Enhancing the Merger Simulation Toolkit with ML/AI

Harold Chiang Jack Collison Lorenzo Magnolfi Christopher Sullivan

October 26, 2024 — Midwest IO Fest

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)
- 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., Bjöornerstedt 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) (Bennett and Kallus, 2023)
  - Uses deep learning + an objective function with instruments
  - Better performance with high-dimensional data than standard nonparametric IV
  - We develop an inference procedure to quantify uncertainty in prediction
- VMM outperforms standard merger simulation and naive neural network predictions
  - Simulations showcase performance differences
  - Application: mergers in airline markets
  - Portable method, computationally manageable

The Merger Simulation Toolkit

Suppose we only observe pre-merger data across products j and markets t:

- (s_t, p_t) endog. outcomes, (x_t, w_t) exog. demand and supply shifters, ownership matrix $\mathcal{H}_t$

1. Estimate demand, obtain $s_t = \mathscr{s}(p_t, \hat{\theta}^D, \cdot)$ and matrix $D_t(p_t, \hat{\theta}^D, \cdot)$ s.t. $D_{jkt} = \frac{\partial \mathscr{s}_{jt}(p_t, \hat{\theta}^D)}{\partial p_{kt}}$
2. Under Nash-Bertrand pricing back out $c_t = p_t - \left(\mathcal{H}_t \odot D_t^{'}\right)^{-1} s_t$
3. Predict post-merger prices as solution to:

$$\tilde{p}_t = c_t + \left(\tilde{\mathcal{H}}_t \odot D_t(\tilde{p}_t, \hat{\theta}^D)^{'}\right)^{-1} \mathscr{s}(\tilde{p}_t, \hat{\theta}^D)$$

where $\tilde{\mathcal{H}}_t$ is post-merger ownership matrix

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 supply model, relaxing Nash-Bertrand and constant cost assumption
- Throughout, we assume $s(\cdot)$ and $D_t = \frac{\partial s_t}{\partial p_t}$ are known/estimated to focus on supply-side

Flexible Models of Supply

In general, can express

$$p_t = \Delta(s_t, p_t, x_t; \mathcal{H}_t) + c(s_t, w_t, \omega_t)$$

as long as the following holds

- Assumption 1: There exists a unique equilibrium, or the equilibrium selection rule is such that the same p_t arises whenever the vector (w_t, x_t, ω_t) is the same.

We also maintain:

- Assumption 2: The cost function is separable in ω_t, or $c(s_t, w_t, \omega_t) = \tilde{c}(s_t, w_t) + \omega_t$.
- Assumption 3: The markup function Δ only depends on s_t and D_t.

so we can write

$$p_t = h(s_t, D_t, w_t; \mathcal{H}_t) + \omega_t$$

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 h does not enforce separability of cost and markup
  - Extension: we can enforce separability with extra regularization steps (not today)
- For merger simulation $\tilde{\mathscr{H}}_t$ (or other counterfactuals), finding prices that solve:

$$\tilde{p}_t - \hat{h}(\mathscr{s}(\tilde{p}_t), D(\tilde{p}_t), w_t; \tilde{\mathscr{H}}_t) - \hat{\omega} = 0$$

where $\hat{h}$ is the VMM model estimate, $\mathscr{s}(\cdot)$ is demand, and $\hat{\omega}_t$ 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 $\mathbb{E}[\omega_{jt} \mid z_{jt}, w_{jt}] = 0$
- Identification follows arguments akin to Berry and Haile (2014)
- Candidate instruments:
  - own and rival prod. characteristics, rival's cost shifters, taxes, etc.
- Must include demand shifters *excluded* from cost
  - If not, w/ logit demand, may just recover inverse demand $h = \mathcal{s}^{-1}$
- But, 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)
- But, curse of dimensionality and instability in classical nonparametric estimation
  - documented in e.g., Bennett et al. (2019); Bennett and Kallus (2020)
- Can use neural networks to fit high-dimensional nonlinear functions with squared loss:

$$\hat{\theta}_N = \mathrm{argmin}_{\theta \in \Theta} \frac{1}{TJ} \sum_{j,t} \left(p_{jt} - h_j(s_t, D_t, w_t; \theta, \mathcal{H}_t)\right)^2$$

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

Variational Method of Moments (VMM)

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

$$\mathbb{E}[p_{jt} - h_j(s_t, D_t, w_t; \theta, \mathcal{H}_t) \mid z_t, w_t] = 0$$

