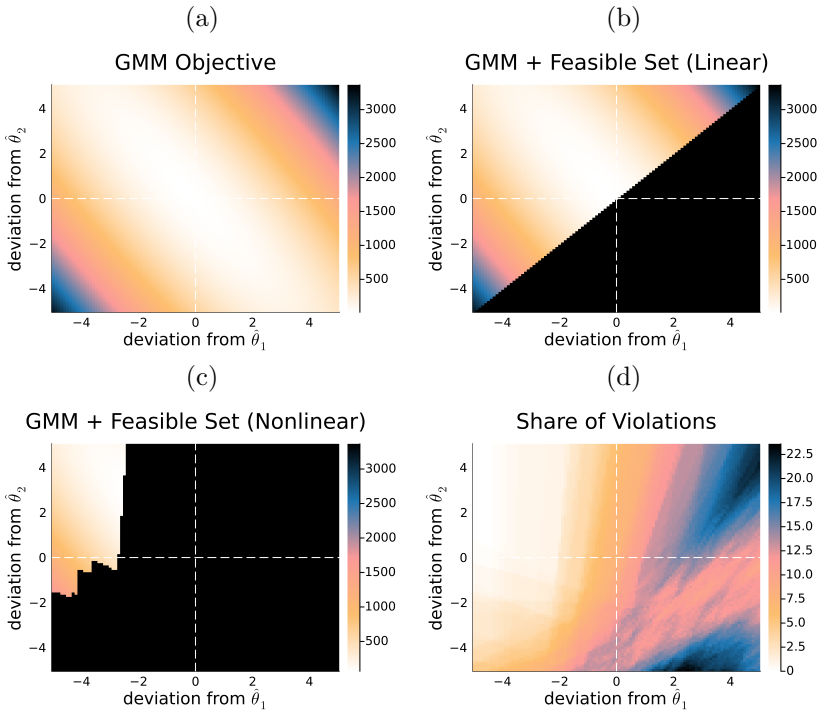
$$\begin{aligned} &\min_\theta \; \mathbf{R}(\theta)'\mathbf{W}\mathbf{R}(\theta) \\ &\text{s.t. } \mathbf{A}\theta \leq 0 \end{aligned} \tag{7}$$

Compiani proves that $\theta \in \mathcal{C}(\Theta) \Rightarrow \mathbf{A}\theta \leq 0$ (i.e., that the linear restrictions are necessary for the desired constraints), but gives no guarantee that $\mathbf{A}\theta \leq 0 \Rightarrow \theta \in \mathcal{C}(\Theta)$. As a result, it is possible that solutions to Equation (7) fail to solve the desired problem

$$\begin{aligned} &\min_\theta \; \mathbf{R}(\theta)'\mathbf{W}\mathbf{R}(\theta) \\ &\text{s.t. } \theta \in \mathcal{C}(\Theta) \end{aligned} \tag{8}$$

A natural option to consider would be to estimate Equation (8) directly using a nonlinear solver. Note, however, that because the constraints are nonconvex, there will be no theoretical guarantees for the convergence of off-the-shelf solvers and local minima may be likely. Practical convergence will depend on the complexity of the desired constraints and on the available data. In our testing, we have found that many high-performance solvers in Julia (e.g., as implemented in NLopt.jl)

Figure 1: Comparison Between Linear and Nonlinear Feasible Sets



Notes: An example of the GMM objective function and both linear and nonlinear feasible sets corresponding to an own-good monotonicty constraint. Data are simulated from a logit model with $J = 2$ goods and $T = 500$ markets. We vary the first two dimensions of the parameter vector while holding the others fixed at the GMM estimate $\hat{\theta}$. The linear feasible set is a superset of the nonlinear feasible set which highlights the limitations of relying on linear restrictions to produce valid nonparametric estimates of demand.

routinely fail to converge when applied to this type of nonconvex problem.

Whether linear restrictions are sufficient for desired constraints is an empirical question, but we can develop some suggestive evidence via simulation. Figure 1 illustrates the potential complexity of $C(\Theta)$ in one simulated example. Here, we generate data from a logit model with $J = 2$ goods and $T = 500$ markets. We then estimate demand nonparametrically using the GMM problem in Equation (7) with linear own-good monotonicity constraints (i.e., downward-sloping demand) introduced by Compiani. The plot on the top left shows the GMM objective value as a function of the first two parameters in our sieve approximation, holding fixed all other parameter values at their linear solution. The plot on the top right shows the same objective function with an overlaid feasible set given by the linear restrictions (requiring $\theta_1 < \theta_2$). The plot on the bottom left overlays the feasible set of the nonlinear constraint requiring the estimated demand function to be downward sloping. We see that the linear feasible set is a superset of the nonlinear feasible set, which implies that there are many values of θ that are admissible under linear restrictions but inadmissible under nonlinear constraints and that this projection of the nonlinear constraints is non-convex, highlighting potential complexities in estimation under these constraints.

Finally, to explore the smoothness of the desired constraints outside of the feasible set, the

bottom right panel of Figure 1 plots the extent of constraint violations on a larger domain. For each pair of parameter values we plot the share of violations of own-good monotonicty across markets. The lightest area in the top left corresponds to the region where constraints are satisfied (highlighted in 1c). We can see here that this surface is highly irregular and many lighter "valleys" can be separated by darker "peaks." Such irregularities will only become more severe when we project back up from two dimensions to the full parameter space and when we impose more complex constraints, posing significant challenges for the optimization of (8) and the practical value of nonparametrics.

3 Quasi-Bayes Methods for Estimation and Inference

In this section we propose a quasi-Bayes approach to estimating nonparametric demand functions which are guaranteed to satisfy potentially complex constraints $C(\Theta)$. The core idea we present is to treat a transformation of the GMM objective function as a "quasi-likelihood," and then to apply Bayesian inference procedures as if this were a standard likelihood function, imposing the constraint that $\theta \in C(\Theta)$ via our priors. Quasi-Bayesian methods (also called "Laplace-type" estimators) were popularized in economics by Chernozhukov and Hong (2003), who demonstrated that modern posterior sampling techniques can alleviate many of the challenges that arise when estimation depends on optimizing complex and irregular target functions.

Quasi-Bayes methods offer several advantages in the context of our problem. First, priors offer a natural way to encode complex constraints on functions of model parameters, such as sign constraints on price effects. Second, and in the spirit of Chernozhukov and Hong (2003), sampling can be a more robust approach to estimation than optimization (Ma et al., 2019). Third, once we generate samples from the target posterior then valid statistical inference about parameters or functionals is automatic, whereas Compiani requires a bootstrap procedure for inference after estimation. Finally, quasi-Bayes procedures arrive at a likelihood without any distributional assumptions on the unobservables, and thus side-step the change-of-variables required to derive a proper likelihood function.

3.1 Quasi-Posterior

Let $\ell_n(\theta) = -\frac{1}{2}\mathbf{R}(\theta)'\mathbf{W}\mathbf{R}(\theta)$ denote a scaled version of the GMM objective function defined in (7) where $n = JT$. Additionally let the transformation $e^{\ell_n(\theta)}$ denote the quasi-likelihood function and $\pi(\theta)$ denote a prior density. Then the quasi-posterior takes the form:

$$\pi(\theta|\text{data}) = \frac{e^{\ell_n(\theta)}\pi(\theta)}{\int e^{\ell_n(\theta)}\pi(\theta)d\theta} \tag{9}$$

and is a valid density over θ.[4] Theorem 1 in Chernozhukov and Hong (2003) establishes a Bernstien-von Mises result for quasi-posteriors, showing that under usual regularity conditions, the quasi-posterior concentrates at rate $1/\sqrt{n}$ around the true parameter value and is approximately Gaussian for large n. As a corollary, estimators based on moments or quantiles of the quasi-posterior also tend to enjoy standard asymptotic guarantees. For example, the quasi-posterior mean

$$\mathbb{E}[\theta|\text{data}] = \int \theta \cdot \frac{e^{\ell_n(\theta)}\pi(\theta)}{\int e^{\ell_n(\theta)}\pi(\theta)d\theta} d\theta \tag{10}$$

is consistent and asymptotically normal (Kim, 2002; Chernozhukov and Hong, 2003). Similar results have also been extended to quasi-posteriors based on nonparametric IV models (Kato, 2013), mixed logit models (Hong et al., 2021), weakly identified GMM models (Andrews and Mikusheva, 2022), and models with constraints (Gallant et al., 2022).

3.2 Imposing Constraints through Reparameterizations and Priors

Our approach to enforcing shape constraints has two features: (1) reparameterizations for linear restrictions, and (2) dogmatic priors for nonlinear constraints. All of the linear restrictions we consider amount to sign and order inequalities on the Bernstein coefficients. Such constraints can be trivially enforced through reparameterizations, as is common in Bayesian models (Gelman, 2004; Pachali et al., 2020; Gallant et al., 2022). For example, in Figure 1 we saw that the linear own-good monotonicty constraint requires $\theta_1 < \theta_2$. To enforce this constraint, we can define a transformation:

$$\begin{pmatrix}\theta_1 \\ \theta_2\end{pmatrix} = g\begin{pmatrix}\theta_1^* \\ \theta_2^*\end{pmatrix} = \begin{pmatrix}\theta_1^* \\ \theta_1^* + e^{\theta_2^*}\end{pmatrix} \tag{11}$$

where θ^* are the new (unconstrained) parameters with quasi-posterior $\pi(\theta^*|\text{data}) \propto e^{\ell_n(g(\theta^*))}\pi(\theta^*)$. Consequently, for any pair $(\theta_1^*, \theta_2^*) \in \mathbb{R}^2$, we are guaranteed to have $\theta_1 < \theta_2$ when evaluating the quasi-likelihood function. While we are using the two-dimensional example for ease of illustration, we can derive the necessary reparameterization for any set of necessary linear restrictions $\mathbf{A}\theta \leq 0$. The value of reparameterizations is that the MCMC sampler will only navigate regions of the parameter space for which the necessary linear restrictions are satisfied.

We know, however, that the linear constraint space is a superset of the parameter space for which nonlinear constraints are satisfied and so a reparameterization alone is not sufficient. To close this gap, we also specify dogmatic priors of the form:

$$\pi(\theta^*) \propto \bar{\pi}(\theta^*)\mathbf{1}_{\mathcal{C}}(\theta). \tag{12}$$

where $\mathbf{1}_{\mathcal{C}}(\theta)$ is an indicator function that only places positive mass on the set $\mathcal{C}(\Theta)$.

[4] The quasi-posterior in (9) bears resemblance to Gibbs posteriors (Zhang, 2006a,b) and the generalized Bayesian inference procedures of Bissiri et al. (2016), where traditional likelihood function-based updating rules are replaced with loss function-based updating rules. Bissiri et al. (2016) demonstrate that such a generalization allows for coherent updating—in a decision-theoretic sense—in the presence of model misspecification.

3.3 Sampling

One virtue of the quasi-Bayesian approach is the ability to repose estimation as a sampling problem. That is, instead of optimizing a GMM objective function, we now seek to sample from the quasi-posterior distribution:

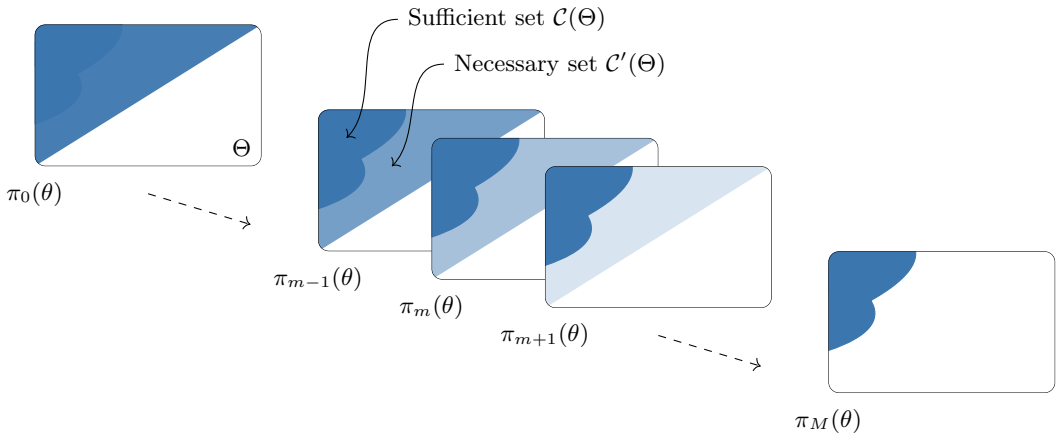
$$\pi(\theta^*|\text{data}) \propto e^{\ell_n(g(\theta^*))}\bar{\pi}(\theta^*)\mathbf{1}_{\mathcal{C}}(\theta). \tag{13}$$

With the advent of modern MCMC algorithms, such sampling tasks can be effectively carried out on high-dimensional non-conjugate models. For example, probabilistic programming languages such as `RStan` (Stan Development Team, 2024), `PyMC` (Abril-Pla et al., 2023), and `Turing.jl` (Ge et al., 2018) offer general-purpose MCMC algorithms that generate samples from the target posterior and do inference—all with limited manual coding. Examples of workhorse algorithms include Metropolis-Hastings (MH) and Hamiltonian Monte Carlo (HMC). In both frameworks, parameters are first initialized at some value and then in each subsequent iteration, a new candidate value will be proposed and either accepted or rejected. This accept/reject decision is made probabilistically using information from both the posterior and the proposal mechanism.[5]

Even with performant off-the-shelf samplers, the size and complexity of the constraint space induced by the dogmatic priors in Equation (12) pose challenges for mixing and convergence. Any general-purpose sampler need not move in directions informed by the constraints, due both to the non-convexity of the constraint space and the coarseness of the indicator function in (12). Consequently, many proposed values will be rejected, and the sampler will struggle to explore regions of high posterior probability. We overcome this challenge in two steps. First, we leverage the reparameterization outlined above which shrinks the sampling space to the smaller "necessary" set $\mathcal{C}'(\Theta) \supset \mathcal{C}(\Theta)$ such that if $\theta \in \mathcal{C}(\Theta)$ then $\theta \in \mathcal{C}'(\Theta)$. Through this step, the reparameterization and the use of Bernstein polynomials, which lend themselves to linear shape restrictions, continue to be valuable. Second, we leverage a novel Sequential Monte Carlo (SMC) algorithm (Chopin, 2002; Del Moral et al., 2006) to sample from a sequence of posterior distributions that converge to (13). Specifically, we adopt the algorithm of Golchi and Campbell (2016) whereby samples from an unconstrained (or, in our case, less-than-fully constrained) posterior are filtered and moved through sampling and resampling steps to obtain draws from the fully constrained target posterior. As such, SMC allows us to repose a hard sampling problem as a sequence of easier sampling problems with larger but shrinking domains. We outline details of our SMC sampling approach below.

