Enhancing the Merger Simulation Toolkit with ML/AI

Harold Chiang Jack Collison Lorenzo Magnolfi Christopher Sullivan

September 27, 2024

Work in Progress

Motivation

- Section 7 of the Clayton Act prohibits mergers if "*[..] the effect of such acquisition[s] may be substantially to lessen competition or to tend to create a monopoly.*"
- From Horizontal Merger Guidelines:
  - "[FTC & DOJ] *seek to identify and challenge competitively harmful mergers while avoiding unnecessary interference with mergers that are either competitively beneficial or neutral.*"
  - "*Most merger analysis is necessarily predictive, requiring an assessment of what will likely happen if a merger proceeds as compared to what will likely happen if it does not.*"
  - "*What sufficient data are available, the Agencies may construct economic models designed to quantify the unilateral price effects resulting from the merger.*"
- How to provide useful predictions on the effects of mergers?

The Merger Simulation Toolkit

- The standard merger simulation method is well-understood and powerful (e.g., Nevo, 2018)
- It focuses on unilateral price effects, and relies on the structure of demand and supply
  - Estimate a matrix of own- and cross-price demand elasticities
  - Typically implemented with two supply-side assumptions:
    1. Nash-Bertrand pricing conduct
    2. Constant marginal cost
  - Can solve for counterfactual post-merger prices
    - holding conduct, demand, and costs fixed or under assumptions, e.g., on efficiencies
- Evidence on the performance of merger simulation retrospectives is mixed (e.g., Björnerstedt and Verboven, 2016)
  - A restrictive supply side is among one of the potential problems (Peters, 2006)

What We Do

- Consider a more flexible, semi/nonparametric supply-side model
  - Nonparametric markup function, depends on endogenous prices and quantities
- Estimate model with AI/ML
  - Adapt Variational Method of Moments (VMM) developed by Bennett and Kallus (2023) for linear IV to model of oligopoly
  - Uses deep learning + an objective function with instruments
  - Potentially better performance with high-dimensional data than standard nonparametrics
  - We provide a bias-corrected inference procedure for multidimensional output
- VMM outperforms standard merger simulation and naive neural network predictions
  - Simulations quantify performance differences
  - Application: mergers in airline markets

The Merger Simulation Toolkit

Suppose we only observe pre-merger data:

- (s, p) endog. outcomes, (x, w) exog. demand and supply shifters, ownership matrix $\mathcal{H}$

1. Estimate demand, obtain matrix $D\left(\hat{\theta}, s, p\right)$ such that $D_{jk}\left(\hat{\theta}, s, p\right) = \frac{\partial s_j}{\partial p_k}\left(\hat{\theta}, s, p\right)$
2. Under Bertrand-Nash pricing back out

$$c = p - \left(\mathcal{H} \odot D\right)^{-1} s$$

3. Use the model predict prices under post-merger ownership matrix $\tilde{\mathcal{H}}$ as solution to:

$$\tilde{p} = c + \left(\tilde{\mathcal{H}} \odot D(\tilde{p}, \cdot)\right)^{-1} s(\tilde{p}, \cdot)$$

A Flexible Model of Supply

- Merger simulation is complex prediction problem with simultaneity
  - Prices are an equilibrium object and correlated with demand
  - Naive prediction approaches will fail to recognize this
- The Nash-Bertrand assumption doesn't always work well
- We develop a flexible model of supply that relaxes Bertrand-Nash and constant cost assumptions
- Throughout, we assume $D = \frac{\partial s}{\partial p}$ is known and focus on the supply-side

Flexible Models of Supply

In general, can express

$$p = \Delta(s, p, x) + c(s, w, \omega)$$

as long as the following holds

- Assumption 1: There exists a unique equilibrium, or the equilibrium selection rule is such that the same p arises whenever the vector (w, x, ω) is the same.

