RAND Journal of Economics
Vol. 51, No. 4, Winter 2020
pp. 1108–1161

Best practices for differentiated products demand estimation with PyBLP

Christopher Conlon*

and

Jeff Gortmaker**

Differentiated products demand systems are a workhorse for understanding the price effects of mergers, the value of new goods, and the contribution of products to seller networks. Berry, Levinsohn, and Pakes (1995) provide a flexible random coefficients logit model which accounts for the endogeneity of prices. This article reviews and combines several recent advances related to the estimation of BLP-type problems and implements an extensible generic interface via the PyBLP package. Monte Carlo experiments and replications suggest different conclusions than the prior literature: multiple local optima appear to be rare in well-identified problems; good performance is possible even in small samples, particularly when "optimal instruments" are employed along with supply-side restrictions. If Python is installed on your computer, PyBLP can be installed with the following command*:* `pip install pyblp`. Up-to-date documentation for the package is available at *https://pyblp.readthedocs.io.*

1. Introduction

■ Empirical models of supply and demand for differentiated products are one of the most important achievements of the New Empirical Industrial Organization (NEIO) literature of the last 30 years. The workhorse model is the Berry, Levinsohn, and Pakes (1995) or BLP approach, which provides an estimator that allows for flexible substitution patterns across products, addresses the potential endogeneity of price, and also provides an algorithm for recovering that estimator. It has the advantage that it both scales well for large numbers of potential products, and can utilize both aggregated and dis-aggregated data. The BLP model and its variants have been used in a wide variety of applications: understanding the value of new goods (Petrin, 2002), evaluating the price effects of mergers (Nevo, 2001, 2000a), and studying two-sided markets (Fan, 2013; Lee, 2013). The BLP approach has been applied to a wide number of different questions and industries including hospital demand and negotiations with insurers (Ho and Pakes,

* New York University, Stern School of Business; cconlon@stern.nyu.edu.
** Harvard University; jgortmaker@g.harvard.edu.
Thanks to Steve Berry, Jeremy Fox, Phil Haile, Mathias Reynaert, and Frank Verboven and seminar participants at NYU, Rochester, and the 2019 IIOC conference. Thanks to the editor Marc Rysman and to three anonymous referees. Daniel Stackman provided excellent research assistance. Any remaining errors are our own.

2014) and students' choice of schools (Bayer, Ferreira, and McMillan, 2007; Nielson, 2017). Moreover, the BLP approach has been extremely influential in the practice of prospective merger evaluation, particularly in recent years.

The model itself is both quite simple to understand, and quite challenging to estimate. At its core, it involves a nonlinear change of variables from the space of observed market shares to the space of mean utilities for products. After this nonlinear change of variables, the BLP problem is simply either a single linear instrumental variables (IV) regression problem (demand alone), or a two equation (supply and demand) linear IV regression problem. This means that a wide variety of tools for that class of problems are available to researchers.

The main disadvantage of the BLP estimator is that parameters governing the nonlinear change of variables are unknown. This results in a nonlinear, nonconvex optimization problem with a simulated (or approximated) objective function. The problem must be solved iteratively using nonlinear optimization software, and because of the nonconvexity, there is no mathematical guarantee that a solution will always be found. This has led to some frustration with the BLP approach (see Knittel and Metaxoglou, 2014). There is also the fear that when estimation is slow or complicated, researchers may cut corners in undesirable ways and sacrifice modeling richness for computational speed.

Despite its popularity, this literature lacks a standardized implementation that is sufficiently general to encompass a wide range of potential problems and use cases. Instead, nearly every researcher implements the estimator on their own with problem-specific tweaks and adjustments. This makes replication extremely challenging, and also makes it hard to evaluate different methodological and statistical improvements to the estimator.

The goal of this article is to present best practices for the estimation of BLP-type models, some of which are well-known in the literature, others of which are lesser known, and others still are novel to this article. In addition to presenting these best practices, we provide a common framework, PyBLP, which offers a general implementation of the BLP approach as a Python package.[1] We recommend installing PyBLP on top of an Anaconda distribution, which comes pre-packaged with PyBLP's primary dependencies.[2] Users of other languages such as MATLAB, Julia, and R can easily use PyBLP from their language of choice with packages that allow for between-language interoperability.[3] The PyBLP software is general, extensible, and open-source so that it can be modified and extended by scholars as new methodological improvements become available. The hope is that these best practices, along with this standardized and extensible software implementation, reduce some of the barriers to BLP-type estimation, making these techniques accessible to a wider range of researchers and facilitating replication of existing results.

This article and the accompanying package build upon a growing literature focused on methodological innovations and econometric properties of the BLP estimator. In Section 3, we discuss several such improvements and evaluate them with Monte Carlo studies in Section 5. Our objective is to compare practices in the literature and arrive at best practices suitable for a large number of use cases. We then implement these best practices as defaults in PyBLP. We organize the best practices around several of the tasks in the BLP estimator: solving the fixed point, optimization, integration, and solving counterfactual pricing equilibria.

[1] We chose Python over other languages because its popularity is growing, its package management systems are well-established, it has a mature scientific computing ecosystem, and as a general purpose language, it is conducive to the development of larger projects.

[2] PyBLP depends on standard packages in Python's scientific ecosystem: NumPy, SciPy, SymPy, and Patsy. It also depends on a companion package PyHDFE (Gortmaker and Tarascina, 2020), which implements algorithms for absorbing high-dimensional fixed effects.

[3] Python can be called from MATLAB with the `py` command, from Julia with `PyCall` (Johnson, 2019), and from R with `reticulate` (Allaire et al., 2017), which we give an example of in Section 6. Scientific computing in all of these high-level languages is backed by similar implementations of numerical linear algebra routines such as LAPACK.

In addition to best practices, we also provide some novel results. We provide a slightly different characterization of the BLP problem in Section 2 which facilitates estimation with simultaneous supply and demand restrictions. We show how this characterization can be made amenable to large numbers of fixed effects, and in Section 4 we characterize an approximation to optimal instruments in the spirit of Amemiya (1977) or Chamberlain (1987). Our characterization of the problem under optimal instruments allows us explore parametric identification with supply and demand in a way that parallels Berry and Haile (2014) and makes explicit cross equation and exclusion restrictions.

On the matter of instruments, our results generally coincide with and build on the existing literature. Gandhi and Houde (2019) construct what they refer to as *differentiation IV*, whereas Reynaert and Verboven (2014) evaluate a *feasible approximation to the optimal IV* in the sense of Amemiya (1977) or Chamberlain (1987).[4] We provide routines to construct both sets of instruments. Our results with respect to differentiation IV are mostly consistent with Gandhi and Houde (2019) in that they outperform other simple, yet commonly used forms of the *BLP instruments* (functions of exogenous product characteristics) such as sums or averages. Our results with respect to approximate optimal instruments are broadly similar to those of Reynaert and Verboven (2014) in that the performance gains under correctly specified supply models are substantial both in terms of bias and efficiency.

Our simulations indicate somewhat more positive results than previously observed in the literature when additional moments from correctly specified models of supply are also included. These findings are somewhat different from those of Reynaert and Verboven (2014), which suggest that once optimal demand-side instruments are included, the addition of a supply side has limited benefit.[5] The explanation for this phenomenon is directly related to our theoretical result in Section 4. Indeed, our simulations indicate that with both a correctly specified supply side and optimal instruments, the finite sample performance of the estimator is good even with relatively weak excluded cost-shifters, which supports the "folklore" around the original Berry, Levinsohn, and Pakes (1995) article: supply restrictions are valuable in improving the econometric performance of the estimator.

Employing best practices and the PyBLP software, we are able to revisit recent findings regarding methodological issues and innovations in BLP-type estimators in large-scale Monte Carlo experiments. Although many of our results confirm previous findings in the literature, we arrive at different conclusions on several occasions. The findings of Knittel and Metaxoglou (2014) suggest that the BLP problem is often characterized by a large number of local optima, and that these local optima can produce a wide range of potential elasticities and welfare effects. In contrast, our experience is that after implementing best practices, parameter estimates and counterfactual predictions are quite stable across starting values and choices of optimization software, both open-source and commercial. Likewise, Armstrong (2016) finds that absent strong cost shifting instruments, as the number of products increases, BLP instruments (characteristics of own and competing products) become weak, and it becomes difficult to reliably estimate demand parameters. Our findings suggest that with a correctly specified supply side and approximations to the optimal instruments, parameter estimates can still be estimated precisely. In general, we struggle to replicate some of the difficulties found in the previous literature, suggesting that the finite sample performance of BLP estimators may be better than previously thought.

There is also a recent literature of alternative approaches to BLP problems employing different algorithms or statistical estimators, which we do not directly address. This is not meant to suggest that there is anything wrong with these approaches, but merely that they are beyond the scope of this article. For example, Dubé, Fox, and Su (2012) propose an alternative estimation algorithm based on the mathematical programming with equilibrium constraints (MPEC)

[4] It is worth mentioning that the actual optimal IV are well-known to be infeasible. See Berry, Levinsohn, and Pakes (1995) or Berry, Levinsohn, and Pakes (1999).

[5] This is most likely because Reynaert and Verboven (2014) examine scenarios with strong cost-shifters.

TABLE 1 Model Notation

j	Products	p_{jt}	Price
t	Markets	c_{jt}	Marginal cost
i	"Individuals"	x_{jt}	Exogenous product characteristic
f	Firms	v_{jt}	Excluded demand-shifter
h	Nesting groups	w_{jt}	Excluded supply-shifter
N	Set/number of products across all markets	U_{ijt}	Indirect utility
T	Set/number of markets	δ_{jt}	Mean utility
J_t	Set/number of products in market t	μ_{ijt}	Heterogeneous utility
I_t	Set/number of individuals in market t	ϵ_{ijt}	Idiosyncratic preference
F_t	Set/number of firms in market t	d_{ijt}	Choice indicator
J_{ft}	Set/number of products of firm f in market t	s_{ijt}	Choice probability
H	Set/number of nesting groups	s_{jt}	Calculated market share
J_{ht}	Set/number of products in nest h and market t	$\mathcal{S}_{jt}$	Observed market share
		ξ_{jt}	Demand-side structural error
θ	Parameters with dimension K		
θ_1	Linear demand-side parameters with dimension K_1	$\mathcal{H}_t$	Ownership or holdings matrix
θ_2	Nonlinear common parameters with dimension K_2	Δ_t	Intra-firm demand derivatives
θ_3	Linear supply-side parameters with dimension K_3	η_{jt}	Multi-product Bertrand markup
		ω_{jt}	Supply-side structural error
β	θ_1 excluding fixed effects		
$\widetilde{\theta}_2$	Parameters in θ_2 governing heterogeneity	Z	Instruments
α	Parameter in θ_2 on price	W	Weighting matrix
γ	θ_3 excluding fixed effects	g	Sample moments
ρ	Nesting parameter in θ_2	q	Objective function

method of Su and Judd (2012) and which Conlon (2017) extends to generalized empirical likelihood estimators. Although the MPEC approach has some advantages, we focus on the more popular nested fixed point problem. Lee and Seo (2015) provide an *approximate BLP* estimator, which is asymptotically similar to the BLP estimator though differs in finite sample. Hong, Li, and Li (Forthcoming) propose a Laplace-type estimator (LTE). Salanie and Wolak (2019) propose another approximate estimator that can be estimated with linear IV, and is helpful for constructing good starting values for optimization. Other common modifications to the BLP model that are beyond the scope of this article include the *pure characteristics* model of Berry and Pakes (2007) and the inclusion of *micro moments*, as in Petrin (2002) and Berry, Levinsohn, and Pakes (2004a).[6]

2. Model and estimation

■ In Table 1 we summarize the notation that we will introduce in this section. Bold font denotes a $J_t \times 1$ vector for all products within a market. For example, $\boldsymbol{s}_t$ and $\boldsymbol{p}_t$ denote vectors of shares and prices for all products in market t. For clarity, we partition the parameter space θ into three parts: the $K_1 \times 1$ vector θ_1 contains the demand parameters β, the $K_3 \times 1$ vector θ_3 contains the supply parameters γ, and the remaining parameters, including the price coefficient α and parameters governing heterogeneous tastes $\widetilde{\theta}_2$, are contained in the $K_2 \times 1$ vector θ_2.

□ **Demand.** Berry, Levinsohn, and Pakes (1995) begin with the following problem. An individual i in market $t = 1, \ldots, T$ receives indirect utility from selecting a particular product j:

$$U_{ijt} = \delta_{jt} + \mu_{ijt} + \epsilon_{ijt}. \tag{1}$$

[6] PyBLP supports an approximation to the pure characteristics model and common forms of micro moments, but we do not analyze econometric performance of these modifications in this article.

Consumers then choose among $J_t = \{0, 1, \ldots, J_t\}$ discrete alternatives—including the outside alternative, denoted $j = 0$, which gives $U_{i0t} = \epsilon_{i0t}$ for all (i, t)—and select the option that provides the most utility:

$$d_{ijt} = \begin{cases} 1 \text{ if } U_{\text{ijt}} > U_{\text{ikt}} \text{ for all } k \neq j, \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

Aggregate market shares are given by integrating over heterogeneous consumer choices:[7]

$$s_{jt} = \int d_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it})\, \mathrm{d}\boldsymbol{\mu}_{it}\, \mathrm{d}\boldsymbol{\epsilon}_{it}.$$

When the ϵ_{ijt} are i.i.d. with the type I extreme value distribution,[8]

$$s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) = \int \frac{\exp(\delta_{jt} + \mu_{ijt})}{\sum_{k \in J_t} \exp(\delta_{kt} + \mu_{ikt})} f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\, \mathrm{d}\boldsymbol{\mu}_{it}. \tag{3}$$

This is often referred to as a *mixed logit* or *random coefficients logit* because each individual i's demands are given by a multinomial logit kernel where $f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)$ denotes the *mixing distribution* over the heterogeneous types i and $\widetilde{\theta}_2$ parameterizes this heterogeneity.[9] Indeed, θ_2 contains all of the parameters in the model related to the endogenous objects that are common to both supply and demand: the parameters $\widetilde{\theta}_2$ governing heterogeneous tastes $\boldsymbol{\mu}_{it}$ and the price coefficient α.

The key insight of Berry (1994) or Berry, Levinsohn, and Pakes (1995) is that we can perform a nonlinear change of variables: $\boldsymbol{\delta}_t \equiv D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$ where $\boldsymbol{\mathcal{S}}_t$ denotes the J_t vector of observed market shares (see Berry and Haile, 2014). For each market t, (3) represents a system of J_t equations in J_t unknowns $\boldsymbol{\delta}_t$. Given the $\boldsymbol{\delta}_t(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$ that solves that system of equations; along with some covariates x_{jt} and v_{jt}, prices p_{jt}, and a structural error ξ_{jt}; under an additivity assumption, one can write the index:[10]

$$\delta_{jt}\left(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2\right) = [x_{jt}, v_{jt}]\beta - \alpha p_{jt} + \xi_{jt}. \tag{4}$$

With the addition of some instruments Z_{jt}^D, which include the exogenous regressors x_{jt} and v_{jt}, one can construct moment restriction conditions of the form $E[\xi_{jt} Z_{jt}^D] = 0$.

□ **Supply.** We can derive an additional set of supply moments from the first order conditions of firms. The conventional approach assumes that multi-product oligopoly firms simultaneously set prices independently for each market t.

Consider the profits of firm f, which for a single market t controls several products J_{ft} and sets prices p_{jt}. We take the first-order conditions of the profit function as follows:

$$\begin{aligned} &\max_{p_{jt}\,:\, j \in J_{ft}} \sum_{j \in J_{ft}} s_{jt}(\boldsymbol{p}_t) \cdot (p_{jt} - c_{jt}), \\ &s_{jt}(\boldsymbol{p}_t) + \sum_{k \in J_{ft}} \frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t) \cdot (p_{kt} - c_{kt}) = 0. \end{aligned}$$

It is helpful to write the first-order conditions in matrix form so that for a single market t,

$$\begin{aligned} \boldsymbol{s}_t(\boldsymbol{p}_t) &= \Delta_t(\boldsymbol{p}_t) \cdot (\boldsymbol{p}_t - \boldsymbol{c}_t), \\ \underbrace{\Delta_t(\boldsymbol{p}_t)^{-1} \boldsymbol{s}_t(\boldsymbol{p}_t)}_{\eta_t(\boldsymbol{p}_t, \boldsymbol{s}_t, \theta_2)} &= \boldsymbol{p}_t - \boldsymbol{c}_t. \end{aligned} \tag{5}$$

[7] Here we define $\boldsymbol{\delta}_t$, $\boldsymbol{\mu}_{it}$, and $\boldsymbol{\epsilon}_{it}$ as the $J_t \times 1$ vectors with elements δ_{jt}, μ_{ijt}, and ϵ_{ijt}.

[8] Identification generally requires normalizing one of the options. As mentioned above, the typical choice is to normalize indirect utility from the outside option: $U_{i0t} = \varepsilon_{i0t}$.

[9] McFadden and Train (2000) show that any random utility model (RUM) can be approximated with some mixture of multinomial logits with a sufficient basis of characteristics x_{jt}. The *mixed multinomial logit* itself dates back to at least Boyd and Mellman (1980) and Cardell and Dunbar (1980).

[10] Well-known special cases are logit without any parameters, in which $D_t^{-1} = \log s_{jt} - \log s_{0t}$, and nested logit with one parameter, in which $D_t^{-1} = \log s_{jt} - \log s_{0t} - \rho \log s_{j|ht}$ where $s_{j|ht}$ is the market share of j in its nest h.

Here the multi-product Bertrand markup $\boldsymbol{\eta}_t(\boldsymbol{p}_t, \boldsymbol{s}_t, \theta_2)$ depends on $\Delta_t(\boldsymbol{p}_t, \boldsymbol{s}_t, \theta_2)$, a $J_t \times J_t$ matrix of intra-firm demand derivatives given by:

$$\Delta_t(\boldsymbol{p}_t) \equiv -\mathcal{H}_t \odot \frac{\partial \boldsymbol{s}_t}{\partial \boldsymbol{p}_t}(\boldsymbol{p}_t), \tag{6}$$

which is the element-wise Hadamard product of two $J_t \times J_t$ matrices: the matrix of demand derivatives with each (j,k) entry given by $\frac{\partial s_{jt}}{\partial p_{kt}}$ and the holdings or ownership matrix $\mathcal{H}_t$ with each (j,k) entry indicating whether the same firm produces products j and k.[11]

This enables us to recover an estimate of marginal costs $c_{jt} = p_{jt} - \eta_{jt}(\theta_2)$, which in turn allows us to construct additional *supply side moments*. We can parametrize marginal cost as[12]

$$f_{MC}(p_{jt} - \eta_{jt}(\theta_2)) = f_{MC}(c_{jt}) = x_{jt}\gamma_1 + w_{jt}\gamma_2 + \omega_{jt} \tag{7}$$

and construct moment conditions of the form $E[\omega_{jt} Z^S_{jt}] = 0$. The idea is that we can use observed prices, along with information on demand derivatives and firm conduct, to recover markups η_{jt} and then marginal costs c_{jt}. This also imposes a functional form restriction on marginal costs, which depends on both product characteristics x_{jt} and the marginal cost shifters w_{jt} that are excluded from demand.

Some researchers may wish to allow marginal costs to depend on the quantity sold (either in market share or total units). This was true in the original Berry, Levinsohn, and Pakes (1995) article, which allowed $\log c_{jt}$ to depend on $\log q_{jt}$ to allow for returns to scale. PyBLP allows for this, but we do not explore quantity dependent marginal costs in our simulations. See Appendix A.3 for more information.

□ **The estimator.** We can construct a GMM estimator using our supply and demand moments. To do so, we stack their sample analogues to form

$$g(\theta) = \begin{bmatrix} g_D(\theta) \\ g_S(\theta) \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{j,t} \xi_{jt} Z^D_{jt} \\ \frac{1}{N}\sum_{j,t} \omega_{jt} Z^S_{jt} \end{bmatrix}$$

and construct a nonlinear GMM estimator for $\theta = [\beta, \alpha, \widetilde{\theta}_2, \gamma]$ with some weighting matrix W:[13]

$$\min_{\theta} q(\theta) \equiv g(\theta)' W g(\theta). \tag{8}$$

[11] Each (j,k) entry equals 1 if both $j,k \in J_{ft}$ for some $f \in F_t$, and equals 0 otherwise. We can easily consider alternative forms of conduct such as Single- or Multi-Product Oligopoly, or Monopoly. Miller and Weinberg (2017) consider estimating a single parameter $\mathcal{H}_t(\kappa)$ and Backus, Conlon, and Sinkinson (2020) use PyBLP test various forms of $\mathcal{H}_t(\kappa)$.

[12] The most common choice for $f_{MC}(\cdot)$ is the identity function. Some authors also consider $f_{MC}(\cdot) = \log(\cdot)$. In practice this constrains marginal costs to be always positive.

[13] Some of the literature implicitly scales the objective by N^2. For example, Nevo (2000b) defines the objective for the demand-only problem as $q(\theta) = \xi' Z_D W Z_D' \xi$. Unless stated otherwise, we leave objective values unscaled throughout this article.

The θ_2 parameters are common to both supply and demand, govern the endogeneous objects, and require at least one excluded instrument each.[14] To be explicit we write the entire program as follows:

$$\begin{aligned}
\min_{\theta} q(\theta) &\equiv g(\theta)' W g(\theta), \\
g(\theta) &= \begin{bmatrix} \frac{1}{N}\sum_{j,t} \xi_{jt} Z^D_{jt} \\ \frac{1}{N}\sum_{j,t} \omega_{jt} Z^S_{jt} \end{bmatrix}, \\
\xi_{jt} &= \delta_{jt} - [x_{jt}, v_{jt}]\beta + \alpha p_{jt}, \\
\omega_{jt} &= f_{MC}(p_{jt} - \eta_{jt}) - [x_{jt}, w_{jt}]\gamma, \\
\boldsymbol{\eta}_t &= \Delta_t(\theta_2)^{-1} \boldsymbol{s}_t, \\
\mathcal{S}_{jt} = s_{jt}(\boldsymbol{\delta}_t, \theta_2) &\equiv \int \frac{\exp(\delta_{jt} + \mu_{ijt})}{\sum_{k \in J_t} \exp(\delta_{kt} + \mu_{ikt})} f(\boldsymbol{\mu}_{it} | \widetilde{\theta}_2) \, \mathrm{d}\boldsymbol{\mu}_{it}.
\end{aligned} \tag{9}$$

This estimator and its econometric properties are discussed in Berry, Levinsohn, and Pakes (1995) and Berry, Linton, and Pakes (2004b). Our focus is not going to be on the econometric properties of $\hat{\theta}$ but on rather various algorithms by which one might obtain $\hat{\theta}$. Technically, we need to solve this program twice. Once to obtain a consistent estimate for W and a second time to obtain the efficient GMM estimator.

Many applied articles omit the supply side from (9), and instead estimate $\theta = [\theta_1, \theta_2]$ using demand moments alone, which is what PyBLP will do if the user does not provide a supply side. An important justification for not including the supply side is that it may be misspecified if the researcher does not know (or is not willing to assume) either the functional form of marginal costs $f_{MC}(\cdot)$ or firm conduct $\mathcal{H}_t$.[15] The program without a supply side is as follows:

$$\begin{aligned}
\min_{\theta} q_D(\theta) &\equiv g_D(\theta)' W g_D(\theta), \\
g_D(\theta) &= \frac{1}{N}\sum_{j,t} \xi_{j,t} Z^D_{jt}, \\
\xi_{jt} &= \delta_{jt} - [x_{jt}, v_{jt}]\beta + \alpha p_{jt}, \\
\mathcal{S}_{jt} = s_{jt}(\boldsymbol{\delta}_t, \theta_2) &\equiv \int \frac{\exp(\delta_{jt} + \mu_{ijt})}{\sum_{k \in J_t} \exp(\delta_{kt} + \mu_{ikt})} f(\boldsymbol{\mu}_{it} | \widetilde{\theta}_2) \, \mathrm{d}\boldsymbol{\mu}_{it}.
\end{aligned} \tag{10}$$

□ **The nested fixed point algorithm.** In addition to providing an estimator, Berry, Levinsohn, and Pakes (1995) provide an algorithm for solving (9), which attempts to simplify the problem. Parameters on exogenous regressors enter the problem linearly; we concentrate out $[\theta_1, \theta_3]$ and perform a nonlinear search over just θ_2 because $[\hat{\theta}_1(\theta_2), \hat{\theta}_3(\theta_2)]$ are implicit functions of other parameters. Our modified algorithm is given below.

Our setup differs slightly from many BLP applications.[16] A key distinction occurs in step (d) where αp_{jt} appears on the left-hand side of (11). The second distinction occurs in step (f), which requires that we stack the supply and demand equations appropriately. We provide a detailed

[14] Berry and Haile (2014) show that $D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$ depends on the endogenous market shares of all products within the market, thus each parameter in $\widetilde{\theta}_2$ requires an additional instrument.

[15] There are other cases where the supply side may be misspecified. Many of these stem from misspecification around the functional form of $\boldsymbol{\eta}_t$ or $\mathcal{H}_t$. Important examples include: possible collusion, Cournot rather than Bertrand competition and double marginalization (Bonnet and Dubois, 2010; Villas-Boas, 2007).

[16] Arguably it is more in line with the original Berry, Levinsohn, and Pakes (1995) article.

Algorithm 1 Nested Fixed Point

For each guess of θ_2:

(a) For each market t, solve $\mathcal{S}_{jt} = s_{jt}(\boldsymbol{\delta}_t, \theta_2)$ for $\hat{\boldsymbol{\delta}}_t(\boldsymbol{\mathcal{S}}_t, \theta_2) \equiv \hat{\boldsymbol{\delta}}_t(\theta_2)$.

(b) For each market t, use the $J_t \times 1$ vector $\hat{\boldsymbol{\delta}}_t(\theta_2)$ to construct the $J_t \times J_t$ matrix $\Delta_t(\boldsymbol{p}_t, \hat{\boldsymbol{\delta}}_t(\theta_2), \theta_2)$

(c) For each market t, recover $\hat{\boldsymbol{\eta}}_t(\theta_2) = \Delta_t(\hat{\boldsymbol{\delta}}_t(\theta_2), \theta_2)^{-1}\boldsymbol{\mathcal{S}}_t$ by solving the $J_t \times J_t$ linear system.

(d) Stack up $\hat{\boldsymbol{\delta}}_t(\mathcal{S}_{jt}, \theta_2)$ and $\hat{c}_{jt}(\hat{\boldsymbol{\delta}}_t(\theta_2), \theta_2) = f_{MC}(p_{jt} - \hat{\eta}_{jt}(\hat{\boldsymbol{\delta}}_t(\theta_2), \theta_2))$ and use linear IV-GMM to recover $[\hat{\theta}_1(\theta_2), \hat{\theta}_3(\theta_2)]$ following the recipe in Appendix A.1. The following is our somewhat different formulation:

$$\begin{aligned} \hat{\delta}_{jt}(\boldsymbol{\mathcal{S}}_t, \theta_2) + \alpha p_{jt} &= [x_{jt}, v_{jt}]\beta + \xi_{jt}, \\ f_{MC}(p_{jt} - \hat{\eta}_{jt}(\theta_2)) &= [x_{jt}, w_{jt}]\gamma + \omega_{jt}. \end{aligned} \tag{11}$$

(e) Construct the residuals:

$$\begin{aligned} \hat{\xi}_{jt}(\theta_2) &= \hat{\delta}_{jt}(\theta_2) - [x_{jt}, v_{jt}]\hat{\beta}(\theta_2) + \alpha p_{jt}, \\ \hat{\omega}_{jt}(\theta_2) &= \hat{c}_{jt}(\theta_2) - [x_{jt}, w_{jt}]\hat{\gamma}(\theta_2). \end{aligned} \tag{12}$$

(f) Stack the sample moments:

$$g(\theta_2) = \begin{bmatrix} \frac{1}{N}\sum_{jt} \hat{\xi}_{jt}(\theta_2) Z^D_{jt} \\ \frac{1}{N}\sum_{jt} \hat{\omega}_{jt}(\theta_2) Z^S_{jt} \end{bmatrix}. \tag{13}$$

(g) Construct the GMM objective: $q(\theta_2) = g(\theta_2)' W g(\theta_2)$.

derivation in Appendix A.1. Later in Section 3, we show how this setup can be adapted to incorporate fixed effects when supply and demand are estimated simultaneously. This requires that the endogenous markup term $\eta_{jt}(\boldsymbol{s}_t, \boldsymbol{p}_t, \boldsymbol{\delta}_t(\theta_2), \theta_2)$ can be written as a function of only the θ_2 parameters and does not depend on $[\theta_1, \theta_3]$.[17]

We provide analytic gradients for the BLP problem with supply and demand in Appendix A.2. One advantage of the BLP algorithm is that is performs a nonlinear search over only K_2 *nonlinear parameters*. Consequently, the Hessian matrix is only $K_2 \times K_2$. This implies relatively minimal memory requirements. Also, the IV-GMM regression in step (d) concentrates out the *linear* parameters $[\theta_1, \theta_3]$. This implies that large numbers of linear parameters can be estimated essentially for free, which is important if one includes a large number of fixed effects such as product- or market-level fixed effects.[18] In fact, other than (a) the remaining steps are computationally trivial. As is well known, (a)-(c) can be performed separately for each market across multiple processors.