[5] For more details on these common sampling algorithms, see for example Andrieu et al. (2003) and Betancourt (2017).

Figure 2: Illustration of the Shrinking Prior Domains in SMC



3.3.1 Sequential Monte Carlo

In order to leverage SMC, we first need to define a sequence of posteriors which will become the new target of sampling. In our case, this sequence is defined as

$$\begin{aligned}
\pi_0(\theta^*|\text{data}) &\propto e^{\ell_n(g(\theta^*))}\bar{\pi}(\theta^*) \\
\pi_1(\theta^*|\text{data}) &\propto e^{\ell_n(g(\theta^*))}\bar{\pi}(\theta^*)\|\theta\|_{\mathcal{C}}^{\lambda_1} \\
&\vdots \\
\pi_m(\theta^*|\text{data}) &\propto e^{\ell_n(g(\theta^*))}\bar{\pi}(\theta^*)\|\theta\|_{\mathcal{C}}^{\lambda_m} \\
&\vdots \\
\pi_M(\theta^*|\text{data}) &\propto e^{\ell_n(g(\theta^*))}\bar{\pi}(\theta^*)\|\theta\|_{\mathcal{C}}^{\lambda_M},
\end{aligned} \tag{14}$$

where $\|\theta\|_{\mathcal{C}}^{\lambda}$ is a smooth penalty function, parameterized by a non-negative penalty parameter λ, measuring the magnitude of market-level constraint violations given θ. We specify this distance metric as

$$||\theta||_{\mathcal{C}}^{\lambda} = \prod_{t=1}^{T} \Phi(-\lambda c_t(\theta)), \tag{15}$$

where $\Phi(\cdot)$ is the standard normal CDF and $c_t(\theta)$ counts of the number of constraints violated by the Jacobian of demand with respect to prices ($\mathbf{J}_p^{\sigma}(\theta)$) in market t. An important feature of this functional form is that $\Phi(-\lambda c_t(\theta)) \to \mathbf{1}_{\mathcal{C}}(\theta)$ as $\lambda \to \infty$, implying that in the limit, the prior will only place positive mass on regions of the parameter space Θ where all constraints are satisfied across all markets in our data. This allows us to define a smooth bridge between the "unconstrained" posterior $\pi_0(\theta^*|\text{data})$ and the original target posterior in (13). An illustration of this domain-shrinking bridge is provided in Figure 2.

Details of our SMC algorithm (à la Golchi and Campbell, 2016) are provided in Algorithm 1. We start by generating a set of draws (or "particles") from the reparameterized model in $\pi_0(\theta^*|\text{data})$

Algorithm 1 Sequentially Constrained Monte Carlo

Generate an initial sample of particles $\{\theta^*_{i(0)}\}_{i=1}^N$ from $\pi_0(\theta^*|\text{data})$ and set initial weights $W_{i(0)} = 1/N$.

For each model $m = 1, \ldots, M$

1. **Reweight**: For each particle $i = 1, \ldots, N$, set $W_{i(m)} = W_{i(m-1)} \cdot w_{i(m-1)}$ where

$$w_{i(m-1)} = \frac{\pi_m\big(\theta^*_{i(m-1)}|\text{data}\big)}{\pi_{m-1}\big(\theta^*_{i(m-1)}|\text{data}\big)} = \frac{\big\|g(\theta^*_{i(m-1)})\big\|_{\mathcal{C}}^{\lambda_m}}{\big\|g(\theta^*_{i(m-1)})\big\|_{\mathcal{C}}^{\lambda_{m-1}}}$$

   and then normalize weights $W_{i(m)} \leftarrow W_{i(m)} / \sum_{i'=1}^N W_{i'(m)}$.

2. **Resample**: If $\text{ESS} = \big(\sum_{i=1}^N W_{i(m)}^2\big)^{-1} < \text{ESS}_{\text{min}}$
   (a) Resample $\{\theta^*_{1(m-1)}, \ldots, \theta^*_{N(m-1)}\}$ with weights $\{W_{1(m)}, \ldots, W_{N(m)}\}$;
   (b) Set $W_{i(m)} = 1/N$.

3. **Move**: Sample new particles from a π_m-invariant transition kernel $\theta^*_{i(m)} \sim K_m(\cdot|\theta^*_{i(m-1)})$.

with no additional penalty over the constrained space. For this initial step, any general-purpose MCMC sampler will suffice.[6] Then, for each subsequent model $m = 1, \ldots, M$, we calculate a set of weights corresponding to the likelihood of each particle under $\pi_m(\theta^*|\text{data})$ relative to $\pi_{m-1}(\theta^*|\text{data})$. Since the only difference between these two posteriors is the soft penalty term $\|\theta\|_{\mathcal{C}}^{\lambda_m}$, and this penalty shrinks the domain with m, values of particles that are closer to satisfying all constraints will be given higher weight. Next, we resample the particles according to these derived weights and then move those particles using an MCMC kernel K_m which is π_m-invariant. This process continues until we sample from the last model in the sequence. Intuitively, SMC can be viewed as a sequential importance sampling algorithm (Neal, 2001; Chopin, 2002). For each step in the sequence, we have a set of draws from $\pi_{m-1}(\theta^*|\text{data})$ and corresponding importance weights that can be used to compute moments of $\pi_m(\theta^*|\text{data})$.

To mitigate problems of particle degeneracy—i.e., when only a few particles are admissible under π_m and a large share of the weights are zero—SMC samplers incorporate two additional steps. The first is a resampling step whereby new particles are sampled with replacement from the set of existing particles using their corresponding importance weights. This simultaneously (i) filters out parameter values that are unlikely under $\pi_m(\theta^*|\text{data})$; and (ii) increases the variation of weights in the sample to help mitigate particle degeneracy. This resampling step is only applied when the variation in weights is sufficiently high (and the number of unique particles starts to shrink

[6] In simulation and in our empirical application, we often find that even simple Metropolis-Hastings algorithms are sufficient here, though mixing is often substantially better when we use more sophisticated samplers (e.g., Hamiltonian Monte Carlo) for this step.

as a result). We can monitor the concentration of weights using the effective sample size (ESS), where $\text{ESS} = (\sum_{i=1}^{N} W_{i(m)}^2)^{-1} \in [1, N]$. When $\text{ESS} = 1$ the distribution of weights is degenerate, when $\text{ESS} = N$ the distribution of weights is uniform. The threshold for resampling is typically taken to be $\text{ESS}_{\text{min}} = N/2$.

The second is a particle jittering step meant to help boost the domain of resampling. In this step, new particles are drawn using an MCMC kernel which is π_m-invariant. The idea here is to use the existing N draws from $\pi_m(\theta^*|\text{data})$ to generate new draws from this same distribution. By doing so, we can increase the number of unique particles and thus boost the domain for the resampling step above. Constructing an efficient transition kernel is also straightforward. In our implementation, for example, we use a random-walk MH where new draws are proposed from a Gaussian distribution centered around the existing draws, with a covariance matrix equal to the covariance of the particles in the current iteration. Since we start sampling with draws from $\pi_m(\theta^*|\text{data})$ in hand (which is different from typical applications of MH), we can also improve our proposals by using the sample covariance matrix of the existing draws as the covariance of the proposal distribution (Chopin, 2002).

3.4 Estimation and Inference

After running the SMC algorithm above, the sampled values for the final model represent draws from the desired constrained posterior and we can trivially summarize the posterior of any parameter or function of parameters using Monte Carlo methods. For example, if we have N draws from the posterior $\{\theta^{(1)}, \ldots, \theta^{(N)}\}$ then our estimate of the quasi-posterior mean of θ is:

$$\hat{\theta}^{\text{qpm}} = \frac{1}{N} \sum_{i=1}^{N} \theta^{(i)}. \tag{16}$$

Similarly, the quasi-posterior mean of the (j, k)-th element of the Jacobian of demand is

$$\hat{\mathbf{J}}_p^{\sigma}(\theta)_{jk}^{\text{qpm}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{J}_p^{\sigma}(\theta^{(i)})_{jk} \tag{17}$$

and associated $100 \times (1 - \alpha)\%$ credible intervals immediately follow using the desired quantiles of the distribution of posterior draws $\{\mathbf{J}_p^{\sigma}(\theta^{(i)})_{jk}\}_{i=1}^{N}$.

4 Implementation in Julia: `NPDemand.jl`

Both GMM and quasi-Bayes approaches described in the previous sections require a substantial coding investment to implement in practice. For example, the supplemental material provided by Compiani (2022) contains more than 1,000 lines of Matlab code to implement the GMM estimator alone. An interested researcher using this code would need to add/modify all relevant constraints for the problem at hand in addition to the custom code required to calculate and process counterfactuals after estimation. Our quasi-Bayes approach—specifically the priors enforcing

constraints and associated SMC sampling techniques—provides another layer of complexity. This type of complexity is increasingly common in methodological work, making adoption of frontier methods difficult for practitioners. To alleviate some of these challenges, we provide the first Julia package for nonparametric demand estimation, `NPDemand.jl`. In this section we briefly describe the package to give the reader a sense of its functionality. More details and examples can be found in Appendix A.

We designed this package to allow sophisticated users to control as much of the estimation and sampling processes as possible, while suppressing all the overhead. As an example, in Figure 3 we show how users can estimate a nonparametric demand function under a monotonicity constraint with a total of two function calls. A user simply runs `define_problem()`, specifying the fields of the data which should be included in the index δ (here *prices* and a second characteristic x) , the order of the desired Bernstein polynomials, and the constraints which should be imposed in estimation, specified as a vector of symbols in Julia. This function returns a `NPDProblem` object with many attributes of the problem stored for future use.

Then, a call to our `estimate!` function solves the constrained problem. The problem is estimated via GMM as a default and relies on `JuMP.jl` (Lubin et al., 2023) as a back-end solver. Quasi-Bayes estimation proceeds in two steps: (i) first running HMC on the problem, which only uses reparameterizations to enforce linear restrictions; (ii) second running SMC which uses priors to enforce all desired constraints. Regardless of estimation approach, `estimate!` updates the problem with the resulting solution and/or draws. Note that in this simple API, we are able to suppress nearly all of the underlying complexities required for estimation.

After a problem has been estimated, we also provide tools which allow users to calculate standard outputs, including price elasticities and predictions of demand at a counterfactual prices, again with minimal knowledge or interaction with the underlying details. We show some simple examples of these functions in Figure 3c. A call to `price_elasticities!` calculates all price elasticities in all markets, which can then be summarized using `summarize_elasticities`. The `compute_demand_function!` takes in a dataframe with alternative values of all product characteristics and calculates counterfactual market shares. For problems which have been estimated via quasi-Bayes, these functions also produce credible intervals for all calculated quantities from these functions. Many of the functions we show in Figure 3 have more involved options available, especially for problems and constraints which require quasi-Bayesian methods. Moreover, the definition of problems and the calculation of counterfactuals both require dataframe inputs with formatting requirements. More detailed examples are given in the Appendix and in our package documentation. We now move on to a series of simulation and empirical exercises to explore the performance of our quasi-Bayes approach, all of which make use of `NPDemand.jl` for estimation and analysis.

Figure 3: Simple Example of `NPDemand.jl` Usage

(a) Problem Definition

```julia
using NPDemand

# Define the estimation problem
problem = define_problem(df;
    index_vars = ["prices", "x"],    # variables in the index
    b0 = 2,                          # order of the Bernstein polynomials
    constraints = [:monotone]        # constraints
)
```

(b) Estimation

```julia
# GMM (default)
estimate!(problem)

# QB Step 1: HMC
# (only using reparameterization to enforce linear restrictions)
estimate!(problem,
    quasi_bayes = true,
    burn_in = 0.25,
    n_samples = 1000,
    sampler = NUTS());

# QB Step 2: SMC
# (using priors to enforce all constraints)
smc!(problem,
    mh_steps = 5,
    ess_threshold = 200,
    smc_method = :adaptive,
    max_violations = 0.05,
    max_iter = 100)
```

(c) Post-Estimation

```julia
# Report share violations of included constraints across markets
report_constraint_violations(problem)

# Calculate price elasticities
price_elasticities!(problem)

# Calculate average of each element of the JxJ price elasticity matrix
summarize_elasticities(problem, "matrix", "mean")

# Compute demand functions at counterfactual prices defined in new_df
compute_demand_function!(new_df, problem)
```

5 Simulations

5.1 Setup

In this section, we run a series of Monte Carlo simulations to evaluate the effectiveness of our quasi-Bayes approach, measured by the fraction of demand estimates which align with the desired constraints and our accuracy in recovering price elasticities. We test our approach on data simulated from the following three data-generating processes (DGPs):

Logit Demand A logit demand model for J goods with the following utility function

$$u_{ijt} = \alpha p_{jt} + x_{jt} + \xi_{jt} + \epsilon_{ijt} \tag{18}$$

where p_{jt}, x_{jt}, and ξ_{jt} denote endogenous prices, an exogenous product characteristic, and i.i.d. demand shocks respectively. We simulate p, x, and ξ largely following Compiani (2022).[7]

Two Disjoint Logits A demand model in which there are two groups of products, each of size $J/2$. Within each group, customers choose between options using the utility function in Equation (18), but there is zero substitution between products in different groups.