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

$$\hat{\theta}_N = \mathrm{argmin}_{\theta \in \Theta} \mathrm{sup}_{f \in \mathcal{F}_N} \frac{1}{TJ} \sum_{j,t} f(z_{jt})^T \omega_{jt}(\theta) - \frac{1}{4TJ} \sum_{j,t} (f(z_{jt})^T \omega_{jt}(\tilde{\theta}_N))^2 - R_N(f)$$

$$\text{s.t.} \quad \omega_{jt}(\theta) = p_{jt} - h_j(s_t, D_t, w_t; \theta, \mathcal{H}_t)$$

- 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:

$$\sqrt{N}(\hat{\theta}_N - \theta_0) \xrightarrow{d} N(0, \Omega_0^{-1})$$

where

$$\Omega_0 = \mathbb{E}\left[\mathbb{E}[\nabla_\theta \omega(\theta_0) \mid z, w]^T \mathbb{E}[\omega(\theta)\omega(\theta)^T \mid z, w]^{-1} \mathbb{E}[\nabla_\theta \omega(\theta_0) \mid z, w]\right],$$

- For inference on d post-merger predicted prices $h(\hat{\theta}_N, \tilde{\mathscr{H}})$, delta method yields:

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

Inference Details

Simulations Setup

- Simple parametric simulations to evaluate performance relative to the baseline
  - Two or three single-product firms in T markets
  - *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 Predictive Performance

- We need a way to compare different (potentially misspecified) models
- We compare implied unobserved cost shocks ω^m under different models m
  - True, Bertrand, monopoly, perfect competition, and flexible models (VMM and naive NN)
  - Cost shocks from the true model are irreducible error (noise)
- We take the mean squared error (MSE) between model implied and true shocks
- Benchmark: how far from the irreducible error is the prediction error

Comparison of Models

- We recover ω^B, ω^M, and ω^P under Bertrand, Monopoly, and perfect competition
- *VMM Model:* For flexible supply-side model, we estimate h and recover $\hat{\omega}_{jt}$:

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

  - VMM instruments: own x, sum of rival x
- *Naive Model:* Ignores endogeneity; we estimate a with NN a flexible h^N and recover $\hat{\omega}^N$:

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

- We compute test sample MSE for different specifications of flexible models:
  - vary neural network architectures, sample sizes, and inclusion of demand derivatives

Bertrand DGP Results Large Network

Table 1: Test sample MSE across models (Bertrand DGP, Small Network)

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

Profit Weight DGP ($\kappa = 0.75$) Results Small Network

Table 2: Test sample MSE across models (Profit Weight, Large Network)

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

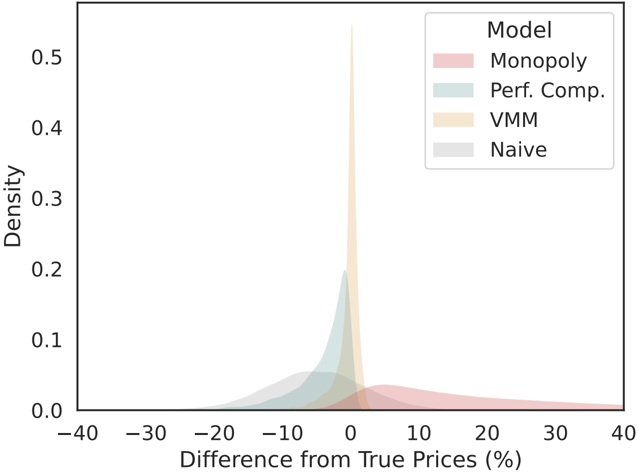
Key Takeaways

- In all simulations, VMM outperforms all but the true model
  - Including the derivative matrix greatly improves performance
- Larger neural networks improve learning in some cases
  - Performance is improved with sample size, especially for the profit weight model
- The naive estimator underperforms VMM

What about predictive performance in out-of-sample 3-to-2 merger simulation?

Merger Simulation for Bertrand DGP

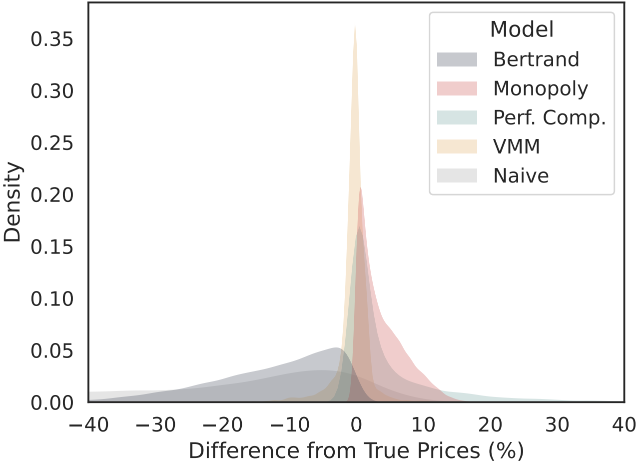
Figure 1: Prediction Error for Bertrand DGP Merger Simulation