The main disadvantage is that all parameters are implicit functions of other parameters, particularly of θ_2. The objective is a complicated implicit function of θ_2. Once we incorporate any heterogeneous tastes, the resulting optimization problem is nonconvex. In general the complexity of this problem grows rapidly with the number of nonlinear θ_2 parameters, whereas a high number of linear $[\theta_1, \theta_3]$ parameters are more or less for free.

□ **Nested logit and RCNL variants.** The random coefficients nested logit (RCNL) model of Brenkers and Verboven (2006) instead assumes that ϵ_{ijt} are not i.i.d. but rather follow the assumptions of a two-level nested logit. This model is popular in applications where the most important set of characteristics governing substitution is categorical. This has made it popular in studies of alcoholic beverages such as distilled spirits (Conlon and Rao, 2017; Miravete, Seim, and Thurk, 2018) and beer (Miller and Weinberg, 2017).

Much like the random coefficients model integrates over a heterogeneous distribution where each individual type follows a logit distribution, the RCNL model integrates over a heterogeneous distribution where each individual now follows a *nested logit*. Within group correlation is

[17] Why? We already know we can invert the shares to solve for $\boldsymbol{\delta}_t(\theta_2)$, and the matrix of demand derivatives $\frac{\partial s_{kt}}{\partial p_{jt}} = -\int \alpha_i s_{ikt}(\boldsymbol{\delta}_t(\theta_2), \boldsymbol{\mu}_{it})[1(j=k) - s_{ijt}(\boldsymbol{\delta}_t(\theta_2), \boldsymbol{\mu}_{it})] f(\boldsymbol{\mu}_{it}, \alpha_i|\theta_2)\, \mathrm{d}\boldsymbol{\mu}_{\mathrm{it}}$ again depends only on θ_2, which contains the price coefficient α.

[18] For example, Nevo (2001, 2000b) includes product fixed effects.

governed by a new parameter ρ. We expand our definition of θ_2 to include the nesting parameter ρ so that $\theta_2 \equiv [\alpha, \rho, \widetilde{\theta}_2]$:[19]

$$U_{ijt} = \delta_{jt} + \mu_{ijt}(\widetilde{\theta}_2) + \epsilon_{ijt}(\rho).$$

The primary difference from the nested logit is that the inclusive value term for all products J_{ht} in nest $h \in H = \{1, \ldots, H\}$ now depends on the consumer's type i:

$$s_{jt}(\boldsymbol{\delta}_t, \theta_2) = \int \frac{\exp[(\delta_{jt} + \mu_{ijt})/(1-\rho)]}{\exp[IV_{iht}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it})/(1-\rho)]} \cdot \frac{\exp IV_{iht}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it})}{1 + \sum_{h \in H} \exp IV_{iht}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it})} f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\, \mathrm{d}\boldsymbol{\mu}_{it}, \quad (14)$$

in which the inclusive value term is

$$IV_{iht}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}) = (1-\rho) \log \sum_{j \in J_{ht}} \exp\left(\frac{\delta_{jt} + \mu_{ijt}}{1-\rho}\right).$$

A challenge for estimation is that $\boldsymbol{\delta}_t \leftarrow\!\!\shortmid \boldsymbol{\delta}_t + \log \mathcal{S}_t - \log s_t(\boldsymbol{\delta}_t, \theta_2)$ is no longer a contraction. Instead, the contraction must be dampened:[20]

$$\boldsymbol{\delta}_t \leftarrow\!\!\shortmid \boldsymbol{\delta}_t + (1-\rho)[\log \mathcal{S}_t - \log s_t(\boldsymbol{\delta}_t, \theta_2)]. \quad (15)$$

This creates an additional challenge because the rate of convergence for the contraction in (15) can become arbitrarily slow as $\rho \to 1$.[21] Thus as more consumers substitute within the nest, this model becomes much harder to estimate. Our simulations will demonstrate that this can be problematic.

As is well known, the relation in (15) has an analytic solution in the absence of random coefficients when $\mu_{ijt} = 0$. This model reduces to the regular nested logit for which the following expression was derived in Berry (1994):

$$\delta_{jt} = \log \mathcal{S}_{jt} - \log \mathcal{S}_{0t} - \rho \log \mathcal{S}_{j|ht}$$

where $\mathcal{S}_{j|ht}$ is the market share of j in its nest h.

3. Algorithmic improvements

■ In each subsection, we review several methods from the literature, including some of our own design. Although we make many of these methods available as options in PyBLP, our focus is on finding best practices that are fastest and most reliable for most users. Later, we provide support for these decisions with our Monte Carlo studies.

□ **Incorporating many fixed effects.** There is a long tradition of extending the demand side utility to incorporate product or market fixed effects. For example, Nevo (2001, 2000a) allows for product fixed effects ξ_j so that

$$\delta_{jt} = [x_{jt}, v_{jt}]\beta - \alpha p_{jt} + \xi_j + \Delta\xi_{jt}.$$

These can manageably be incorporated as dummy variables in the linear part of the regression as there are only $J = 24$ products in these articles.

With weekly UPC-store level Nielsen scanner data it is not uncommon for there to be $J_t > 3500$ products (true in both distilled spirits and ready-to-eat cereal). There are approximately $T = 500$ weeks t of Nielsen scanner data from 2006 to 2016. Incorporating store-week

[19] The nesting parameter can also be indexed as ρ_h so as to vary by group h. We support both types of nesting parameters in PyBLP.

[20] See Grigolon and Verboven (2014) for a derivation. The expression in (15) does not precisely match Grigolon and Verboven (2014) because of a minor typesetting error.

[21] This can be formalized in terms of the modulus of the contraction mapping or the Lipschitz constant. See Dubé, Fox, and Su (2012) for more details.

fixed effects ξ_{st} for only 100 stores could reach the order of 50,000 such fixed effects. Allowing for UPC-store fixed effects ξ_{js} can imply 100,000 or more fixed effects.[22] Clearly, the *least squares dummy variable* (LSDV) approach will not scale with tens or hundreds of thousands of fixed effects. We might consider the *within transformation* to remove the fixed effects, though we cannot directly incorporate both a within transformation and a supply side without re-writing the problem because of endogenous prices p_{jt}. We show how to re-write the problem in Appendix A.1. Define Y_{jt}^D, Y_{jt}^S, X_{jt}^D, and X_{jt}^S as follows:

$$Y_{jt}^D \equiv \hat{\delta}_{jt}(\theta_2) + \alpha p_{jt} = [x_{jt}, v_{jt}]\beta + \xi_{jt} \equiv X_{jt}^D\beta + \xi_{jt},$$
$$Y_{jt}^S \equiv p_{jt} - \hat{\eta}_{jt}(\theta_2) = [x_{jt}, w_{jt}]\gamma + \omega_{jt} \equiv X_{jt}^S\gamma + \omega_{jt}.$$

Stacking the system yields

$$\underbrace{\begin{bmatrix} Y_D \\ Y_S \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} X_D & 0 \\ 0 & X_S \end{bmatrix}}_{X} \begin{bmatrix} \beta \\ \gamma \end{bmatrix} + \begin{bmatrix} \xi \\ \omega \end{bmatrix}. \tag{16}$$

After re-arranging terms and re-stacking, this is just a conventional linear IV problem in terms of (Y, X) where the endogenous parameters have been incorporated into Y. This means that the *within transform* can be used to absorb a single-dimensional fixed effect. Consider two dimensions of fixed effects $j = 1, \ldots, J$ and $t = 1, \ldots, T$:

$$\widetilde{Y}_{jt} = Y_{jt} - \overline{Y}_{j\cdot} - \overline{Y}_{\cdot t},$$
$$\widetilde{X}_{jt} = X_{jt} - \overline{X}_{j\cdot} - \overline{X}_{\cdot t}.$$

The simplest approach might be to *iteratively demean*: remove the product mean $\overline{X}_{j\cdot}$, update X_{jt}, remove the market mean $\overline{X}_{\cdot t}$, and repeat this process until $\overline{X}_{j\cdot} = \overline{X}_{\cdot t} = 0$. This *method of alternating projections* (MAP) can be done in a single iteration if $\text{Cov}(\overline{X}_{\cdot t}, \overline{X}_{j\cdot}) = 0$. However, if the two dimensions of fixed effects are correlated this can require many iterations and can be quite burdensome.

The LSDV approach handles the burden of correlation but requires constructing the annihilator matrix to remove all the fixed effects. This approach requires inverting a $(J + T) \times (J + T)$ matrix. Constructing and inverting such a large matrix is often infeasible because of memory requirements. Several algorithms have been proposed to deal with this problem. The most popular algorithm is perhaps that of Correia (2016), based on the accelerated MAP approach of Guimarães and Portugal (2010).[23]

The BLP application is unusual in that we re-run regressions using the same X variables many times. However, the left hand side Y variables $\hat{\delta}_{jt}(\theta_2) + \alpha p_{jt}$ and $\hat{c}(\theta_2)$ change with θ_2, which means the entire procedure needs to be repeated for each guess of θ_2. For more than a single dimension of fixed effects, PyBLP supports a number of different algorithms.[24] These algorithms have the advantage that the linear explanatory variables in X can be residualized only once, whereas the left hand side variables Y still need to be residualized for each guess of θ_2. The savings are particularly large if the dimensions of X_D or X_S are large.

□ **Solving for the shares.** The main challenge of the Nested Fixed Point (NFXP) algorithm is solving the system of market shares: $\mathcal{S}_{jt} = s_{jt}(\boldsymbol{\delta}_t, \theta_2)$. Because the NFXP approach holds θ_2

[22] See Backus, Conlon, and Sinkinson (2020) and Conlon and Rao (2017) for examples of product-chain or store-week fixed effects.

[23] The Correia (2016) algorithm is implemented for the linear IV model in the Stata command `ivreghdfe`.

[24] PyBLP supports a number of different MAP acceleration schemes, the LSMR solver of Fong and Saunders (2011), and for two-dimensional fixed effects, the algorithm of Somaini and Wolak (2016). Comparison of these different approaches is beyond the scope of this article—for a more in-depth discussion, refer to Correia (2016).

fixed, rather than solve a system of N nonlinear equations and N unknowns, we solve T systems of J_t equations and J_t unknowns in $\boldsymbol{\delta}_t$ in parallel.[25]

Consider a single market t where we search for the J_t vector $\boldsymbol{\delta}_t$ which satisfies:

$$\mathcal{S}_{jt} = s_{jt}(\boldsymbol{\delta}_t|\theta_2) = \int \frac{\exp(\delta_{jt} + \mu_{ijt})}{\sum_{k\in J_t}\exp(\delta_{kt} + \mu_{ikt})} f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\,\mathrm{d}\boldsymbol{\mu}_{it}. \tag{17}$$

Although mathematically there is a unique solution, it is impossible, numerically speaking, to choose a vector $\boldsymbol{\delta}_t$ that solves (17) exactly. Instead, we must solve the system of equations to some tolerance. We express the tolerance in terms of the log difference in shares:

$$\left\|\log \mathcal{S}_{jt} - \log s_{jt}(\boldsymbol{\delta}_t, \theta_2)\right\|_\infty \leq \epsilon^{tol}. \tag{18}$$

There is a tradeoff with regard to the tolerance of the *inner loop* in (18). If the tolerance is too loose, the numerical error propagates to the rest of the estimation routine.[26] It is also possible to set a tolerance which is too tight and thus can never be satisfied. This is particularly problematic when summing over a large number of elements. We prefer to set ϵ^{tol} between `1E-14` and `1E-12` as the *machine epsilon* or detectable difference between two double precision floating point numbers is around `1E-16` (on 64-bit architectures).

Jacobian based approaches. A direct approach would be to solve the system of J_t equations and J_t unknowns using Newton-type methods. Consider the following Newton-Raphson iteration:[27]

$$\boldsymbol{\delta}_t^{h+1} \leftarrow\!\!\!\!\rightarrow \boldsymbol{\delta}_t^h - \lambda \cdot \Psi_t^{-1}(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2) \cdot \boldsymbol{s}_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2). \tag{19}$$

Each Newton-Raphson iteration would require computation of the J_t vector of market shares $\boldsymbol{s}_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)$, the $J_t \times J_t$ Jacobian matrix $\Psi_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2) = \frac{\partial \boldsymbol{s}_t}{\partial \boldsymbol{\delta}_t}(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)$, and its inverse $\Psi_t^{-1}(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)$.

There are some alternative *quasi-Newton* methods which solve variants of (19). These variants generally involve modifying the step-size λ or approximating $\Psi_t^{-1}(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)$ in ways that avoid calculating the inverse Jacobian at each step. Quasi-Newton methods are often relatively fast when they work, though they need not converge (e.g., they may oscillate or not reach a resting point) and may be sensitive to starting values.[28]

Our experience indicates that the Levenberg–Marquardt (LM) algorithm is the fastest and most reliable Jacobian-based solution method.[29] LM minimizes the following least-squares problem in order to solve the following $J_t \times J_t$ system of nonlinear equations:

$$\min_{\boldsymbol{\delta}_t} \sum_{j\in J_t} [\mathcal{S}_{jt} - s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2)]^2.$$

The idea is to update our guess of $\boldsymbol{\delta}_t^h$ to $\boldsymbol{\delta}_t^h + \boldsymbol{x}_t$ where $\boldsymbol{x}_t$ is a $J_t \times 1$ vector. The LM update is given by the solution to the following linear system of equations:

$$[\Psi_t'\Psi_t + \lambda\,\mathrm{diag}(\Psi_t'\Psi_t)] \cdot \boldsymbol{x}_t = \Psi_t'[\boldsymbol{\mathcal{S}}_t - \boldsymbol{s}_t(\boldsymbol{\delta}_t, \theta_2)] \tag{20}$$

[25] This same idea provides the sparsity of share constraints in the MPEC approach of Dubé, Fox, and Su (2012).

[26] Dubé, Fox, and Su (2012) show how error propagates from (18) to the estimates of $\hat{\theta}$. Lee and Seo (2016) provide a more precise characterization of this error when using Newton's method.

[27] In practice it is generally faster to solve the linear system: $\Psi_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)(\boldsymbol{\delta}_t^{h+1} - \boldsymbol{\delta}_t^h) = -\boldsymbol{s}_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2)$.

[28] Though the BLP problem is a nonconvex system of nonlinear equations, there are some properties which make it amenable to quasi-Newton methods. The market share function $s_{jt}(\boldsymbol{\delta}_t|\widetilde{\theta}_2)$ is $\mathbb{C}^\infty$ with respect to $\boldsymbol{\delta}_t$, is bounded between (0,1), and agrees with its Taylor approximation at any $\boldsymbol{\delta}_t$. Fox et al. (2012) and Iaria and Wang (2019) establish the real analyticity of the mixtures of logits under different conditions on the mixing distribution. Within some *basin of attraction*, quasi-Newton methods will be quadratically convergent. For an example of a quasi-Newton solution to (19), see Houde (2012). Another useful property is that with a strictly positive outside good share, $\frac{\partial s_{0t}}{\partial p_{kt}} > 0$ for all k, which guarantees strict diagonal dominance of Ψ_t and hence that it is always nonsingular.

[29] Specifically we use the `lm` option of `scipy.optimize.root`, which calls the LM routine from `MINPACK` (More, Garbow, and Hillstrom, 1980).

where $\boldsymbol{x}_t$ is multiplied by an approximation to the Hessian. This has the advantage that for $\lambda = 0$ the algorithm takes a full Gauss-Newton step and for λ large it takes a step in the direction of the gradient. The additional diagonal term also guarantees the invertibility of the approximate Hessian, even as it becomes nearly singular.

As in all iterative solution methods there are two primary costs: the cost per iteration and the number of iterations until convergence. The cost per iteration is driven primarily by the cost of computing (rather than inverting) the Jacobian matrix which involves $J_t \times J_t$ numerical integrals.[30]

Fixed point approaches. Berry, Levinsohn, and Pakes (1995) also propose a fixed point approach to solve the $J_t \times J_t$ system of equations in (17). They show that the following fixed point relation $f(\boldsymbol{\delta}_t) = \boldsymbol{\delta}_t$ is a contraction mapping:[31]

$$f : \boldsymbol{\delta}_t^{h+1} \leftarrow\!\shortmid \boldsymbol{\delta}_t^h + \log \boldsymbol{\mathcal{S}}_t - \log \boldsymbol{s}_t(\boldsymbol{\delta}_t^h, \widetilde{\theta}_2). \tag{21}$$

This kind of contraction mapping is linearly convergent with a rate of convergence that is proportional to $L(\widetilde{\theta}_2)/[1 - L(\widetilde{\theta}_2)]$ where $L(\widetilde{\theta}_2)$ is the Lipschitz constant. Because (21) is a contraction, we know that $L(\widetilde{\theta}_2) < 1$. Dubé, Fox, and Su (2012) show that for the BLP contraction the Lipschitz constant is $L(\widetilde{\theta}_2) = \max_{\delta_t} ||I_{J_t} - \frac{\partial \log \boldsymbol{s}_t}{\partial \boldsymbol{\delta}_t}(\boldsymbol{\delta}_t, \widetilde{\theta}_2)||_\infty$.

A smaller Lipschitz constant implies that (21) converges more rapidly. Dubé, Fox, and Su (2012) show in simulation that all else being equal, a larger outside good share generally implies a smaller Lipschitz constant.[32] Conversely, as the outside good share becomes smaller, the convergence of the fixed point relationship takes increasingly many steps.

Accelerated fixed points. Given a fixed point relationship there may be faster ways to obtain a solution to $f(\boldsymbol{\delta}_t) = \boldsymbol{\delta}_t$ than direct iteration on the fixed point relation as in (21). There is a large literature on acceleration methods for fixed points. Most of these methods use information from multiple iterations ($\boldsymbol{\delta}_t^h$, $\boldsymbol{\delta}_t^{h+1}$, $\boldsymbol{\delta}_t^{h+2}$, $f(\boldsymbol{\delta}_t^h)$, $f(f(\boldsymbol{\delta}_t^h))$) to approximate Ψ_t or its inverse.[33]

Reynaerts, Varadhan, and Nash (2012) conduct extensive testing of various fixed point acceleration methods and find that the SQUAREM algorithm of Varadhan and Roland (2008) works well on the BLP contraction in (21). The intuition is to form a residual $\boldsymbol{r}^h$ which is determined by the change between the current iteration $\boldsymbol{\delta}_t^h$ and the next iteration $f(\boldsymbol{\delta}_t^h)$, as well as the change in the residual from this iteration to the next $\boldsymbol{v}^h$ to form an estimate of the Jacobian. The residual and the curvature can also be used to construct a step-size α^h. The exact algorithm is described below:[34]

$$\begin{gathered} \boldsymbol{\delta}_t^{h+1} \leftarrow\!\shortmid \boldsymbol{\delta}_t^h - 2\alpha^h \boldsymbol{r}^h + (\alpha^h)^2 \boldsymbol{v}^h, \\ \alpha^h = \tfrac{(\boldsymbol{v}^h)'\boldsymbol{r}^h}{(\boldsymbol{v}^h)'\boldsymbol{v}^h}, \quad \boldsymbol{r}^h = f(\boldsymbol{\delta}_t^h) - \boldsymbol{\delta}_t^h, \quad \boldsymbol{v}^h = f(f(\boldsymbol{\delta}_t^h)) - 2f(\boldsymbol{\delta}_t^h) + \boldsymbol{\delta}_t^h. \end{gathered} \tag{22}$$

In general the SQUAREM method is 3 to 6 times faster than direct iteration on the BLP contraction in (21). The idea is to take roughly the same number of steps as Newton-Raphson iteration, but to reduce the cost of steps by avoiding calculating the Jacobian directly. In fact, all of the terms in (22) are computed as a matter of course, because these are just iterations of $\boldsymbol{\delta}_t^h$ and $f(\boldsymbol{\delta}_t^h)$. Unlike direct iteration on (21), there is technically no convergence guarantee as the iteration on (22) is no longer a contraction.

---

[30] A typical entry is $\frac{\partial s_{jt}}{\partial \delta_{kt}} = \int [1(j = k) s_{ijt}(\boldsymbol{\mu}_{it}) - s_{ijt}(\boldsymbol{\mu}_{it}) s_{ikt}(\boldsymbol{\mu}_{it})] f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\, \mathrm{d}\boldsymbol{\mu}_{it}$. The primary cost arises from numerical integration over heterogeneous types. Even for a large market with $J_t = 1{,}000$ products, inverting a $1{,}000 \times 1{,}000$ matrix is easy relative to numerically computing J_t^2 integrals.

[31] Here $f(\cdot)$ defines a contraction iteration and is not to be confused with the functional form for marginal costs $f_{MC}(\cdot)$ or the distribution of heterogeneity $f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)$.

[32] A simple but novel derivation. Consider the matrix $\frac{\partial \log \boldsymbol{s}_t}{\partial \boldsymbol{\delta}_t} = I_{J_t} - \mathrm{diag}^{-1}(\boldsymbol{s}_t)\Gamma_t(\widetilde{\theta}_2)$ in which element (j, k) is $1(j = k) - s_{jt}^{-1} \int s_{ijt}(\boldsymbol{\mu}_{it}) s_{ikt}(\boldsymbol{\mu}_{it}) f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\, \mathrm{d}\boldsymbol{\mu}_{it}$. This implies that $L(\widetilde{\theta}_2) = \max_{\delta_t}[\max_j s_{jt}^{-1} \sum_k |\Gamma_{jkt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2)|]$. A rough approximation to the term inside the square braces is $\max_j \sum_k |s_{kt}| \cdot |\mathrm{Corr}(s_{ijt}, s_{ikt})| < 1 - s_{0t}$.

[33] Many of these algorithms are special cases of *Anderson Mixing*.

[34] PyBLP includes a Python port of the `SQUAREM` package from R.

There are alternative acceleration methods in addition to SQUAREM. Reynaerts, Varadhan, and Nash (2012) also consider DF-SANE which takes the form $\delta_t^{h+1} \leftarrow \delta_t^h - \alpha^h f(\delta_t^h)$ with a different choice of the step-size α^h. They find performance is similar to SQUAREM though it can be slightly slower and less robust. Consistent with Reynaerts, Varadhan, and Nash (2012), we find the SQUAREM to be the fastest and most reliable accelerated fixed point approach. We find that the Jacobian-based Levenberg–Marquardt approach gives similar reliability and slightly better performance, though comparisons can be problem dependent.

□ **Optimization.** Optimization of the GMM objective function for the BLP problem can be challenging. The BLP problem in (9) represents a nonconvex optimization problem.[35] This has some important implications. First, the Hessian need not be positive semidefinite at all values of θ_2. This also means that no optimization routine is guaranteed to find a global minimum in a fixed amount of time. This critique applies both to derivative-based quasi-Newton approaches and to simplex-based Nelder-Mead type approaches. Well-known recommendations involve considering a number of different starting values and optimization routines, verifying that $\hat{\theta}$ satisfies both the first order conditions (the gradient is within a tolerance of zero) and second order conditions (the Hessian matrix has all positive eigenvalues).[36] Both of these are reported by default in PyBLP.

The PyBLP package has built-in support for the optimization routines implemented in the open-source SciPy library and can also interface with commercial routines such as Artleys Knitro. In Section 5, we compare different routines implemented in SciPy and Knitro. The optimization interface to PyBLP is generic in the sense that any optimizer can be used if it is implemented as a Python function, or, with the help of packages that allow for between-language interoperability, a function in most other languages.[37] Though nonderivative based routines such as Nelder-Mead have been frequently used in previous articles, they are not recommended.[38] Indeed, the most important aspect of PyBLP optimization is that it calculates the analytic gradient for any user-provided model, including models that incorporate both supply and demand moments or fixed effects.[39] Analytic gradients provide a major speedup in computational time and also generally improve the chances that the algorithm converges to a valid minimum.[40]

Some optimization routines allow the user to input constraints on parameters, which can speed up estimation and prevent the optimization routine from considering "unreasonable" values. An important example are *box constraints* on the parameter space, which restrict components of θ_2 to $\theta_2^{(\ell)} \in [\underline{\theta}_2^{(\ell)}, \overline{\theta}_2^{(\ell)}]$. Some typical constraints are that demand slopes down and that random coefficients have nonnegative but bounded variances.[41] This is particularly helpful because large values for random coefficients can generate some of the numerical issues that we discuss below.

[35] Absent unobserved heterogeneity or random coefficients the problem is globally convex.

[36] When there are parameter bounds, these conditions are based on the projected gradient and the reduced Hessian instead.

[37] In PyBLP's documentation we give an example of such a "custom" routine by constructing a brute force solver that searches over a grid of parameter values. This flexibility should allow users to experiment with routines without much difficulty or "upgrade" if better routines are developed. For more on interoperability, see Footnote 3.

[38] Both Dubé, Fox, and Su (2012) and Knittel and Metaxoglou (2014) find that derivative-based routines outperform the simplex-based Nelder-Mead routine both in terms of speed and reliability. Our own tests concur with this prior opinion.

[39] We could not find any examples of simultaneous estimation of supply and demand with analytic gradients in the literature. A likely explanation is that the derivatives of the markup term $\frac{\partial \eta_{jt}}{\partial \theta_2}$ are quite complicated. See Appendix A.2 for a derivation.

[40] A promising alternative is automatic differentiation (AD). We chose to implement analytic gradients because ceding control to an AD library can limit one's ability to handle numerical errors, which are pervasive in the BLP problem. However, AD is a promising technique for structural estimation and we are optimistic going forward.

[41] Requiring nonnegative variances on random coefficients is not absolutely necessary because the objective function should be symmetric about zero so that $g(\sigma) = g(-\sigma)$. This is because we optimize over the Cholesky root of the covariance for the random coefficients $LL' = \Sigma$ rather than the covariance matrix itself.

In practice, a common issue is that the default termination tolerances in optimization software can be relatively loose. We provide some examples in Section 6. For termination conditions that are sensitive to the scale of the GMM objective value, this problem is made worse when N is large.[42] This highlights the importance of trying different optimizer configurations, particularly if the algorithm terminates surprisingly early.