We also maintain:

- Assumption 2: The cost function is separable in ω, or $c(s, w, \omega) = \tilde{c}(s, w) + \omega$.
- Assumption 3: The markup function Δ only depends on s and D.

so we can write

$$p = h(s, D, w) + \omega$$

Remarks

- More general than workhorse model!
    - Assumption 1 amounts to static model describing the data
    - Assumption 2 is almost without loss
    - Assumption 3 satisfied for very broad range of conduct models (e.g., Bertrand, Cournot, Stackelberg, many collusive models, models where firms max profits + consumer surplus)
- Notice that formulation of $\tilde{\Delta}$ does not enforce separability of cost and markup
    - Extension: we can enforce separability with extra regularization steps (not today)
- Can be used for merger simulation (or other counterfactuals), finding prices that solve:

$$\tilde{p} - \hat{h}(s(\tilde{p}), D(\tilde{p}), w) - \hat{\omega} = 0$$

where $\hat{h}$ is the VMM model estimate, $s(\cdot)$ is demand, and $\hat{\omega}$ are estimated residuals

Identification

- We rely on a moment condition with instruments z for identification
  - Instruments are of the right dimension, assume completeness
  - Exogeneity moment condition $\mathbb{E}[\omega|z, w] = 0$
- Candidate instruments include demand shifters
  - Sets of competing products, cost shifters of competitors, etc.
- Identification of both models follows arguments akin to Berry and Haile (2014)
- Standard nonparametric techniques are unlikely to perform well in finite samples

Estimation

- Classic nonparametric estimators are well studied for GMM type setups: see reviews by Carrasco et al. (2007); Chen (2007)
- Curse of dimensionality and instability in classical nonparametric estimation under current environment: documented in e.g., Bennett et al. (2019); Bennett and Kallus (2020)
- One can use neural networks to fit high-dimensional nonlinear functions with squared loss:

$$\hat{\theta}_n = \mathrm{argmin}_{\theta \in \Theta} \frac{1}{\mathrm{TJ}} \sum_{\mathrm{j,t}} (\mathrm{p_{jt}} - \mathrm{h}(\mathrm{s_t}, \mathrm{D_t}, \mathrm{w_{jt}}; \theta))^2$$

- However, standard neural networks ignore endogeneity
  - Cannot correctly recover the markup function $h(\cdot)$

Variational Method of Moments (VMM)

- Inherently, we have a moment condition for the structural markup:

$$\mathbb{E}[p_{jt} - h(s_t, D_t, w_{jt}; \theta)|z] = 0$$

- Given preliminary estimate $\tilde{\theta}_n$, reformulate Bennett and Kallus (2023) to:

$$\hat{\theta}_n = \mathrm{argmin}_{\theta\in\Theta} \sup_{\mathrm{f}\in\mathcal{F}_\mathrm{n}} \frac{1}{\mathrm{TJ}} \sum_{\mathrm{j,t}} \mathrm{f}(\mathrm{z_{jt}})^\mathrm{T} \rho(\mathrm{x_t}; \theta) - \frac{1}{4\mathrm{TJ}} \sum_{\mathrm{j,t}} (\mathrm{f}(\mathrm{z_{jt}})^\mathrm{T} \rho(\mathrm{x_t}; \tilde{\theta}_\mathrm{n}))^2 - \mathrm{R_n}(\mathrm{f})$$

$$\text{s.t.} \quad \rho(x_t; \theta) = p_{jt} - h(s_t, D_t, w_{jt}; \theta)$$

- Both f and h are neural networks, allowing flexible controls of model complexity to cope with the curse of dimensionality
- $R_n(\cdot)$ is a penalty term that regularize the complexity of f
- We can use the estimate of the structural object h for merger simulation

Inference

- If $\tilde{\theta}_n \xrightarrow{p} \theta_0$, under regularity conditions, Theorems 2-3 in Bennett and Kallus (2023) imply:

$$\begin{aligned}\sqrt{n}(\hat{\theta}_n - \theta_0) = &- \Omega_0^{-1}\sqrt{n}\psi_n + o_p(1) \\ &\xrightarrow{d} N(0, \Omega_0^{-1}),\end{aligned}$$

where

$$\begin{aligned}\Omega_0 =& \mathbb{E}\left[\mathbb{E}[D(X;\theta_0)|Z]^T V(Z;\theta_0)^{-1}\mathbb{E}[D(X;\theta_0)|Z]\right], \\ V(Z;\theta) =& \mathbb{E}[\rho(X;\theta)\rho(X;\theta)^T|Z], \\ D(X;\theta) =& \nabla_\theta \rho(X;\theta)\end{aligned}$$