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

Merger Simulation for Profit Weight DGP ($\kappa = 0.75$)

Figure 2: Prediction Error for Profit Weight Merger Simulation



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

Peeking Inside the Black Box: Interpretation via Pass-through

- **Key question:** How do we interpret the flexible $\hat{h}$ we recover?
- A useful object for comparison is the pass-through matrix implied by $\hat{h}$
- To compute pass-through:
    - Pick median post-merger market by inside share from simulations
    - Increase costs c by 10%, loading increases on the residual $\hat{\omega}$
    - Solve for equilibrium prices under different models of conduct
    - Compare price pre and post cost change, report price change/cost change

Bertrand DGP Pass-through

Table 3: Bertrand DGP Pass-through 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 and cost functions that imply correct pass-throughs

Profit Weight DGP ($\kappa = 0.75$) Pass-through

Table 4: Profit Weight DGP Pass-through Comparison
$c_1 = 13.75, \ c_2 = 12.96, \ s_1 = 0.61, \ s_2 = 0.04$

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

0.39	-0.44
-0.00	0.97

(b) VMM

0.40	-0.31
-0.00	0.88

- The flexible model learns markup and cost functions that imply correct pass-throughs

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
Bertrand	N = 253	21.014	21.092	0.817	0.059	1.415	[18.673, 23.512]
Bertrand	N = 2,579	20.341	20.474	0.057	0.042	0.071	[20.305, 20.642]
Profit Weight	N = 253	17.321	12.907	0.309	0.174	0.648	[11.991, 13.822]
Profit Weight	N = 2,579	17.375	15.554	0.099	0.069	0.158	[15.261, 15.847]

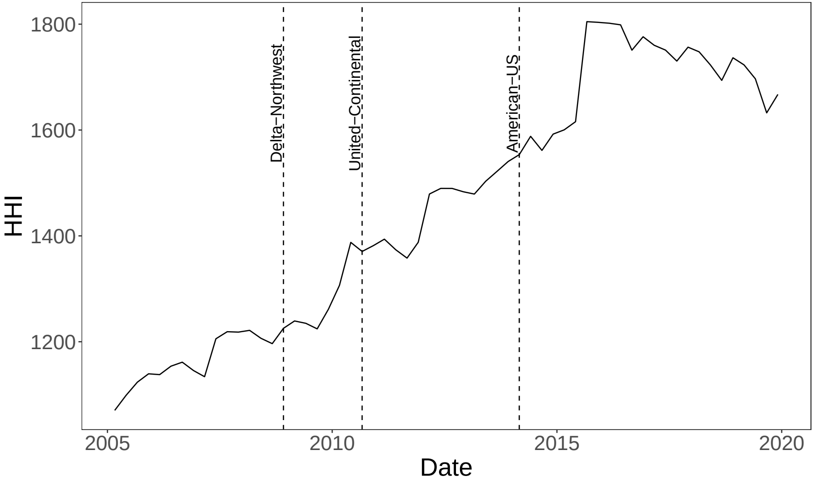
- Intuitively: when predicting price at a particular market structure, uncertainty is (i) quantifiable, (ii) reasonable already at a low sample size of $T = 100$, and (iii) decreasing with sample size

Application to Airline Mergers

- Good environment to test our method: airline markets in the US have rich data from DB1B
  - Fares, passenger counts, distances, carrier identifiers, etc.
  - Origin and destinations of trips
  - Several large mergers in sample
- Goal: predict unilateral price effects of American-US Airways merger
  - Zoom in on markets that move from $3 \rightarrow 2$ firms post-merger
  - Treated markets are markets in which both merging firms are present
- (We abstract from many interesting aspects of the industry here...)