Consistent with the prior literature, we recommend trying multiple optimizers and starting values to check for agreement. Our recommendation is that researchers try Knitro's Interior/Direct algorithm first if it is available,[43] and then try a BFGS-based routine, ideally with constraints on the parameter space such as SciPy's L-BFGS-B solver. An advantage of commercial solvers is that they work well out of the box, and do not require much in the way of configuration. However, our simulations indicate that when properly configured, most optimizers arrive at the same parameter estimates, satisfying both first and second order conditions. For the NFXP algorithm, the choice of solver seems much less important than proper configuration of parameter bounds, analytic gradients, and tight termination tolerances, which are all implemented by default in PyBLP.[44]

□ **Numerical issues and tricks.** There are several numerical challenges posed by the BLP problem, most of which are related to the exponentiated indirect utilities and the logit denominator: $\sum_j \exp(\delta_{jt} + \mu_{ijt})$. If some values in this summation are on the order of $\exp(-5) \approx 0.0067$ and others are $\exp(30) > 10^{13}$, their sum may be rounded. This rounding occurs because most computers follow the IEEE-754 standard for floating point arithmetic and on most 64-bit architectures this means that floating point operations have around 15 significant digits. A *loss of precision* arises when taking summations of many numbers with different scales, and depending on the situation, may or may not be problematic. A related problem is *overflow*, which is likely to arise when attempting to compute large values such as $\exp(800)$. This can mean that $s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) \to 1$ whereas $s_{kt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) \to 0$ for $k \neq j$, leading optimization routines to fail.[45]

There are a number of solutions to these problems. One solution is to avoid large values in the argument of $\exp(\cdot)$ by limiting the magnitudes of random coefficients through the *box constraints* described above. Another simple solution involves working market by market and avoiding very large summations. One additional method would be to use a summation algorithm that attempts to preserve precision such as *Kahan Summation*.[46] We found this to be substantially slower and thus do not implement it by default. As suggested in Skrainka (2012b), yet another approach is to use *extended precision arithmetic*. Again, we found this to be substantially slower without improving our statistical performance.[47]

Another way to guarantee overflow safety is to use a protected version of the log-sum-exp function: $\text{LSE}(\boldsymbol{x}) = \log \sum_k \exp x_k = a + \log \sum_k \exp(x_k - a)$. By choosing $a = \max\{0, \max_k x_k\}$, use of this function helps ensure overflow safety when evaluating the multinomial logit function. This is a well-known trick from the applied mathematics and machine learning literatures, but there does not appear to be evidence of it in the literature on BLP models. This is implemented by default, as the additional computational cost appears to be trivial.

[42] In theory, the scale of the objective $q(\theta) = g(\theta)'Wg(\theta)$ should be unimportant. By default, PyBLP scales by N so that after two-step GMM the objective equals the Hansen (1982) J statistic, the scale of which is invariant to problem size. We leave objective values unscaled throughout this article unless stated otherwise.

[43] The main disadvantage of Knitro is that it is not freely available—it must be purchased and installed by end-users.

[44] Commercial solvers likely have other advantages for other formulations of the problem, such as the MPEC approach of Dubé, Fox, and Su (2012), for which they recommend using Knitro.

[45] In this case $\boldsymbol{\delta}_t$ cannot be solved for in $s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) = \mathcal{S}_{jt}$, so the inversion for the mean utilities fails.

[46] In Python this is implemented as `math.fsum`.

[47] PyBLP supports `numpy.longdouble` with the `pyblp.options.dtype` setting. On most Unix platforms, it is implemented with a 128-bit long double. For problems very with very small shares, or for problems with very large numbers of products, additional precision might still be valuable.

After implementing these suggestions, numerical issues are less common but can still occur. For example, the log-sum-exp function does not guard against occasional *underflow*, and in edge cases the weighting matrix W or the intra-firm matrix of demand derivatives Δ_t may be nearly singular. If not dealt with, issues like these can cause the optimization routine to fail, which highlights the importance of robust and verbose error handling.[48]

Implementations of the BLP method have used a number of "tricks" to speed up the algorithm. One well-known example involves working with $\exp(\delta_{jt})$ in place of δ_{jt} in the contraction mapping to avoid transcendental functions like $\exp(\cdot)$. This has little benefit on modern architectures as transcendentals are highly optimized and $\exp(\cdot)$ of a billion numbers takes less than a second. Another trick is using $\boldsymbol{\delta}_t$ from a previous guess of θ_2 as a starting value. The SQUAREM and LM algorithms we suggest for solving the system of equations are relatively insensitive to starting values, so these sorts of speedups are not particularly useful. A "vectorization" speedup used in Nevo (2000b) and Knittel and Metaxoglou (2014) is to stack all markets together and construct cumulative sums of $\exp(\delta_{jt} + \mu_{ijt})$. We find that this is approach is dominated by parallelization across markets T; furthermore, this approach will result in a loss of precision as $T \to \infty$.

□ **Heterogeneity and integration.** An important aspect of the BLP model is that it incorporates heterogeneity via random coefficients. There are several ways to redefine the integral in (3):[49]

$$s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) = \int \frac{\exp(\delta_{jt} + \mu_{ijt})}{\sum_{k \in J_t} \exp(\delta_{kt} + \mu_{ikt})} f(\boldsymbol{\mu}_{it} | \widetilde{\theta}_2) \, \mathrm{d}\boldsymbol{\mu}_{it} \tag{23a}$$

$$= \int \frac{\exp[\delta_{jt} + \mu_{ijt}(\widetilde{\nu}_{it}(\nu_{it}), \widetilde{\theta}_2)]}{\sum_{k \in J_t} \exp[\delta_{kt} + \mu_{ikt}(\widetilde{\nu}_{it}(\nu_{it}), \widetilde{\theta}_2)]} \phi(\nu_{it}) \, \mathrm{d}\nu_{it} \tag{23b}$$

$$= \int_{[0,1]^{K_2}} \frac{\exp[\delta_{jt} + \mu_{ijt}(\widetilde{\nu}_{it}(u_{it}), \widetilde{\theta}_2)]}{\sum_{k \in J_t} \exp[\delta_{kt} + \mu_{ikt}(\widetilde{\nu}_{it}(u_{it}), \widetilde{\theta}_2)]} \, \mathrm{d}u_{it}. \tag{23c}$$

The first approach in (23a) draws directly from the distribution of unobserved tastes $f(\boldsymbol{\mu}_{it} | \widetilde{\theta}_2)$. The second approach in (23b) integrates over the K_2-dimensional standard normal $\phi(\nu_{it})$ and transforms the draws into a correlated normal using the Cholesky root $L(\widetilde{\theta}_2)$ of the covariance matrix so that $\widetilde{\nu}_{it} = L(\widetilde{\theta}_2) \cdot \nu_{it}$ and $\mu_{ijt} = \sum_{k \in K_2} x_{jt}^{(k)} \cdot \widetilde{\nu}_{it}^{(k)}$. The third method in (23c) integrates over the K_2-dimensional hypercube $[0, 1]^{K_2}$ and transforms the draws using the inverse CDF of the mixing distribution: $\widetilde{\nu}_{it} = F^{-1}(u_{it} | \widetilde{\theta}_2)$. Under all three formulations the integral is "well behaved" in the sense that the integrand is bounded on [0,1], smooth, and infinitely continuously differentiable on $\mathbb{C}^\infty$.[50]

We can approximate the integral at a finite set of I_t *nodes* ν_{it} and *weights* w_{it}. Together the nodes and weights (ν_{it}, w_{it}) define an *integration rule*. We use $s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2)$ to denote the true value of the integral and use $s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)$ to denote the approximation with I_t nodes:

$$s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) \approx s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t) \equiv \sum_{i \in I_t} w_{it} \cdot s_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}(\nu_{it}, \widetilde{\theta}_2)). \tag{24}$$

[48] When PyBLP encounters such a numerical error, it replaces the problematic object with a "reasonable" counterpart (e.g., corresponding values from the last optimization iteration or the Moore-Penrose pseudo-inverse of a near-singular matrix) and provides an informative warning or error message.

[49] We assume that heterogeneity is normally distributed to simplify some expressions. PyBLP currently accommodates normal and lognormal distributions of parameters. Others such as half-normal or exponential distributions can be implemented with a different change of variables.

[50] For a proof that all the derivatives are bounded see Iaria and Wang (2019).

Pseudo-Monte Carlo. The simplest integration rule is pseudo-Monte Carlo (pMC) integration. Here pseudo-random draws ν_{it} are taken from one of the three candidate distributions in (23a)-(23c) and are used to calculate $s_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}(\nu_{it}, \widetilde{\theta}_2))$.[51] Each draw ν_{it} is equally weighted: $w_{it} = I_t^{-1}$.

How accurate is the approximation in (24) under the pMC approach? Define the simulation error as $\epsilon_{I_t}^{pMC} = s_{jt}(\boldsymbol{\delta}_t, \widehat{\theta}_2) - s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)$. A straightforward application of the Central Limit Theorem shows that:[52]

$$\epsilon_{I_t}^{pMC} \xrightarrow{d} N\left(0, \sqrt{V(s_{jt})/I_t}\right) \quad \text{with} \quad V(s_{jt}) = \int \left(s_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}) - s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2)\right)^2 f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)\, \mathrm{d}\boldsymbol{\mu}_{it}.$$

The main advantage of pMC integration is that it avoids the *curse of dimensionality* because the simulation error is not directly related to the dimension of the integral. Its main disadvantage is that the simulation error declines slowly in the number of draws at an $O(I_t^{-1/2})$ rate. For a more extensive discussion of the pMC approach for BLP problems including bias correction and standard error adjustments, consult Freyberger (2015).

Quasi-Monte Carlo. Quasi-Monte Carlo (qMC) rules use deterministic sequences to more evenly cover the unit hypercube $[0, 1]^{K_2}$ in (23c). Integration error can be reduced by choosing *low discrepancy sequences*.[53] A popular choice are Halton (1960) sequences. Train (2000), Bhat (2001), and Train (2009) report success with qMC sequences (Halton draws in particular) for maximum simulated likelihood (MSL) estimators of mixed logit models. Nevo (2001) reports success with Halton sequences for the BLP problem.

For fixed dimension of integration K_2, the $O(I_t^{-1} \cdot (\log I_t)^{K_2})$ rate of qMC is expected to outperform the $O(I_t^{-1/2})$ rate of pMC as the number of nodes I_t becomes large.[54] Owen (1997) shows that randomizing qMC sequences in a particular manner can give a $O(I_t^{-3/2} \cdot (\log I_t)^{K_2})$ rate for smooth integrands as in (23c). For these reasons, it is common to *scramble* qMC sequences. Other common practices involve *discarding* the first few points and *skipping* others in-between selected points rather than taking points sequentially.[55]

Variance reduction and importance sampling. There are a number of additional techniques that have been used to reduce simulation error or variance. For example, Hess, Train, and Polak (2006) and Brunner et al. (2017) employ Modified Latin Hypercube Sampling (MLHS) for MSL and BLP problems, respectively. The MLHS approach takes a stratified sample by cutting the unit hypercube into smaller hypercubes and then sampling within each hypercube with pMC. Another common variance reduction technique is *antithetic sampling* which exploits the symmetry of $\phi(\nu_{it}) = \phi(-\nu_{it})$ by taking each standard normal draw and reflecting it through the origin.

Importance sampling is also common. Rather than drawing from the standard multivariate normal distribution $\phi(\nu_{it})$, importance sampling instead proposes to draw from a biased distribution $q(\nu_{it})$. The idea is to draw more points where $s_{ijt}(\boldsymbol{\delta}_t, \nu_{it}, \widetilde{\theta}_2) = s_{ijt}(\boldsymbol{\delta}_t, \boldsymbol{\mu}_{it}(\nu_{it}(\nu_{it}), \widetilde{\theta}_2))$ is large and fewer where it is close to zero. To keep the value of the integral the same, the oversampled part of the distribution places less weight $w(\nu_{it})$ on each point, and more on the undersampled

[51] By default PyBLP follows (23b), which is popular because it allows one to fix the draws ν_{it} and simply rescale $\widetilde{\nu}_{it}$ as $\widetilde{\theta}_2$ changes without adding additional randomness to the estimation procedure. If the draws change with each estimation iteration this can create "chattering" because the simulated objective is not identical at every evaluation.

[52] See Chapter 7 of Judd (1998) for the derivation. It is easy to bound the variance term $V(s_{jt}) < 1$.

[53] These are *low discrepancy* in that $D_{I_t}^*(u_{1,t}, \ldots u_{I_t,t}) \leq c \cdot I_t^{-1} \cdot (\log I_t)^{K_2}$ where the *star discrepancy* $D_{I_t}^*$ bounds integration error in the Koksma-Hlawka inequality: $|s_{jt}(\boldsymbol{\delta}_t, \widehat{\theta}_2) - s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)| \leq V_{HK}(s_{jt}) \cdot D_{I_t}^*(u_{1,t}, \ldots u_{I_t,t})$. Here V_{HK} denotes bounded variation in the sense of Hardy-Krause. See Judd (1998) and Owen (2005) for more details.

[54] Because the constant term can be large, this is by no means a guarantee. Also one requires that $2^{K_2} < I_t$.

[55] By default, PyBLP scrambles Halton sequences with the recipe in Owen (2017) and discards the first 1000 points in each dimension. Otherwise the first few points can exhibit a high degree of correlation between dimensions.

parts:

$$s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2) = \int s_{ijt}(\boldsymbol{\delta}_t, \nu_{it}, \widetilde{\theta}_2) \cdot \frac{\phi(\nu_{it})}{\underbrace{q(\nu_{it})}_{w(\nu_{it})}} \cdot q(\nu_{it})\, \mathrm{d}\nu_{it}.$$

One challenge is that the optimal $q_{jt}(\nu_{it})$ is product specific.[56] Berry, Levinsohn, and Pakes (1995) oversample consumers with a small outside good by drawing from $q_t(\nu_{it}) = \phi(\nu_{it}) \cdot \frac{1-s_{i0t}(\boldsymbol{\delta}_t, \nu_{it}, \widetilde{\theta}_2)}{1-\mathcal{S}_{0t}}$ where $\widetilde{\theta}_2$ and $\boldsymbol{\delta}_t$ are replaced by consistent estimates.[57]

Quadrature. The other approach is *Gaussian quadrature*. Theoretically, the integrand is approximated with a polynomial and then integrated exactly as polynomial integration is trivial. This amounts to a weighted sum in (24) over a particular choice of (ν_{it}, w_{it}). The main choice to make is the *polynomial order* of the rule. As the order grows, more nodes are required but the accuracy of the approximation improves.

Gaussian quadrature works best when certain conditions are met: that the integrand is bounded and continuously differentiable. Thankfully, the logit kernel in (24) is always bounded between $(0, 1)$ and is infinitely continuously differentiable. There are a number of different flavors of quadrature rules designed to best approximate integrals under different weighting schemes. The *Gauss-Hermite* family of rules work best when $f(\nu_{it}) \propto \exp(-\nu_{it}^2)$, which (with a change of variables) includes integrals over a normal density. *Nested* rules offer an alternative where for a given polynomial order p, they reuse the set of nodes from the rule with order $p-1$.[58] Both the advantage and disadvantage of Gaussian quadrature rules is that they do a better job covering the "tail" of the probability distribution. Although this increases the accuracy of the approximation, it can also lead to very large values which create overflow issues.[59] We prefer to use quadrature rules, and to be careful of potential numerical issues when computing shares.

The Gaussian quadrature rules apply only to a single dimension. One way to estimate higher dimensional integrals is to construct the product of single dimensional integrals. The disadvantage of *product rules* is that if one needs I_t points to approximate the integral in dimension one, then one needs I_t^d points to approximate the integral in dimension d. This is the so-called *curse of dimensionality*.

The curse of dimensionality is a well-known problem in numerical analysis and several off-the-shelf solutions exist. There are several clever algorithms for improving upon the product rule for higher dimensional integration. Judd and Skrainka (2011) explore *monomial cubature rules* whereas Heiss and Winschel (2008) use *sparse grid* methods to selectively eliminate nodes from the product rules. One disadvantage of these methods is that they often involve negative weights weights $w_{it} < 0$, which can create problems during estimation or when trying to decompose the distribution of heterogeneity (particularly for counterfactuals).

Integration in PyBLP. Though PyBLP allows for flexible, user-supplied distributions of random coefficients, by far the most commonly employed choices in the literature are the independent normal, correlated normal, and lognormal distributions for $f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2)$. PyBLP supports all of these distributions and provides some specialized routines to handle these integrals with limited user

[56] Heiss (2010) proposes an "adaptive importance sampler" for probit-type models. Brunner (2017) adapts this to the BLP-type problem and reports success. This adaptive, optimal importance sampling requires updating the weights w_{it} as θ_2 varies. For this reason, we don't currently implement this approach as part of PyBLP.

[57] At the consistent estimate for $\widetilde{\theta}_2$, the $\boldsymbol{\delta}_t$ needs to be computed only once, so it is feasible to use a higher-precision integration rule than is used for estimation. Berry, Levinsohn, and Pakes (1995) draw from $q_t(\nu_{it})$ with rejection sampling, and this same procedure is implemented in PyBLP.

[58] In theory, this allows for *adaptive* accuracy without wasting calculations. When the error from numerical integration is large, the polynomial degree can be expanded to reduce the error.

[59] Essentially for some simulated individual i we have that $s_{ijt} \to 1$ and $s_{ikt} \to 0$. This problem has been previously documented by Judd and Skrainka (2011) and Skrainka (2012b).

intervention. There are a number of different methods one can use to generate (ν_{it}, w_{it}) for the normal case: pMC, Halton sequences, MLHS, product rules, and sparse grids.

In general, we find that the best practice in low dimensions is to use product rules to a relatively high degree of polynomial accuracy. In higher dimensions, Halton sequences and in particular sparse grids appear to scale the best, both in our own Monte Carlo studies and in those of Judd and Skrainka (2011) and Heiss and Winschel (2008).

□ **Solving for pricing equilibria.** Many counterfactuals of BLP-type problems involve perturbing either the market structure, marginal costs, or both, and solving for counterfactual equilibrium prices. Being able to solve for equilibrium prices quickly and accurately is also crucial to generating the optimal instruments in the next section. The Bertrand-Nash first order conditions are defined by (5) for each market t:[60]

$$\boldsymbol{p}_t = \boldsymbol{c}_t + \underbrace{\Delta_t(\boldsymbol{p}_t, \mathcal{H}_t)^{-1} s(\boldsymbol{p}_t)}_{\boldsymbol{\eta}_t(\boldsymbol{p}_t, \mathcal{H}_t)}$$

In order to recover marginal costs during estimation, one need only invert the $J_t \times J_t$ matrix $\Delta_t(\boldsymbol{p}_t, \mathcal{H}_t)$. Solving for counterfactual pricing equilibria is more difficult as it requires solving the $J_t \times J_t$ nonlinear system of equations, often after replacing the ownership matrix $\mathcal{H}_t$ with a post-merger counterpart $\mathcal{H}_t^*$:

$$\boldsymbol{p}_t = \boldsymbol{c}_t + \boldsymbol{\eta}_t(\boldsymbol{p}_t, \mathcal{H}_t^*). \tag{25}$$

In general, solving this problem is difficult because it represents a nonconvex, nonlinear system of equations where one must simulate in order to compute $\boldsymbol{\eta}_t(\boldsymbol{p}_t, \mathcal{H}_t^*)$ and its derivatives. Once one incorporates both multi-product firms and arbitrary coefficients into the problem, both existence and uniqueness of an equilibrium become challenging to establish.[61]

One approach might be to solve the system using Newton's method, which requires calculating the $J_t \times J_t$ Jacobian $\frac{\partial \boldsymbol{\eta}_t}{\partial \boldsymbol{p}_t}$. The expression for the Jacobian involves the Hessian matrix of demand $\frac{\partial^2 s_{kt}}{\partial p_{jt}^2}$ as well as tensor products, and can be computationally challenging.[62]

The second, and perhaps most common approach in the literature is treating (25) as a fixed point and iterating on $\boldsymbol{p}_t \leftarrow\!\shortmid \boldsymbol{c}_t + \boldsymbol{\eta}_t(\boldsymbol{p}_t, \mathcal{H}_t^*)$.[63] The problem is that although a fixed point of (25) may represent the Bertrand-Nash equilibrium of (7), it is not necessarily a contraction. In fact, as part of Monte Carlo experiments conducted in Armstrong (2016), the author finds that iterating on (25) does not always lead to a solution and at least some fraction of the time leads to cycles. We were able to replicate this finding for similarly constructed Monte Carlo experiments between 1-5% of the time. Our preferred approach borrows from the engineering literature and does not appear to be well known in Industrial Organization, but in our experiments appears to be highly effective. We follow Morrow and Skerlos (2011) who reformulate the solution to (5) by breaking up the matrix of demand derivatives into two parts: a $J_t \times J_t$ diagonal matrix Λ_t, and a $J_t \times J_t$

[60] We suppress the dependence on the parameters θ_2 as we hold everything fixed and solve for $\boldsymbol{p}_t$.

[61] Caplin and Nalebuff (1991) and Gallego et al. (2006) have results that apply to single product firms and linear in price utility under logit demands. Konovalov and Sandor (2010) generalizes these results to logit demands with linear in price utility and multi-product firms. With the addition of random coefficients, it is possible that the resulting model will violate the quasi-concavity of the profit function that these results require. Morrow and Skerlos (2010) avoid some of these restrictions but place other restrictions on indirect utilities. Existence and uniqueness are beyond the scope of this article—we instead focus on calculating solutions to the system of first order conditions, assuming such solutions exist.

[62] For example, Knittel and Metaxoglou (2014) do not update $\boldsymbol{s}_t(\boldsymbol{p}_t)$ and thus avoid fully solving the system of equations. An Newton-type alternative with a finite-differenced Jacobian would be slow and ill-advised as there are J_t derivatives for J_t markups and every derivative involves integration in order to compute both $\Delta_t(\boldsymbol{p}_t)$ and $\boldsymbol{s}_t(\boldsymbol{p}_t)$.

[63] The "folklore" solution is to dampen this expression with some ρ and consider $\boldsymbol{p}_t \leftarrow\!\shortmid \rho \cdot \boldsymbol{p}_t + (1-\rho) \cdot [\boldsymbol{c}_t + \eta_t^*(\boldsymbol{p}_t, \mathcal{H}_t^*)]$. This tends to be slower and more reliable, though we cannot find any theoretical justification for convergence.

dense matrix Γ_t:

$$\frac{\partial \boldsymbol{s}_t}{\partial \boldsymbol{p}_t}(\boldsymbol{p}_t) = \Lambda_t(\boldsymbol{p}_t) - \Gamma_t(\boldsymbol{p}_t),$$
$$\Lambda_{jj,t} = \int \alpha_i \, s_{ijt}(\boldsymbol{\mu}_{it}) f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2) \, \mathrm{d}\boldsymbol{\mu}_{it}, \tag{26}$$
$$\Gamma_{jk,t} = \int \alpha_i \, s_{ijt}(\boldsymbol{\mu}_{it}) s_{ikt}(\boldsymbol{\mu}_{it}) f(\boldsymbol{\mu}_{it}|\widetilde{\theta}_2) \, \mathrm{d}\boldsymbol{\mu}_{it},$$

in which $\alpha_i = \frac{\partial u_{ijt}}{\partial p_{jt}}$ is the marginal dis-utility of price. Morrow and Skerlos (2011) then reformulate the problem as a different fixed point that is specific to mixed logit demands:[64]

$$\boldsymbol{p}_t \leftarrow\!\shortmid \boldsymbol{c}_t + \boldsymbol{\zeta}_t(\boldsymbol{p}_t) \quad \text{where} \quad \boldsymbol{\zeta}_t(\boldsymbol{p}_t) = \Lambda_t(\boldsymbol{p}_t)^{-1}[\mathcal{H}_t^* \odot \Gamma_t(\boldsymbol{p}_t)](\boldsymbol{p}_t - \boldsymbol{c}_t) - \Lambda_t(\boldsymbol{p}_t)^{-1}\boldsymbol{s}_t(\boldsymbol{p}_t). \tag{27}$$

The fixed point in (27) is entirely different from that in (25) and coincides only at resting points. Consistent with results reported in Morrow and Skerlos (2011), we find that (27) is around 3-12 times faster than Newton-type approaches and reliably finds an equilibrium.[65]

Perhaps most consequentially, the ability to solve for a pricing equilibrium rapidly and reliably makes it possible to generate the Amemiya (1977) or Chamberlain (1987) *feasible approximation to the optimal instruments*.

4. Supply and demand: optimal instruments and overidentifying restrictions

■ *In this section, we focus on joint estimation of supply and demand under optimal instruments in order to clarify the precise role of overidentifying restrictions in the parametric identification of parameters. Absent information on firm conduct or the precise functional form for marginal costs, many researchers estimate a demand side only*.

As a way to improve performance, we can construct approximations to the optimal instruments in the spirit of Amemiya (1977) or Chamberlain (1987). Approximations to the optimal instruments were featured in both Berry, Levinsohn, and Pakes (1995) and Berry, Levinsohn, and Pakes (1999) but are not commonly employed in many subsequent studies using the BLP approach in part because they are challenging to construct. Reynaert and Verboven (2014) show that approximations to the optimal instruments can improve the econometric performance of the estimator, particularly with respect to the θ_2 parameters. The form we derive is somewhat different from their expression, and arrives at different instruments for supply and demand. Although the procedure itself is quite involved, the good news is that it does not require much in the way of user input, and is fully implemented by the PyBLP software.

□ **Derivation.** Recall the GMM moment conditions are given by $E[\xi_{jt}|Z_{jt}^D] = 0$ and $E[\omega_{jt}|Z_{jt}^S] = 0$ and the asymptotic GMM variance depends on $(D' \, \Omega^{-1} D)$ where the expressions are given below:

$$D = E\left[\left(\frac{\partial \xi_{jt}}{\partial \theta}, \; \frac{\partial \omega_{jt}}{\partial \theta}\right) \middle| \mathrm{Z_t}\right], \quad \Omega = E\left[\begin{pmatrix}\xi_{jt} \\ \omega_{jt}\end{pmatrix}\begin{pmatrix}\xi_{jt} & \omega_{jt}\end{pmatrix} \middle| \mathrm{Z_t}\right].$$

Chamberlain (1987) showed that the approximation to the optimal instruments are given by the expected Jacobian contribution for each observation (j, t): $E[D_{jt}(\mathrm{Z_t})\Omega_{jt}^{-1}|\mathrm{Z_t}]$. We use the word

[64] This resembles a well known "folklore" solution to the pricing problem, which is to rescale each equation by its own share s_{jt} (see Skrainka, 2012a). For the plain logit, $\Lambda_{jj,t} = \alpha s_{jt}$.

[65] In PyBLP, iteration is terminated when the firms' first order condition $||\Lambda(\boldsymbol{p}_t)(\boldsymbol{p}_t - \boldsymbol{c}_t - \boldsymbol{\zeta}_t(\boldsymbol{p}_t))||_\infty$ is less than a fixed tolerance. Morrow and Skerlos (2011) refer to this as the problem's *numerical simultaneous stationarity condition*.