Inference

- By the delta method, for a transformation $h : \mathbb{R}^b \supset \Theta \to \mathbb{R}^d$, denote

$$h(\hat{\theta}_n) = [h_{x_1}(\hat{\theta}_n), ..., h_{x_d}(\hat{\theta}_n)]^T = [h_1(\hat{\theta}_n), ..., h_d(\hat{\theta}_n)]^T,$$

  it holds that

$$\sqrt{N}(h(\hat{\theta}_n) - h(\theta_0)) \xrightarrow{d} N(0, \nabla_{\theta'} h(\theta_0)\Omega_0^{-1}\nabla_{\theta'} h(\theta_0)^T)$$

Inference: Simplest Case ($d = 1$)

- Note that $\nabla_{\theta'} h(\theta_0)$ is $d \times b$; in the simplest case, suppose that $d = 1$
- Lemma 9 in Bennett and Kallus (2023) states that for any $\beta \in \mathbb{R}^b$, we have:

$$\beta^T \Omega_0^{-1} \beta = -\frac{1}{4} \inf_{\gamma \in \mathbb{R}^b} \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}[f(Z)^T \nabla_\theta \rho(X; \theta_0) \gamma] - \frac{1}{4} \mathbb{E}[(f(Z)^T \rho(X; \theta_0))^2] - 4\gamma^T \beta - R_n(f) \right\} \quad (1)$$

- Take $\beta = \nabla_\theta h_x(\theta_0)$ and the above solution to the optimization problem becomes:

$$\sigma_x^2 = \nabla_\theta h_x(\theta_0) \Omega_0^{-1} \nabla_\theta h_x(\theta_0)^T$$

- This is the asymptotic variance for $\sqrt{N}(h_x(\hat{\theta}_n) - h_x(\theta_0))$
  - $\nabla_\theta h_x(\theta_0)$ can be difficult to compute analytically
  - Numerical differentiation can be employed (e.g., Hong et al. (2015))
  - Expectations can be replaced by sample means, $\hat{\theta}_n$ can be used in place of θ_0
  - These together yield a feasible version of Equation (1) which provides an estimator $\hat{\sigma}_x^2$ for σ_x^2

Inference: Extending to $d \geq 2$

- The approach above cannot obtain a covariance matrix when $d \geq 2$
- Holm's Step-Down procedure using the estimates for $\hat{\sigma}^2_{x_j}$ and $h(\hat{\theta})$ for each $j = 1, ..., d$
- The set of critical values T_α is known for significance levels $\frac{\alpha}{d+1-k}$ and $k = 1, ..., d$
    - We can use a folded normal distribution with $t = 1$ to account for bias
- For any ordering of x and fixed ordering T_α, we can compute the confidence interval:

$$h_x(\hat{\theta}) \pm N^{-\frac{1}{2}} \hat{\sigma}_x T_\alpha$$

- We compute this for all permutations of $j = 1, ..., d$, resulting in $d!$ permutations of x
- This is because we must consider any possible ordering of the p-values of $x_1, ..., x_d$

Inference Algorithm

1. Estimate $\hat{\sigma}^2_{x_j}$ for $\sigma^2_{x_j}$ for $j \in \{1, ..., d\} \equiv J$ by solving the feasible version of Equation (1)
2. Fix values $T_\alpha = \{T_{\alpha_k} : k = 1, ..., d\}$ where $\alpha_k = \frac{\alpha}{d+1-k}$
3. For each permutation $\tilde{J}$ of J:
    3.1 Arrange values $\tilde{x}$ and $\hat{\sigma}_{\tilde{x}}$ with permuted indices $\tilde{J}$
    3.2 Construct bounds as $h_{\tilde{x}}(\hat{\theta}) \pm n^{-\frac{1}{2}} \hat{\sigma}_{\tilde{x}} T_\alpha$ with fixed T_α
4. Simultaneous confidence interval as the union of $2 \times d \times d!$ linear constraints from Step (3)