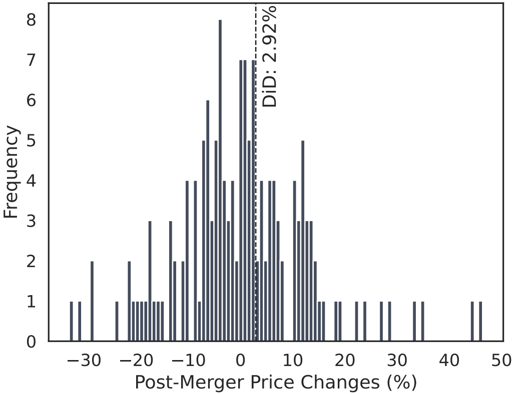
Airline Concentration

Figure 3: HHI in the Airline Industry



Observed Price Changes after AA-US Merger

Figure 4: Price Change Distribution



- Price changes after the AA-US merger in 3 $\rightarrow$ 2 markets

Demand

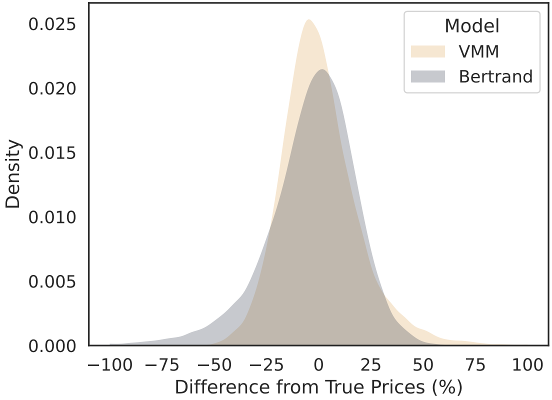
Table 6: Demand Estimates

	$\log(s_{jt})$ - $\log(s_{0t})$
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

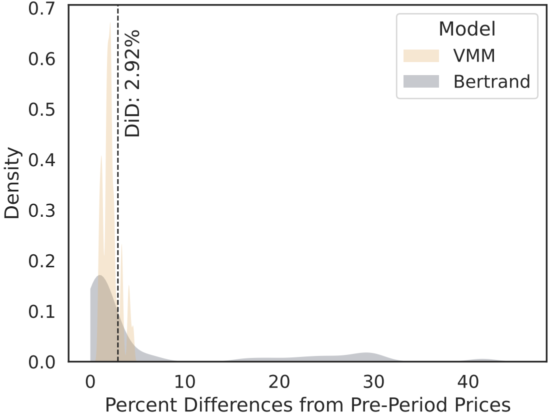


Model	Sample	MSE
Bertrand	All	1949.59
VMM	All	1242.95
Bertrand	Train	1932.70
VMM	Train	1235.39
Bertrand	Test	2016.33
VMM	Test	1272.82

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

Merger Simulation: Predicted Price Changes

Figure 6: Predicted Price Change Distribution

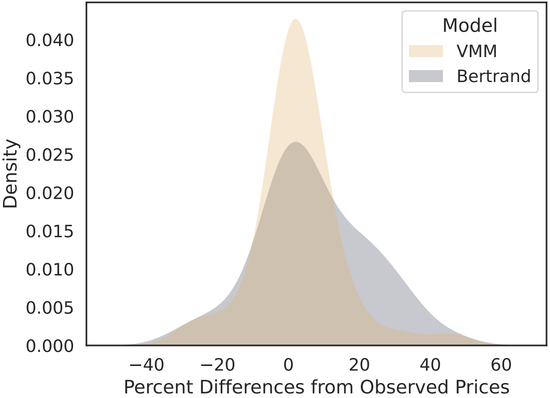


	Price Increases (%)	
Model	Median	Mean
Bertrand	1.45	6.66
VMM	2.05	2.16
DiD	-	2.92

- In theory, VMM can predict price decreases but it doesn't here

Merger Simulation: Comparing Predicted and Observed Post-merger Prices

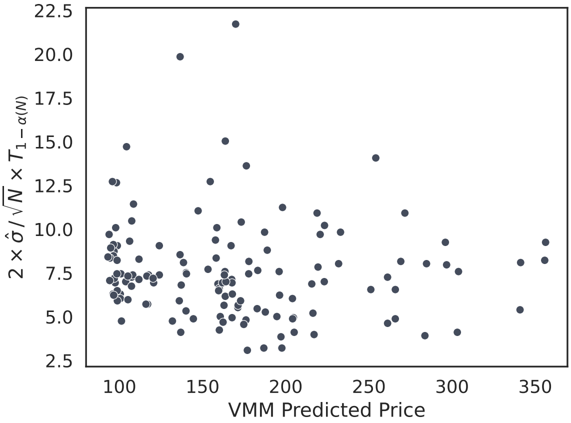
Figure 7: Merger Simulation Comparison



Model	MSE
Bertrand	365.71
VMM	66.93

Merger Simulation: Inference

Figure 8: Width of Confidence Intervals



Thank You!

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:

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

- 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

Back

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 $x_1, ..., x_d$

Back

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)

Back

Bertrand DGP Results

Back

Table 7: Test sample MSE across models (Bertrand DGP, Large Network)

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

Profit Weight DGP ($\kappa = 0.75$) Results Back

Table 8: Test sample MSE across models (Profit Weight DGP, Small Network)

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