Complementary Groups of Products A demand model in which there are two groups of products, each of size $J/2$. Demand is determined by the following equations, rather than by a utility function and a model of consumer choice:

$$\delta_{jt} = \alpha p_{jt} + x_{jt} + \xi_{jt} \tag{19}$$

$$q_{jt} = \exp(\delta_{jt} + \gamma_{own}\bar{\delta}_{own} + \gamma_{other}\bar{\delta}_{other}) \tag{20}$$

We choose parameters γ_{own} and γ_{other} such that products within the same group are substitutes and products in different groups are complements.

These DGPs were chosen to highlight the flexibility of NPD and the contributions of our new approach. For each DGP, our NPD estimation imposes only the constraints which are correctly specified. In the first DGP, we can impose a high level of symmetry on our estimates, including the full exchangeability of the demand function across all products, montonicity (negative own-price elasticities), diagonal dominance, and substitution between all products. In our second DGP ("two disjoint logits"), we weaken our constraints and impose both exchangeability and substitution only within the two subgroups of products. In our final DGP, which includes both complements and substitutes, we can again impose exchangeability within each group of products, as well as the signs of within- and across-group price elasticities of demand. This is meant to mimic a setting like

[7] For all DGPs, we generate an instrument $z_{jt} = 0.9 * U[0,1] + 0.05$, and the demand shifter $\xi_{jt} = N(0, v)$ for a chosen v. Then, $p_{jt} = 2(z_{jt} + U[0, 0.1]) + \xi_{jt}$ and $x_{jt} = U[0,1]$.

studies of multicategory demand (similar to those we study in Section 6), where individual products within a category are substitutes but products in different categories are complements. As we show below, both in this section and in our empirical estimates, these latter constraints demonstrate the importance of the SMC algorithm we introduced in the previous section.

For each DGP, we estimate multiple nonparametric demand functions, varying both the number of constraints imposed and the way in which they are imposed. For each simulation, we estimate demand via both the original GMM procedure introduced by Compiani (2022), as well as our quasi-Bayesian procedure ("QB"), which includes both our linearly constrained quasi-Bayesian sampling and the SMC procedure described in Algorithm 1. We also estimate demand under three different sets of constraints for each simulation, increasing the number of imposed constraints each time. This allows us to study the extent to which increasing the number of constraints explicitly imposed during estimation increases the fraction of markets in which all constraints are satisfied. In principle, it is possible that imposing a simple constraint like monotonicity (with respect to own prices) in estimation is sufficient to generate demand estimates which also satisfy more complex constraints like diagonal dominance. Establishing whether this happens in our chosen DGPs and in real data is an important input for practitioners, as imposing more complex constraints often comes at a substantial computational cost.

5.2 Simulation Results

We show our main set of simulation results in Table 2. We start by focusing first on the first row of every sub-table, which presents the fraction of markets in which any of the imposed constraints are violated. The first and clearest takeaway from our set of simulations, which is visible in the final column of each sub-table, is that our QB procedure is much better than GMM at ensuring that all constraints are satisfied in all three of the DGPs we study. Even when our data is simulated from a simple logit model and we impose all linear restrictions introduced by Compiani (2022), we find that at least one constraint (monotonicity, diagonal dominance, or that all products are substitutes) is violated in 15% of the 500 markets in the data. When we instead impose constraints via QB, that number drops to 0.3%.[8] We see a similar level of improvement in our two-logits DGP, and for our DGP with complements we find that nearly 98% of markets violate at least one of the model's constraints when we estimate demand via GMM, whereas less than 6% of markets violate any constraints after QB estimation. This large discrepancy is due in part to the fact that we cannot impose two of the three constraints in our complements DGP (complementarity between groups and substitution within groups) with linear restrictions, meaning that for this DGP the only correctly specified constraint we can impose is monotonicity. Thus, one contribution of our approach is that it allows us to enforce these new and nonlinear constraints.[9]

[8] Readers may wonder how we can have 0.3% of markets violating constraints, when 0.3% of 500 markets would imply that 1.5 markets violate constraints. This arises because all calculations of constraint violations from our QB approach report the average violations over our posterior. Although, for each value of the posterior, the number of markets with violations will be an integer, this average need not be.

[9] We note here that the only thing preventing us from producing estimates with exactly zero violations is computational time. The time cost of these simulations increases nonlinearly as a function of the maximum acceptable level

Table 2: Simulation Results

(a) DGP: Logit demand

	mono		+ diag dom		+ subs	
	GMM	QB	GMM	QB	GMM	QB
Share of markets violating constraints (%)	50.0	25.47	27.5	9.52	15.0	0.28
MSE of Price Elasticities						
– Own-price	0.531	0.058	0.045	0.023	0.025	0.021
– Cross-price	0.447	0.025	0.035	0.003	0.009	0.001

(b) DGP: Two disjoint logits

	mono		+ diag dom		+ subs	
	GMM	QB	GMM	QB	GMM	QB
Share of markets violating constraints (%)	47.5	12.11	13.0	0.42	13.0	0.29
MSE of Price Elasticities						
– Own-price	4.39	0.092	0.217	0.068	0.217	0.068
– Cross-price	4.253	0.009	0.148	0.005	0.148	0.005

(c) DGP: Complements

	mono		+ subs		+ comps	
	GMM	QB	GMM	QB	GMM	QB
Share of markets violating constraints (%)	97.5	69.03	97.5	65.91	97.5	5.65
MSE of Price Elasticities						
– Own-price	18.863	0.665	18.863	0.632	18.863	0.596
– Cross-price	17.669	0.026	17.669	0.025	17.669	0.013

Notes: Simulation results with $J = 4$ products. All results are calculated from 50 simulations. For results concerning constraint violations, we average across simulations. For price elasticities, we take the median value across simulations.

The rest of the columns in Table 2 explore the impact of iteratively imposing stronger constraints on our estimates' adherence to the full set of applicable constraints. Here we find that for each level of constraints we impose, our methods significantly out-perform GMM in terms of constraint adherence. In our logit simulation, imposing monotonicity alone results in more than 25% of markets violating at least one constraint (among montonicity, diagonal dominance, and positive cross-price elasticities). In our simulation of complements, imposing only monotonicity leaves nearly 70% of markets with at least one violation of constraints. In all cases, adding a second constraint reduces the share of violations, but this effect is heterogeneous. In our logit simulation, adding diagonal dominance reduces the share of violations from approximately 25% to less than 10%. When we simulate two disjoint logits, we find an even more dramatic reduction, from 12% to less than 1% (i.e., below our threshold for stopping SMC). In contrast, in our simulation with complements, we

of violations, so we allow for up to 1% of markets to violate constraints even in our QB procedure. For completeness, our empirical results in Section 6 present examples of estimates in which we force the number of constraint violations to exactly zero.

find that a second constraint (substitution within exchangeable group) only reduces violations from 69% to 66%. This highlights the importance of enabling practitioners to flexibly combine and test constraints, as some contexts may require stronger assumptions (and corresponding constraints) than others to generate reasonable demand estimates.

The other rows of the table compare the accuracy of estimates of own- and cross-price elasticities across methods and under different sets of constraints. We show here that our QB approach reduces the MSE of our estimated price elasticities on average, often dramatically, in each DGP and in each combination of constraints we study. Beginning again with the final two columns of the table, we find that QB reduces the median MSE of our own price elasticities in our logit demand simulation by more than 15%, and reduces the MSE of our cross-price elasticities by nearly 90%. The other two DGPs show even bigger improvements. For example, our cross-price elasticity MSEs improve by more than 95% in our DGP with disjoint logits, and by more than 99% in our complements DGP, relative to the corresponding GMM estimates. Thus, when a researcher's data produces estimates of demand which violate many economic constraints, we find that imposing those constraints both improves the compliance of the estimated demand curve with economic theory and, perhaps as important to many practitioners, improves the accuracy of the elasticities implied by the demand system.

The final dimension we show in this table is the way in which constraint violations and MSEs change, holding the data fixed, as we increase the number of constraints imposed in estimation. In columns 1 and 2, we show the prevalence of violations of *any* of the economic constraints satisfied by our DGPs, while only imposing monotonicty with respect to own-prices. Although monotonicity is the simplest constraint we can impose, we find that fully enforcing monotonicity reduces the prevalence of any violations (by nearly or more than 50%) and reduces MSEs as well. Columns 3 and 4 show the results of a similar exercise, where we now impose either diagonal dominance (logit DGPs) or substitution within exchangeable groups (complements DGP) in addition to monotonicity. In the first two DGPs, we see that our GMM estimates improve substantially on both violations and MSE when we add diagonal dominance, although QB still outperforms GMM on MSE in both cases.[10] These findings suggest both that fully enforcing even simple constraints can improve the quality of nonparametric demand estimates and that, intuitively, imposing stronger constraints yields further improvements.

5.3 Practical Considerations and the Impact of SMC

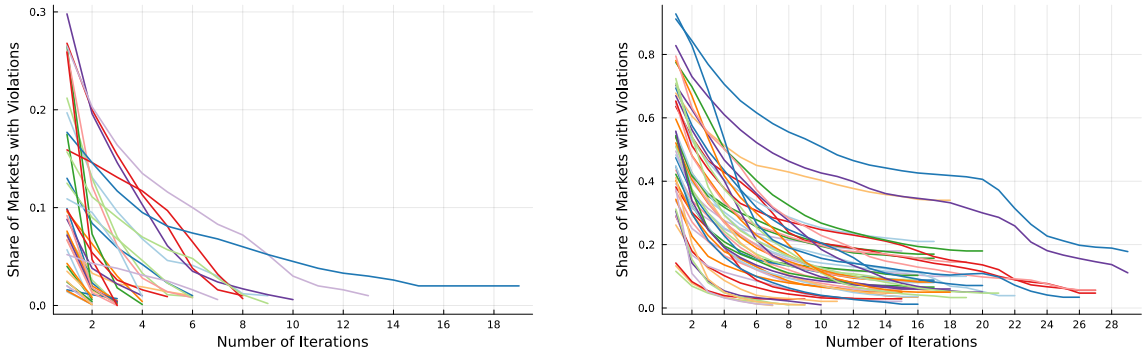
Although our main results above demonstrate dramatic improvements from QB relative to GMM, both in terms of compliance with constraints and in terms of accuracy of price elasticities, other important and interesting features of our two-step approach are worth highlighting before moving on to our empirical application. In Figures 4a and 4b we plot the fraction of markets with any

[10] In our complements DGP, we are unable to compare GMM and QB under the same constraints in columns 3-6 because we do not know of any linear restrictions for imposing substitution within groups nor complements across groups.

Figure 4: Impact of SMC on Violations and Posteriors

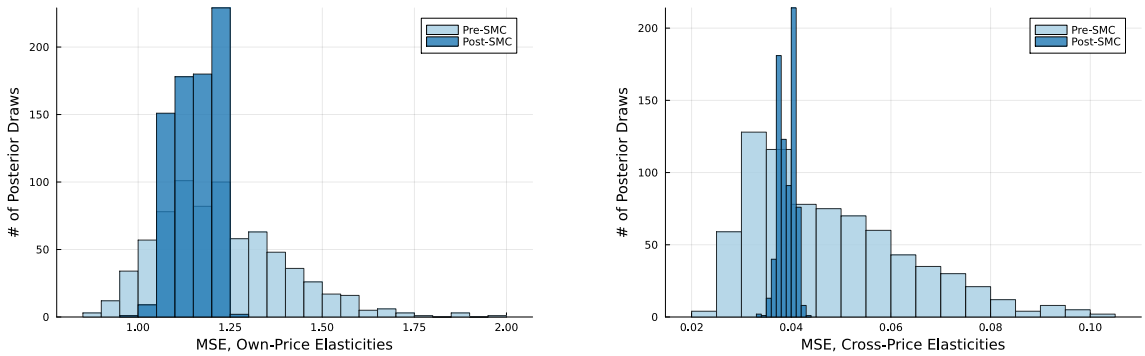
(a) Constraint Violations, One Logit

(b) Constraint Violations, Complements



(c) Posterior MSE, Own-Price Elasticities

(d) Posterior MSE, Cross-Price Elasticities



Notes: Figures (a) and (b) plot the fraction of markets in which any of the imposed constraints are violated as a function of SMC iterations. Results shown for both the logit and complements DGPs over 50 simulations. There are fewer than 50 lines in figure (a) because unconstrained elasticities in some simulations of the logit DGP already satisfy all constraints without SMC, and some lines end because we reached the maximum feasible penalty. This can be avoided by increasing the number of Metropolis-Hastings steps in our algorithm, but we do not do this in the interest of computational time. Figures (c) and (d) present the posterior distribution, before and after SMC, of the MSE of our estimates of own- and cross- price elasticities for a single representative simulation of the complements DGP, respectively.

constraint violations as a function of SMC iterations for our logit and complements DGPs. These figures demonstrate that SMC quickly and consistently reduces the fraction of markets in which constraints are violated, though there is substantial heterogeneity across simulations in the rate of convergence. In many cases, and even in our relatively complex complements DGP, the fraction of markets with any violations is reduced by more than 50% after 10 iterations. However, this heterogeneity we observe is one key reason that `NPDemand.jl` calculates λ_m, the sequence of penalties within SMC, adaptively; doing so allows us to take the largest step possible for each iteration in each problem, without taking too large a step and thereby making our Markov chain degenerate.