"approximation" because the aforementioned expectation over the unobserved (ξ_{jt}, ω_{jt}) lacks a closed form. We derive the components of the approximation below:[66]

$$D_{jt} \equiv \underbrace{\begin{bmatrix} \frac{\partial \xi_{jt}}{\partial \beta} & \frac{\partial \omega_{jt}}{\partial \beta} \\ \frac{\partial \xi_{jt}}{\partial \alpha} & \frac{\partial \omega_{jt}}{\partial \alpha} \\ \frac{\partial \xi_{jt}}{\partial \theta_2} & \frac{\partial \omega_{jt}}{\partial \theta_2} \\ \frac{\partial \xi_{jt}}{\partial \gamma} & \frac{\partial \omega_{jt}}{\partial \gamma} \end{bmatrix}}_{(K_1+K_2+K_3)\times 2} = \begin{bmatrix} -x_{jt} & 0 \\ -v_{jt} & 0 \\ \frac{\partial \xi_{jt}}{\partial \alpha} & \frac{\partial \omega_{jt}}{\partial \alpha} \\ \frac{\partial \xi_{jt}}{\partial \theta_2} & \frac{\partial \omega_{jt}}{\partial \theta_2} \\ 0 & -x_{jt} \\ 0 & -w_{jt} \end{bmatrix}, \quad \Omega_t \equiv \underbrace{\begin{bmatrix} \sigma_{\xi}^2 & \sigma_{\xi_t \omega_t} \\ \sigma_{\xi_t \omega_t} & \sigma_{\omega_t}^2 \end{bmatrix}}_{2\times 2}. \tag{28}$$

A little calculation shows that for each market t and product j,

$$D_{jt}\Omega_t^{-1} = \frac{1}{\sigma_\xi^2 \sigma_\omega^2 - \sigma_{\xi\omega}^2} \cdot \begin{bmatrix} -\sigma_\omega^2 x_{jt} & \sigma_{\xi\omega} x_{jt} \\ -\sigma_\omega^2 v_{jt} & \sigma_{\xi\omega} v_{jt} \\ \sigma_\omega^2 \frac{\partial \xi_{jt}}{\partial \alpha} - \sigma_{\xi\omega} \frac{\partial \omega_{jt}}{\partial \alpha} & \sigma_\xi^2 \frac{\partial \omega_{jt}}{\partial \alpha} - \sigma_{\xi\omega} \frac{\partial \xi_{jt}}{\partial \alpha} \\ \sigma_\omega^2 \frac{\partial \xi_{jt}}{\partial \theta_2} - \sigma_{\xi\omega} \frac{\partial \omega_{jt}}{\partial \theta_2} & \sigma_\xi^2 \frac{\partial \omega_{jt}}{\partial \theta_2} - \sigma_{\xi\omega} \frac{\partial \xi_{jt}}{\partial \theta_2} \\ \sigma_{\xi\omega} x_{jt} & -\sigma_\xi^2 x_{jt} \\ \sigma_{\xi\omega} w_{jt} & -\sigma_\xi^2 w_{jt} \end{bmatrix}. \tag{29}$$

For each column vector, the first and fifth entries are linear functions of x_{jt}. Let Θ be a conformable matrix of zeros and ones such that

$$(D_{jt}\Omega_t^{-1}) \odot \Theta = \frac{1}{\sigma_\xi^2 \sigma_\omega^2 - \sigma_{\xi\omega}^2} \cdot \begin{bmatrix} -\sigma_\omega^2 x_{jt} & 0 \\ -\sigma_\omega^2 v_{jt} & \sigma_{\xi\omega} v_{jt} \\ \sigma_\omega^2 \frac{\partial \xi_{jt}}{\partial \alpha} - \sigma_{\xi\omega} \frac{\partial \omega_{jt}}{\partial \alpha} & \sigma_\xi^2 \frac{\partial \omega_{jt}}{\partial \alpha} - \sigma_{\xi\omega} \frac{\partial \xi_{jt}}{\partial \alpha} \\ \sigma_\omega^2 \frac{\partial \xi_{jt}}{\partial \theta_2} - \sigma_{\xi\omega} \frac{\partial \omega_{jt}}{\partial \theta_2} & \sigma_\xi^2 \frac{\partial \omega_{jt}}{\partial \theta_2} - \sigma_{\xi\omega} \frac{\partial \xi_{jt}}{\partial \theta_2} \\ 0 & -\sigma_\xi^2 x_{jt} \\ \sigma_{\xi\omega} w_{jt} & -\sigma_\xi^2 w_{jt} \end{bmatrix}. \tag{30}$$

We can partition our instrument set by column into "demand" and "supply" instruments:[67]

$$Z_{jt}^{Opt,D} \equiv \underbrace{E[(D_{jt}(Z_t)\Omega_t^{-1} \odot \Theta)_{\cdot 1} | Z_t]}_{K_1+K_2+(K_3-K_x)}, \quad Z_{jt}^{Opt,S} \equiv \underbrace{E[(D_{jt}(Z_t)\Omega_t^{-1} \odot \Theta)_{\cdot 2} | Z_t]}_{K_2+K_3+(K_1-K_x)}. \tag{31}$$

Here, we have $K - K_x$ (where K_x denotes the dimension of common exogenous parameters x_{jt}) instruments for both supply and demand, though it is evident from (30) that the instruments for the θ_2 parameters are not the same.[68] The optimal instruments from the *linear* portions of demand and supply are simply exogenous regressors re-scaled by covariances, whereas the optimal instruments from the θ_2 parameters are *nonlinear* functions of the data.

Two sets of overidentifying restrictions arise from *exclusion restrictions*, which are made explicit in (30) where w_{jt} and v_{jt} show up in one equation but not the other. There are $K_3 - K_x$ overidentifying restrictions from cost shifters w_{jt} that are excluded from demand and $K_1 - K_x$ demand shifters v_{jt} that are excluded from supply. When we include simultaneous supply and demand moments we also have *cross equation* restrictions. As shown in (30), we have

[66] In our Monte Carlo exercises we assume that (ξ_{jt}, ω_{jt}) are jointly i.i.d. across all j and t so that $\Omega_{jt} = \Omega$. This is merely a matter of convenient notation, as extensions to heteroskedastic or clustered covariances are straightforward.

[67] Here $(\,)_{\cdot 1}$ and $(\,)_{\cdot 2}$ denote the first and second column of the matrix in (30) and $\odot$ is the elementwise Hadamard product.

[68] This is true except for the knife-edge case where $\frac{\partial \xi_{jt}}{\partial \theta_\ell} / \frac{\partial \omega_{jt}}{\partial \theta_\ell} \propto \frac{\sigma_\xi^2 + \sigma_{\xi\sigma}}{\sigma_\omega^2 + \sigma_{\xi\sigma}}$. In fact, (30) highlights that the set of instruments Z_D and Z_S should never be the same because of the different excluded variables.

$2 \times (K - K_x)$ restrictions and K parameters. This gives us $K - 2K_x$ overidentifying restrictions. The additional K_2 overidentifying restrictions in (30) come from the fact that we have two restrictions for each of the θ_2 parameters, including the price coefficient α.

Remark 1. Our version of the optimal instruments makes explicit the exclusion restrictions in the BLP model. Perhaps most importantly, this tells us precisely where to find exclusion restrictions: *something that enters the other equation*. Both the supply and demand moments are informative for the θ_2 parameters. The role of the exogenous cost-shifters w_{jt} is now explicit: they provide overidentifying restrictions for demand that are informative about θ_2 (including the term on price α). Likewise, the role of the exogenous demand-shifters v_{jt} is also made explicit: they provide overidentifying restrictions for supply which are informative about θ_2 (and markups).[69] The link between the supply and demand side is the endogenous markup $\eta_{jt}(\theta_2, \boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$, which depends on the common θ_2 parameters. This has led $(\frac{\partial \xi_{jt}}{\partial \theta_2}, \frac{\partial \omega_{jt}}{\partial \theta_2})$-type instruments to be described as *quantity shifters* or *markup shifters*.

Remark 2. It is worth pointing out that our derivation of the optimal instruments appears to vary from derivations in the prior literature. Some of the prior literature using optimal instruments for BLP-type problems suggests the resulting problem is *just identified* rather than *over identified* and relies on the same set of instruments for both supply and demand. Reynaert and Verboven (2014) appear to construct their version of optimal instruments by summing across the rows of (29) and excluding either the first or third row.[70] This gives $K = K_1 + K_2 + K_3$ instruments and K unknowns so that the model is just identified. However, because they stack $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$ they effectively have $2N$ rather than N observations. Conceptually, one way to view their formulation is that it imposes $E[\xi_{jt} Z^D_{jt}] + E[\omega_{jt} Z^S_{jt}] = 0$ rather than separately imposing $E[\xi_{jt} Z^D_{jt}] = 0$ and $E[\omega_{jt} Z^S_{jt}] = 0$.

Remark 3. One alternative to the Chamberlain (1987) approximation to the optimal instruments is to instead construct a semiparametric basis that spans the same vector space as the optimal instruments. This approach was suggested by Newey (1990) and applied to conditional moment restriction models in Ai and Chen (2003) and Donald, Imbens, and Newey (2009). Given a conditional moment condition of the form $E[\xi_{jt}|Z_t] = 0$, one can instead write $E[\xi_{jt} A(Z_t)] = 0$ for some choice of $A(\cdot)$. One could impose that the moments hold at several quantiles of Z_t or one can construct a polynomial sieve basis in Z_t. This approach also suffers from a curse of dimensionality as in high dimensions the number of needed interaction terms explodes. The bases must also be chosen carefully as it is easy to generate new instruments $A(Z_t)$ that are highly correlated with one another, which leads to the "many moments" problem of Newey and Smith (2004).[71] To highlight this approach, we consider all quadratic interactions of (x_{jt}, w_{jt}) when deriving instruments in "Own" characteristics. Gandhi and Houde (2019) derive a second-order polynomial basis in the *differences* of product characteristics $d_{jkt} = x_{kt} - x_{jt}$, and show that this basis has desireable properties and avoids several of the aforementioned problems.

□ **Constructing feasible instruments.** The main challenge with implementing the optimal instruments is that we must take the expectation of the Jacobian over the joint distribution of unobservables $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$ for all products within a market. We need to compute the following

[69] The idea of using demand shifters as overidentifying restrictions to identify conduct has a long history in industrial organization going back at least as far as Bresnahan (1982), and was treated nonparametrically in Berry and Haile (2014). In related work, Backus, Conlon, and Sinkinson (2020) show how to use (30) to test for firm conduct.

[70] We should mention that Reynaert and Verboven (2014) do not consider joint supply and demand as their main specification.

[71] For example, the polynomial basis $(1, x^2, x^3, x^4)$ exhibits a high degree of correlation.

expectation:[72]

$$E\left[\frac{\partial \xi_{jt}}{\partial \theta_2}, \frac{\partial \omega_{jt}}{\partial \theta_2}\middle| Z_t\right] = \int \left[\frac{\partial \xi_{jt}}{\partial \theta_2}, \frac{\partial \omega_{jt}}{\partial \theta_2}\right](\boldsymbol{\xi}_t, \boldsymbol{\omega}_t, Z_t; \theta_2) f\left(\xi_{1,t}, \ldots, \xi_{J_t,t}, \omega_{1,t}, \ldots, \omega_{J_t,t}\middle| Z_t, \theta_2\right) \mathrm{d}\boldsymbol{\xi}_t \, \mathrm{d}\boldsymbol{\omega}_t.$$

One challenge is that this is an integral over $2J_t$ dimensions. A second challenge is that without additional assumptions, $f(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t | Z_t, \theta_2)$ is unknown. The third is that $(\boldsymbol{s}_t, \boldsymbol{p}_t)$ are endogenous in that they depend on $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$. This means we must construct estimates of $E[p_{jt}|Z_t]$ and $E[s_{jt}|Z_t]$ in order to calculate $\frac{\partial \xi_{jt}}{\partial \theta_2}$. One approach would be to construct $E[p_{jt}|Z_t] = \hat{p}_{jt}$ by regressing the endogenous prices p_{jt} on a series of exogenous regressors in a "first stage" regression (either linearly or nonlinearly) and evaluating $\hat{\boldsymbol{s}}_t(\hat{\boldsymbol{p}}_t, \theta_2)$.[73] The other approach is to solve the nonlinear system of equations for $(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t)$ directly. Here is what Berry, Levinsohn, and Pakes (1995) say about optimal instruments:

> *Unfortunately $D_j(z)$ is typically very difficult, if not impossible, to compute. To calculate $D_j(z)$ we would have to calculate the pricing equilibrium for different (ξ_j, ω_j) sequences, take derivatives at the equilibrium prices, and then integrate out over the distribution of such sequences. In addition, this would require an assumption that chooses among multiple equilibria when they exist, and either additional assumptions on the joint distribution of (ξ, ω), or a method for estimating that distribution.*

The appendix of the NBER working paper version of Berry, Levinsohn, and Pakes (1999) is even less positive:

> *Calculating a good estimate of $E[p|z]$ then requires (i) knowing or estimating the density of the unobservables and (ii) solving at some initial guess for θ the fixed point that defines equilibrium prices for each (ξ, ω) and then integrating this implicit function with respect to the density of the unknown parameters. This process is too complicated to be practical.*

In Algorithm 2, we follow the possibly more accurate but costly recipe proposed by Berry, Levinsohn, and Pakes (1999) and show that with other computational advances in PyBLP it is feasible to implement.

In general, "asymptotic" and "empirical" approaches are not believed to be computationally feasible, particularly when there are large numbers of draws for $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$. The costly step is Item 4 above, which involves solving for a new equilibrium $(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t)$ for each set of draws. These improved optimal instruments are feasible primarily because of the advances we describe at the end of Section 3, which drastically reduce the amount of time it takes to solve for equilibria. For relatively large problems, constructing optimal instruments may take several minutes. For smaller problems such as Berry, Levinsohn, and Pakes (1995) or Nevo (2000b) it takes only several seconds. Our simulations indicate that "approximate" performs as well as the more expensive options. Updating results with optimal instruments in PyBLP requires only the last two lines of code in Figure 6.

This approach is not without its limitations. It requires both a way to generate $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ and an estimate of its covariance. This is straightforward if $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ are i.i.d. or follow some other known structure (e.g., clustered at the product level), but this requires additional assumptions. Furthermore, the resulting $2J_t$ dimensional integral may be hard to approximate accurately for distributions of $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ that are highly skewed or otherwise do not lend well to approximation.

[72] Here $f(\cdot)$ denotes the joint distribution of unobservables $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$ and is not to be confused with the distribution of heterogeneity $f(\boldsymbol{\mu}_{it}, \widetilde{\theta}_2)$.

[73] Reynaert and Verboven (2014) suggest computing the approximation to the optimal instruments under an additional assumption of perfect competition, so that $E[p_{jt}|Z_t] = E[c_{jt}|Z_t] = [x_{jt}, w_{jt}]\gamma + \hat{\omega}_{jt}$. Likewise Gandhi and Houde (2019) suggest using $E[p_{jt}|Z_t] = \hat{p}_{jt}$ via linear or nonlinear regression in order to construct $d_{jkt} = \hat{p}_{kt} - \hat{p}_{jt}$.

Algorithm 2 Feasible Approximation to Optimal IV (Berry, Levinsohn, and Pakes, 1999)

After obtaining an initial estimate $\hat{\theta} = [\hat{\beta}, \hat{\alpha}, \hat{\theta}_2, \hat{\gamma}]$, for each market t we can:

1. Obtain an initial estimate of $\widehat{\Omega}_{jt}^{-1}$ by computing covariances of $(\hat{\xi}_{jt}, \hat{\omega}_{jt})$. This can be i.i.d. or clustered at any desired level.
2. Draw the $J_t \times 2$ matrix of structural errors $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ according to one of the below options.
3. Compute $\hat{Y}_{jt}^S = \hat{c}_{jt} = [x_{jt}, w_{jt}]\hat{\gamma} + \omega_{jt}^*$ and the exogenous portion of utility $\hat{Y}_{jt}^D = [x_{jt}, v_{jt}]\hat{\beta} + \xi_{jt}^*$.
4. Use $(\hat{\boldsymbol{Y}}_t^D, \hat{\boldsymbol{Y}}_t^S, \boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{\nu}_t)$ to solve for equilibrium prices and quantities $(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t)$ with the ζ-markup approach in (27). Note that this does not involve any endogenous quantities.
5. Treating $(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t, \boldsymbol{x}_t, \boldsymbol{w}_t)$ as data, solve for $\hat{\boldsymbol{\xi}}_t = \boldsymbol{\xi}_t(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t, \boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{\nu}_t, \hat{\theta}_2)$ and $\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t, \boldsymbol{x}_t, \boldsymbol{w}_t, \boldsymbol{\nu}_t, \hat{\theta}_2)$.
6. Construct the Jacobian terms $\frac{\partial \hat{\xi}_{jt}}{\partial \theta_2}(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$, $\frac{\partial \hat{\omega}_{jt}}{\partial \theta_2}(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$, and $\hat{D}_{jt}(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ using the analytic formulas in Appendix A.2.
7. Average $\hat{D}_{jt}(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ over several draws of $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ to construct an estimate of $E[\hat{D}_{jt}|Z_t]$.

There are three options for generating $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ suggested by Berry, Levinsohn, and Pakes (1999) and PyBLP makes all three available:

(a) "Approximate": Replace $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ with their expectation: $(E[\boldsymbol{\xi}_t], E[\boldsymbol{\omega}_t]) = (0, 0)$. This is what Berry, Levinsohn, and Pakes (1999) do. Because the function is nonlinear, this is a good approximation only if $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$ are very small.

(b) "Asymptotic": Estimate an asymptotic normal distribution for $(\xi_{jt}, \omega_{jt}) \sim N(0, \hat{\Omega})$ and then draw $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ from that distribution. This assumes that a normal approximation is reasonable.

(c) "Empirical": Draw $(\boldsymbol{\xi}_t^*, \boldsymbol{\omega}_t^*)$ from the joint empirical distribution of $(\hat{\xi}_{jt}, \hat{\omega}_{jt})$. This requires an assumption of *exchangeability*.

Finally, this approach leverages the fact that we have correctly specified the supply side. This may be problematic if we assume a multi-product oligopoly price setting game and the true equilibrium is described by collusive pricing, for example.

□ **Demand side only.** Most empirical applications of the BLP approach do not include a supply side, but estimate demand alone. This has some advantages and some disadvantages. One important implication is that we lose the *cross-equation* overidentifying restrictions, though we manage to retain the $K_3 - K_x$ *exclusion restrictions* for demand from the cost-shifters w_{jt}. Absent the supply side, we no longer have a model for marginal costs and cannot solve for equilibrium $(\hat{\boldsymbol{p}}_t, \hat{\boldsymbol{s}}_t)$. Instead, the user can supply a vector of expected prices $E[\boldsymbol{p}_t|Z_t^D]$ or allow PyBLP to construct the vector with a first-stage regression. Item 4 reduces to computing the market shares at the expected prices. The value of the approximate optimal IV then depends on how well price is explained by the exogenous instruments Z_t^D in the first stage.

One reason to omit the supply side is that including a misspecified supply side may be worse than no supply side at all. The most controversial aspect of the supply side is often the *conduct assumption* $\mathcal{H}_t$ used to recover the markup $\eta_{jt}(\mathcal{H}_t, \theta_2)$, which may not be known to the researcher prior to estimation. The good news is that testing the validity of the supply side moments amounts to a test of over-identifying restrictions. The simplest test involves estimating the full model with supply and demand to obtain $\hat{\theta}$, re-estimating the model using only demand moments g_D to obtain $g_D(\hat{\theta}_D)$ along with the optimal demand-only weighting matrix W_D, and then comparing GMM objectives in a Hausman manner (see Newey, 1985):

$$\text{LR} = N[g(\hat{\theta})'Wg(\hat{\theta}) - g_D(\hat{\theta}_D)'W_D g_D(\hat{\theta}_D)] \sim \chi^2_{K-K_x}. \tag{32}$$

There are of course alternatives based on the LM (Score) and Wald tests, which are also supported by PyBLP.

Remark 4. In our Monte Carlo study, we find a substantial additional benefit when incorporating both supply side moments as well as the approximation to the optimal IV. These benefits substantially exceed those of the demand moments with optimal IV or demand and supply moments with other instruments. We offer two explanations: first, the difference between the nonlinear form of $E[\boldsymbol{p}_t|Z_t^D]$ under the approximation to the optimal IV and its linear projection; and second, the

value of *cross equation restrictions* in (30), which use different expressions for $(\frac{\partial \xi_{jt}}{\partial \theta_2}, \frac{\partial \omega_{jt}}{\partial \theta_2})$. In our Monte Carlo exercises, we are generally able to reject misspecified conduct assumptions, but not correctly specified ones.

5. Monte Carlo experiments

■ Here we provide some Monte Carlo experiments to illustrate some of the best practices laid out in Sections 3 and 4.

□ **Monte Carlo configuration.** Our simulation configurations are loosely based on those of Armstrong (2016). The distinguishing feature is that we randomly draw utilities and costs, but solve for equilibrium prices and shares.[74] Below, we describe our baseline configurations, and in the following sections describe how we modify these configurations to compare different aspects of the problem.

For each configuration, we construct and solve 1000 different synthetic datasets. In each of $T = 20$ markets we randomly choose the number of firms from $F_t \in \{2, 5, 10\}$ and have each firm produce a number of products chosen randomly from $J_{ft} \in \{3, 4, 5\}$. This procedure generates variation in the number of firms and products across markets, which provides variation in our instruments. Sample sizes are generally between $200 < N < 600$.

We draw the structural error terms (ξ_{jt}, ω_{jt}) from a mean-zero bivariate normal distribution with variances $\sigma_\xi^2 = \sigma_\omega^2 = 0.2$ and covariance $\sigma_{\xi\omega} = 0.1$. Linear demand characteristics are $[1, x_{jt}, p_{jt}]$ and supply characteristics are $[1, x_{jt}, w_{jt}]$. The one nonlinear characteristic is x_{jt} and heterogeneity is parameterized by $\mu_{ijt} = \sigma_x x_{jt} \nu_{it}$ where we draw ν_{it} from the standard normal distribution for 1000 different individuals in each market. We draw the two exogenous characteristics (x_{jt}, w_{jt}) from the standard uniform distribution and compute the endogenous (p_{jt}, s_{jt}) with the ζ-markup approach in (27). Demand-side parameters, $[\beta_0, \beta_x, \alpha] = [-7, 6, -1]$ and $\sigma_x = 3$, generate realistic outside shares generally between $0.8 < s_{0t} < 0.9$. Supply-side parameters, $[\gamma_0, \gamma_x, \gamma_w] = [2, 1, 0.2]$, enter into a linear functional form for marginal costs: $c_{jt} = [1, x_{jt}, w_{jt}]\gamma + \omega_{jt}$. For our baseline specification, $\text{Corr}(p_{jt}, w_{jt}) \approx 0.2$ is relatively low, which implies that our cost-shifting instruments are relatively weak.[75]

In our different Monte Carlo experiments we modify this baseline problem in a number of ways. In most experiments, we consider three variants:

(a) "Simple" is the baseline problem described above.
(b) "Complex" adds a random coefficient on price: $\sigma_p = 0.2$. Nonlinear coefficients are $[x_{jt}, p_{jt}]$.
(c) "RCNL" adds a nesting parameter: $\rho = 0.5$. Each of the J_{ft} products produced by a firm is randomly assigned to one of $H = 2$ nesting groups.

We estimate two broad classes of models: demand-only ones, which we estimate with single equation GMM, and models that also include the supply side, which we estimate with multiple equation GMM.

To numerically integrate choice probabilities, we use Gauss-Hermite product rules that exactly integrate polynomials of degree 17 or less.[76] In some specifications, we compare quadrature with the other integration methods described in Section 3.

[74] The first Monte Carlo studies to solve for price and quantity as an equilibrium for BLP-type models are likely Skrainka (2012a), Armstrong (2016), and Conlon (2017). Without solving for equilibrium prices and quantities when generating the data, markups are not "endogenous" and the relevance condition for many (BLP) IV are violated.

[75] With very strong cost shifters or when the variance of (ξ_{jt}, ω_{jt}) is very small, the estimator always performs well and we can obtain very low bias and median absolute error for nearly any choice of instruments.

[76] In the Simple specification when there is only one dimension of integration, the product rule has $(17 + 1)/2 = 9$ nodes. In the Complex specification there are $9^2 = 81$ nodes. If our simulations were of higher dimension, it would be more efficient to use sparse grid integration.

TABLE 2 Various Forms of Instrumental Variables

$Z_{jt}^{Own} = \{1, x_{jt}, w_{jt}, x_{jt}^2, w_{jt}^2, x_{jt} \cdot w_{jt}\}$
$Z_{jt}^{Sums} = \{Z_{jt}^{Own}, \sum_{k \in J_{ft} \setminus \{j\}} 1, \sum_{k \notin J_{ft}} 1, \sum_{k \in J_{ft} \setminus \{j\}} x_{kt}, \sum_{k \notin J_{ft}} x_{kt}\}$
$Z_{jt}^{Local} = \{Z_{jt}^{Own}, \sum_{k \in J_{ft} \setminus \{j\}} 1(|d_{jkt}| < \text{SD}(d)), \sum_{k \notin J_{ft}} 1(|d_{jkt}| < \text{SD}(d))\}$
$Z_{jt}^{Quad} = \{Z_{jt}^{Own}, \sum_{k \in J_{ft} \setminus \{j\}} d_{jkt}^2, \sum_{k \notin J_{ft}} d_{jkt}^2\}$

Note: Optimal instruments, $Z_{jt}^{Opt,D}$ and $Z_{jt}^{Opt,D}$ in (31), are approximated with Algorithm 2 using initial estimates from a single GMM step under Z_{jt}^{Sums}. We define the difference in characteristic space from product j as $d_{jkt} = d_{kt} - d_{jt}$ for each characteristic in x_{jt}. For the Complex simulation we also include a measure of expected price $E[p_{jt}|Z_t]$ as an additional x_{jt}: fitted values from a linear regression of endogenous prices onto all exogenous variables, including the above instruments. For the RCNL simulation, we also include counts of products within the same nest in each market.

To solve the standard fixed point for $\boldsymbol{\delta}_t$ in each market, we use the SQUAREM acceleration method of Varadhan and Roland (2008) with a L^∞ tolerance of `1E-14` and limit the number of contraction evaluations to 1000. When evaluating the multinomial logit function, we use the log-sum-exp function described in Section 3 to improve numerical stability. During the first GMM step, SQUAREM starts at the solution to the simple logit (or nested logit) model; in the second step, it starts at the estimated first-stage $\hat{\boldsymbol{\delta}}_t$.

To optimize, we supply objective values and analytic gradients to a bounded limited-memory BFGS routine (L-BFGS-B), which is made available by the open-source SciPy library. We use an L^∞ projected gradient-based tolerance of `1E-5` and limit the number of major iterations to 1000.[77] Drawing different starting values from a uniform distribution with support 50% above and below the true parameter values, we solve each simulation three times and keep the solution with the smallest objective value. For each simulation, we use box constraints 1000% above and below the true values with the following exceptions: $\sigma_x \geq 0$, $\sigma_p \geq 0$, $\rho \in [0, 0.95]$, and when a supply side is estimated, $\alpha \leq -0.001$.[78]

Instruments are constructed in two stages. We list the various sets of instruments in Table 2. We begin by exclusively using "own product" characteristics and consider all quadratic interactions of (x_{jt}, w_{jt}).[79] The "BLP instruments" expand this set to include characteristics of other products. These are meant to be correlated both with the endogenous markup $\eta_{jt}(\theta_2, \boldsymbol{x}_t, \boldsymbol{w}_t)$, as well as the inverse mean utility $D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$, both of which depend on characteristics of all products in market t. The second set of instruments, Z_{jt}^{Sums}, incorporates sums of characteristics of other products, and separates products owned by the same brand f from products owned by competing brands in the same market t. The next two sets of instruments consider the *differentiation IV* of Gandhi and Houde (2019). These are variants on the BLP instruments as they represent different functions of own and rival product characteristics. Pooling products across all markets, for each pair of products (j, k) we construct the difference $d_{jkt} = x_{kt} - x_{jt}$ of the exogenous regressor. The *differentiation IV* come in two flavors: *Local* and *Quadratic*. The Local measure, Z_{jt}^{Local}, counts the number of products within a standard deviation of product j, whereas the Quadratic measure, Z_{jt}^{Quad}, sums up the aggregate distance between j and other products. Following Gandhi and Houde (2019), for the Complex simulation where there is a random coefficient on price, we construct an additional instrument using fitted values from a regression of endogenous prices onto all exogenous variables, including the constructed instruments above. For the RCNL simulation, we follow the suggestions of Berry (1994) or Gandhi and Houde (2019) and include the number of products within the same nest and market.

[77] MATLAB solvers report absolute tolerances and SciPy solvers report relative ones. This means magnitudes are not directly comparable. We discuss termination tolerances in more detail in Section 6. In practice, we recommend trying different tolerances to ensure that the optimization routine is not terminating prematurely.

[78] When the optimization software considers $\alpha = 0$, the matrix of intra-firm demand derivatives Δ_t becomes singular.

[79] One way to view this approach is as a sieve basis to approximate the optimal IV (Ai and Chen, 2003).