Simulations Setup

- Simple parametric simulations to evaluate performance relative to the baseline
  - *Demand*: Logit with two independent product characteristics
  - *Supply*: Linear costs with two independent cost shifters
- We simulate data under two different assumptions on conduct
  - *Bertrand*: Identity ownership matrix
  - *Profit Weight*: Off-diagonal weights of 0.75

Evaluating Performance

- We need a way to compare different (potentially misspecified) models
- A relevant comparison is looking at the residuals ω under different assumptions
    - True, Bertrand, monopoly, perfect competition, and flexible models
    - Residuals from the true model are irreducible
- We take the mean squared difference between model residuals and true residuals
- The idea is to see how far off the prediction error is from the irreducible error

Comparison of Models

- For known demand system, under Bertrand, Monopoly, and perfect competition we recover residuals ω^B, ω^M, and ω^P
- *Flexible Model* In the flexible supply-side model, we estimate a flexible function h and recover $\hat{\omega}_{jt}$:

$$p_{jt} = h_j\left(s_t, D_t, w_{jt}\right) + \hat{\omega}_{jt}$$

- *Naive Model* A naive flexible supply-side model ignores endogeneity; we estimate a flexible function h^N and recover $\hat{\omega}^N$:

$$p_{jt} = h_j^N\left(s_t, D_t, w_{jt}\right) + \hat{\omega}_{jt}^N$$

- We evaluate performance for different neural network architectures, sample sizes, and inclusion of demand derivatives

Bertrand Results Large Network

Table 1: Model Comparison (Bertrand, Small Network)

Sample Size	Derivatives	ω	ω^B	ω^M	ω^P	$\hat{\omega}$	$\hat{\omega}^N$
N = 100	No	0.005	0.005	583.409	6.518	0.892	1.693
N = 100	Yes	-	-	-	-	0.556	1.319
N = 1,000	No	0.001	0.001	979.962	5.977	1.390	1.800
N = 1,000	Yes	-	-	-	-	0.348	0.978
N = 10,000	No	0.000	0.000	1693.914	6.317	1.221	1.743
N = 10,000	Yes	-	-	-	-	0.170	1.047

Profit Weight ($\theta = 0.75$) Results Large Network

Table 2: Model Comparison (Profit Weight, Small Network)

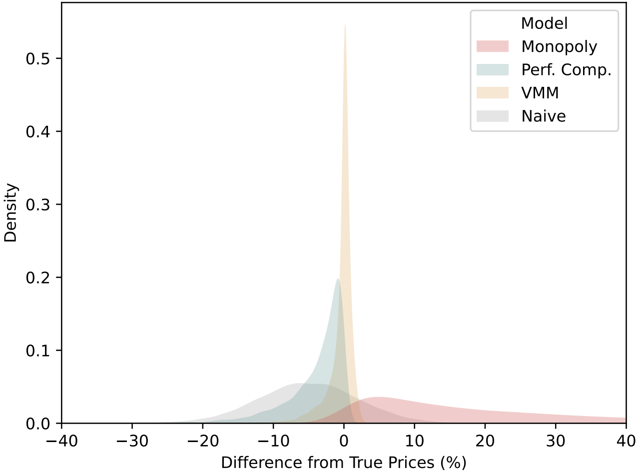
Sample Size	Derivatives	ω	ω^B	ω^M	ω^P	$\hat{\omega}$	$\hat{\omega}^N$
N = 100	No	0.005	8.765	5.077	11.474	2.330	2.934
N = 100	Yes	-	-	-	-	2.749	2.512
N = 1,000	No	0.001	7.058	6.264	7.802	2.385	2.314
N = 1,000	Yes	-	-	-	-	1.176	1.747
N = 10,000	No	0.000	7.965	6.289	8.690	1.855	2.563
N = 10,000	Yes	-	-	-	-	1.112	0.892