Comparing Figures 4a and 4b also makes clear how the practical and computational challenge of enforcing constraints via SMC scales with constraint complexity. In logit data, our Markov chains

often satisfy all imposed constraints even before we run SMC. In contrast, this is almost never the case when we simulate data with complements, and very few simulations of the latter finish SMC in less than 10 iterations (and many do not terminate within the iteration and time constraints we impose here for speed).

In the rest of Figure 4 we also demonstrate the value of SMC for reducing uncertainty in our posterior distributions. Although we found in Table 2 that the improvements in constraint compliance that QB produces also improve MSE with respect to price elasticities, there is also natural intuition that suggests that conducting inference within a more narrowly constrained domain should reduce posterior uncertainty. Indeed, this is what we often find. Figures 4c and 4d plot the posterior distribution of MSEs for own- and cross-price elasticities, respectively, for a single representative simulation. Here, the variance of posterior MSEs after SMC is dramatically smaller than before SMC. Although no post-SMC posterior MSEs are as small as the smallest pre-SMC MSEs, they provide substantial improvements on average because, after SMC, we never sample from parameters that generate the long right tail of MSEs present before SMC. Although we do not show this in figures here, it is important to note that even our linear restrictions (imposed via QB) substantially improve upon the MSE of estimated price elasticities over GMM estimates. In the simulation shown in Figures 4c and 4d, for example, the corresponding GMM MSEs are 51.1 and 20.1, respectively, whereas MSEs in these figures are always less than 2. We view this as demonstrating the value of regularization through Bayesian priors. Although the order of the Bernstein polynomial sieves we use to approximate demand functions acts as a natural regularization parameter, our priors offer much more flexible regularization which appear to improve multiple aspects of our estimates.

6 Empirical Analysis

6.1 Data

The data set used in this analysis is comprised of point-of-sale transaction data from a regional supermarket chain in the United States with nearly 500 stores spanning five states.[11] Our raw data consist of every transaction made in each store between January 1, 2015 and December 31, 2017. Specifically, we observe the quantity, price, and promotional activity for each UPC scanned at the register. Purchase records are also accompanied by a customer rewards ID when available. Because we observe all transactions and not just those from a subset of enrolled panelists, we are able to aggregate up to a coarser market-level unit of analysis.

For the purposes of our analysis, we aggregate from the UPC×date×customer level to the subcategory×brand×week×ZIP3 level. Examples of product categories include Baking Goods, Frozen Ice Cream, and Snacks. Within each category, we select all non-fringe brands from the most popular subcategories. Examples of subcategories include Cake Mix and Frosting (Baking Goods). Finally, we aggregate across forms and pack sizes and define products at the brand-level within each subcategory. We choose to model a wider assortment at a more aggregated, brand-level than

[11] The data are provided by DecaData (https://decadata.io).

Table 3: Description of Product Categories

	Category	Subcategories (number of products)
1	Baking Goods	Cake Mix (5), Frosting (3)
2	Beer	Premium (4), Economy (3)
3	Butter, Margarine, Spreads	Margarine and Spreads (4), Butter (2)
4	Cookies	Regular (4)
5	Fish Canned	Light Tuna (3), White Tuna (2)
6	Franks	Meat (2), Beef (3)
7	Frozen Ice Cream	Premium (4)
8	Frozen Pizza	Value (1), Core (4)
9	Jams, Jellies, Peanut Butter	Peanut Butter (4), Jams, Jellies, Preserves (2)
10	Ketchup	Regular (2)
11	Mayonnaise	Regular (3), Light (3)
12	Snacks	Potato Chips (4), Tortilla Chips (4)

a narrower assortment at a more granular, UPC-level for three reasons. First, aggregation helps us sidestep the zero shares problem that would exist when modeling demand at the UPC level. Second, aggregation helps reduce the size of the choice set which is practically important given the curse of dimensionality inherent in NPD systems (Compiani, 2022). Finally, wider assortments require models admitting more flexible substitution patterns given the more complex types of product differentiation across subcategories. We believe this is a better setting to showcase the flexibility of NPD systems instead of a narrow class of substitutes. Further details about aggregation and product selection steps are provided in Appendix B.

Table 3 provides a list of the 12 product categories used in our analysis. We estimate an NPD demand system for each category separately. We use a relatively large set of categories in order to provide more generalizable evidence that constraints are helpful in practice. Our 12 chosen categories are differentiated in several ways. The first is in the number of goods, ranging from $J = 2$ (Ketchup) to $J = 8$ (Baking Goods and Snacks). This allows us to examine if and how the size of the estimation problems interacts with violations of economic constraints. Second, some categories have only one subcategory while others have two. This dimension captures some of the nuances in product differentiation across wider assortments. Lastly, 10 of the 12 categories are comprised of substitutable subcategories and products (e.g., Regular vs. Light Mayonnaise), while the remaining two categories exhibit some cross-subcategory complementarity (e.g., Cake Mix vs. Frosting) and within-subcategory substitutability (e.g., Duncan Hines vs. Betty Crocker Cake Mix). Such complexity allows us to highlight both the benefits of flexible NPD systems as well as the importance of judicious constraints.

6.2 Econometric Specification

Demand

We apply our general framework from Section 2 to our data as a model of demand for brands within a three-digit ZIP code ("ZIP3") in a given week. We use two years of data from 2015-2016

for estimation, which yields a total of 5,565 ZIP3-week markets for each product category. Let demand for brand $j = 1, \ldots, J$ in each market $t = 1, \ldots, T$ be defined as

$$s_{jt} = \sigma_j\big(\delta_1(\mathbf{x}_{1t}, p_{1t}, \xi_{1t}), \ldots, \delta_J(\mathbf{x}_{Jt}, p_{Jt}, \xi_{Jt})\big) \tag{21}$$

where $\mathbf{x}_{jt}$ is a vector of product-market characteristics, p_{jt} is price, and ξ_{jmt} is an unobserved demand shock. Specifically, we model the product indicies $\delta_j(\cdot)$ as linear functions of product characteristics:

$$\delta_j(x_{jt}, p_{jt}, \xi_{jt}) = \alpha p_{jmt} + \beta x_{jt}^{\text{promo}} + \mathbf{x}_{jt}^{\text{FE}\prime}\gamma + \xi_{jt} \tag{22}$$

where x_{jt}^{promo} is a feature promotion variable and $\mathbf{x}_{jmt}^{\text{FE}}$ is a vector of controls including year, quarter, state, holiday, and product dummies. We normalize $\alpha = 1$ and drop one level from each variable contained in $\mathbf{x}_{jmt}^{\text{FE}}$. Then, as described in Section 2, we can invert to get a system of inverse demand equations.

$$p_{jt} = \sigma_j^{-1}(s_{1t}, \ldots, s_{Jt}) - \beta x_{jt}^{\text{promo}} - \mathbf{x}_{jt}^{\text{FE}\prime}\gamma - \xi_{jt} \tag{23}$$

Note that this regression has two sets of endogenous variables. Shares are expressed as functions of all ξ_j's in (22) and are thus endogenous by construction. Prices will be endogenous if they were set as a function of any of the unobserved product-market characteristics contained in ξ. As discussed in Berry and Haile (2021), such a model will require $J+1$ instruments: one set of J instruments for market shares and another excluded instrument for prices in our inverted index. We use the vector x_{jt}^{promo} as instruments for market shares, and Hausman IVs as our instrument for price.[12] In principle, other than σ^{-1}, we need only identify the ratio between β and the (implicit) coefficient on p_{jt} in the index, meaning we could use x^{promo} itself as our final instrument. However, we have already used all variation in x^{promo} to identify σ^{-1}.

Nonparametric approximation

We approximate σ^{-1} using a tensor product of d-order Bernstein basis functions. Note that because each argument of σ^{-1} is approximated with a d-order polynomial, the order of the entire function is dJ. Throughout our analysis, we fix the degree of the polynomials to $d = 2$. For simplicity, we do not address the problem of selecting the "best" value of d, which is nontrivial in nonparametric IV regression problems (Chen et al., 2024). As we will show, the estimation problem already tends to have an unfavorable signal-to-noise ratio even with $d = 2$, and so the variance of GMM estimates and the violation of constraints should only become more egregious with higher-order approximations. Moreover, our package includes d as a user-specified argument of the `define_problem( )` function (see Figure 3), so other values of d can easily be tested.

[12] At the most granular UPC level, we observe x_{jt}^{promo} as a discrete indicator of promotional status for each transaction. However, aggregating to the subcategory-brand-ZIP3-week level induces significantly more variation in this instrument.

Constraints and priors

For each category, we estimate two NPD systems: one without any shape constraints and one with shape constraints. For the 10 categories whose subcategories are imperfect substitutes, we include the following shape constraints: monotonicity, diagonal dominance, and all substitutes. For the remaining two categories whose subcategories are complements, we enforce monotonicity, that all products within a subcategory are substitutes, and that all products in different subcategories are complements (specifically, that cross-price elasticities between different categories are negative). In all models, we impose exchangeability for all products within, but not across, subcategories. We then estimate each NPD system using both GMM and QB methods. Thus, for each category, we estimate four demand systems in total: unconstrained GMM, unconstrained QB, constrained GMM, and constrained QB.

The constrained GMM model incorporates linear restrictions in the GMM problem.[13] In our QB approach, we incorporate liner restrictions via a reparameterization and then specify priors of the form $\pi(\theta) \propto \bar{\pi}(\theta^*)\mathbf{1}_{\mathcal{C}}(\theta)$ to ensure that constraints are satisfied everywhere. Here it becomes important to specify what is meant by *everywhere*—i.e., the exact form of the constraint set $C(\Theta)$. Due to the complexity and volatility of the constraint space shown in Figure 1d, the only way we have to verify whether a candidate parameter vector θ satisfies the desired shape constraints is to invert $\hat{\sigma}^{-1}$ at θ on some available candidate data and evaluate the desired behavior on the resulting price elasticities. This candidate data could be, for example, a holdout set from the available estimation data or a grid of hypothetical market shares and prices on which the constraints should hold. Both of these options follow best practices in statistics and machine learning—separating the data on which constraints are imposed from that on which the likelihood is evaluated. However, these approaches work best in situations when the constraint space is much smoother than what we see in Figure 1d. Given the complexity of our constraints, we instead define $C(\Theta)$ directly on our main data; a candidate θ is said to be in $C(\Theta)$ if there are no violations of any desired constraints in the markets in our data.[14]

Sampling details

Estimation of constrained QB models proceeds as follows. We start by estimating the model using only linear restrictions that can be encoded using reparameterizations. We sample from this model's posterior distribution using the No-U-Turn-Sampler (NUTS) (Hoffman and Gelman, 2014) implemented in `Turing.jl`. We use Turing's default values of 1000 adaption steps and a target accept ratio 0.65 and run the sampler for 5,000 iterations, keeping every other draw to reduce autocorrelation. We also discard the first 20% of draws as burn-in, resulting in a final set of

[13] Monotonicity, diagonal dominance, and all substitutes can all be represented as linear restrictions on the Bernstein coefficients. Substitutes within/across groups and complements across groups do not have such a representation and would require nonlinear restrictions, giving rise to a non-convex optimization problem.

[14] Nothing about our proposal requires that we construct constraints in this way. We do so only to ensure that constraints are satisfied in our data and as a useful first demonstration of our proposed methods, but this can be easily generalized.

2,000 draws for our inference. After HMC, we implement a second sampling step using the SMC methods outlined in Section 3.3.1. For this, we follow the adaptive approach popularized by Jasra et al. (2011) which solves for an optimal sequence of penalty values λ subject to a minimum ESS threshold, which we set to 500. We stop the sampler whenever the share of violations across all markets falls below 1%. For the final particle rejuvination step in Algorithm 1, we use 20 iterations of a random-walk MH algorithm with Gaussian proposals centered at zero and with a covariance matrix set to the sample covariance of particles at the current iteration.

6.3 Estimation Results

In this section, we provide three sets of results. First, we document the share of constraint violations across both estimators and model specifications. Second, we focus on the constrained specifications and explore the distribution of own and cross-price elasticity estimates across estimators (GMM vs. QB). Third, we quantify statistical uncertainty in the QB own-price elasticity estimates and report credible intervals for market-level elasticities and functions thereof. In total, we summarize and compare estimates of more than 2 million market-level price elasticities (361,725 own elasticities and 1,780,800 cross elasticities) across 12 product categories.