TABLE 3 Fixed Effect Absorption

Dimensions	Levels	Absorbed	Seconds	Megabytes
1	216	No	56	721
1	216	Yes	20	39
2	216 × 216	No	112	1414
2	216 × 216	Yes	25	43
2	36 × 1296	No	330	2498
2	36 × 1296	Yes	25	43
3	36 × 36 × 36	No	46	375
3	36 × 36 × 36	Yes	24	47

Note: This table documents the impact of fixed effect (FE) absorption on estimation speed and memory usage during one GMM step. Reported values are medians across 100 different simulations. When not absorbed, FEs are included as dummy variables. A single FE is absorbed with de-meaning; multiple, with iterative de-meaning, also known as the Method of Alternating Projections (MAP). To accommodate FEs in the Simple simulation, we first set $T = 6^4$ and $F_t = J_f = 6$ so that there are $N = 6^6 = 46,656$ products. We then randomly assign each $n \in \{1, \ldots, N\}$ to a product and add FEs drawn from the standard uniform distribution. The 1-D case has $\beta_{n \bmod 216}$. For the "square" 2-D case, we add $\beta_{n \text{ div } 216}$. The "uneven" 2-D case has $\beta_{n \bmod 36}$ and $\beta_{n \text{ div } 36}$. The 3-D case has $\beta_{n \bmod 36}$, $\beta_{n(\text{div } 36) \bmod 36}$, and $\beta_{n \text{div } 216}$.

Using parameter estimates from a single GMM step under Z_{jt}^{Sums}, we construct feasible optimal instruments with the "approximate" variant described in Algorithm 2 and compare with the "asymptotic" and "empirical" alternatives. When a supply side is included, we compute the vector of expected prices with the ζ-markup approach in (27). Absent a supply side, we estimate $E[\boldsymbol{p}_t|Z_t^D]$ with a regression of endogenous prices onto all exogenous variables, including constructed instruments.

□ **Monte Carlo results.** When reporting our Monte Carlo results, we focus on the median bias and median absolute error (MAE) of the parameter estimates. In the online appendix we provide additional results measuring the performance of standard errors and various counterfactual predictions, such as elasticities, merger effects, and welfare estimates. Computation was done on the NYU HPC cluster.[80]

□ **Fixed effects.** In Table 3, we add fixed effects to the Simple simulation and compare computational time and memory usage for experiments where we either absorb the fixed effects or include them as dummy variables. As one might expect, absorbing fixed effects dramatically reduces memory requirements by multiple orders of magnitude and can speed up computation by as much as 10 times. As expected, the largest improvements for the two-dimensional case are when one dimension is much larger than the other. Even absorbing relatively small numbers of fixed effects (216 in a single dimension) leads to a substantial reduction in memory usage and computational time. This is a likely advantage of PyBLP going forward.

□ **Fixed point iteration algorithms.** To highlight the effects of different iteration schemes from Section 3, we explore several methods of solving the system of share equations for $\boldsymbol{\delta}_t(\theta_2)$. We compare the conventional fixed-point iteration scheme proposed by Berry, Levinsohn, and Pakes (1995) to two alternative methods of solving the system of nonlinear equations without the Jacobian: DF-SANE and SQUAREM. We also compare with two Jacobian-based methods: Powell's method and Levenberg–Marquardt (LM).[81]

[80] We uniformly distribute our computation across five types of Intel Xeon processors: 71% on a E5-2690 v2 @ 3.00GHz, 13% on a E5-2660 v3 @ 2.60GHz, 10% on a E5-2690 v4 @ 2.60GHz, 5% on a Gold 6248 @ 2.50GHz, and 1% on a Gold 6148 @ 2.40GHz.

[81] Powell's method and LM are implemented in MINPACK (More, Garbow, and Hillstrom, 1980) as `HYBRJ` and `LMDER`, respectively. We do not report results for other SciPy root-finding methods such as Broyden's Method or Anderson acceleration because we found them too slow and unreliable to be worth considering.

We focus mainly on computational burden and report the results in Table 4. For the Simple and Complex simulations we vary the coefficient on the constant term β_0. A smaller value of β_0 leads to a larger share for the outside good. As shown by Dubé, Fox, and Su (2012), a smaller outside good share implies a larger value for the contraction's Lipschitz constant, which generally increases the number of required iterations. Several patterns emerge. As we shrink the outside good share from $s_{0t} = 0.91 \rightarrow 0.27$, thereby increasing the Lipschitz constant, the number of required iterations increases approximately by a factor of five times for the Simple and Complex simulations. As expected, the Jacobian-based routines are unaffected by variation of the Lipschitz constant.[82] With the SQUAREM iteration scheme the number of iterations increases, but by only 110%.

In general, we find Powell's method reliable for relatively loose tolerances, but it struggled with a tolerance of `1E-14`, and thus we do not recommend it. The DF-SANE algorithm performs substantially better than standard fixed point iteration, but less well than our two preferred algorithms: the SQUAREM fixed point acceleration scheme and the Jacobian-based LM algorithm.

In terms of other speedups, the effects of the SQUAREM iteration scheme are substantial. It reduces the number of iterations between 3-8 times without substantially increasing the cost per iteration. This is because it approximates the Newton step without actually computing the costly Jacobian. It performs well across all settings, but does particularly well in the Berry, Levinsohn, and Pakes (1995, 1999) example which includes a supply side.

The LM algorithm reduces the number of iterations, but at a higher cost per iteration. On some of the more difficult problems with a small outside good share, it performs as much as 10 times faster than fixed point iteration, and at other times roughly as fast as SQUAREM. The speedup is particularly large for the RCNL model, where the contraction is dampened by $(1 - \rho)$. When $\rho \rightarrow 1$ the contraction becomes arbitrarily slow. In our experiments, LM performs somewhat better than quasi-Newton routines were reported to perform in Reynaerts, Varadhan, and Nash (2012).[83]

□ **Fixed point iteration tricks.** We also consider some commonly employed tricks in the literature to speed up the contraction mapping and report our results in Table B1. In all cases, we use the SQUAREM algorithm. We consider two common "tweaks" from the literature: working with $\exp(\delta_{jt})$ rather than δ_{jt} to avoid taking logs and eliminating a division in the share computation, and using a "hot start" where the starting value for the iterative procedure is the δ_{jt}^{n-1} that solved the system of equations for the previous guess of θ_2. In addition, we consider the potential cost of our overflow safe modification to the log-sum-exp function.

The results in Table B1 are largely underwhelming. Once we adopt SQUAREM acceleration, additional tricks to speed up the problem seem to have little benefit. The "hot start" approach was able to reduce the number of iterations between by between 10-20%, so it is worth considering.[84] One clear recommendation is the log-sum-exp trick described in Section 3, which reduces the chance of overflow problems. This seems relatively low-cost and reduces the possibility that the iteration routine fails to find a solution to the system of equations.

□ **Numerical integration.** Given extensive past work by Heiss and Winschel (2008), Judd and Skrainka (2011), Skrainka (2012b), and Freyberger (2015), it is not surprising that the choice of integration method has a striking effect. In Figure 1 we add random coefficients to the Simple simulation and compare integration error under pseudo-Monte Carlo

[82] This is consistent with the findings of Dubé, Fox, and Su (2012) using the MPEC method.

[83] One possible explanation is that Reynaerts, Varadhan, and Nash (2012) employ a standard *Newton-Raphson* solver, whereas the MINPACK implementation of Levenberg–Marquardt, `LMDER`, is designed to be more robust to poor starting values.

[84] This introduces a potential numerical problem where the GMM objective of iteration n need not evaluate to precisely the same quantity depending on the last iteration's θ^{n-1}, although for sufficient fixed point tolerances this should not be an issue. In practice, we still recommend a tight tolerance of `1E-14`.

TABLE 4 Fixed Point Algorithms

Problem	Median s_{0t}	Algorithm	Jacobian	Termination	Mean Milliseconds	Mean Evaluations	Percent Converged
Simple simulation ($\beta_0 = -7$)	0.91	Iteration	No	Absolute L^∞	5.95	41.85	100.00%
Simple simulation ($\beta_0 = -7$)	0.91	DF-SANE	No	Absolute L^∞	3.43	16.27	100.00%
Simple simulation ($\beta_0 = -7$)	0.91	SQUAREM	No	Absolute L^∞	2.56	15.94	100.00%
Simple simulation ($\beta_0 = -7$)	0.91	SQUAREM	No	Relative L^2	2.75	15.26	100.00%
Simple simulation ($\beta_0 = -7$)	0.91	Powell	Yes	Relative L^2	3.48	16.33	28.56%
Simple simulation ($\beta_0 = -7$)	0.91	LM	Yes	Relative L^2	2.31	8.91	100.00%
Simple simulation ($\beta_0 = -1$)	0.27	Iteration	No	Absolute L^∞	29.35	212.09	100.00%
Simple simulation ($\beta_0 = -1$)	0.27	DF-SANE	No	Absolute L^∞	7.10	35.28	100.00%
Simple simulation ($\beta_0 = -1$)	0.27	SQUAREM	No	Absolute L^∞	5.40	34.58	100.00%
Simple simulation ($\beta_0 = -1$)	0.27	SQUAREM	No	Relative L^2	5.73	33.91	100.00%
Simple simulation ($\beta_0 = -1$)	0.27	Powell	Yes	Relative L^2	3.67	17.21	11.38%
Simple simulation ($\beta_0 = -1$)	0.27	LM	Yes	Relative L^2	2.35	8.92	100.00%
Complex simulation ($\beta_0 = -7$)	0.91	Iteration	No	Absolute L^∞	8.35	45.02	100.00%
Complex simulation ($\beta_0 = -7$)	0.91	DF-SANE	No	Absolute L^∞	4.52	17.67	100.00%
Complex simulation ($\beta_0 = -7$)	0.91	SQUAREM	No	Absolute L^∞	3.32	16.14	100.00%
Complex simulation ($\beta_0 = -7$)	0.91	SQUAREM	No	Relative L^2	3.50	15.50	100.00%
Complex simulation ($\beta_0 = -7$)	0.91	Powell	Yes	Relative L^2	4.03	14.89	32.29%
Complex simulation ($\beta_0 = -7$)	0.91	LM	Yes	Relative L^2	2.85	8.98	100.00%
Complex simulation ($\beta_0 = -1$)	0.28	Iteration	No	Absolute L^∞	39.35	216.80	100.00%
Complex simulation ($\beta_0 = -1$)	0.28	DF-SANE	No	Absolute L^∞	8.92	36.23	100.00%
Complex simulation ($\beta_0 = -1$)	0.28	SQUAREM	No	Absolute L^∞	7.03	34.75	100.00%
Complex simulation ($\beta_0 = -1$)	0.28	SQUAREM	No	Relative L^2	7.51	34.09	100.00%
Complex simulation ($\beta_0 = -1$)	0.28	Powell	Yes	Relative L^2	4.80	17.70	9.71%
Complex simulation ($\beta_0 = -1$)	0.28	LM	Yes	Relative L^2	2.90	8.93	100.00%
RCNL simulation ($\rho = 0.5$)	0.92	Iteration	No	Absolute L^∞	21.63	93.40	100.00%
RCNL simulation ($\rho = 0.5$)	0.92	DF-SANE	No	Absolute L^∞	9.62	32.30	100.00%
RCNL simulation ($\rho = 0.5$)	0.92	SQUAREM	No	Absolute L^∞	8.45	33.49	100.00%
RCNL simulation ($\rho = 0.5$)	0.92	SQUAREM	No	Relative L^2	8.53	31.33	100.00%
RCNL simulation ($\rho = 0.5$)	0.92	Powell	Yes	Relative L^2	4.67	14.44	60.65%
RCNL simulation ($\rho = 0.5$)	0.92	LM	Yes	Relative L^2	3.27	8.89	100.00%
RCNL simulation ($\rho = 0.8$)	0.92	Iteration	No	Absolute L^∞	58.01	250.34	100.00%
RCNL simulation ($\rho = 0.8$)	0.92	DF-SANE	No	Absolute L^∞	16.30	55.50	99.92%

(*Continued*)

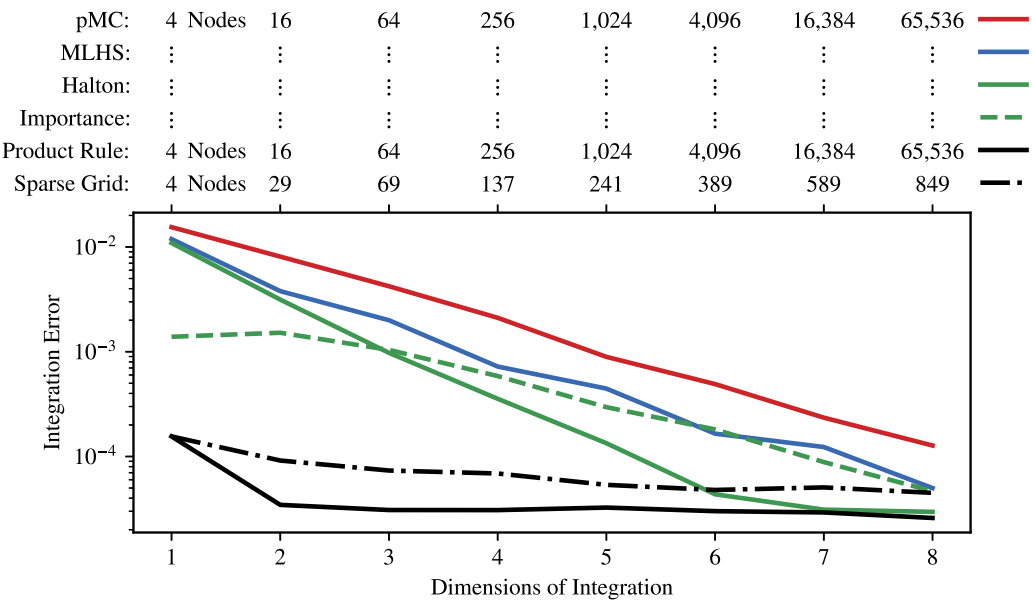
TABLE 4 Continued

Problem	Median s_{0t}	Algorithm	Jacobian	Termination	Mean Milliseconds	Mean Evaluations	Percent Converged
RCNL simulation ($\rho = 0.8$)	0.92	SQUAREM	No	Absolute L^∞	14.04	55.54	100.00%
RCNL simulation ($\rho = 0.8$)	0.92	SQUAREM	No	Relative L^2	13.95	51.70	100.00%
RCNL simulation ($\rho = 0.8$)	0.92	Powell	Yes	Relative L^2	6.48	20.16	61.48%
RCNL simulation ($\rho = 0.8$)	0.92	LM	Yes	Relative L^2	3.62	9.64	100.00%
NEVO example	0.54	Iteration	No	Absolute L^∞	12.95	86.64	100.00%
NEVO example	0.54	DF-SANE	No	Absolute L^∞	5.50	25.40	100.00%
NEVO example	0.54	SQUAREM	No	Absolute L^∞	4.05	24.06	100.00%
NEVO example	0.54	SQUAREM	No	Relative L^2	4.26	22.80	100.00%
NEVO example	0.54	Powell	Yes	Relative L^2	3.84	17.06	29.20%
NEVO example	0.54	LM	Yes	Relative L^2	2.62	9.34	100.00%
BLP example	0.89	Iteration	No	Absolute L^∞	152.72	203.04	100.00%
BLP example	0.89	DF-SANE	No	Absolute L^∞	34.93	42.37	100.00%
BLP example	0.89	SQUAREM	No	Absolute L^∞	31.54	40.61	100.00%
BLP example	0.89	SQUAREM	No	Relative L^2	31.43	39.38	100.00%
BLP example	0.89	Powell	Yes	Relative L^2	26.40	19.31	9.34%
BLP example	0.89	LM	Yes	Relative L^2	17.64	8.71	100.00%

Note: This table documents the impact of algorithm choice on solving the nested fixed point. Reported values are medians across 100 different simulations and 10 identical runs of the example problems. We report the number of milliseconds and contraction evaluations needed to solve the fixed point, averaged across all markets and one GMM step's objective evaluations. We also report each algorithm's convergence rate: the percent of times no numerical errors were encountered and a limit of 1000 iterations was not reached. We configure simple iteration with an absolute L^∞ norm tolerance of `1E-14` and compare it with DF-SANE and SQUAREM. The MINPACK (More, Garbow, and Hillstrom, 1980) implementations of algorithms that do require a Jacobian—a modification of the Powell hybrid method and Levenberg–Marquardt (LM)—only support relative L^2 norm tolerances, so for comparison's sake we also include SQUAREM with the same termination condition. Simulations are configured as described in Section 5, except for the coefficient on the constant term, β_0, and the nesting parameter, ρ, which we vary to document the effects of decreasing the outside share s_{0t} and of dampening the contraction in the RCNL model. The example problems from Nevo (2000b) and Berry, Levinsohn, and Pakes (1995, 1999) are the replications described in Section 6.

FIGURE 1

INTEGRATION ERROR [Color figure can be viewed at wileyonlinelibrary.com]



This plot documents the performance of different numerical integration methods in terms of root mean square error of market shares. Reported values are medians across 100 different markets and are on a logarithmic scale. To document the curse of dimensionality, as we increase the number of random coefficients K_2, we also increase the number of integration nodes I_t to match that of a Gauss-Hermite product rule that exactly integrates polynomials of degree 7 or less: $I_t = 4^{K_2}$. To accommodate $K_2 > 1$ dimensions of integration in the Simple simulation, we draw additional exogenous characteristics from the standard uniform distribution. On these new characteristics, we add uncorrelated and mean-zero random coefficients. For comparability's sake, we use the unaltered Simple simulation's market shares $\mathcal{S}_{jt}$ for all K_2, and set each random coefficient's variance to $1/K_2$ so that the distribution of utility is invariant to K_2. We use 1 million pseudo-Monte Carlo (pMC) draws to precisely compute $\boldsymbol{\delta}_t = D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$. With this, we compute the integration error $||\boldsymbol{\mathcal{S}}_t - \boldsymbol{s}_t(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)||_2$ of $s_{jt}(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)$ from (24) under various integration methods and nodes I_t. In addition to a less-precise pMC rule, we also consider Modified Latin Hypercube Sampling (MLHS); Halton draws where in each dimension we use a different prime (2, 3, etc.), discard the first 1000 points, and scramble the sequence according to the recipe in Owen (2017); importance sampling of Halton draws according to the Berry, Levinsohn, and Pakes (1995) procedure described in Section 3 with $\widetilde{\theta}_2$ and $\boldsymbol{\delta}_t$ are replaced by their true values; a Gauss-Hermite product rule that exactly integrates polynomials of degree 7 or less; and sparse grids with the same polynomial order (Heiss and Winschel, 2008).

(pMC), Modified Latin Hypercube Sampling (MLHS), scrambled Halton draws, importance sampling, and Gauss-Hermite quadrature that exactly integrates polynomials of degree 7 or less.[85] As we increase the dimension of integration K_2, we decrease the variance of the random coefficients so that the variance of $\mu_{ijt}(\widetilde{\theta}_2)$ remains fixed. As a measure of integration error for I_t nodes, we compute $||\boldsymbol{\mathcal{S}}_t - \boldsymbol{s}_t(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)||_2$ where $\boldsymbol{\mathcal{S}}_t$ is the vector of true market shares and the vector of mean utilities $\boldsymbol{\delta}_t = D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$ is precisely computed with one million pMC draws.[86]

To document the curse of dimensionality (CoD), as we add random coefficients we also increase the number of integration nodes I_t to match the size of the product rule. For the same number of draws, the quadrature routines outperform the qMC and pMC routines. As I_t increases, the CoD kicks in and the relative performance of pMC, MLHS, Halton draws, and importance

[85] For each dimension of Halton draws we use a different prime (2, 3, etc.), discard the first 1000 points, and scramble the sequence (Owen, 2017). When doing importance sampling, we use Halton draws. In the online appendix we find little difference between importance sampling procedures based on pMC, MLHS, or Halton draws.

[86] In the online appendix we find similar results for a relative measure of integration error.

sampling improves relative to quadrature. Of the nonquadrature methods, scrambled Halton draws perform the best, particularly for a larger number of random coefficients. In our setting, MLHS and importance sampling seem largely underwhelming.[87]

Particularly for a small number of random coefficients, the product rule provides the most accuracy. For more than $K_2 = 5$ random coefficients, sparse grids provide similar accuracy and require fewer than 10% as many nodes. In Table B2 we confirm that these conclusions translate to performance of parameter estimates. When compared to the product rule, 10 times as many pMC, MLHS, Halton, or importance sampling draws provide similar accuracy, but take 3-20 times as long to estimate.

Our recommendation is to use a product rule for estimating models with only a few random coefficients. For larger numbers of random coefficients we recommend either sparse grids or scrambled Halton draws. However, performance is likely to be context-dependent. After obtaining an estimate of $\widetilde{\theta}_2$, we recommend verifying that the chosen integration rule performs well relative to feasible alternatives in that setting. This can be done with a procedure like the one used to construct Figure 1: precisely compute $\boldsymbol{\delta}_t = D_t^{-1}(\boldsymbol{\mathcal{S}}_t, \widetilde{\theta}_2)$ using an integration rule with many more nodes than would be feasible during estimation, compute approximations $\boldsymbol{s}_t(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)$ with feasible numbers of integration nodes I_t for various integration rules, and compare integration errors $||\boldsymbol{\mathcal{S}}_t - \boldsymbol{s}_t(\boldsymbol{\delta}_t, \widetilde{\theta}_2; I_t)||_2$.[88]

□ **Instruments and supply moments.** We also consider several different choices of instruments as well as models which include both supply and demand moments or demand moments only. In Section 4, we present a slightly different construction of the Chamberlain (1987) optimal instruments for the BLP problem than the construction proposed in Reynaert and Verboven (2014). In Table 5, we present simulation results using own characteristics only, sums of characteristics for other products, both the Local and Quadratic forms of the Gandhi and Houde (2019) differentiation IV, and the "approximate" version of the feasible optimal instruments from equation (31).

We find that in most settings the feasible approximation to the optimal instruments performs best, which is consistent with the findings of Reynaert and Verboven (2014). We also find that the differentiation IV outperform the sums of characteristics BLP instruments, as Gandhi and Houde (2019) suggest.

Including moments from a correctly specified supply side substantially improves performance for most sets of instruments with the exception of the "Own" instruments, where the performance gains are limited. We also find that the feasible approximation to the optimal instruments performs much better when a correctly specified supply side is included. In fact, with both optimal instruments and a supply side, the bias is all but eliminated in most of our Monte Carlo experiments, and the MAE is substantially reduced, particularly for the random coefficients. This does not match Reynaert and Verboven (2014) who find that including the supply side has little effect once feasible optimal instruments are used.[89] We highlight the main theoretical difference in Section 4.[90] The gains from including a correctly specified supply side also translate to nonlinear functions of model parameters such as average elasticities and counterfactual simulations such as price effects of mergers. We provide additional details in the online appendix.

[87] In the online appendix we find that importance sampling performs slightly better when there is a larger outside share and a random coefficient on the constant term, although this is sensitive to how we measure integration error. Importance sampling is expected to perform worse when done at an estimate of $\widetilde{\theta}_2$ instead of the true parameters.

[88] PyBLP makes this type of procedure quick to set up with post-estimation methods that compute mean utilities and market shares under different integration configurations.

[89] A likely important distinction is that their simulations all appear to include "strong cost shifters" in which case all estimators perform quite well, whereas our base specification includes a "weak cost shifter" w_{jt} with $\text{Corr}(p_{jt}, w_{jt}) \approx 0.2$.

[90] See Remark 3. The punchline is that $E[\boldsymbol{p}_t|Z_t]$ can be approximated with a linear projection onto instruments, or with the nonlinear estimate from solving for $(\boldsymbol{p}_t^*, \boldsymbol{s}_t^*)$ in equilibrium. The addition of the supply side to the optimal IV problem facilitates the latter.

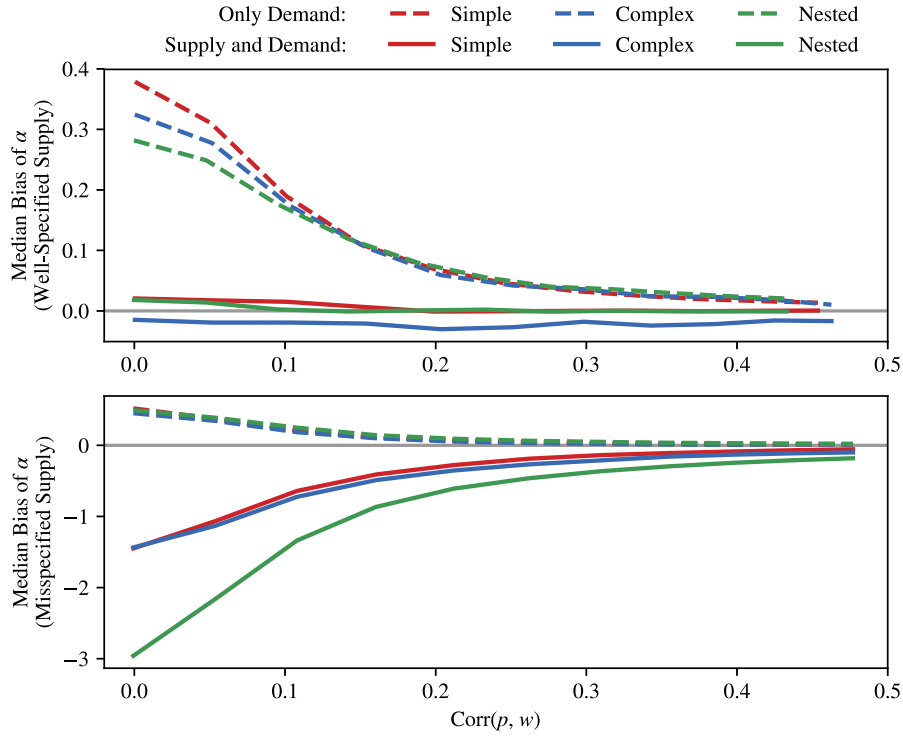
TABLE 5 Alternative Instruments

			True Value				Median Bias				Median Absolute Error				
Simulation	Supply	Instruments	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	No	Own	0.6	−1	3			0.126	−0.045			0.238	0.257		
Simple	No	Sums	0.6	−1	3			0.224	−0.076			0.257	0.208		
Simple	No	Local	0.6	−1	3			0.181	−0.056			0.242	0.235		
Simple	No	Quadratic	0.6	−1	3			0.206	−0.085			0.263	0.239		
Simple	No	Optimal	0.8	−1	3			0.218	−0.049			0.250	0.174		
Simple	Yes	Own	1.4	−1	3			0.021	0.006			0.226	0.250		
Simple	Yes	Sums	1.5	−1	3			0.054	−0.020			0.193	0.196		
Simple	Yes	Local	1.4	−1	3			0.035	−0.006			0.207	0.229		
Simple	Yes	Quadratic	1.4	−1	3			0.047	−0.022			0.217	0.237		
Simple	Yes	Optimal	2.2	−1	3			0.005	0.012			0.170	0.171		
Complex	No	Own	1.1	−1	3	0.2		−0.025	0.000	−0.200		0.381	0.272	0.200	
Complex	No	Sums	1.1	−1	3	0.2		0.225	−0.132	−0.057		0.263	0.217	0.200	
Complex	No	Local	1.0	−1	3	0.2		0.184	−0.107	−0.085		0.274	0.236	0.200	
Complex	No	Quadratic	1.0	−1	3	0.2		0.200	−0.117	−0.198		0.299	0.243	0.200	
Complex	No	Optimal	1.6	−1	3	0.2		0.191	−0.119	0.001		0.274	0.195	0.200	
Complex	Yes	Own	3.9	−1	3	0.2		−0.213	0.060	0.208		0.325	0.263	0.208	
Complex	Yes	Sums	3.3	−1	3	0.2		0.018	−0.104	0.052		0.203	0.207	0.180	
Complex	Yes	Local	3.4	−1	3	0.2		−0.043	−0.078	0.135		0.216	0.225	0.200	
Complex	Yes	Quadratic	3.5	−1	3	0.2		−0.028	−0.067	0.116		0.237	0.227	0.200	
Complex	Yes	Optimal	4.9	−1	3	0.2		−0.024	−0.036	−0.002		0.193	0.171	0.191	
RCNL	No	Own	5.1	−1	3		0.5	0.423	−1.235		0.199	0.463	1.393		0.218
RCNL	No	Sums	3.4	−1	3		0.5	0.222	−0.187		0.013	0.237	0.300		0.034
RCNL	No	Local	3.4	−1	3		0.5	0.216	−0.194		0.016	0.247	0.324		0.039
RCNL	No	Quadratic	3.4	−1	3		0.5	0.211	−0.231		0.018	0.252	0.354		0.042
RCNL	No	Optimal	4.3	−1	3		0.5	0.217	−0.031		−0.008	0.230	0.155		0.021
RCNL	Yes	Own	9.7	−1	3		0.5	0.205	−0.955		0.162	0.301	1.201		0.189
RCNL	Yes	Sums	7.0	−1	3		0.5	0.047	−0.154		0.018	0.148	0.275		0.034
RCNL	Yes	Local	7.1	−1	3		0.5	0.038	−0.148		0.020	0.172	0.298		0.039
RCNL	Yes	Quadratic	7.1	−1	3		0.5	0.036	−0.169		0.022	0.168	0.330		0.042
RCNL	Yes	Optimal	10.0	−1	3		0.5	0.008	−0.003		0.002	0.111	0.136		0.017