Key Takeaways

- In the Bertrand simulations, we outperform the monopolist and perfect competition
    - Without the derivative matrix, performance holds roughly constant
    - Including the derivative matrix greatly improves performance
- In the profit weight simulations, we outperform all but the true model
    - Performance is expectedly not as good as in the baseline model
    - Our estimator scales better with sample size with the derivative matrix
- Larger neural networks improve learning in some cases
    - Performance is improved with sample size, especially for the profit weight model
    - Performance does not improve upon the addition of the derivative matrix
- The naive estimator underperforms the variational method of moments

Merger Simulation for Bertrand

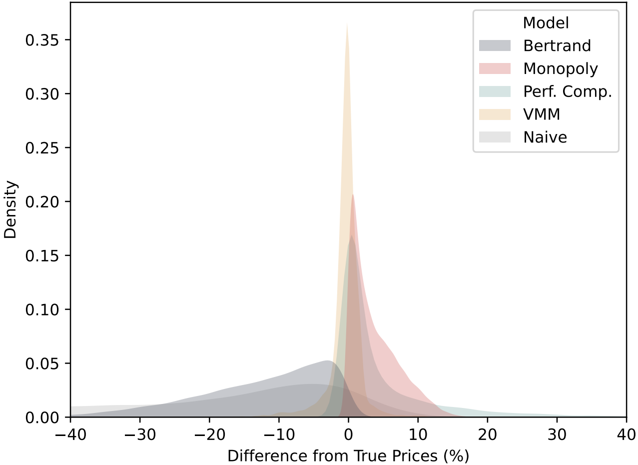
Figure 1: Bertrand Merger Simulation



Model	MSE
Bertrand	0.00
Monopoly	26.26
Perf. Comp.	1.80
VMM	0.27
Naive	3.92

Merger Simulation for Profit Weight ($\theta = 0.75$)

Figure 2: Profit Weight Merger Simulation



Model	MSE
Bertrand	15.19
Monopoly	1.55
Perf. Comp.	3.57
VMM	0.24
Naive	23.79

Interpretation via Passthrough

- **Key question:** How do we interpret the flexible markup function?
- A useful object for comparison is the passthrough matrix
  - Increase costs c by 10%, loading increases on the residual $\hat{\omega}$
  - Solve for equilibrium prices under different models of conduct
- Compare post-merger prices for markets with median inside market share
  - This holds market structure constant across models

Bertrand Passthrough

Table 3: Bertrand Passthrough Comparison
$c_1 = 15.85,\ c_2 = 12.54,\ s_1 = 0.54,\ s_2 = 0.15$

(a) True Model (Bertrand)

0.49	0.05
0.14	0.88

(b) VMM

0.49	0.10
0.10	0.66

- The flexible model learns markup functions that imply correct passthroughs

Profit Weight ($\theta = 0.75$) Passthrough

Table 4: Profit Weight Passthrough Comparison
$c_1 = 13.75,\ c_2 = 12.96,\ s_1 = 0.61,\ s_2 = 0.04$

(a) True Model ($\theta = 0.75$)

0.39	-0.44
-0.00	0.97

(b) VMM

0.40	-0.31
-0.00	0.88

- The flexible model learns markup functions that imply correct passthroughs

Inference

Table 5: Inference Comparison by Sample Size (Small Network)

Model	Sample Size	ψ	$\hat{\psi}$	Avg. $\hat{\sigma}/\sqrt{N}$	Min. $\hat{\sigma}/\sqrt{N}$	Max. $\hat{\sigma}/\sqrt{N}$	Interval
Baseline	N = 100	21.014	19.566	3.654	0.870	6.315	[8.748, 30.385]
Complex	N = 100	17.321	14.570	2.650	0.309	3.753	[6.726, 22.415]

- Intuitively: when predicting price at a particular market structure, uncertainty is (i) quantifiable, and (ii) reasonable already at a low sample size of $N = 100$