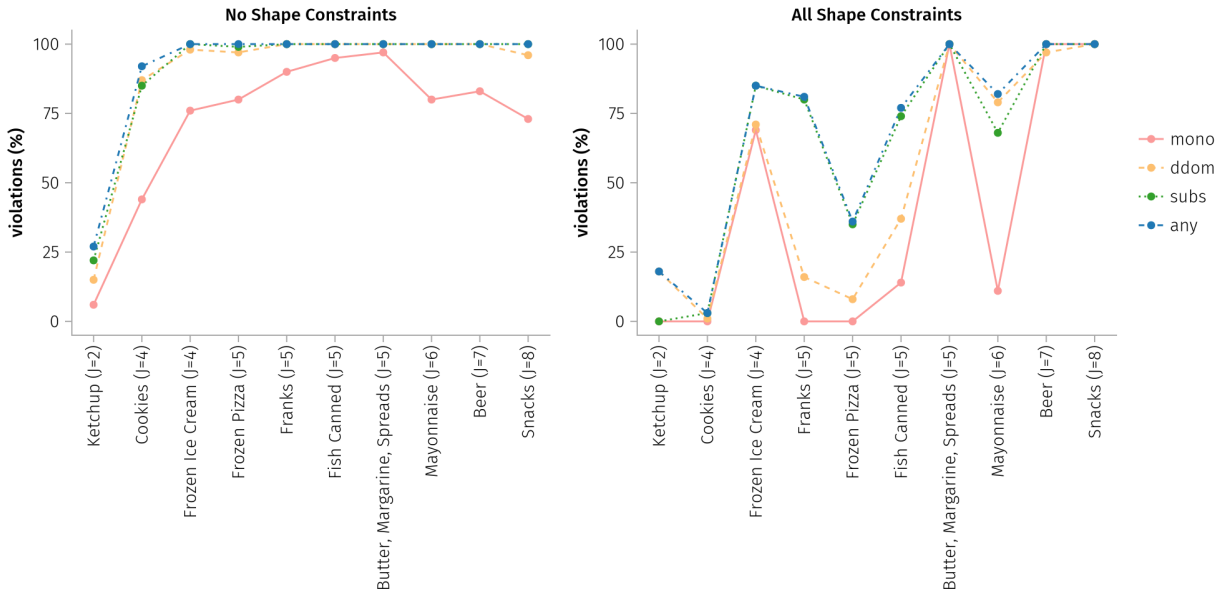
6.3.1 Constraint Violations

We start with a summary of constraint violations across all estimated models. To motivate the problem, consider the GMM constraint violations reported in Figure 5. The figure on the left reports violations from the unconstrained model without shape constraints and the figure on the right reports violations from the constrained model with all shape constraints. The y-axis reports the share of markets for which the given constraint is violated. Product categories are arranged in ascending order along the x-axis according to J. We focus on the 10 categories whose subcategories are imperfect substitutes and for which the linear restrictions could, in theory, enforce the desired constraints everywhere. Ketchup is the smallest category with $J = 2$ and Snacks is the largest category with $J = 8$.

We find that estimates from larger demand systems, on average, tend to be noisier and violate more of the desired economic properties. For example, without imposing any constraints, only 6% of markets exhibit upward-sloping demand curves with $J = 2$ products (Ketchup). However, with $J = 8$ products (Snacks), 73% of markets exhibit monotonicity violations. In fact, we find that for all estimated NPD systems with $J \geq 5$, 100% of markets exhibit at least one (of the three possible) shape constraint violation. This points to the need to discipline nonparametric estimators.

The plot on the right of Figure 5 shows that incorporating shape constraints as linear restrictions in the GMM problem is often not enough. While such constraints tend to help—i.e., the share of markets exhibiting violations decreases for most categories—there is still an appreciably large share of violations remaining. For example, in seven of the 10 categories, we find that more than 50% of markets exhibit at least one shape constraint violation even after enforcing such linear restrictions in estimation. This is the problem that our QB approach to estimation is designed to solve.

Figure 5: Constraint Violations in GMM Estimates



In Table 4, we report constraint violations for GMM estimates (as reported in Figure 5) alongside the QB estimates. The top panel reports violations for the unconstrained specification and the bottom panel reports violations for the constrained specification. There are two noticeable findings. The first is that in unconstrained QB model, the statistical regularization imbued by the priors of a Bayesian model can, and sometimes does, help reduce violations of economic constraints. For example, in the Cookies category the share of "any" violations is reduced from 92% to 63%. In most other categories, QB can show some improvements on each constraint alone, but the improvements on all constraints together is marginal. Turning to the constrained specification in panel (b), we find that GMM violations are reduced compared to panel (a), but still non-negligible and often appreciably large. In comparison, all QB columns are filled with zeros, demonstrating that our approach effectively enforces all desired constraints.

We have also explored the impact of data volume and the number of markets T on the share of violations. In particular, we estimated each category's constrained and unconstrained NPD systems via GMM using one year ($T = 2,756$), two years ($T = 5,565$), and three years ($T = 8,321$) of data. The results are omitted here for brevity but are shown in Figure C6 in the appendix. In short, we find that more data does not "solve" the problem. The share of market-level violations does tend to decrease with data volume in many categories, but also *increases* with data volume in others. This serves as a useful reminder of the weak signals in consumer data, and that the quality of the variation in the data often matters more than the quantity of data.

Table 4: Share of Markets Violating Constraints

(a) No shape constraints imposed in estimation

	(I) Monotonicity		(II) Diag dom		(III) Subs		Any of (I), (II), (III)	
Category	GMM	QB	GMM	QB	GMM	QB	GMM	QB
Beer	83	77	100	90	100	97	100	100
Butter, Margarine, Spreads	97	81	100	86	100	95	100	100
Cookies	44	20	87	34	85	49	92	63
Fish Canned	95	44	100	86	100	91	100	99
Franks	90	39	100	81	100	80	100	99
Frozen Ice Cream	76	55	98	74	100	76	100	89
Frozen Pizza	80	45	97	75	99	92	100	99
Ketchup	6	44	15	36	22	55	27	79
Mayonnaise	80	54	100	82	100	95	100	99
Snacks	73	74	96	89	100	98	100	100

(b) All shape constraints imposed in estimation

	(I) Monotonicity		(II) Diag dom		(III) Subs		Any of (I), (II), (III)	
Category	GMM	QB	GMM	QB	GMM	QB	GMM	QB
Beer	100	0	97	0	100	0	100	0
Butter, Margarine, Spreads	100	0	100	0	100	0	100	0
Cookies	0	0	1	0	3	0	3	0
Fish Canned	14	0	37	0	74	0	77	0
Franks	0	0	16	0	80	0	81	0
Frozen Ice Cream	69	0	71	0	85	0	85	0
Frozen Pizza	0	0	8	0	35	0	36	0
Ketchup	0	0	18	0	0	0	18	0
Mayonnaise	11	0	79	0	68	0	82	0
Snacks	100	0	100	0	100	0	100	0

6.3.2 Elasticity Estimates

Next, we summarize price elasticity estimates. We again start by focusing on the 10 categories of imperfect substitutes. Table 5 reports the 10th, 50th, and 90th percentiles of the distribution of own-price elasticity estimates across markets. Panels (a) and (b) report estimates from the unconstrained and constrained specifications, respectively. Table 6 follows the same structure and reports the distribution of cross-price elasticity estimates.

The results from Tables 5 and 6 provide a similar but complementary message to the earlier results on constraint violations. We find that without any constraints, estimated price elasticities are somewhat reasonable at the median—e.g., appropriately signed—but exhibit unwieldy variation across markets. For example, if we focus on the GMM columns from panel (a), we can see that the 90th percentile of the own elasticity distribution is positive for nine of the 10 categories, and larger than 10 for half of the categories. The same is true for cross-price elasticities in Table 6, where the

Table 5: Distribution of Own-Price Elasticity Estimates Across Markets

(a) No shape constraints imposed in estimation

	GMM			QB		
Category	10th	50th	90th	10th	50th	90th
Beer	-87.28	-15.97	47.47	-12.82	-0.05	12.69
Butter, Margarine, Spreads	-42.10	-1.46	8.94	-149.79	0.89	101.72
Cookies	-15.80	-3.74	2.06	-90.28	-10.24	30.04
Fish Canned	-18.23	0.17	27.11	-104.81	-8.63	33.79
Franks	-17.00	-0.42	16.60	-102.49	-13.86	30.61
Frozen Ice Cream	-153.27	-2.72	24.90	-108.59	-1.77	71.70
Frozen Pizza	-14.80	-2.56	3.39	-84.31	-8.18	8.38
Ketchup	-14.52	-3.68	-1.01	-23.23	-0.62	18.41
Mayonnaise	-49.01	-2.94	20.27	-229.51	-11.36	88.66
Snacks	-15.36	-3.73	9.08	-146.88	-6.67	73.91

(b) All shape constraints imposed in estimation

	GMM			QB		
Category	10th	50th	90th	10th	50th	90th
Beer	-72.92	-20.59	176.96	-10.18	-3.60	-1.55
Butter, Margarine, Spreads	-423.19	10.18	114.38	-154.39	-16.74	-3.29
Cookies	-58.12	-15.48	-5.42	-83.70	-20.35	-6.19
Fish Canned	-81.07	-12.21	-2.88	-46.75	-10.27	-3.31
Franks	-68.85	-25.42	-4.72	-55.83	-19.94	-3.54
Frozen Ice Cream	-157.96	-6.76	351.05	-176.53	-9.96	-1.74
Frozen Pizza	-41.23	-9.64	-3.17	-46.53	-8.27	-2.20
Ketchup	-75.24	-17.20	-3.42	-18.26	-4.52	-1.22
Mayonnaise	-159.04	-24.56	-2.79	-129.76	-23.01	-3.91
Snacks	-58.21	6.76	70.95	-126.14	-32.78	-10.91

10th percentile (GMM unconstrained) is negative for all categories and larger than 10 in absolute value for six of the 10 categories.

After incorporating linear restrictions in the GMM estimation problem, the distribution of elasticities shifts. The 90th percentile of own elasticities becomes negative for eight of the nine categories for which it had been positive. The 10th percentile of cross elasticities remains negative for most categories, however the values are much smaller and closer to zero. We know from Table 4 that the GMM linear restrictions are not enough to ensure that the demand functions are well-behaved everywhere. Tables 5 and 6 provide corroborating evidence.

The QB constrained specification leads to estimated elasticities which obey the shape constraints everywhere in the distribution across markets. For example, own elasticities are negative and cross elasticities are positive for all three reported quantiles. The median elasticities also tend to be large relative to estimates in the literature based on more restrictive logit-like specifications of demand. This is perhaps not surprising giving that the functional form itself will regularize. Our goal in this paper is to pare back the assumptions on functional form while retaining economic

Table 6: Distribution of Cross-Price Elasticity Estimates Across Markets

(a) No shape constraints imposed in estimation

	GMM			QB		
Category	10th	50th	90th	10th	50th	90th
Beer	-32.85	0.24	36.94	-4.97	-0.03	4.77
Butter, Margarine, Spreads	-12.06	0.13	19.14	-44.50	-0.29	47.65
Cookies	-4.69	0.93	6.70	-31.90	1.18	39.02
Fish Canned	-19.55	-0.03	16.65	-29.31	0.96	42.71
Franks	-10.71	0.05	11.26	-28.05	2.97	47.27
Frozen Ice Cream	-42.62	0.92	51.51	-54.01	0.19	56.60
Frozen Pizza	-4.55	0.20	4.89	-15.07	1.10	27.05
Ketchup	-0.11	1.06	4.50	-20.30	-0.01	22.67
Mayonnaise	-16.87	0.49	21.95	-57.47	0.71	72.38
Snacks	-7.19	0.23	7.83	-31.34	-0.12	38.24

(b) All shape constraints imposed in estimation

	GMM			QB		
Category	10th	50th	90th	10th	50th	90th
Beer	-38.88	0.78	12.20	0.12	0.40	1.56
Butter, Margarine, Spreads	-39.49	0.00	2.81	0.59	2.46	10.91
Cookies	0.24	0.67	1.55	1.64	4.61	13.84
Fish Canned	-0.30	0.20	7.49	0.41	1.43	6.37
Franks	-0.00	0.00	2.41	0.57	2.33	6.10
Frozen Ice Cream	-152.50	2.24	34.11	0.56	2.59	31.98
Frozen Pizza	-0.00	0.26	1.37	0.40	1.27	5.51
Ketchup	0.71	1.84	5.59	0.58	1.61	8.42
Mayonnaise	-0.50	1.20	13.63	0.36	2.88	14.84
Snacks	-17.03	0.00	10.20	0.87	3.04	13.78

validity. Doing so should produce different—and possibly larger—estimates than other existing specifications. Indeed, Compiani (2022) compares NPD estimates to a mixed logit model and also finds NPD produces larger elasticities.

So far we have only summarized the elasticity estimates from categories with imperfect substitutes. In Table 7, we summarize the own and cross-price elasticity estimates for the final two categories that exhibit within-subcategory substitutes and cross-subcategory complements. This type of block substitution pattern is novel to our estimation approach and was not addressed by Compiani (2022). In Table 7, we report the same quantiles of the distribution of elasticity estimates, but we now do so for own elasticities, within-subcategory cross elasticities, and across-subcategory cross elasticities. By partitioning the cross elasticities we can better see the desired behavior: the "within" elasticities should be positive and the "across" elasticities should be negative.