Note: This table documents bias and variance of parameter estimates over 1000 simulated datasets for different instruments. Own instruments are $[1, x_{jt}, w_{jt}, x_{jt}^2, w_{jt}^2, x_{jt} \cdot w_{jt}]$. Sums include own and competitor product characteristics. Local and Quadratic instruments follow the definitions in Gandhi and Houde (2019). Optimal instruments are the "approximate" version from Algorithm 2. All instruments are defined in Table 2. For all problems, we use a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

FIGURE 2

INSTRUMENT STRENGTH AND MISSPECIFICATION [Color figure can be viewed at wileyonlinelibrary.com]



Each plot documents how bias of the linear parameter on price, α, decreases with the strength of the cost shifter w_{jt}, which is included as a demand-side instrument. To weaken or strengthen the instrument, we vary its supply-side parameter from $\gamma_w = 0$ to $\gamma_w = 1$, and report the correlation this induces between w_{jt} and prices p_{jt}. Reported bias values are medians across 1000 different simulations. The top plot reports results for the simulation configurations described in Section 5. In the bottom plot, we simulate data according to perfect competition (i.e., prices are set equal to marginal costs instead of those that satisfy Bertrand-Nash first order conditions), but continue to estimate the model under the assumption of imperfect competition. For all problems, we use the "approximate" version of the feasible optimal instruments and a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

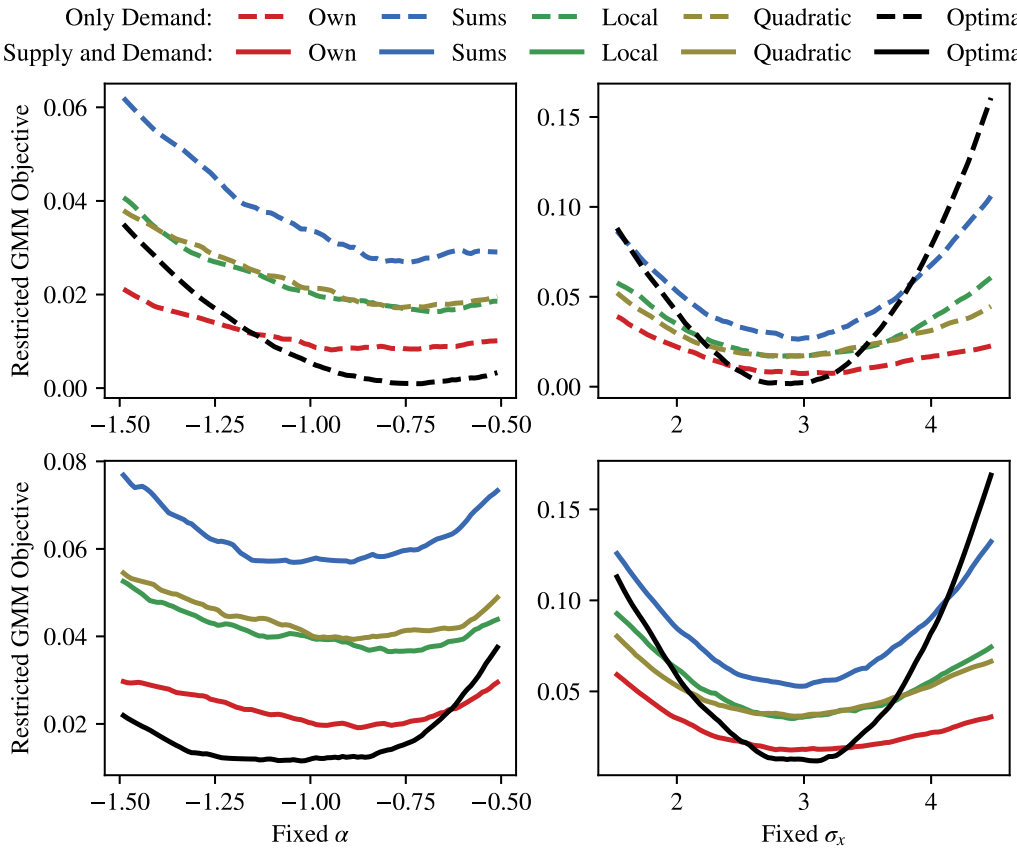
We also show that the incremental value of the supply-side restrictions is largest when the exogenous cost shifter w_{jt} is weakest. We plot results in Figure 2 where we vary the magnitude of the coefficient γ_w, which governs how responsive marginal costs are to the exogenous cost shifter. Reducing this coefficient reduces the correlation between p_{jt} and w_{jt}. When the cost-shifting instrument is very weak with $\text{Corr}(p_{jt}, w_{jt}) \approx 0.05$, this increases both the bias and the variance of the price parameter α consistent with Armstrong (2016). However, if we include the correctly specified supply restrictions (with optimal instruments) we are able to eliminate the bias and substantially reduce the variance of the estimates.[91] This is consistent with the "folklore" around Berry, Levinsohn, and Pakes (1995), where the parameters were difficult to estimate absent supply moments, and it is in contrast to Reynaert and Verboven (2014) who do not find substantial benefits of including supply-side restrictions once optimal instruments are employed.[92] The

[91] We should be cautious because Berry and Haile (2014) suggest that absent any cost-shifters, the model may not be nonparametrically identified. In our Monte Carlo examples, even as the coefficient γ_w becomes small, $\gamma_x > 0$ so that the observed characteristics do a good job explaining p_{jt} (the effective "first-stage" for price remains nontrivial).

[92] There are several caveats/differences: simultaneous supply and demand is not the main focus of their work; they construct optimal instruments so that the model is just-identified, whereas our approach constructs them to be over-identified; their excluded cost-shifter is effectively stronger than ours.

FIGURE 3

PROFILED GMM OBJECTIVE WITH ALTERNATIVE INSTRUMENTS [Color figure can be viewed at wileyonlinelibrary.com]



Each plot profiles the GMM objective with respect to a single parameter for the Simple simulation. We fix either α or σ_x, re-optimize over other parameters, and plot median restricted objective values over 100 different simulations. The left column profiles the objective over the price parameter α, whereas the right column profiles over the random coefficient σ_x. The top row uses moments from demand alone, whereas the bottom row uses both supply and demand moments. Own instruments are $[1, x_{jt}, w_{jt}, x_{jt}^2, w_{jt}^2, x_{jt} \cdot w_{jt}]$. Sums include own and competitor product characteristics. Local and Quadratic instruments follow the definitions in Gandhi and Houde (2019). Optimal instruments are the "approximate" version from Algorithm 2. All instruments are defined in Table 2. For all problems, we use a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

second panel of Figure 2 suggests that we should be cautious. We generate the data from an assumption of perfect competition, and then impose the Bertrand-Nash multiproduct-oligopoly price setting assumption when construct the supply moments. We illustrate that incorporating moments from an incorrectly specified supply side is worse than not incorporating supply moments because it induces bias in the α parameter.

We find that that estimates are less sensitive to the precise method used to compute the feasible approximation to the optimal instruments, and we report those results in Table B3. We also find that computing feasible optimal IV with the recipe of Berry, Levinsohn, and Pakes (1999) under the wrong model of firm conduct outperforms computing $E[p_{jt}|Z_t]$ via a "first stage" linear regression when *only* demand moments are used. We document this in Figure B1.

We further illustrate these advantages in Figure 3 where we plot the profiled GMM objective function using our Simple simulation. In this exercise, we hold fixed either α or σ_x and

re-optimize the GMM objective over the other parameters. We then plot the profiled objective over a grid of values for α or σ_x. We repeat this exercise for various sets of instruments, and with or without supply moments. The resulting plots indicate that the approximation to the optimal instruments and the inclusion of the supply moments makes the resulting objective function steeper about the minimum. This suggests that stronger instruments may aid both in numerical optimization and parametric identification of parameters as well as improved efficiency.[93] The minimum under the optimal instruments is generally closer to zero. This allows us to reject misspecified models of supply and fail to reject correctly specified models with the LR test in (32) under the approximation to the optimal IV.

Consistent with Gandhi and Houde (2019), our recommendation is to start with differentiation IV in a first stage including some version of "expected price" and, assuming firm conduct is known, to compute feasible optimal instruments in a second stage. The substantial small sample benefits of including optimal instruments suggest that they should be employed more widely, particularly when there are multiple random coefficients.

□ **Problem size.** We might also be interested in how the BLP problem scales as we vary its size. We display our results in Figure 4. As we increase the number of markets T, we can substantially improve both the bias and the efficiency of our estimates for both α and σ_x. For a small number of markets and without additional moments from the supply side, the bias in α can be substantial. For more than $T > 100$ markets we get similar (asymptotic) performance for models with and without the additional supply-side restrictions.

In the online appendix we describe how the both the computational time and econometric performance of the estimator is affected by the scale of the problem. We find that computational time is roughly linear in the number of markets T, and that it grows at a rate closer to $\sqrt{J_t}$ as we increase the number of products. When we include both supply and demand the computational time appears to grow more quickly than J_t.

Our Monte Carlo exercises are fairly simple and benefit from variation in the number of products per market, but when using only demand-side restrictions, it appears as if $T = 40$ markets is roughly enough to obtain "reasonable" parameter estimates in terms of bias and efficiency. These results are somewhat in contrast to those in Armstrong (2016) who finds that when T is small, as J_t becomes large the instruments become weak and the estimator performs poorly. We find that when we increase the size of J_t the estimator performs better rather than worse. We attribute this to three main differences: we allow for variation in the number of products per firm across markets, we use the feasible approximation to the optimal instruments, and in some specifications we include the additional supply moments.

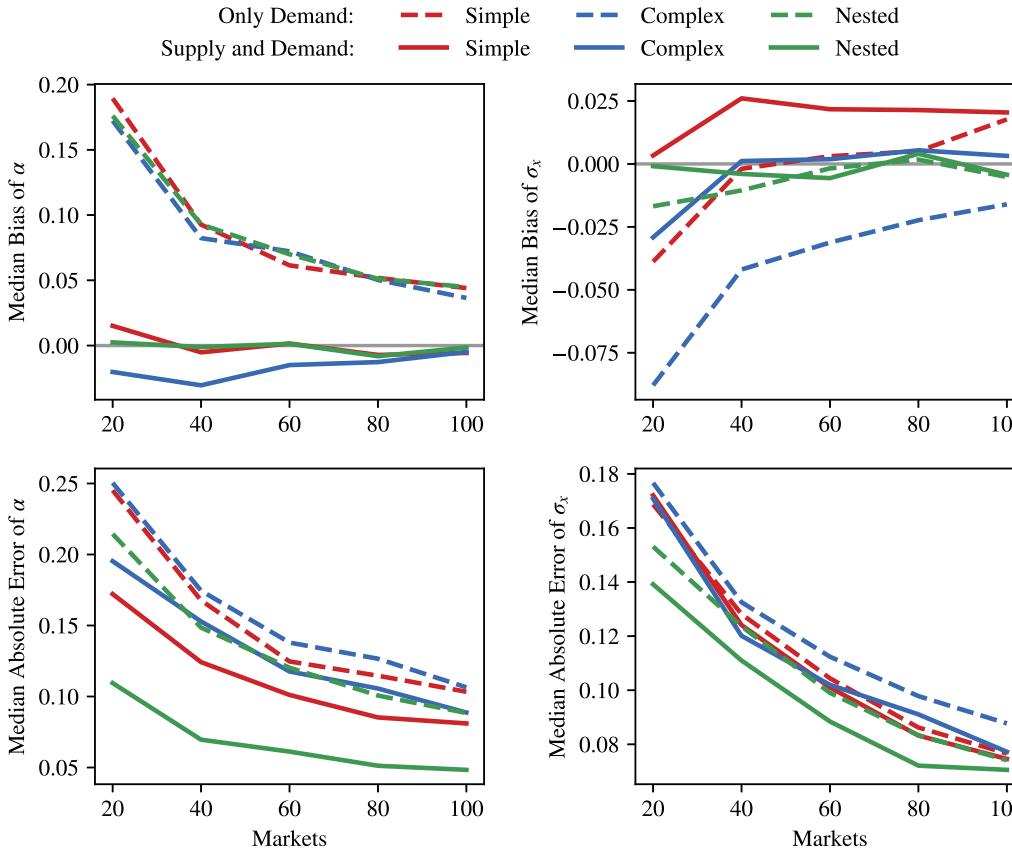
□ **Optimization algorithms.** We consider an array of different optimization algorithms and report our results regarding convergence in Table 6. With the exception of the derivative-free Nelder-Mead algorithm, most of the optimizers reliably find an optimum that satisfies first and second order conditions. Our preferred algorithms are Knitro's Interior/Direct algorithm and BFGS-based algorithms in SciPy as they provide the best speed and reliability. In the online appendix we consider an even wider array of algorithms and report parameters estimates.

Our findings are somewhat different from those of Knittel and Metaxoglou (2014), who report that different optimization algorithms find a variety of local minima and often fail to converge to a valid minima. We obtain essentially the opposite result. For all derivative-based optimization algorithms in Table 6, we find that more than 99% of all simulation runs converge to a local minimum, which we define as having an L^∞ norm of the gradient sufficiently close to zero and a positive semi-definite Hessian matrix. We should caution that our simulations are simple enough to be run thousands of times, so it may not be surprising that most optimization

[93] In practice, "weak identification" in nonlinear models arises when the objective function becomes flat with respect to the parameters (Stock and Wright, 2000).

FIGURE 4

PROBLEM SCALING [Color figure can be viewed at wileyonlinelibrary.com]



These plots document how bias and variance of parameter estimates decrease with the number of markets T. The top row plots median bias across 1000 simulations, whereas the bottom row plots median absolute error across the same simulations. The left column reports results for α and the right column reports results for σ_x. For all problems, we use the "approximate" version of the feasible optimal instruments and a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

software packages appear to work well in low dimensions. It may also be the case that various numerical fixes and improvements to the fixed point iteration problem may have resolved some of the issues with optimization.[94] In general, strong instruments and minimal numerical error lead to steep and smooth objective functions, which are easier to optimize.

In practice, we recommend placing box constraints on parameters, using gradient-based optimization routines, and setting tight optimization tolerances. We also recommend trying multiple optimizers and starting values to check for agreement.

6. Replication exercises

■ Here we provide replications using PyBLP for well-known BLP applications.

[94] We use the approximation to the optimal instruments, which tends to make the objective steep about the maximum as demonstrated in Figure 3. We also substantially reduce numerical error by using a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

TABLE 6 Optimization Algorithms

							Percent of Runs		Median, First GMM Step			
Simulation	Supply	$\|\theta_2\|$	Software	Algorithm	Gradient	Termination	Converged	PSD Hessian	Seconds	Evaluations	$q = \bar{g}W\bar{g}$	$\|\|\nabla q\|\|_\infty$
Simple	No	1	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	0.2	4	1.10E-08	8.30E-07
Simple	No	1	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	0.2	4	8.19E-09	7.28E-07
Simple	No	1	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	0.6	11	1.58E-08	1.03E-06
Simple	No	1	SciPy	TNC	Yes	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	99.9%	99.8%	0.6	10	3.89E-24	9.61E-15
Simple	No	1	SciPy	Nelder-Mead	No	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	66.5%	100.0%	19.6	115	1.08E-24	4.70E-15
Simple	Yes	2	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	0.6	5	2.17E-06	3.54E-06
Simple	Yes	2	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	0.4	4	2.18E-06	3.32E-06
Simple	Yes	2	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	2.0	11	2.61E-06	5.26E-06
Simple	Yes	2	SciPy	TNC	Yes	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	99.8%	100.0%	2.2	18	2.15E-06	5.12E-11
Simple	Yes	2	SciPy	Nelder-Mead	No	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	53.3%	100.0%	31.7	251	2.14E-06	9.69E-13
Complex	No	3	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	96.9%	0.7	6	1.92E-07	4.11E-06
Complex	No	3	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	93.6%	0.4	6	1.83E-07	3.59E-06
Complex	No	3	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	96.5%	1.9	26	2.25E-07	5.14E-06
Complex	No	3	SciPy	TNC	Yes	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	100.0%	86.9%	1.8	20	5.08E-20	1.61E-12
Complex	No	3	SciPy	Nelder-Mead	No	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	54.0%	74.6%	30.1	275	1.18E-24	1.39E-14
Complex	Yes	4	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	93.7%	1.9	9	3.20E-06	6.11E-06
Complex	Yes	4	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	94.1%	1.4	9	3.12E-06	5.57E-06
Complex	Yes	4	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	94.2%	4.6	28	3.19E-06	6.36E-06
Complex	Yes	4	SciPy	TNC	Yes	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	99.5%	99.5%	5.7	31	2.87E-06	4.02E-10
Complex	Yes	4	SciPy	Nelder-Mead	No	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	45.5%	99.5%	64.6	480	2.80E-06	1.83E-12
RCNL	No	2	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	1.7	10	1.67E-08	5.72E-06
RCNL	No	2	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	99.9%	1.9	11	1.52E-09	1.47E-06
RCNL	No	2	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	4.3	25	2.64E-09	1.95E-06
RCNL	No	2	SciPy	TNC	Yes	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	100.0%	99.6%	4.1	22	3.56E-19	1.16E-11
RCNL	No	2	SciPy	Nelder-Mead	No	$\|\|\theta_2^n - \theta_2^{n-1}\|\|_\infty$	56.4%	98.7%	60.5	243	1.27E-25	2.53E-14
RCNL	Yes	3	Knitro	Interior/Direct	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	3.4	13	2.83E-06	9.95E-06
RCNL	Yes	3	SciPy	L-BFGS-B	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	3.2	12	2.66E-06	3.85E-06
RCNL	Yes	3	SciPy	BFGS	Yes	$\|\|\nabla q\|\|_\infty$	100.0%	100.0%	9.3	37	2.73E-06	4.41E-06

(*Continued*)

TABLE 6 Continued

							Percent of Runs		Median, First GMM Step			
Simulation	Supply	$\lvert\theta_2\rvert$	Software	Algorithm	Gradient	Termination	Converged	PSD Hessian	Seconds	Evaluations	$q = \bar{g}'W\bar{g}$	$\lvert\lvert\nabla q\rvert\rvert_\infty$
RCNL	Yes	3	SciPy	TNC	Yes	$\lvert\lvert\theta_2^n - \theta_2^{n-1}\rvert\rvert_\infty$	100.0%	100.0%	7.2	24	2.95E-06	2.24E-09
RCNL	Yes	3	SciPy	Nelder-Mead	No	$\lvert\lvert\theta_2^n - \theta_2^{n-1}\rvert\rvert_\infty$	39.0%	100.0%	115.2	423	2.93E-06	4.60E-12

Note: This table documents optimization convergence statistics over 1000 simulated datasets for different optimization algorithms. For comparison's sake, we only perform one GMM step and do not set parameter bounds. We report each algorithm's convergence rate (the percent of times the algorithm reported that it successfully found an optimum before reaching 1000 iterations) and the percent of times the final Hessian matrix was positive semidefinite. We also report medians for the number of seconds needed to solve each problem, the number of objective evaluations, the final GMM objective value, and the L^∞ norm of the gradient. We configure three algorithms to terminate with gradient-based L^∞ norms of `1E-5`: Interior/Direct from Knitro, along with L-BFGS-B (limited-memory BFGS) and BFGS from SciPy. Because SciPy's derivative-free Nelder-Mead algorithm does not support gradient-based termination, we instead configure it to terminate with an absolute parameter-based L^∞ norm of `1E-5` and for comparison's sake use the same configuration for SciPy's TNC (truncated Newton) algorithm. For all problems, we use the "approximate" version of the feasible optimal instruments and a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

FIGURE 5

NEVO (2000b) REPLICATION CODE [Color figure can be viewed at wileyonlinelibrary.com]

```
import numpy as np
import pandas as pd

import pyblp

problem = pyblp.Problem(
    product_formulations=(
        pyblp.Formulation('0 + prices', absorb='C(product_ids)'),                    # Linear demand
        pyblp.Formulation('1 + prices + sugar + mushy'),                             # Nonlinear demand
    ),
    agent_formulation=pyblp.Formulation('0 + income + income_squared + age + child'),  # Demographics
    product_data=pd.read_csv(pyblp.data.NEVO_PRODUCTS_LOCATION),
    agent_data=pd.read_csv(pyblp.data.NEVO_AGENTS_LOCATION)
)

results = problem.solve(
    sigma=np.diag([0.3302, 2.4526, 0.0163, 0.2441]),         # Starting values for unobserved heterogeneity
    pi=[
        [ 5.4819,  0,      0.2037, 0     ],                  # Starting values for observed heterogeneity
        [15.8935, -1.2000, 0,      2.6342],
        [-0.2506,  0,      0.0511, 0     ],
        [ 1.2650,  0,     -0.8091, 0     ]
    ],
    method='1s',                                             # One-step GMM
    optimization=pyblp.Optimization('bfgs', {'gtol': 1e-5})  # Gradient-based termination tolerance
)

elasticities = results.compute_elasticities()
markups = results.compute_markups()
```

This Python code demonstrates how to construct and solve the problem from Nevo (2000b) in PyBLP. Names in the formulation objects correspond to variable names in the datasets, which are packaged with PyBLP and in this example are loaded into memory with the Python package `pandas`. Although not an explicit dependency of PyBLP, `pandas` is a convenient package for loading data into data frames, and comes pre-packaged in Anaconda installations. In addition to `pandas` data frames, PyBLP can also handle other data types such as NumPy structured arrays or simple dictionaries. Most estimation outputs are stored as attributes of the problem results class. Post-estimation outputs such as elasticities and markups can be computed with class methods. Estimation results are reported in Table 7.

□ **Nevo (2000b).** In our first replication we estimate the model of Nevo (2000b) on its publicly available "fake data." This problem is notable because it includes a combination of observable demographics, unobserved heterogeneity, and product fixed effects. We demonstrate how to construct and solve the problem with PyBLP in Figure 5.

We estimate the model three times: once following the original Nevo (2000b) example,[95] a second time with an optimization tolerance of `1E-5` instead of `1E-4`,[96] and a third time using the demand side only feasible optimal instruments described above. We obtain nearly identical estimates of mean own-price elasticities and markups for each specification despite different parameter estimates for the Price × Income terms that are nearly collinear in the data.

□ **Berry, Levinsohn, and Pakes (1995, 1999).** For our second replication, we consider the problem in Berry, Levinsohn, and Pakes (1995), which lacks demographic interactions and product fixed effects but adds a supply side and allows the price coefficient to vary with income. We provide the PyBLP formulation of the problem in Figure 6. This time, we access Python through the R package `reticulate` (Allaire et al., 2017) to demonstrate how PyBLP can be used in R.

[95] Replication is straightforward because the original instruments are provided along with the data and the reported estimates are from one-step GMM with the two-stage least squares (2SLS) weighting matrix $(Z'Z)^{-1}$. The data we use are distributed with PyBLP.

[96] It is well known that the code accompanying Nevo (2000b) set tolerances too loose. However, with a scaled GMM objective value of $Nq(\hat{\theta}) = 4.56$, our results with a tighter tolerance are identical to those reported by Dubé, Fox, and Su (2012) using the MPEC approach.

FIGURE 6

BERRY, LEVINSOHN, AND PAKES (1995, 1999) REPLICATION CODE [Color figure can be viewed at wileyonlinelibrary.com]

```
library(readr)
library(reticulate)

pyblp <- import('pyblp')

problem <- pyblp$Problem(
    product_formulations = tuple(
        pyblp$Formulation('1 + hpwt + air + mpd + space'),                      # Linear demand
        pyblp$Formulation('1 + prices + hpwt + air + mpd + space'),             # Nonlinear demand
        pyblp$Formulation('1 + log(hpwt) + air + log(mpg) + log(space) + trend')  # Supply
    ),
    agent_formulation = pyblp$Formulation('0 + I(1 / income)'),                 # Price interaction
    costs_type = 'log',                                                         # Log-linear costs
    product_data = read_csv(pyblp$data$BLP_PRODUCTS_LOCATION),
    agent_data = read_csv(pyblp$data$BLP_AGENTS_LOCATION)
)

results <- problem$solve(
     sigma = diag(c(3.612, 0, 4.628, 1.818, 1.050, 2.056)),  # Starting values for unobserved heterogeneity
     pi = rbind(0, -43.501, 0, 0, 0, 0),                     # Starting value for the term on price
     initial_update = TRUE,                                  # Update weight matrix at starting values
     costs_bounds = tuple(0.001, NULL),                      # Constrain marginal costs to be positive
     W_type = 'clustered',                                   # Cluster by automobile model
     se_type = 'clustered'
)

elasticities = results$compute_elasticities()
markups = results$compute_markups()

instrument_results = results$compute_optimal_instruments(method = 'approximate')
updated_problem = instrument_results$to_problem()
```

This R code demonstrates how to construct and solve the problem from Berry, Levinsohn, and Pakes (1995, 1999) in PyBLP. Python functions are called with the R package `reticulate`. Similar packages that allow for Python interoperability are available in many other languages. Names in the formulation objects correspond to variable names in the datasets, which are are packaged with PyBLP and in this example are loaded into memory with the R package `readr`. Most estimation outputs are stored as attributes of the problem results class. Post-estimation outputs such as elasticities and markups can be computed with class methods. Here, the "approximate" technique from Algorithm 2 is used to construct feasible optimal instruments. This gives an optimal instruments results object that is converted into an updated problem, which can be solved like any other. Estimation results are reported in Table 8.

The widespread availability of such packages for between-language interoperability means that it is often straightforward to incorporate functionality from other languages.[97]

Our configuration for the Berry, Levinsohn, and Pakes (1995) problem differs more substantially from the original article because parts of the original configuration are not included with the data.[98] We estimate the model once with the original article's sums of characteristics BLP instruments and importance sampling integration rule, and a second time with feasible optimal instruments and 10,000 scrambled Halton draws in each market. We report our results in Table 8. We obtain broadly similar parameter estimates. Estimates under Halton draws and the feasible approximation to the optimal instruments suggest somewhat less preference heterogeneity, which leads to slightly less elastic demand and larger markups than in the original specification.

□ **Knittel and Metaxoglou (2014).** With the Nevo (2000b) configuration and a demand-only version of the Berry, Levinsohn, and Pakes (1995) problem, we conduct a more extensive

[97] For more on interoperability, see Footnote 3.

[98] We obtain the data from the replication package for Andrews, Gentzkow, and Shapiro (2017) and obtain very similar results as this earlier replication. Following Berry, Levinsohn, and Pakes (1999), we replace the original article's $\log(y_i - p_j)$ term with its first-order linear approximation p_j/y_i. Otherwise, there are individuals for whom $p_j > y_i$, creating a host of problems.