Application

- Airline markets in the US have rich data from DB1B
    - Fares, passenger counts, distances, carrier identifiers, etc.
    - Origin and destinations of trips
- Rich data on competition for a long panel with several large mergers
- Difference-in-differences estimates to approximate unilateral price effects of mergers
    - Zoom in on markets that move from $3 \rightarrow 2$ firms post-merger
    - Treated markets are concentrated[1] markets in which both merging firms are present
- Nested logit demand, as usual for the airline industry
- Merger simulation focuses on American-US Airways merger
    - Zoom in on markets that move from $3 \rightarrow 2$ firms post-merger
    - Condition on similar share of nonstop flights in the pre- and post-period

[1] HHI $\in [1,000, 1,800]$ with ΔHHI ≥ 100 or HHI $> 1,800$ with ΔHHI ≥ 50

Airline Concentration

Figure 4: HHI in the Airline Industry

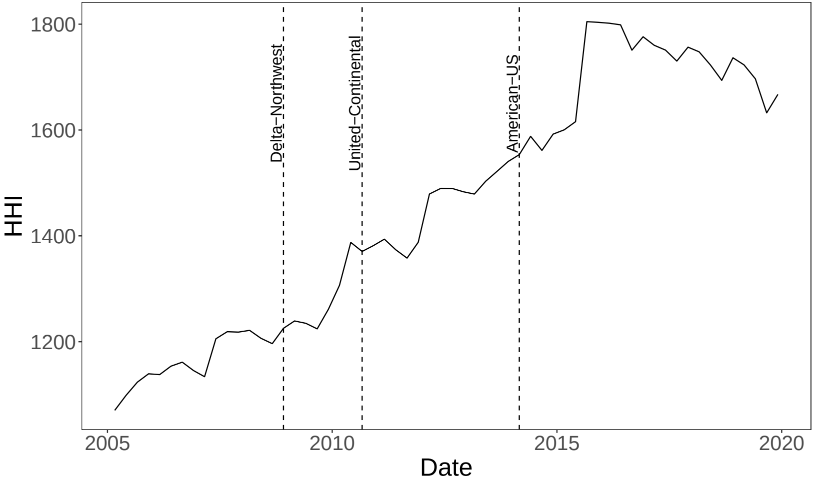


Table 6: Difference-in-Differences Estimates

	log(Fare)		
	DL-NW	UA-CO	AA-US
	(1)	(2)	(3)
Treated × Post	0.0108	-0.0401	0.0517**
	(0.0192)	(0.0283)	(0.0247)
Share Nonstop	-0.2389***	-0.2218***	-0.1424***
	(0.0197)	(0.0204)	(0.0186)
R^2	0.52375	0.53233	0.44646
Observations	15,536	12,849	22,389
Treated (%)	11.251	5.0120	5.4800
Origin-destination fixed effects	✓	✓	✓
Year-quarter fixed effects	✓	✓	✓

Demand

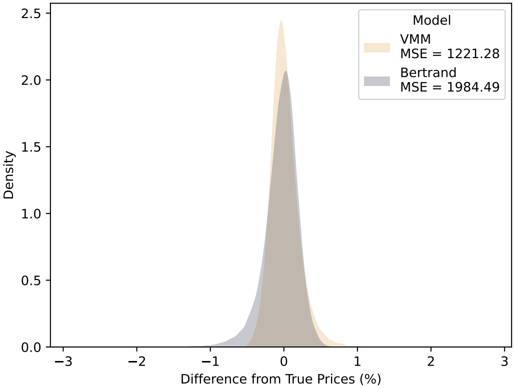
Table 7: Demand Estimates

	$\log(s_{jt})$ - $\log(s_{0t})$
Average Fare	-0.0048***
	(0.0004)
$\log(S_t)$	0.8356***
	(0.0133)
Share Nonstop	0.4030***
	(0.0282)
Average Distance (1,000's)	-0.4881***
	(0.0498)
Average Distance^2 (1,000's)	0.0485***
	(0.0045)
$\log(1 + \text{Num. Fringe})$	-0.2642***
	(0.0057)
R^2	0.94238
Observations	1,283,472
Own-price elasticity	-5.1652
Origin-destination fixed effects	✓

- Elasticities broadly in line with literature (e.g., Berry and Jia, 2010)