The baseline GMM specification cannot accommodate such complex within/across subcategory constraints using linear restrictions. We therefore only impose own-good monotonicity when estimating the model via GMM. While comparing the resulting elasticities between GMM and QB is

Table 7: Distribution of Elasticity Estimates for Categories with Complements

		GMM			QB		
Category		10th	50th	90th	10th	50th	90th
Baking Goods	own	-37.80	-2.45	14.29	-40.39	-8.09	-1.52
	within	-12.39	0.98	16.92	0.59	2.51	9.14
	across	-8.36	-0.16	7.61	-4.43	-1.39	-0.36
Jams, Jellies, Peanut Butter	own	-0.63	-0.18	-0.04	-51.28	-6.34	-1.67
	within	-0.04	0.01	0.23	0.88	5.23	25.20
	across	0.01	0.07	0.36	-6.26	-1.43	-0.34

Notes: GMM only enforces monotonicity while QB enforces monotonicity, substitutes within group, and complements across group.

no longer an apples-to-apples comparison, we still include GMM estimates as a benchmark.

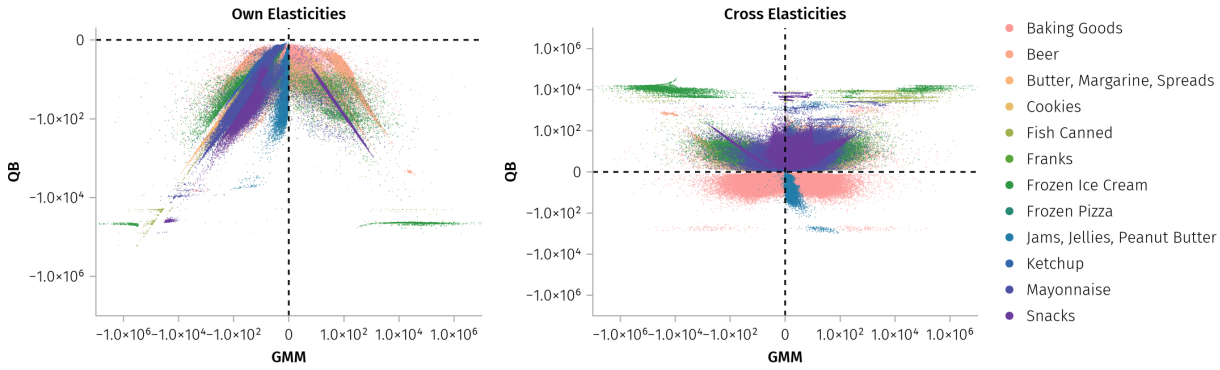
In the Baking Goods category, the GMM estimates are reasonable at the median. The median own elasticity is -2.45 and the within and across elasticities are positive and negative respectively. In Jams, Jellies, and Peanut Butter, the own elasticity is negative but much smaller in magnitude (-0.18), and the across-subcategory elasticities are not correctly signed. In both categories, we again find that the tails of the distribution of estimates are quite long, leading to a large share of incorrectly signed estimates. In contrast, the QB estimates nail the complex, block-wise substitution patterns for all markets in the data.

Figure 6a provides a view of the joint distribution of constrained elasticity estimates. The figure on the left plots the own elasticities and the figure on the right plots the cross elasticities. GMM estimates are on the x-axis and QB estimates are on the y-axis. Each point is a single product-market elasticity and is colored according to its category. This figure is useful for visualizing the impact of constraints and Bayesian regularization on high-variance estimates. For example, if QB produced the same estimates of own elasticities as GMM, we would see a cloud of points along the 45 degree line. Instead, we see two clouds of points. The first is in the lower left quadrant and consists of elasticities that are estimated to be negative by both GMM and QB. Many elasticities fall on or above the 45 degree line, indicating that the QB estimates tend to be shrunk towards zero. The second cloud of points sits in the lower right quadrant and consists of elasticities estimated to be positive by GMM but negative by QB. In this case, the QB constraints have enough bite to flip the estimated sign. The right-hand plot in Figure 6a tells a similar story. GMM estimates exhibit wide dispersion ranging from -10^6 to 10^6, while QB estimates range from 0 to 10^4 for pairs of substitutes and -10^3 to 0 for pairs of complements.

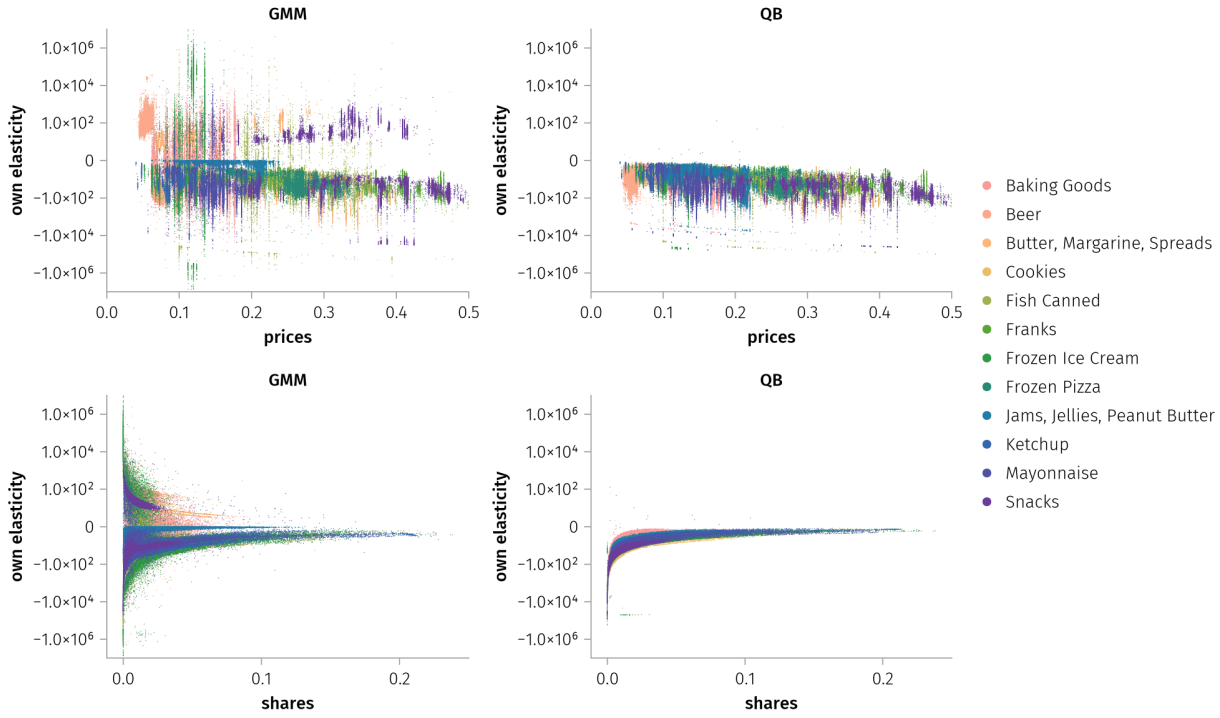
Finally, we explore how the distribution of own elasticities varies with prices and shares. Figure 6b plots all market-level own elasticity estimates from the constrained specification against prices in the top row and against shares in the bottom row. The column on the left-hand side plots GMM estimates and the column on the right-hand side plots QB estimates. In theory, we should expect the magnitude of own elasticities to increase in prices and decrease in shares. This is indeed the pattern we find across both GM and QB estimates, although we again notice the large cloud of

Figure 6: Market-Level Elasticity Estimates

(a) GMM vs. QB

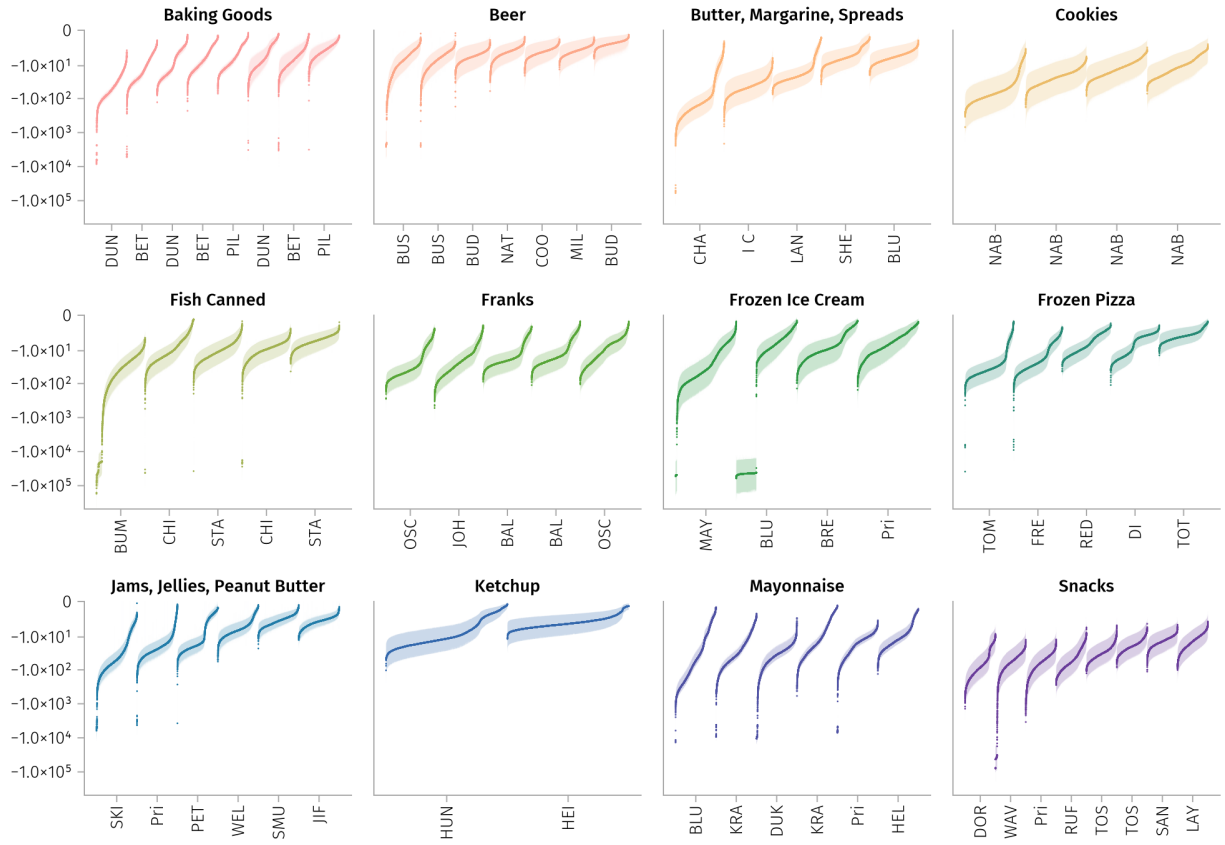


(b) Own Elasticity Estimates as a Function of Prices and Shares



incorrectly signed GMM estimates whose values become more egregious with larger prices and/or smaller shares. It is also worth noting that the extremely large elasticity estimates tend to come from markets with very small shares—e.g., see the left tail of the distribution of estimates show in the bottom-right plot of Figure 6b.

Figure 7: Own Elasticity Estimates and 95% Credible Intervals



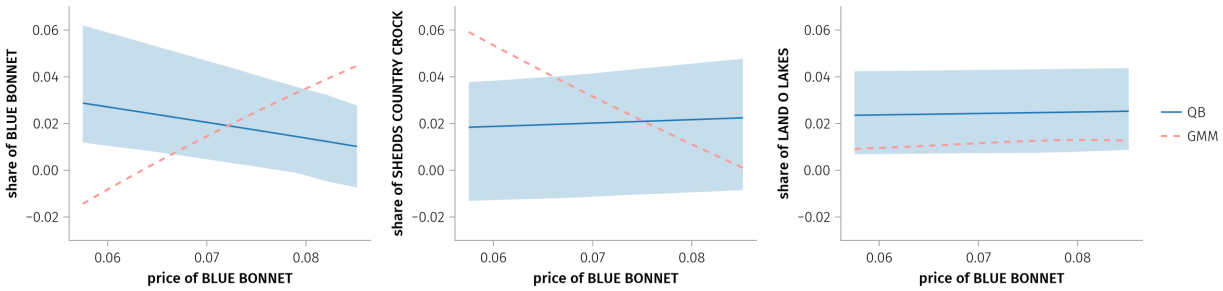
6.3.3 Uncertainty Quantification

So far we have summarized several million market-level price elasticity estimates. Our last set of estimation results focuses on the statistical uncertainty associated with those point estimates. One virtue of our quasi-Bayes approach is that inference is automatic. That is, after using the sampling algorithms outlined in Section 3.3 to produce draws from the target posterior distribution, we can then use those draws to construct interval estimates for any quantity of interest.

To highlight the simplicity of inference in our framework, we compute 95% credible intervals for all 361,725 market-level own-price elasticity estimates. Figure 7 visualizes the distribution of all point estimates and accompanying credible intervals across all 12 categories. Within each category, estimates are arranged by product and placed in ascending order. The distribution shown for each product is comprised of 5,565 market-level estimates. The shaded bands represent 95% credible intervals.

In addition to inferences about granular, market-level elasticities, researchers may also be interested in various summary statistics such as the mean or median elasticity. We suppress the results here for the sake of brevity, but include two figures in the appendix which report estimated quantiles of own-price elasticities for each product, and associated credible intervals. Figure C7

Figure 8: Demand Curves (Butter, Margarine, Spreads)



plots the median own elasticities for all products across all categories. Figure C8 zooms in on the Frozen Pizza category and plots the own elasticity quantile function for each product.

6.4 Counterfactual Demand Predictions

As a final exercise, we show how our QB-NPD methodology can be used to construct counterfactual demand functions. Suppose we want to predict s_j, the market share of good j, at a vector counterfactual prices $\mathbf{p}^* = (p_1^*, \ldots, p_J^*)$. For simplicity, assume that all other observed and unobserved product characteristics are held fixed. In our approach, this prediction is made via the quasi-posterior predictive distribution which naturally integrates demand predictions over the posterior uncertainty in model parameters.