TABLE 7 Nevo (2000b) Replication

		Published Estimates	Replication	Tighter Tolerance	Best Practices
Means	Price	−32.433	−32.404	−62.729	−27.489
		(7.743)	(7.729)	(14.803)	(4.383)
Standard Deviations	Price	1.848	1.851	3.313	2.910
		(1.075)	(1.070)	(1.340)	(0.669)
	Constant	0.377	0.376	0.558	0.196
		(0.129)	(0.129)	(0.163)	(0.085)
	Sugar	0.004	0.003	0.006	0.028
		(0.012)	(0.012)	(0.014)	(0.008)
	Mushy	0.081	0.080	0.093	0.324
		(0.205)	(0.204)	(0.185)	(0.110)
Interactions	Price × income	16.598	16.457	588.318	15.957
		(172.334)	(172.237)	(270.441)	(98.164)
	Price × income squared	−0.659	−0.655	−30.192	−1.282
		(8.955)	(8.951)	(14.101)	(5.119)
	Price × child	11.625	11.543	11.054	4.551
		(5.207)	(5.166)	(4.123)	(2.405)
	Constant × income	3.089	3.100	2.292	6.253
		(1.213)	(1.203)	(1.209)	(0.541)
	Constant × age	1.186	1.172	1.284	0.162
		(1.016)	(1.001)	(0.631)	(0.207)
	Sugar × income	−0.193	−0.193	−0.385	−0.289
		(0.005)	(0.045)	(0.121)	(0.037)
	Sugar × age	0.029	0.030	0.052	0.046
		(0.036)	(0.036)	(0.026)	(0.014)
	Mushy × income	1.468	1.462	0.748	0.998
		(0.697)	(0.693)	(0.802)	(0.303)
	Mushy × age	−1.514	−1.502	−1.353	−0.523
		(1.103)	(1.091)	(0.667)	(0.188)
Mean own-price elasticity			−3.700	−3.618	−3.685
Mean markup			0.360	0.364	0.363
GMM objective		6.60E-03	6.61E-03	2.02E-03	2.03E-04
GMM objective scaled by N		1.49E+01	1.49E+01	4.56E+00	4.59E-01

Note: This table reports replication results for the model of Nevo (2000b) described in Section 6. From left to right, we report estimates from the original article, our replication results, additional results for when we reduce the BFGS optimization algorithm's gradient-based L^∞ norm from `1E-4` to a slightly tighter `1E-5`, and a final set of results using best estimation practices: a tighter termination tolerance and the "approximate" version of the feasible optimal instruments. Standard errors are in parentheses.

replication exercise meant to mimic Knittel and Metaxoglou (2014). We solve these two problems with 50 random starting values and multiple optimization algorithms.[99]

We first replicate some of the difficulties encountered by Knittel and Metaxoglou (2014) by using very loose optimization tolerances.[100] Early termination of optimization algorithms creates dispersion across recovered parameter estimates, objective values, and implied elasticities across optimizers and starting values. However, tighter optimization tolerances suffice to eliminate all dispersion for the problem in Nevo (2000b).[101] In addition to tight tolerances, using quadrature instead of a small number of pseudo-Monte Carlo draws also eliminates all dispersion in the

[99] For a longer description of our replication exercise, please refer to the online appendix.

[100] Knittel and Metaxoglou (2014) use a tolerance of `1E-3` for changes in the parameter vector and the objective function. Because of the loose objective function tolerance in particular, the optimization routines often terminate too early. We replicate this behavior with loose L^∞ gradient- and parameter-based tolerances of `1E-1`.

[101] We use L^∞ gradient- and parameter-based tolerances of `1E-4`. In their online appendix, Knittel and Metaxoglou (2014) also report that when using a tighter tolerance and a gradient-based routine, they manage to eliminate essentially all dispersion for the problem in Nevo (2000b) but not Berry, Levinsohn, and Pakes (1995).

TABLE 8 Berry, Levinsohn, and Pakes (1995, 1999) Replication

		Published Estimates	Replication	Best Practices
Means	Constant	−7.061	−7.284	−6.679
		(0.941)	(2.807)	(1.304)
	HP/weight	2.883	3.460	2.774
		(2.019)	(1.415)	(0.833)
	Air	1.521	−0.999	0.572
		(0.891)	(2.101)	(0.349)
	MP$	−0.122	0.421	0.340
		(0.320)	(0.250)	(0.098)
	Size	3.460	4.178	3.920
		(0.610)	(0.658)	(0.322)
Standard deviations	Constant	3.612	2.025	2.962
		(1.485)	(6.065)	(1.637)
	HP/weight	4.628	6.101	1.388
		(1.885)	(2.200)	(2.107)
	Air	1.818	3.956	1.424
		(1.695)	(2.110)	(0.435)
	MP$	1.050	0.254	0.072
		(0.272)	(0.549)	(1.002)
	Size	2.056	1.908	0.231
		(0.585)	(1.108)	(3.837)
Term on price	$\ln(y - p)$	43.501	44.842	45.898
		(6.427)	(9.216)	(11.748)
Supply-side terms	Constant	0.952	2.760	2.785
		(0.194)	(0.116)	(0.104)
	ln(HP/weight)	0.477	0.897	0.731
		(0.056)	(0.072)	(0.071)
	Air	0.619	0.423	0.528
		(0.038)	(0.087)	(0.040)
	ln(MPG)	−0.415	−0.525	−0.651
		(0.055)	(0.073)	(0.071)
	ln(size)	−0.046	−0.261	−0.472
		(0.081)	(0.210)	(0.125)
	Trend	0.019	0.027	0.018
		(0.002)	(0.003)	(0.002)
Mean own-price elasticity			−3.928	−3.461
Mean markup			0.316	0.346
GMM objective			2.24E-01	1.06E-01
GMM objective scaled by N			4.97E+02	2.36E+02

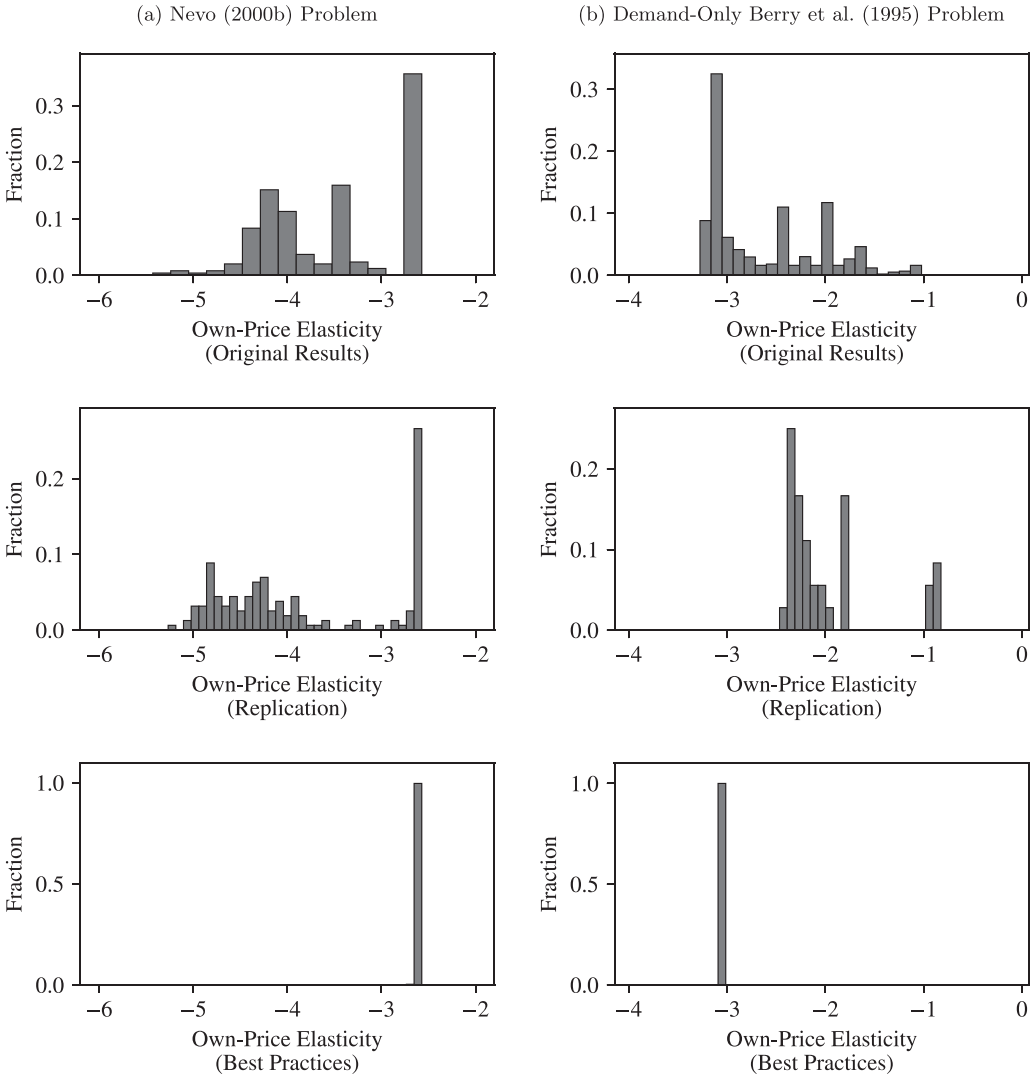
Note: This table reports replication results for the model of Berry, Levinsohn, and Pakes (1995, 1999) described in Section 6. From left to right, we report estimates from the original article, our replication results, and results using best estimation practices: 10,000 scrambled Halton draws in each market and the "approximate" version of the feasible optimal instruments. Standard errors are in parentheses and are clustered by automobile model.

demand-only version of the Berry, Levinsohn, and Pakes (1995) problem. This is consistent with Brunner et al. (2017), who find that simulation error contributes substantial instability to this particular configuration.

In other words, our best practices eliminate much of the difficulties encountered by Knittel and Metaxoglou (2014). In line with our Monte Carlo experiments, when properly configured, the choice of optimization routine (open-source or otherwise) does not seem to be particularly important for the nested fixed point algorithm. To demonstrate this result graphically, in Figure 7 we present histograms of own prices elasticities obtained from our replication exercise below the corresponding figures from Knittel and Metaxoglou (2014). The "Best Practices" histograms exhibit essentially no dispersion across optimization routines and starting values.

FIGURE 7

KNITTEL AND METAXOGLOU (2014) HISTOGRAMS FOR MEDIAN PRODUCT OWN-PRICE ELASTICITIES [Color figure can be viewed at wileyonlinelibrary.com]



All figures report the own price elasticity for the median product. The three figures on the left are for the problem in Nevo (2000b). Figures on the right are for a demand-only version of the problem in Berry, Levinsohn, and Pakes (1995) described in Knittel and Metaxoglou (2014). The top two figures are reproduced as fair use from Knittel and Metaxoglou (2014). The bottom four are produced by PyBLP where each observation is one trial from 50 starting values and seven optimization algorithm configurations supported by Knitro and SciPy. The middle figures replicate some of the difficulties in Knittel and Metaxoglou (2014) by using loose optimization tolerances. The bottom two figures eliminate these difficulties with best estimation practices. It suffices to use tight optimization tolerances for the problem in Nevo (2000b). Difficulties for the demand-only version of the problem in Berry, Levinsohn, and Pakes (1995) can be eliminated by additionally using a Gauss-Hermite product rule that exactly integrates polynomials of degree 11 of less instead of 50 pseudo-Monte Carlo draws in each market. For additional details, please consult the online appendix.

7. Conclusion

■ Our goal has been to review recent methodological developments related to the BLP problem, and to collect and evaluate them not only in a single article, but also in a single software package, PyBLP. We have provided a list of best practices with numerical evidence to support our recommendations, and we have implemented them as defaults in PyBLP. Our hope is that these practices (and estimation of BLP models in general) can now be made available to more researchers, and provide a common platform that should facilitate replication. For researchers who wish to implement and estimate similar models that are not among the wide range of BLP-type models supported by PyBLP, we hope that this article and our well-documented code will serve as a good starting point.

In addition, we present some methodological results that we believe to be novel. We show how with a slight reformulation of the nested fixed point problem, it is possible to include high dimensional fixed effects in models with simultaneous supply and demand. We also provide a somewhat different expression for optimal instruments than the prior literature (Reynaert and Verboven, 2014), which makes clear the over-identifying restrictions implied by the supply side. Also novel, we find that optimal instruments when combined with a correctly specified supply side are extremely valuable. Consistent with prior work, we find the gains to optimal instruments to be substantial such that they should nearly always be employed. Thankfully, we have made this process extremely straightforward in PyBLP.

Somewhat reassuringly, we find that under our best practices, including correctly specified supply restrictions and approximations to the optimal instruments, finite sample performance of the BLP estimator appears to be quite good and perhaps better than previously believed.

Appendix A: Derivations

□ **Concentrating out linear parameters.** Our objective is to concentrate out $\hat{\beta}(\theta_2)$ and $\hat{\gamma}(\theta_2)$. Define Y_{jt}^D, Y_{jt}^S, X_{jt}^D, and X_{jt}^S as follows:

$$\begin{aligned} Y_{jt}^D &\equiv \hat{\delta}_{jt}(\theta_2) + \alpha p_{jt} = [x_{jt}, v_{jt}]\beta + \xi_{jt} \equiv X_{jt}^D\beta + \xi_{jt}, \\ Y_{jt}^S &\equiv p_{jt} - \hat{\eta}_{jt}(\theta_2) = [x_{jt}, w_{jt}]\gamma + \omega_{jt} \equiv X_{jt}^S\gamma + \omega_{jt}. \end{aligned} \tag{A1}$$

Stacking the system across observations yields:[102]

$$\underbrace{\begin{bmatrix} Y_D \\ Y_S \end{bmatrix}}_{2N\times 1} = \underbrace{\begin{bmatrix} X_D & 0 \\ 0 & X_S \end{bmatrix}}_{2N\times(K_1+K_3)} \underbrace{\begin{bmatrix} \beta \\ \gamma \end{bmatrix}}_{(K_1+K_3)\times 1} + \underbrace{\begin{bmatrix} \xi \\ \omega \end{bmatrix}}_{2N\times 1}. \tag{A2}$$

Adding the $N \times M_D$ instruments for demand, Z_D, and the $N \times M_S$ instruments for supply, Z_S, yields $M = M_D + M_S$ moment restrictions

$$E\begin{bmatrix} Z_{jt}^{D\prime}(Y_{jt}^D - X_{jt}^D\beta) \\ Z_{jt}^{S\prime}(Y_{jt}^S - X_{jt}^S\gamma) \end{bmatrix} = 0, \tag{A3}$$

which have sample analogues

$$\underbrace{\frac{1}{N}\underbrace{\begin{bmatrix} Z_D' & 0 \\ 0 & Z_S' \end{bmatrix}}_{M\times 2N}\underbrace{\begin{bmatrix} Y_D \\ Y_S \end{bmatrix}}_{2N\times 1}}_{\widetilde{Y}} - \underbrace{\frac{1}{N}\underbrace{\begin{bmatrix} Z_D'X_D & 0 \\ 0 & Z_S'X_S \end{bmatrix}}_{M\times(K_1+K_3)}\underbrace{\begin{bmatrix} \beta \\ \gamma \end{bmatrix}}_{(K_1+K_3)\times 1}}_{\widetilde{X}} \tag{A4}$$

Now we can simply perform a GMM regression of $\widetilde{Y}$ on $\widetilde{X}$ with the same $M \times M$ weighting matrix W used in the overall problem:

$$\begin{bmatrix} \hat{\beta}(\theta_2) \\ \hat{\gamma}(\theta_2) \end{bmatrix} = (\widetilde{X}'W\widetilde{X})^{-1}\widetilde{X}'W\widetilde{Y}. \tag{A5}$$

□ **Analytic derivative calculations.** *This derivation appears to be novel to the literature for the case of simultaneous estimation of supply and demand.*

[102] Note that cannot perform independent regressions unless we are willing to assume that $\text{Cov}(\xi_{jt}, \omega_{jt}) = 0$.

The gradient of the GMM objective function is

$$\nabla q(\theta_2) = 2G(\theta_2)'Wg(\theta_2).$$

The challenging piece here is the Jacobian of the GMM objective:

$$\underbrace{G(\theta_2)}_{M\times K_2} = \frac{1}{N}\underbrace{\begin{bmatrix} Z_D' & 0 \\ 0 & Z_S' \end{bmatrix}}_{M\times 2N}\underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial \theta_2} \\ \frac{\partial \omega}{\partial \theta_2} \end{bmatrix}}_{2N\times K_2}.$$

We can write:[103]

$$\begin{bmatrix} \frac{\partial \xi}{\partial \theta_2} \\ \frac{\partial \omega}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \delta}{\partial \theta_2} \\ -f_{MC}'(\cdot)\frac{\partial \eta}{\partial \theta_2} \end{bmatrix}. \tag{A6}$$

The $f_{MC}'(\cdot)$ in (A6) comes from (7) where $f_{MC}(\cdot)$ is typically linear with $f_{MC}'(c_{jt}) = 1$, or logarithmic with $f_{MC}'(c_{jt}) = 1/c_{jt}$.

For the demand moments, after invoking the implicit function theorem, this has a convenient block structure that can be separated market by market t:[104]

$$\underbrace{\frac{\partial \delta_t}{\partial \theta_2}(\theta_2)}_{J_t\times K_2} = -\left(\underbrace{\left[\frac{\partial \mathbf{s}_t}{\partial \delta_t}(\theta_2)\right]}_{J_t\times J_t}\right)^{-1}\underbrace{\left[\frac{\partial \mathbf{s}_t}{\partial \theta_2}(\theta_2)\right]}_{J_t\times K_2}.$$

Differentiating the supply moments is challenging because demand derivatives $\frac{\partial s_{jt}}{\partial p_{kt}}(\boldsymbol{\xi}_t(\theta_2), \theta_2)$ in the matrix $\Delta_t(\boldsymbol{\xi}_t(\theta_2), \theta_2)$ depend on both $\boldsymbol{\xi}_t(\theta_2)$ and θ_2 directly. To avoid excessive tensor product notation, consider the derivative with respect to an element within θ_2 labeled θ_ℓ:[105]

$$\begin{aligned}\underbrace{\frac{\partial \boldsymbol{\eta}_t}{\partial \theta_\ell}}_{J_t\times 1} &= -\Delta_t^{-1}\frac{\partial \Delta_t}{\partial \theta_\ell}\Delta_t^{-1}\boldsymbol{s}_t - \Delta_t^{-1}\frac{\partial \Delta_t}{\partial \boldsymbol{\xi}_t}\frac{\partial \boldsymbol{\xi}_t}{\partial \theta_\ell}\Delta_t^{-1}\boldsymbol{s}_t + \Delta_t^{-1}\underbrace{\frac{\partial \boldsymbol{s}_t}{\partial \theta_\ell}}_{0} \\ &= -\underbrace{\Delta_t^{-1}}_{(J_t\times J_t)}\underbrace{\frac{\partial \Delta_t}{\partial \theta_\ell}}_{(J_t\times J_t)}\underbrace{\boldsymbol{\eta}_t}_{(J_t\times 1)} - \underbrace{\Delta_t^{-1}}_{(J_t\times J_t)}\underbrace{\frac{\partial \Delta_t}{\partial \boldsymbol{\xi}_t}}_{(J_t\times J_t\times J_t)}\underbrace{\frac{\partial \boldsymbol{\xi}_t}{\partial \theta_\ell}}_{(J_t\times 1)}\underbrace{\boldsymbol{\eta}_t}_{(J_t\times 1)}.\end{aligned}$$

This is expression is complicated because the supply-side structural error term $\boldsymbol{\omega}_t$ and the markup $\boldsymbol{\eta}_t$ depend both directly on θ_2 and indirectly on θ_2 through $\boldsymbol{\xi}_t$.

□ **Quantity dependent marginal costs.** The original Berry, Levinsohn, and Pakes (1995) paper incorporates quantity dependent marginal costs, which allow for increasing or decreasing returns to scale:

$$\log(p_{jt} - \eta_{jt}(\theta_2)) = \log c_{jt} = [x_{jt}, w_{jt}]\gamma + \gamma_q \log q_{jt} + \omega_{jt}.$$

The obvious problem is that $\log q_{jt}$ is endogenous in that it depends on $(\boldsymbol{\xi}_t, \boldsymbol{\omega}_t)$. A second question is whether the firm takes into account the fact that selling an additional unit *changes the marginal cost*, which would imply an additional term in the first-order condition:

$$s_{jt}(\boldsymbol{p}_t) + \sum_{k\in J_{ft}}\frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t)\cdot(p_{kt} - c_{kt}) - s_{kt}(\boldsymbol{p}_t)\cdot\frac{\partial c_{kt}}{\partial q_{kt}}\cdot\frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t)\cdot M_t = 0,$$

$$s_{jt}(\boldsymbol{p}_t) + \sum_{k\in J_{ft}}\frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t)\cdot\left(p_{kt} - c_{kt} - \underbrace{M_t\cdot s_{kt}(\boldsymbol{p}_t)}_{q_{kt}(\boldsymbol{p}_t)}\cdot\frac{\partial c_{kt}}{\partial q_{kt}}\right) = 0,$$

$$s_{jt}(\boldsymbol{p}_t) + \sum_{k\in J_{ft}}\frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t)\cdot\left(p_{kt} - c_{kt}\cdot(1+\gamma_q)\right) = 0.$$

[103] During optimization, this equality does not hold because we concentrate out the linear parameters. Abbreviating (A6) as $\frac{\partial L}{\partial \theta_2} = \frac{\partial R}{\partial \theta_2}$ and using notation from Appendix A.1, $\frac{\partial L}{\partial \theta_2} = (I - X(\tilde{X}'W\tilde{X})^{-1}\tilde{X}'WZ')\frac{\partial R}{\partial \theta_2} \neq \frac{\partial R}{\partial \theta_2}$. We thank Luis Armona and Daniel Stackman for pointing this out. Fortunately, one can use orthogonality between L and the projection matrix to show $\nabla q \propto \frac{\partial L'}{\partial \theta_2}ZWZ'L = \frac{\partial R'}{\partial \theta_2}ZWZ'L$, i.e. it is fine to use (A6) when computing the gradient.

[104] The matrix inverse of $\frac{\partial s_t}{\partial \delta_t}(\theta_2)$ is guaranteed by the diagonal dominance of system of equations with respect to δ_{jt}. As long as the outside good has a positive share, we have that for each j, $|\frac{\partial s_{jt}}{\partial \delta_{jt}}| > \sum_{k\neq j}|\frac{\partial s_{kt}}{\partial \delta_{jt}}|$. In practice, as shares become small, there may still be numerical issues.

[105] In the markup η_{jt}, the $\boldsymbol{s}_t$ is data and thus does not depend on parameters.

If $\log q_{jt}$ enters the (log) marginal cost function, the implied marginal costs are all increased proportionally by $(1+\gamma_q)$. Otherwise the first order condition depends both on the functional form $f_{MC}(\cdot)$ and how q_{jt} enters the equation:

$$s_{jt}(\boldsymbol{p}_t) + \sum_{k \in J_{ft}} \frac{\partial s_{kt}}{\partial p_{jt}}(\boldsymbol{p}_t) \cdot \left(p_{kt} - c_{kt} - q_{kt} \frac{\partial c_{kt}}{\partial q_{kt}} \right) = 0.$$

This does not appear to be how the existing literature such as Berry, Levinsohn, and Pakes (1995, 1999) address quantity dependent marginal costs. Instead, the first-order condition treats the marginal cost as if it were constant (i.e., assuming $\frac{\partial c_{kt}}{\partial q_{kt}} = 0$ above), but then when the parameters γ are recovered allows for an extra term on q_{jt} or $\log q_{jt}$. Thus the firm treats marginal costs as if they were constant when setting prices, but the marginal costs are still quantity dependent:[106]

$$f_{MC}(p_{jt} - \eta_{jt}(\theta_2)) = [x_{jt}, w_{jt}]\gamma + \gamma_q \log q_{jt} + \omega_{jt}.$$

Also note that q_{jt} cannot be included in the set of instruments for supply, Z_t^S, and as an endogenous variable, it increases the number of required instruments. The good news is that the conventional quantity shifters (BLP instruments) should be relevant here.

Appendix B: Additional tables and figures

TABLE B1 Fixed Point Tricks

Problem	Median s_{0t}	Overflow Safe	Type	Initial $\boldsymbol{\delta}_t$	Mean Milliseconds	Mean Evaluations	Percent Converged
Simple simulation	0.91	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	2.56	15.94	100.00%
Simple simulation	0.91	No	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	1.66	15.94	100.00%
Simple simulation	0.91	Yes	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	2.57	15.94	100.00%
Simple simulation	0.91	No	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	1.66	15.94	100.00%
Simple simulation	0.91	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^{n-1}$	1.78	13.93	100.00%
Complex simulation	0.91	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	3.32	16.14	100.00%
Complex simulation	0.91	No	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	2.32	16.15	100.00%
Complex simulation	0.91	Yes	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	3.33	16.16	100.00%
Complex simulation	0.91	No	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	2.34	16.15	100.00%
Complex simulation	0.91	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^{n-1}$	2.74	14.74	100.00%
RCNL simulation	0.92	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	8.45	33.49	100.00%
RCNL simulation	0.92	No	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	6.06	33.49	100.00%
RCNL simulation	0.92	Yes	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	8.24	33.49	100.00%
RCNL simulation	0.92	No	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	6.09	33.50	100.00%
RCNL simulation	0.92	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^{n-1}$	6.37	27.32	100.00%
Nevo example	0.54	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	4.05	24.06	100.00%
Nevo example	0.54	No	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	2.67	24.06	100.00%
Nevo example	0.54	Yes	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	4.07	24.06	100.00%
Nevo example	0.54	No	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	2.69	24.06	100.00%
Nevo example	0.54	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^{n-1}$	3.10	18.27	100.00%
BLP example	0.89	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	31.54	40.61	100.00%
BLP example	0.89	No	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^0$	26.97	40.69	100.00%
BLP example	0.89	Yes	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	31.64	40.61	100.00%
BLP example	0.89	No	$\exp(\boldsymbol{\delta}_t)$	$\boldsymbol{\delta}_t^0$	26.57	40.69	100.00%
BLP example	0.89	Yes	$\boldsymbol{\delta}_t$	$\boldsymbol{\delta}_t^{n-1}$	29.08	37.95	100.00%

Note: This table documents the impact of common tricks used in the literature on solving the nested fixed point. Reported values are medians across 100 different simulations and 10 identical runs of the two example problems. We report the number of milliseconds and contraction evaluations needed to solve the nested fixed point, averaged across all markets and objective evaluations of one GMM step. We also report convergence rates: the percent of times no numerical errors were encountered a limit of 1000 iterations was not reached. Overflow safe results are those that use the log-sum-exp (LSE) function. The two fixed point types are the standard linear contraction over δ_{jt} and the exponentiated version over $\exp(\delta_{jt})$. An initial $\boldsymbol{\delta}_t^0$ means that the contraction always starts at the solution to the logit model, whereas $\boldsymbol{\delta}_t^{n-1}$ is the "hot-start" version where starting values are those that solved the fixed point for the previous guess of θ_2. We use the SQUAREM algorithm with an absolute L^∞ norm tolerance of `1E-14`. Simulations are configured as in Section 5. The example problems from Nevo (2000b) and Berry, Levinsohn, and Pakes (1995, 1999) are the replications described in Section 6.

[106] Allowing for extra term in the first-order condition also likely violates the conditions necessary for uniqueness of the pricing equilibrium, particularly if γ_q is negative and there are increasing returns to scale.