Fit: Pooled In-Sample and Out-of-Sample Results

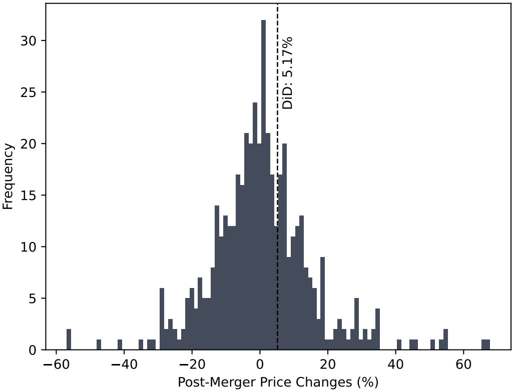
Figure 5: Model Comparison



- Reduction of $\sim 40\%$ in passenger-weighted MSE relative to Bertrand with constant costs

Merger Simulation: Observed Price Changes

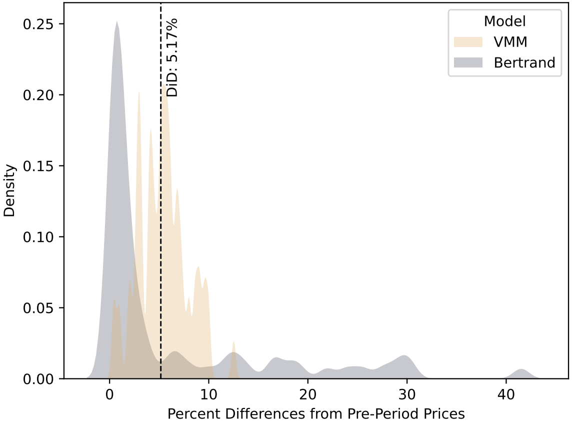
Figure 6: Price Change Distribution



- Price changes after the AA-US merger in 3 $\to$ 2 markets with $> 80\%$ nonstop flights

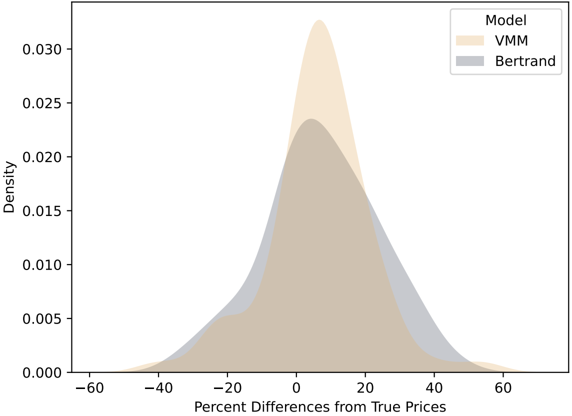
Merger Simulation: Predicted Price Changes

Figure 7: Predicted Price Change Distribution



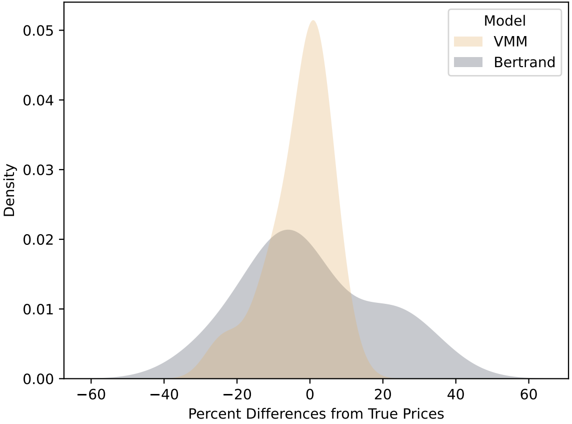
Merger Simulation: All Markets

Figure 8: Merger Simulation Comparison



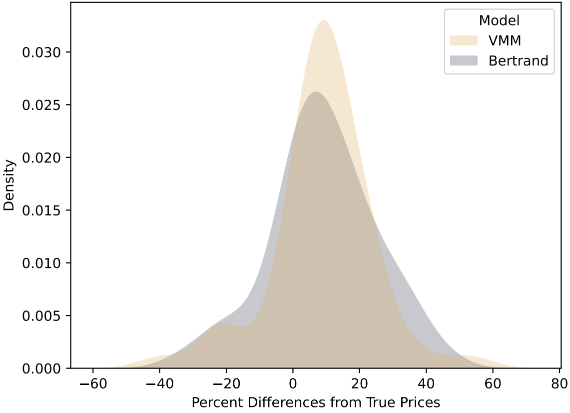
Merger Simulation: Markets with Price Increases