To illustrate, we construct a few example demand curves using data from the Butter, Margarine, and Spreads category. We consider a grid of 10 prices for Blue Bonnet margarine that range from a 10% reduction in the minimum observed Blue Bonnet price to a 10% increase in the maximum observed Blue Bonnet price. All other prices are fixed at their median price. We then predict market shares for three products: Blue Bonnet margarine, Shedd's Country Crock margarine, and Land O'Lakes butter. The resulting demand curves are shown in Figure 8. The solid line corresponds to the quasi-posterior mean prediction and the shaded region is a 95% credible interval. The dashed line corresponds to the predicted demand curve based on the GMM estimates. In both cases we use parameter estimates from the fully constrained specification.

We find that GMM estimates lead to a demand function for Blue Bonet which is *increasing* in the price of Blue Bonet and a demand function for Shedd's Country Crock which is *decreasing* in the price of Blue Bonnet (an obvious substitute). Both demand functions are directionally inconsistent with the law of demand—a result that may be expected, given the large share of constraint violations documented in Section 6.3. What is also to be expected is that the QB demand curves are all directionally consistent with economic theory by construction.

Figure 8 provides one of the more egregious illustrations of the differences between QB and GMM predictions in our data. It is possible that an aggregate demand curve can still be downward-sloping despite many markets exhibiting monotonicity violations.[15] Ultimately this will depend on both

[15] Although we haven't computed the demand functions associated with every possible pair of products in our

the center and spread of the distribution of estimates. Our perspective is that we should not just hope that we can pin down reasonable mean or median estimates of model parameters so that our predictions of counterfactual demand satisfy economic theory. This may in fact be a lot to ask given the inherent noise in consumer data. Instead, it can be useful to enforce the desired constraints at the most granular unit of analysis to ensure the validity of any counterfactual predictions.

7 Conclusion

In this paper, we propose a quasi-Bayes approach to estimating nonparametric demand systems subject to complex shape constraints. We build on the pioneering work of Compiani (2022), who uses Bernstein polynomials to nonparametrically estimate inverse demand functions via linearly constrained GMM. Our quasi-Bayes approach transforms this GMM objective function to construct a quasi-likelihood and then defines priors which rule out regions of the parameter space violating the desired constraints. To sample from the induced posterior, we employ a novel Sequential Monte Carlo algorithm which targets a sequence of posteriors with soft-constraints converging to the desired hard-constraint.

We then use both simulation experiments and retail scanner data to demonstrate that our estimation approach works and can flexibly estimate demand subject to complex shape constraints. In our empirical analyses, we estimate nonparametric demand systems for 12 separate product categories which together span 66 products and more than 2 million market-level price elasticities. Our results show that (i) constraints can improve the quality of estimates regardless of the estimation approach; (ii) an appreciable share of markets exhibit violations even after linear restrictions are imposed in the GMM problem; and (iii) our QB method can close this gap and accommodate even more complex constraints which cannot be represented as linear restrictions on model parameters.

Throughout the paper, we showcase our accompanying Julia package `NPDemand.jl`. The package allows researchers to estimate nonparametric demand systems using both GMM and QB estimators—all with a single function call. The back-end of our package is powered by `Turing.jl`, Julia's state of the art library for Bayesian computation. We hope that our methods and tools—alongside generalizable evidence of their value in practice—provide a useful step forward and encourage other researchers to test nonparametric approaches in their analysis of consumer demand.

data, we do find that GMM often produces reasonably signed elasticities at the median or mean values. For example, roughly 85% (77%) of median (mean) own-price elasticity estimates are negative under the constrained specification. This suggests that many of the predicted demand functions may have the correct slope in aggregate despite a large share of market-level violations.

References

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., et al. (2023). PyMC: A Modern, and Comprehensive Probabilistic Programming Framework in Python. *PeerJ Computer Science*, 9:e1516.

Andrews, I. and Mikusheva, A. (2022). Optimal Decision Rules for Weak GMM. *Econometrica*, 90(2):715–748.

Andrieu, C., Freitas, N. D., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43.

Berry, S., Gandhi, A., and Haile, P. (2013). Connected Substitutes and Invertibility of Demand. *Econometrica*, 81(5):2087–2111.

Berry, S. T. and Haile, P. A. (2014). Identification in Differentiated Products Markets using Market Level Data. *Econometrica*, 82(5):1749–1797.

Berry, S. T. and Haile, P. A. (2021). Foundations of Demand Estimation. In Ho, K., Hortaçsu, A., and Lizzeri, A., editors, *Handbook of Industrial Organization, Volume 4*, pages 1–62. Elseiver.

Berry, S. T., Levinsohn, J., and Pakes, A. (1995). Automobile Prices in Market Equilibrium. *Econometrica*, 63(4):841–890.

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130.

Blundell, R., Horowitz, J., and Parey, M. (2017). Nonparametric Estimation of a Nonseparable Demand Function under the Slutsky Inequality Restriction. *Review of Economics and Statistics*, 99(2):291–304.

Blundell, R., Horowitz, J. L., and Parey, M. (2012). Measuring the Price Responsiveness of Gasoline Demand: Economic Shape Restrictions and Nonparametric Demand Estimation. *Quantitative Economics*, 3(1):29–51.

Brand, J. (2021). Differences in Differentiation: Rising Variety and Markups in Retail Food Stores.

Chen, X., Christensen, T., and Kankanala, S. (2024). Adaptive Estimation and Uniform Confidence Bands for Nonparametric Structural Functions and Elasticities. *Review of Economic Studies*, 92(1):162–196.

Chernozhukov, V. and Hong, H. (2003). An MCMC Approach to Classical Estimation. *Journal of Econometrics*, 115(2):293–346.

Chopin, N. (2002). A Sequential Particle Filter Method for Static Models. *Biometrika*, 89(3):539–552.

Compiani, G. (2022). Market Counterfactuals and the Specification of Multiproduct Demand: A Nonparametric Approach. *Quantitative Economics*, 13(2):545–591.

Conlon, C. and Gortmaker, J. (2020). Best Practices for Differentiated Products Demand Estimation with PyBLP. *The RAND Journal of Economics*, 51(4):1108–1161.

Deaton, A. and Muellbauer, J. (1980). An Almost Ideal Demand System. *American Economic Review*, 70(3):312 326.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

Farrell, M. H., Liang, T., and Misra, S. (2021). Deep Learning for Individual Heterogeneity: An Automatic Inference Framework.

Fosgerau, M., Monardo, J., and Palma, A. d. (2024). The Inverse Product Differentiation Logit Model. *American Economic Journal: Microeconomics*, 16(4):329–370.

Gallant, A. R., Hong, H., Leung, M. P., and Li, J. (2022). Constrained Estimation using Penalization and MCMC. *Journal of Econometrics*, 228(1):85–106.

Ge, H., Xu, K., and Ghahramani, Z. (2018). Turing: A Language for Flexible Probabilistic Inference. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1682–1690.

Gelman, A. (2004). Parameterization and Bayesian Modeling. *Journal of the American Statistical Association*, 99(466):537–545.

Ghosal, R., Ghosh, S., Urbanek, J., Schrack, J. A., and Zipunnikov, V. (2023). Shape-Constrained Estimation in Functional Regression with Bernstein Polynomials. *Computational Statistics & Data Analysis*, 178:107614.

Golchi, S. and Campbell, D. A. (2016). Sequentially Constrained Monte Carlo. *Computational Statistics & Data Analysis*, 97:98–113.

Haag, B. R., Hoderlein, S., and Pendakur, K. (2009). Testing and Imposing Slutsky Symmetry in Nonparametric Demand Systems. *Journal of Econometrics*, 153(1):33–50.

Hoderlein, S. and Lewbel, A. (2012). Regression Dimension Reduction with Economic Constraints: The Example of Demand Systems with Many Goods. *Econometric Theory*, 28(5):1087–1120.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623.

Hong, H., Li, H., and Li, J. (2021). BLP Estimation using Laplace Transformation and Overlapping Simulation Draws. *Journal of Econometrics*, 222(1):56–72.

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2011). Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38(1):1–22.

Kankanala, S. (2023). Quasi-Bayes in Latent Variable Models.

Kato, K. (2013). Quasi-Bayesian Analysis of Nonparametric Instrumental Variables Models. *The Annals of Statistics*, 41(5):2359–2390.

Kim, J.-Y. (2002). Limited Information Likelihood and Bayesian Analysis. *Journal of Econometrics*, 107(1-2):175–193.

Kong, X., Dubé, J.-P., and Daljord, Ø. (2024). Nonparametric Estimation of Demand with Switching Costs: The Case of Habitual Brand Loyalty.

Lewbel, A. (1995). Consistent Nonparametric Hypothesis Tests with an Application to Slutsky Symmetry. *Journal of Econometrics*, 67(2):379–401.

Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., and Vielma, J. P. (2023). JuMP 1.0: Recent Improvements to a Modeling Language for Mathematical Optimization. *Mathematical Programming Computation*, 15:581–589.

Ma, Y.-A., Chen, Y., Jin, C., Flammarion, N., and Jordan, M. I. (2019). Sampling Can be Faster than Optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885.

Matzkin, R. L. (1994). Restrictions of Economic Theory in Nonparametric Methods. volume 4 of *Handbook of Econometrics*, pages 2523–2558. Elsevier.

Mehta, N. (2015). A Flexible Yet Globally Regular Multigood Demand System. *Marketing Science*, 34(6):843 863.

Monardo, J. (2021). Measuring Substitution Patterns with a Flexible Demand Model.

Monardo, J. (2024). Functional Form and Shape Restrictions in Discrete Choice Models.

Neal, R. M. (2001). Annealed Importance Sampling. *Statistics and Computing*, 11(2):125–139.

Newey, W. K. and Powell, J. L. (2003). Instrumental Variable Estimation of Nonparametric Models. *Econometrica*, 71(5):1565–1578.

Pachali, M. J., Kurz, P., and Otter, T. (2020). How to Generalize from a Hierarchical Model? *Quantitative Marketing and Economics*, 18(4):343–380.

Romeo, C. J. (2016). Incorporating Prior Information into a GMM Objective for Mixed Logit Demand Systems. *The Journal of Industrial Economics*, 64(2):336–363.

Singh, A., Liu, Y., and Yoganarasimhan, H. (2023). Choice Models and Permutation Invariance: Demand Estimation in Differentiated Products Markets.

Stan Development Team (2024). RStan: The R Interface to Stan. R package version 2.32.6.

Sun, Y. and Ishihara, M. (2019). A Computationally Efficient Fixed Point Approach to Dynamic Structural Demand Estimation. *Journal of Econometrics*, 208(2):563–584.

Wang, A. (2023). Sieve BLP: A Semi-Nonparametric Model of Demand for Differentiated Products. *Journal of Econometrics*, 235(2):325–351.

Wang, J. and Ghosh, S. (2012). Shape Restricted Nonparametric Regression with Bernstein Polynomials. *Computational Statistics & Data Analysis*, 56(9):2729–2741.

Wang, J. and Huang, Y. (2024). Scalable Just-in-Time Price Elasticity Estimation.

Zhang, T. (2006a). From ε-entropy to KL-entropy: Analysis of Minimum Information Complexity Density Estimation. *The Annals of Statistics*, 34(5):2180–2210.

Zhang, T. (2006b). Information-Theoretic Upper and Lower Bounds for Statistical Estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321.

APPENDIX

A NPDemand.jl Package Usage and Examples

A.1 Input DataFrame

In order to begin using `NPDemand.jl`, one has to structure data s in Table A1. Here, we show an example dataframe for a problem in which the index contains prices and a second characteristic x. For each product (indexed starting at 0), the dataframe should have a numbered column for `shares`, `prices`, `x`, `share_iv`, and `price_iv`. The first three are straightforward, and the last two denote the instruments for market shares (in σ^{-1}) and prices (in the index). The instruments for market shares must be full rank, because we require J dimensions of variation to identify σ^{-1}. In contrast, the instruments for price require much less variation as they jointly identify only a single coefficient (the implicit coefficient on price in the index relative to other covariates).

Table A1: Example of Dataframe for NPDemand Problem Construction

shares0	prices0	x0	price_iv0	shares1	prices1	x1	...
0.151	1.584	0.070	0.615	0.259	1.737	0.490	...
0.174	0.000	0.528	0.176	0.232	2.006	0.894	...
0.183	0.707	0.441	0.284	0.192	0.923	0.783	...
0.291	1.068	0.236	0.201	0.190	1.211	0.814	...
...							