TABLE B2 Impact of Alternative Integration Methods on Parameter Estimates

				True Value				Median Bias				Median Absolute Error				
Simulation	Supply	Integration	I_t	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	No	Monte Carlo	100	1.0	−1	3			0.233	−0.691			0.298	0.691		
Simple	No	Monte Carlo	1000	3.1	−1	3			0.198	−0.132			0.251	0.191		
Simple	No	MLHS	1000	3.2	−1	3			0.188	−0.051			0.241	0.167		
Simple	No	Halton	1000	3.2	−1	3			0.186	−0.050			0.241	0.165		
Simple	No	Importance	1000	21.6	−1	3			0.181	0.018			0.242	0.169		
Simple	No	Product rule	9^1	0.8	−1	3			0.189	−0.039			0.245	0.169		
Simple	Yes	Monte Carlo	100	2.7	−1	3			0.113	−0.705			0.243	0.705		
Simple	Yes	Monte Carlo	1000	8.8	−1	3			0.021	−0.102			0.180	0.182		
Simple	Yes	MLHS	1000	8.8	−1	3			0.020	−0.015			0.172	0.162		
Simple	Yes	Halton	1000	9.2	−1	3			0.020	−0.015			0.170	0.162		
Simple	Yes	Importance	1000	27.2	−1	3			−0.001	0.065			0.176	0.181		
Simple	Yes	Product rule	9^1	2.2	−1	3			0.015	0.003			0.172	0.172		
Complex	No	Monte Carlo	100	1.9	−1	3	0.2		0.304	−0.776	−0.091		0.317	0.778	0.110	
Complex	No	Monte Carlo	1000	5.3	−1	3	0.2		0.193	−0.190	−0.012		0.253	0.223	0.098	
Complex	No	MLHS	1000	5.1	−1	3	0.2		0.191	−0.103	−0.018		0.254	0.182	0.102	
Complex	No	Halton	1000	5.7	−1	3	0.2		0.141	−0.120	0.033		0.241	0.190	0.121	
Complex	No	Importance	1000	28.3	−1	3	0.2		0.180	−0.017	−0.028		0.254	0.173	0.108	
Complex	No	Product rule	9^2	1.6	−1	3	0.2		0.172	−0.088	−0.011		0.250	0.177	0.169	
Complex	Yes	Monte Carlo	100	5.2	−1	3	0.2		0.114	−0.702	−0.137		0.250	0.713	0.141	
Complex	Yes	Monte Carlo	1000	15.0	−1	3	0.2		0.029	−0.126	−0.040		0.194	0.204	0.106	
Complex	Yes	MLHS	1000	15.2	−1	3	0.2		0.015	−0.051	−0.050		0.195	0.171	0.146	
Complex	Yes	Halton	1000	17.1	−1	3	0.2		−0.025	−0.053	0.024		0.194	0.162	0.127	
Complex	Yes	Importance	1000	38.6	−1	3	0.2		−0.050	0.071	−0.020		0.208	0.190	0.112	
Complex	Yes	Product rule	9^2	4.7	−1	3	0.2		−0.020	−0.029	0.004		0.195	0.171	0.169	

(*Continued*)

TABLE B2 (Continued)

				True Value				Median Bias				Median Absolute Error				
Simulation	Supply	Integration	I_t	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
RCNL	No	Monte Carlo	100	5.7	−1	3		0.5	0.236	−0.645		0.046	0.268	0.645		0.051
RCNL	No	Monte Carlo	1000	18.9	−1	3		0.5	0.182	−0.131		0.001	0.219	0.182		0.022
RCNL	No	MLHS	1000	19.0	−1	3		0.5	0.177	−0.026		−0.007	0.216	0.160		0.021
RCNL	No	Halton	1000	19.9	−1	3		0.5	0.174	−0.024		−0.008	0.214	0.155		0.021
RCNL	No	Importance	1000	40.2	−1	3		0.5	0.086	0.854		−0.110	0.241	0.854		0.110
RCNL	No	Product rule	9[1]	4.3	−1	3		0.5	0.176	−0.017		−0.007	0.214	0.153		0.021
RCNL	Yes	Monte Carlo	100	12.5	−1	3		0.5	0.072	−0.573		0.046	0.124	0.573		0.048
RCNL	Yes	Monte Carlo	1000	45.8	−1	3		0.5	0.020	−0.096		0.007	0.112	0.156		0.019
RCNL	Yes	MLHS	1000	45.8	−1	3		0.5	0.008	−0.005		0.000	0.109	0.138		0.017
RCNL	Yes	Halton	1000	47.6	−1	3		0.5	0.010	−0.008		0.000	0.109	0.139		0.017
RCNL	Yes	Importance	1000	66.0	−1	3		0.5	−0.071	0.862		−0.099	0.130	0.863		0.099
RCNL	Yes	Product rule	9[1]	9.3	−1	3		0.5	0.002	−0.001		0.000	0.109	0.139		0.018

Note: This table documents bias and variance of parameter estimates over 1000 simulated datasets for different numerical integration methods and numbers of integration nodes I_t. Importance sampling is based on Halton draws. For descriptions of the integration rules, please refer to Figure 1. For all problems, we use the "approximate" version of the feasible optimal instruments.

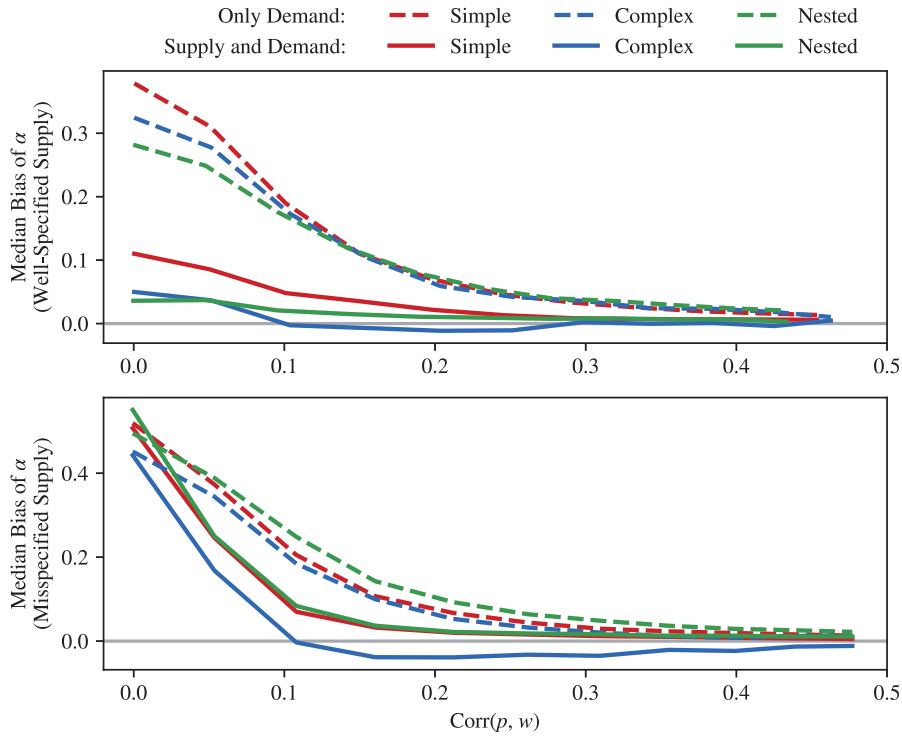
TABLE B3 Form of Feasible Optimal Instruments

			True Value				Median Bias				Median Absolute Error				
Simulation	Supply	Approach	Seconds	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ	α	σ_x	σ_p	ρ
Simple	No	Approximate	0.8	−1	3			0.189	−0.039			0.245	0.169		
Simple	No	Asymptotic	4.8	−1	3			0.189	−0.038			0.245	0.169		
Simple	No	Empirical	4.8	−1	3			0.188	−0.035			0.245	0.168		
Simple	Yes	Approximate	2.2	−1	3			0.015	0.003			0.172	0.172		
Simple	Yes	Asymptotic	18.1	−1	3			0.021	0.003			0.192	0.172		
Simple	Yes	Empirical	18.2	−1	3			0.026	0.007			0.181	0.171		
Complex	No	Approximate	1.6	−1	3	0.2		0.172	−0.088	−0.011		0.250	0.177	0.169	
Complex	No	Asymptotic	6.5	−1	3	0.2		0.177	−0.084	−0.008		0.251	0.175	0.165	
Complex	No	Empirical	6.5	−1	3	0.2		0.171	−0.085	−0.012		0.246	0.175	0.168	
Complex	Yes	Approximate	4.7	−1	3	0.2		−0.020	−0.029	0.004		0.195	0.171	0.169	
Complex	Yes	Asymptotic	29.7	−1	3	0.2		0.001	−0.085	−0.029		0.210	0.209	0.168	
Complex	Yes	Empirical	29.5	−1	3	0.2		−0.008	−0.074	−0.016		0.211	0.199	0.168	
RCNL	No	Approximate	4.3	−1	3		0.5	0.176	−0.017		−0.007	0.214	0.153		0.021
RCNL	No	Asymptotic	9.4	−1	3		0.5	0.175	−0.022		−0.007	0.214	0.153		0.021
RCNL	No	Empirical	9.6	−1	3		0.5	0.173	−0.020		−0.007	0.215	0.151		0.021
RCNL	Yes	Approximate	9.3	−1	3		0.5	0.002	−0.001		0.000	0.109	0.139		0.018
RCNL	Yes	Asymptotic	46.4	−1	3		0.5	0.010	0.005		−0.000	0.113	0.142		0.018
RCNL	Yes	Empirical	46.2	−1	3		0.5	0.011	−0.004		−0.000	0.113	0.145		0.017

Note: This table documents bias and variance of parameter estimates over 1000 simulated datasets for different forms of optimal instruments. The "approximate" approach replaces the structural errors with their unconditional expectations of zero. For the "asymptotic" approach we take 100 draws from the estimated normal distribution for $(\xi_{jt}, \omega_{jt}) \sim N(0, \hat{\Omega})$. For the "empirical" approach we sample with replacement from the joint distribution of $(\hat{\xi}_{jt}, \hat{\omega}_{jt})$. For all problems, we use a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

FIGURE B1

IMPACT OF SUPPLY-SIDE MOMENTS ON OPTIMAL IV FOR DEMAND-ONLY PROBLEMS [Color figure can be viewed at wileyonlinelibrary.com]



Each plot documents how bias of the linear parameter on price, α, decreases with the strength of the cost shifter w_{jt}. Reported bias values are medians across 1000 different simulations. Unlike in Figure 2, once feasible optimal instruments have been constructed, only demand moments are used during estimation. Here, dashed lines mean that instruments were constructed only with demand moments, and solid lines mean that supply moments were used as well when constructing instruments. In the bottom plot we simulate data according to perfect competition (i.e., prices are set equal to marginal costs instead of those that satisfy Bertrand-Nash first order conditions), but continue to construct feasible optimal instruments under the assumption of imperfect competition. For all problems, we use the "approximate" version of the feasible optimal instruments and a Gauss-Hermite product rule that exactly integrates polynomials of degree 17 or less.

References

Ai, C. and Chen, X. "Efficient Estimation of Models with Conditional Moment Restrictions Containing Unknown Functions." *Econometrica*, Vol. 71 (2003), pp. 1795–1843. https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00470.

Allaire, J., Ushey, K., Tang, Y. and Eddelbuettel, D. Reticulate: R Interface to Python. https://github.com/rstudio/reticulate, Accessed March 01, 2020.

Amemiya, T. "The Maximum Likelihood and the Nonlinear Three-Stage Least Squares Estimator in the General Nonlinear Simultaneous Equation Model." *Econometrica*, Vol. 45 (1977), pp. 955–968. http://www.jstor.org/stable/1912684.

Andrews, I., Gentzkow, M. and Shapiro, J.M. "Measuring the Sensitivity of Parameter Estimates to Estimation Moments." *Quarterly Journal of Economics*, Vol. 132 (2017), pp. 1553–1592.

Armstrong, T. "Large Market Asymptotics for Differentiated Product Demand Estimators with Economic Models of Supply." *Econometrica*, Vol. 84 (2016), pp. 1961–1980.

Backus, M., Conlon, C. and Sinkinson, M. "Common Ownership and Competition in the Ready-To-Eat Cereal Industry." Working Paper, 2020.

Bayer, P., Ferreira, F. and McMillan, R. "A Unified Framework for Measuring Preferences for Schools and Neighborhoods." *Journal of Political Economy*, Vol. 115 (2007), pp. 588–638. https://doi.org/10.1086/522381.

Berry, S. "Estimating Discrete-Choice Models of Product Differentiation." *RAND Journal of Economics*, Vol. 25 (1994), pp. 242–261.

Berry, S., Levinsohn, J. and Pakes, A. "Automobile Prices in Market Equilibrium." *Econometrica*, Vol. 63 (1995), pp. 841–890.

———. "Voluntary Export Restraints on Automobiles: Evaluating a Trade Policy." *American Economic Review*, Vol. 89 (1999), pp. 400–430. http://www.aeaweb.org/articles?id=10.1257/aer.89.3.400.

———. "Differentiated Products Demand Systems from a Combination of Micro and Macro Data: The New Car Market." *Journal of Political Economy*, Vol. 112 (2004a), pp. 68–105.

Berry, S., Linton, O.B. and Pakes, A. "Limit Theorems for Estimating the Parameters of Differentiated Product Demand Systems." *Review of Economic Studies*, Vol. 71 (2004b), pp. 613–654. http://ideas.repec.org/a/bla/restud/v71y2004ip613-654.html.

Berry, S. and Pakes, A. "The Pure Characteristics Demand Model." *International Economic Review*, Vol. 48 (2007), pp. 1193–1125.

Berry, S.T. and Haile, P.A. "Identification in Differentiated Products Markets Using Market Level Data." *Econometrica*, Vol. 82 (2014), pp. 1749–1797. https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA9027.

Bhat, C.R. "Quasi-Random Maximum Simulated Likelihood Estimation of the Mixed Multinomial Logit Model." *Transportation Research Part B: Methodological*, Vol. 35 (2001), pp. 677–693. http://www.sciencedirect.com/science/article/pii/S019126150000014X.

Bonnet, C. and Dubois, P. "Inference on Vertical Contracts Between Manufacturers and Retailers Allowing for Nonlinear Pricing and Resale Price Maintenance." *The RAND Journal of Economics*, Vol. 41 (2010), pp. 139–164. http://www.jstor.org/stable/40649274.

Boyd, J. and Mellman, R.E. "The Effect of Fuel Economy Standards on the U.S. Automotive Market: An Hedonic Demand Analysis." *Transportation Research Part A: General*, Vol. 14 (1980), pp. 367–378. http://www.sciencedirect.com/science/article/pii/0191260780900552.

Brenkers, R. and Verboven, F. "Liberalizing a Distribution System: The European Car Market." *Journal of the European Economic Association*, Vol. 4 (2006), pp. 216–251. https://onlinelibrary.wiley.com/doi/abs/10.1162/jeea.2006.4.1.216.

Bresnahan, T.F. "The Oligopoly Solution Concept is Identified." *Economics Letters*, Vol. 10 (1982), pp. 87–92. http://www.sciencedirect.com/science/article/pii/0165176582901215.

Brunner, D. Numerical Integration in Random Coefficient Models of Demand. Master's thesis, Heinrich-Heine-Universtat Dusseldorf, 2017.

Brunner, D., Heiss, F., Romahn, A. and Weiser, C. "Reliable Estimation of Random Coefficient Logit Demand Models." Technical report, University of Düsseldorf, 2017.

Caplin, A. and Nalebuff, B. "Aggregation and Imperfect Competition: On the Existence of Equilibrium." *Econometrica*, Vol. 59 (1991), pp. 25–59. http://www.jstor.org/stable/2938239.

Cardell, N. and Dunbar, F.C. "Measuring the Societal Impacts of Automobile Downsizing." *Transportation Research Part A: General*, Vol. 14 (1980), pp. 423–434. http://www.sciencedirect.com/science/article/pii/0191260780900606.

Chamberlain, G. "Asymptotic Efficiency in Estimation with Conditional Moment Restrictions." *Journal of Econometrics*, Vol. 34 (1987), pp. 305–334.

Conlon, C. "The MPEC Approach to Empirical Likelihood Estimation of Demand." Working Paper, 2017.

Conlon, C. and Rao, N. "The Price of Liquor is Too Damn High: The Effects of Post and Hold Pricing." Working Paper, 2017.

Correia, S. "Linear Models with High-Dimensional Fixed Effects: An Efficient and Feasible Estimator." Working Paper, 2016.

Donald, S.G., Imbens, G.W. and Newey, W.K. "Choosing Instrumental Variables in Conditional Moment Restriction Models." *Journal of Econometrics*, Vol. 152 (2009), pp. 28–36. http://www.sciencedirect.com/science/article/pii/S0304407609000566.

Dubé, J.P.H., Fox, J.T. and Su, C.L. "Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation." *Econometrica*, Vol. 80 (2012), pp. 2231–2267.

Fan, Y. "Ownership Consolidation and Product Characteristics: A Study of the US Daily Newspaper Market." *American Economic Review*, Vol. 103 (2013), pp. 1598–1628. http://www.aeaweb.org/articles?id=10.1257/aer.103.5.1598.

Fong, D.C.L. and Saunders, M. "LSMR: An Iterative Algorithm for Sparse Least-Squares Problems." *SIAM Journal on Scientific Computing*, Vol. 33 (2011), pp. 2950–2971.

Fox, J.T., il Kim, K., Ryan, S.P. and Bajari, P. "The Random Coefficients Logit Model is Identified." *Journal of Econometrics*, Vol. 166 (2012), pp. 204–212. http://www.sciencedirect.com/science/article/pii/S0304407611001655.

Freyberger, J. "Asymptotic Theory for Differentiated Products Demand Models with Many Markets." *Journal of Econometrics*, Vol. 185 (2015), pp. 162–181. http://www.sciencedirect.com/science/article/pii/S0304407614002474.

Gallego, G., Huh, W.T., Kang, W. and Phillips, R. "Price Competition with the Attraction Demand Model: Existence of Unique Equilibrium and Its Stability." *Manufacturing & Service Operations Management*, Vol. 8 (2006), pp. 359–375. https://pubsonline.informs.org/doi/abs/10.1287/msom.1060.0115.

Gandhi, A. and Houde, J. "Measuring Substitution Patterns in Differentiated Products Industries." Working Paper, 2019.
Gortmaker, J. and Tarascina, A. *PyHDFE*. https://github.com/jeffgortmaker/pyhdfe. Accessed on March 1 2020.
Grigolon, L. and Verboven, F. "Nested Logit or Random Coefficients Logit? A Comparison of Alternative Discrete Choice Models of Product Differentiation." *The Review of Economics and Statistics*, Vol. 96 (2014), pp. 916–935. https://doi.org/10.1162/REST_a_00420.
Guimarães, P. and Portugal, P. "A Simple Feasible Procedure to Fit Models with High-Dimensional Fixed Effects." *Stata Journal*, Vol. 10 (2010), pp. 628–649. https://ideas.repec.org/a/tsj/stataj/v10y2010i4p628-649.html.
Halton, J.H. "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals." *Numerische Mathematik*, Vol. 2 (1960), pp. 84–90. https://doi.org/10.1007/BF01386213.
Hansen, L.P. "Large Sample Propertes of Generalized Method of Moments Estimators." *Econometrica*, Vol. 50 (1982), pp. 1029–1054.
Heiss, F. "The Panel Probit Model: Adaptive Integration on Sparse Grids." *Advances in Econometrics*, Vol. 26 (2010), pp. 41–64.
Heiss, F. and Winschel, V. "Likelihood Approximation by Numerical Integration on Sparse Grids." *Journal of Econometrics*, Vol. 144 (2008), pp. 62–80. http://www.sciencedirect.com/science/article/B6VC0-4RJYVBD-1/2/12335dcfcbca363c96edf0931cc1b02f.
Hess, S., Train, K.E. and Polak, J.W. "On the Use of a Modified Latin Hypercube Sampling (MLHS) Method in the Estimation of a Mixed Logit Model for Vehicle Choice." *Transportation Research Part B: Methodological*, Vol. 40 (2006), pp. 147–163. https://ideas.repec.org/a/eee/transb/v40y2006i2p147-163.html.
Ho, K. and Pakes, A. "Hospital Choices, Hospital Prices, and Financial Incentives to Physicians." *American Economic Review*, Vol. 104 (2014), pp. 3841–84. http://www.aeaweb.org/articles?id=10.1257/aer.104.12.3841.
Hong, H., Li, H. and Li, J. "BLP Estimation Using Laplace Transformation and Overlapping Simulation Draws." *Journal of Econometrics*, (Forthcoming).
Houde, J.F. "Spatial Differentiation and Vertical Mergers in Retail Markets for Gasoline." *American Economic Review*, Vol. 102 (2012), pp. 2147–82. http://www.aeaweb.org/articles?id=10.1257/aer.102.5.2147.
Iaria, A. and Wang, A. "Identification and Estimation of Demand for Bundles." Working Paper, 2019.
Johnson, S.G. *PyCall* (2019). https://github.com/JuliaPy/PyCall.jl.
Judd, K. *Numerical Methods in Economics*, Vol. 1, 1 ed. Cambridge, Mass.: The MIT Press, 1998. https://EconPapers.repec.org/RePEc:mtp:titles:0262100711.
Judd, K.L. and Skrainka, B. "High Performance Quadrature Rules: How Numerical Integration Affects a Popular Model of Product Differentiation." CeMMAP Working Papers CWP03/11, Centre for Microdata Methods and Practice, Institute for Fiscal Studies, 2011.
Knittel, C.R. and Metaxoglou, K. "Estimation of Random-Coefficient Demand Models: Two Empiricists' Perspective." *Review of Economics and Statistics*, Vol. 96 (2014), pp. 34–59.
Konovalov, A. and Sandor, Z. "On Price Equilibrium with Multi-Product Firms." *Economic Theory*, Vol. 44 (2010), pp. 271–292. http://www.jstor.org/stable/40864781.
Lee, J. and Seo, K. "A Computationally Fast Estimator for Random Coefficients Logit Demand Models Using Aggregate Data." *The RAND Journal of Economics*, Vol. 46 (2015), pp. 86–102. http://www.jstor.org/stable/43895583.
———. "Revisiting the nested fixed-point algorithm in BLP random coefficients demand estimation." *Economics Letters*, Vol. 149 (2016), pp. 67–70.
Lee, R.S. "Vertical Integration and Exclusivity in Platform and Two-Sided Markets." *American Economic Review*, Vol. 103 (2013), pp. 2960–3000. http://www.aeaweb.org/articles?id=10.1257/aer.103.7.2960.
McFadden, D. and Train, K. "Mixed MNL Models for Discrete Response." *Journal of Applied Econometrics*, Vol. 15 (2000), pp. 447–470.
Miller, N. and Weinberg, M. "Understanding the Price Effects of the MillerCoors Joint Venture." *Econometrica*, Vol. 85 (2017), pp. 1763–1791.
Miravete, E.J., Seim, K. and Thurk, J. "Market Power and the Laffer Curve." *Econometrica*, Vol. 86 (2018), pp. 1651–1687. https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA12307.
More, J.J., Garbow, B.S. and Hillstrom, K.E. "User Guide for MINPACK-1." Technical report, Argonne National Laboratory, 1980.
Morrow, W.R. and Skerlos, S.J. "On the Existence of Bertrand-Nash Equilibrium Prices Under Logit Demand." *CoRR*, vol. abs/1012.5832 (2010).
———. "Fixed-Point Approaches to Computing Bertrand-Nash Equilibrium Prices Under Mixed-Logit Demand." *Operations Research*, Vol. 59 (2011), pp. 328–345. http://or.journal.informs.org/content/59/2/328.abstract.
Nevo, A. "Mergers with Differentiated Products: The Case of the Ready-to-Eat Cereal Industry." *RAND Journal of Economics*, Vol. 31 (2000a), pp. 395–421.
———. "A Practitioner's Guide to Estimation of Random Coefficients Logit Models of Demand (Including Appendix)." *Journal of Economics and Management Strategy*, Vol. 9 (2000b), pp. 513–548.
———. "Measuring Market Power in the Ready-to-Eat Cereal Industry." *Econometrica*, Vol. 69 (2001), pp. 307–342.
Newey, W. "Generalized Method of Moments Specification Testing." *Journal of Econometrics*, Vol. 29 (1985), pp. 229–256. https://EconPapers.repec.org/RePEc:eee:econom:v:29:y:1985:i:3:p:229-256.

Newey, W.K. "Semiparametric Efficiency Bounds." *Journal of Applied Econometrics*, Vol. 5 (1990), pp. 99–135. https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.3950050202.

Newey, W.K. and Smith, R.J. "Higher Order Properties of GMM and Generalized Empirical Likelihood Estimators." *Econometrica*, Vol. 72 (2004), pp. 219–255.

Nielson, C. "Targeted Vouchers, Competition Among Schools, and the Academic Achievement of Poor Students." Working Paper, 2017.

Owen, A. "Scrambled Net Variance for Integrals of Smooth Functions." *Annals of Statistics*, Vol. 25 (1997), pp. 1541–1562.

Owen, A.B."Multidimensional Variation for Quasi-Monte Carlo." *World Scientific* (2005), pp. 49–74. https://www.worldscientific.com/doi/abs/10.1142/9789812567765_0004.

———. "A Randomized Halton Algorithm in R." 2017.

Petrin, A. "Quantifying the Benefits of New Products: The Case of the Minivan." *Journal of Political Economy*, Vol. 110 (2002), pp. 705–729.

Reynaert, M. and Verboven, F. "Improving the Performance of Random Coefficients Demand Models: The Role of Optimal Instruments." *Journal of Econometrics*, Vol. 179 (2014), pp. 83–98. https://EconPapers.repec.org/RePEc:eee:econom:v:179:y:2014:i:1:p:83-98.

Reynaerts, J., Varadhan, R. and Nash, J.C. "Enhancing the Convergence Properties of the BLP (1995) Contraction Mapping." Vives Discussion Paper Series 35, Katholieke Universiteit Leuven, Faculteit Economie en Bedrijfswetenschappen, Vives, 2012. http://ideas.repec.org/p/ete/vivwps/35.html.

Salanie, B. and Wolak, F. "Fast, Robust, and Approximately Correct: Estimating Mixed Demand Systems." Working Paper, 2019.

Skrainka, B. *Three Essays on Product Differentiation*. PhD dissertation, University College London, 2012a.

Skrainka, B.S. "A Large Scale Study of the Small Sample Performance of Random Coefficient Models of Demand." Working Paper, 2012b.

Somaini, P. and Wolak, F. "An Algorithm to Estimate the Two-Way Fixed Effects Model." *Journal of Econometric Methods*, Vol. 5 (2016), pp. 143–152.

Stock, J.H. and Wright, J.H. "GMM with Weak Identification." *Econometrica*, Vol. 68 (2000), pp. 1055–1096.

Su, C.L. and Judd, K.L. "Constrained Optimization Approaches to Estimation of Structural Models." *Econometrica*, Vol. 80 (2012), pp. 2213–2230.

Train, K. "Halton Sequences for Mixed Logit." Technical report, University of California, Berkeley, 2000.

———. *Discrete Choice Methods with Simulation*. Cambridge, UK: Cambridge University Press, 2009. https://EconPapers.repec.org/RePEc:cup:cbooks:9780521766555.

Varadhan, R. and Roland, C. "Simple and Globally Convergent Methods for Accelerating the Convergence of Any EM Algorithm." *Scandinavian Journal of Statistics*, Vol. 35 (2008), pp. 335–353. http://www.jstor.org/stable/41548597.

Villas-Boas, S.B. "Vertical Relationships between Manufacturers and Retailers: Inference with Limited Data." *The Review of Economic Studies*, Vol. 74 (2007), pp. 625–652. http://www.jstor.org/stable/4626153.

Supporting information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Figure OA1: Histograms for Median Product Own-Price Elasticities
Figure OA2: Histograms for Top Product Own-Price Elasticities
Figure OA3: Histograms for Median Product Markups
Figure OA4: Histograms for Top Product Markups
Figure OA5: Histograms for GMM Objective Values
Figure OA6: Integration Error: Importance Sampling
Figure OA7: Integration Error: Importance Sampling and Relative RMSE
Figure OA8: Instrument Strength and Misspecification: Variance
Figure OA9: Impact of Supply-Side Moments on Optimal IV for Demand-Only Problem: Variance
Figure OA10: Instrument Strength and Misspecification: Non-Optimal Instruments, Bias
Figure OA11: Instrument Strength and Misspecification: Non-Optimal Instruments, Variance
Figure OA12: Profiled GMM Objective with Alternative Instruments: Complex Simulation
Figure OA13: Profiled GMM Objective with Alternative Instruments: RCNL Simulation
Figure OA14: Problem Scaling: σ_p and ϕ