Figure 9: Merger Simulation Comparison



Merger Simulation: Markets with Price Decreases

Figure 10: Merger Simulation Comparison



Easy to Use Framework

Figure 11: Using VMM

```
# Import VMM modules
from VMM.experiment_setups.inference_experiment_setups import *
from VMM.experiment_setups.estimation_experiment_setups import *
from VMM.utils.hyperparameter_optimization import *
from VMM.scenarios.abstract_parametric_scenario import ParametricDataset
from VMM.scenarios.logit_scenario import EmpiricalLogitScenario
from VMM.scenarios import *
from VMM.predictor import *

# Import simulation modules
from Simulation.utils import *
from Simulation.simulate import *
from Simulation.mergers import *
from Simulation.diagnostics import *

# Import libraries
from argparse import ArgumentParser, ArgumentDefaultsHelpFormatter
from sklearn.model_selection import GroupShuffleSplit
import pandas as pd
import pickle
import random

# Load data
data = pd.read_csv("Data/data.csv")

# Set variables
exog = ['average_distance_thousands']
chars = ['share_nonstop', 'average_distance_thousands_squared']
ins = ['average_distance_rival', 'average_num_markets_rival', 'rival_carriers']
F = np.max(data.groupby('market_ids')['firm_ids'].nunique())

# Set up scenario
scenario = EmpiricalLogitScenario()
scenario.setup(data, exog=exog, chars=chars, ins=ins, F=F)
train = scenario.get_dataset("train")
dev = scenario.get_dataset("dev")
test = scenario.get_dataset("test")

# Fit model
n = len(np.unique(data.iloc[train.index].market_ids))
predictor = get_estimator(scenario, train.z, n=n, use_gpu=True)
predictor.fit(x=train.x, z=train.z, m=data.iloc[train.index].market_ids, x_dev=dev.x, z_dev=dev.z)
```

Conclusion

- We propose modern methodology to estimate flexible models of supply
  - Sidesteps the curse of dimensionality of nonparametric methods, avoids misspecification
  - Introduce tractable inference for multidimensional output
- Simulations show that the flexible method outperforms misspecified models
  - Performs well in-sample and for *ex ante* post-merger predictions
- We take the model to the data in the airline industry
  - Evaluate the American-US Airways merger using
  - Outperform the original merger simulation toolkit

Table 8: Model Comparison (Bertrand, Large Network)

Sample Size	Derivatives	ω	ω^B	ω^M	ω^P	$\hat{\omega}$	$\hat{\omega}^N$
N = 100	No	0.005	0.005	583.409	6.518	2.127	0.848
N = 100	Yes	-	-	-	-	1.234	1.259
N = 1,000	No	0.001	0.001	979.962	5.977	0.645	0.802
N = 1,000	Yes	-	-	-	-	0.690	0.791
N = 10,000	No	0.000	0.000	1693.914	6.317	0.352	0.875
N = 10,000	Yes	-	-	-	-	0.506	0.875

Results

Back

Table 9: Model Comparison (Profit Weight, Large Network)

Sample Size	Derivatives	ω	ω^B	ω^M	ω^P	$\hat{\omega}$	$\hat{\omega}^N$
N = 100	No	0.005	8.765	5.077	11.474	1.359	1.847
N = 100	Yes	-	-	-	-	2.381	2.233
N = 1,000	No	0.001	7.058	6.264	7.802	1.213	0.812
N = 1,000	Yes	-	-	-	-	0.814	0.820
N = 10,000	No	0.000	7.965	6.289	8.690	0.324	0.887
N = 10,000	Yes	-	-	-	-	0.301	0.892