A.2 Defining and Solving the Linear Problem (GMM)

Figure A1: Simple Example of NPDemand.jl Usage

```
using NPDemand, Turing

problem = define_problem(df;
    index_vars = ["prices", "x"],
    b0 = 2
)

# Estimate via GMM
estimate!(problem)
```

A.3 Constraint Implementation

Figure A2: Simple Example of NPDemand.jl Usage

```
using NPDemand, Turing

problem_constraints =
[:exchangeability, # some products are exchangeable in the demand function
:monotone, # Demand function is monotonic in own-price
:diagonal_dominance_all] # diagonal dominance of Jacobian of demand

problem = define_problem(df;
    exchange = [[1 2], [3 4]],
    index_vars = ["prices", "x"],
    constraints = problem_constraints,
    b0 = 2
)
```

A.4 Options for Quasi-Bayes and Sequential Monte-Carlo

Figure A3: Simple Example of NPDemand.jl Usage

```
using NPDemand, Turing

# ---------------------
# Define a problem in the same way as shown above
# ---------------------

# Define some function inputs
skip      = 10;
burn_in   = 0.5;
mh_steps  = 15;
step_size = 0.01;

# Alternatively, estimate via Quasi-Bayes with a custom sampler
estimate!(problem,
    quasi_bayes = true,
    penalty = 0,
    burn_in = burn_in,
    n_attempts = 500,
    n_samples = 20_000,
    step = step_size,
    sampler = HMC(0.01, 3; adtype = Turing.AutoForwardDiff()));

# After producing an unconstrained Markov chain, we can run SMC on it
smc!(problem,
    burn_in = burn_in,
    skip = skip,
    max_penalty = 6,
    step = step_size,
    mh_steps = mh_steps,
    ess_threshold = 200,
    smc_method = :adaptive,
    max_violations = 0.01
    )
```

A.5 Post-Estimation Processing

Figure A4: Calculating Counterfactual Demand Predictions

```
using NPDemand, TidierPlots
n_temp = 15
alt_price_df = DataFrame(); # Initialize new dataframe
alt_price_df.prices1 = 1.1 .* ones(n_temp);
alt_price_df.prices2 .= 1.1;
alt_price_df.prices3 .= 1.1;

alt_price_df.prices0 .= collect(range(0.01,15.8, length=n_temp));
for j = 0:3
    # shares set to zero to initialize
    # compute_demand_function! will ignore and replace
    alt_price_df[!,"shares$j"] .=0;
    alt_price_df[!,"x$j"] .= 0.5;
end
alt_price_df[!,"dummyFE"] .= 1

compute_demand_function!(problem_qb,
    alt_price_df;
    max_iter = 50, show_trace = false,
    n_draws = 50, CI = 0.95,
    average_over = ["dummyFE"]);

# Plot predictions with credible intervals
# Reshape alt_price_df
melted_df = DataFrame(
    prices = repeat(alt_price_df.prices0, outer = 4),
    shares = vcat(alt_price_df.shares0, alt_price_df.shares1,
    ↪  alt_price_df.shares2, alt_price_df.shares3),
    x = repeat(alt_price_df.x0, inner = 4),
    lb = vcat(alt_price_df.shares0_lb, alt_price_df.shares1_lb,
    ↪  alt_price_df.shares2_lb, alt_price_df.shares3_lb),
    ub = vcat(alt_price_df.shares0_ub, alt_price_df.shares1_ub,
    ↪  alt_price_df.shares2_ub, alt_price_df.shares3_ub),
    product = repeat(["Product 1", "Product 2", "Product 3", "Product 4"],
    ↪  inner = n_temp)
)

# Plot using ggplot
ggplot(melted_df) +
    geom_line(aes(x = :prices, y = :shares, color = :product), size = 1.5)
    ↪  +
    geom_errorbar(aes(x = :prices, ymin = :lb, ymax = :ub, color =
    ↪  :product), width = 0.0) +
    labs(x = "Price 1, Others fixed at median", y = "Market Share") +
    theme_minimal()
```

Figure A5: Price Elasticity Calculations

```
using NPDemand, Turing

# ---------------------
# Define a problem in the same way as shown above
# ---------------------

# ---------------------
# Estimate problem however desired
# Note: confidence intervals in functions below
# only work if problem estimated via QB
# ---------------------

# Calculate price elasticities of demand in every market,
# with 95% confidence intervals
price_elasticities!(problem, CI = 0.95)

# Calculate the 20th percentile price elasticity between each
# pair of products, with 90% confidence intervals
summarize_elasticities(problem, "quantile", q=0.2, CI = 0.9)
```

B Additional Data Details

Aggregation

Our raw data consists of every unique transaction made in each store, and so for each UPC×date we have repeated observations of transacted prices. The following steps outline our approach to aggregating from UPC×date×household to brand×week×ZIP3.

1. UPC×date×**household** → UPC×date×**store**

   We compute store-level prices for each UPC×date by taking the median price across transactions (there is rarely any within-date, cross-household variation in prices).

2. UPC×**date**×**store** → UPC×**week**×**ZIP3**

   Next, we aggregate to the week×ZIP3 level by taking total quantity sold and the median UPC prices across all date×stores within a week×ZIP3.

3. **UPC**×week×ZIP3 → **brand**×week×ZIP3

   Finally, we aggregate from UPCs to brands by summing the total equivalent unit sales for all UPCs within a brand. We measure prices as a quantity-weighted averages of UPC-level prices (per equivalent unit), where the weights are computed separately for each year.

Market sizes

To compute market sizes for each category, we leverage the fact that our raw data is at the household-trip level. Following Brand (2021), we start by calculating a ratio of the number households who ever purchased from the category in a given year to the number of households who purchased from the category in a week. This ratio is informative about the relative size of outside good demand—i.e., for every one household who we observed make a purchase in a given week, there were x additional households who were in the market but did not make a purchase. We compute this ratio at the weekly level for each category-year, and then take an average for the year. We then use this average ratio as a scaling factor for the maximum weekly unit sales for the category, where the maximum is taken across all cross-sectional units (e.g., ZIP3s).

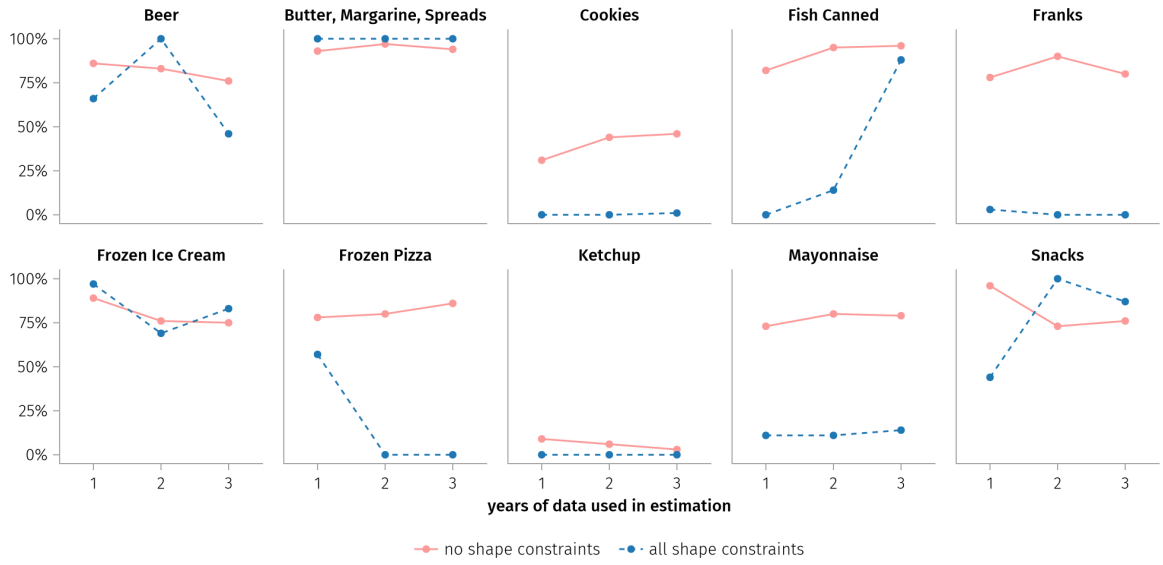
Product selection

The last step is to select a focal set of products within each category to use in our analysis. Because of the high-dimensional nature of our estimator, we only consider products from up to two subcategories. For each category, we first compute subcategory-level market shares. If the top subcategory has greater than 65% share, then we only consider products from that single subcategory. Otherwise, we consider products from the top two subcategories. Then, among the chosen 1-2 subcategories, we compute product-level market shares and select all products with greater than 5% share.

C Additional Empirical Results

Figure C6: GMM Estimates and Data Volume

(a) Share of markets with own-good monotonicty violations



(b) Own-price elasticities (10th and 90th percentiles)

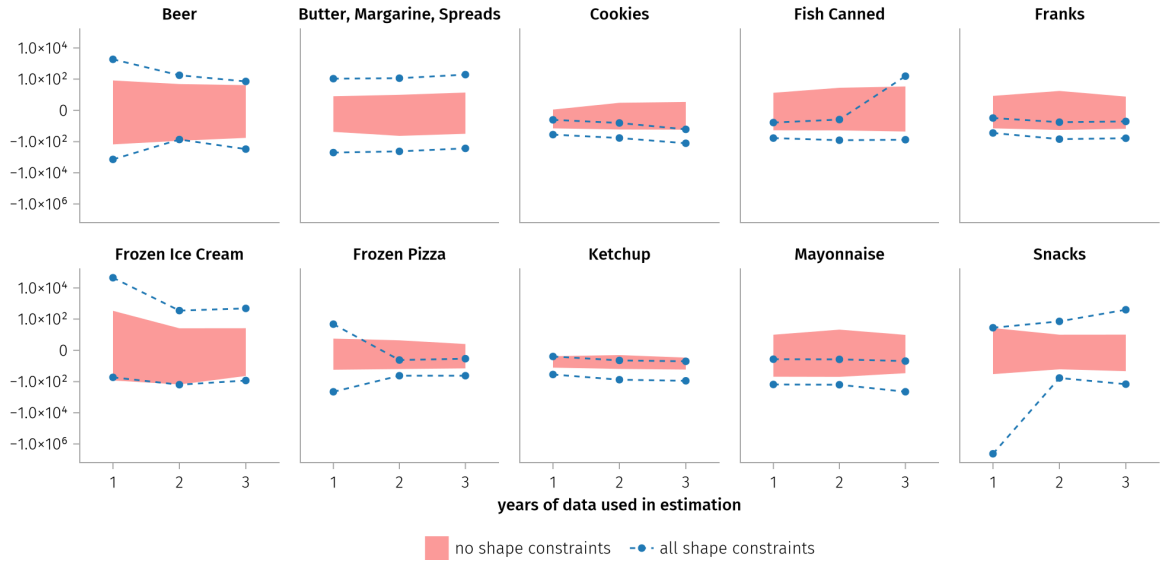


Figure C7: Median Own-Price Elasticity Estimates and 95% Credible Intervals

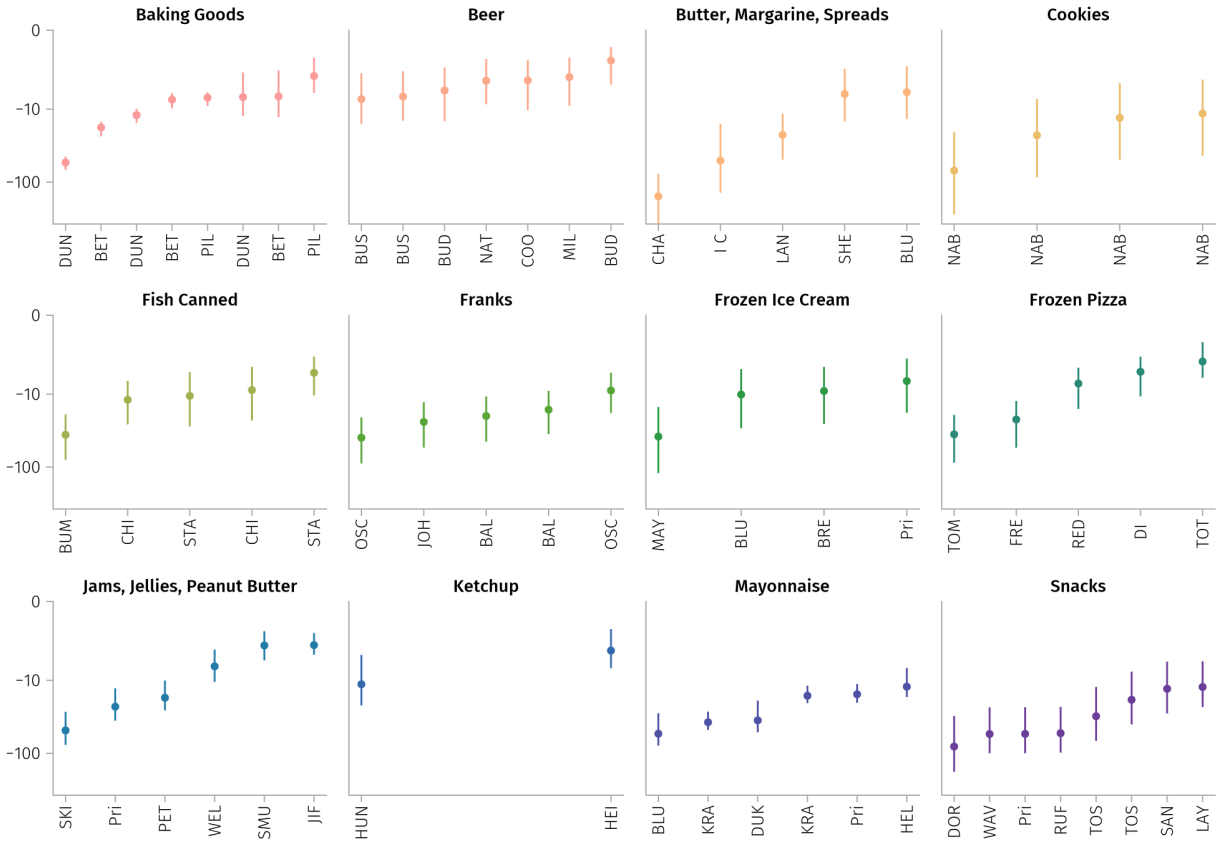


Figure C8: Own Elasticity Quantile Functions (Frozen Pizza)