Nonparametric inference on counterfactuals in first-price auctions

P. Andreyanov (HSE) and G. Franguridi (USC)

EEA-ESEM and EARIE August 2022

Introduction

Structural auctions

For those who work with auction data...

If you want to publish well, you might consider a structural approach.

Whenever a policy change may lead to a change in total surplus, and/or change of the division of surplus between bidders and auctioneer, you would try to estimate them, and maybe test some natural hypotheses.

Structural auctions

Then, you could reinforce your arguments with some numbers, e.g.,

- letting one more bidder in increases TS by 5%
- setting reserve price optimally decreases BS by 15%

You might want to do it non-parametrically, following Guerre, E., I. Perrigne, and Q. Vuong (2000): Optimal nonparametric estimation of first-price auctions, Econometrica, or as I will call it GPV ("gee-pee-vee")

First price auctions and GPV

The GPV estimator is a sample analog of inverse elasticity formula

$$\hat{v} = b + \frac{1}{M-1}\frac{\hat{F}(b)}{\hat{f}(b)}, \quad \text{where } F, f \text{ are the cdf, pdf of bids}$$

and M is the number of bidders. Only private values, of course.

With estimated private values $\hat{v}$ at hand, attached to each observation, the researcher should, in principle, be able to re-sample any target of interest: revenue, bidder surplus, strategy...

But after reading GPV carefully, you can spot a few problems down the road.

Problem 1

- GPV: *"... program is written in FORTRAN. For each replication, the execution time lasted less than one minute ..."*
- ME: *"On a modern computer, the same program that took 1 minute to run in 2000 could run anywhere from under a second to a few seconds in Fortran or C++. That is too slow, your bootstraps will take anywhere from a few hours to a few days. If you want to be serious in 2024, a single replication should be measured in microseconds or milliseconds."*

Problem 2

- GPV: *"... the number of potential bidders can vary across auctions. These considerations ... raise new technical difficulties ..."*
- ME: *"Every applied researcher would, of course, want to use auctions with the varying number of bidders, to absorb more data into the model."*

Problem 3

- GPV: *"... Each replication gives us two estimated functions: the equilibrium strategy and the estimated density function... "*
- ME: *"Who cares about the density or strategy? I want revenue; I want bidder surplus. These are nontrivial functionals of a density."*

Our goal

Our goal

We wanted to create an alternative estimator that is

- computationally friendly
- random-bidder friendly
- would somehow carry to as many targets of interest as possible: revenue, total surplus and bidder's surplus (estimation and inference, both pointwise and uniform)

How do we do it?

Step 0: quantiles

Let there be M bidders.

$$v = b + \frac{F^{M-1}(b)}{(M-1)F^{M-2}f(b)}, \quad \text{where } F, f \text{ are the cdf, pdf of bids}$$

making it linear in quantile density

$$v(u) = Q(u) + \frac{u^{M-1}q(u)}{(M-1)u^{M-2}}, \quad \text{where } Q, q \text{ are the cqdf, qdf of bids}$$

So $v(u)$ - the quantile function of valuations is linear in $q(u)$ - the quantile density of bids. You can sample from $v(u)$ by supplying a uniform r.v. as it's argument.

Step 1: quantiles + random bidders

Say, we have 2 and 3 bidders with equal probability $\frac{1}{2} : \frac{1}{2}$.

But subjectively, the bidder assigns beliefs $\frac{2}{5} : \frac{3}{5}$, because he gets sampled more often into large auctions than into small auctions.

Moving towards quantiles, we simplify the inverse elasticity formula...

$$v = b + \frac{\frac{2}{5}F(b) + \frac{3}{5}F^2(b)}{\frac{2}{5} + \frac{6}{5}F(b)} \frac{1}{f(b)}, \quad \text{where } F, f \text{ are the cdf, pdf of bids}$$

making it linear in quantile density

$$v(u) = Q(u) + \frac{\frac{2}{5}u + \frac{3}{5}u^2}{\frac{2}{5} + \frac{6}{5}u} q(u), \quad \text{where } Q, q \text{ are the cqdf, qdf of bids}$$

Failing to tune the beliefs properly will break revenue equivalence and lead to various paradoxes.

Step 2: estimation

Estimation is "putting hats" over all fs and qs. For example, $\hat{Q}$ is simply a sorted array of bids, and $\hat{q}$ is the smoothed spacings of the latter.

Here is how it looks like in python

```
import numpy as np
bids = [100, 250, 180, 300, 150, 220, 190, 400, 450]
sorted_bids = np.sort(bids)
spacings = np.diff(sorted_bids)
window_size = 3
kernel = np.ones(window_size)/window_size
smooth_incs = np.convolve(spacings, kernel, mode='same')
values = sorted_bids + smth_increments/(M-1)
```

Note that both sorting and convolving have a stunning $N \cdot log(N)$ computational complexity. Most importantly, the estimator is computed "at all points (bids) simultaneously". Boundary correction – as usual.

Step 3: inference

We are interested in uniform inference, which is hard because there is no limiting distribution for the $\max_u$ of the estimated density (either f or q).

To achieve this feat, we use something called Bahadur-Kiefer expansion and the theory of local empirical processes, see Rio (1994), Chernozhukov, Chetverikov & Kato (2014, 2016), to show that

$$\frac{\hat{q}_h(u)}{q(u)} - 1 \approx 1 - \hat{f}_h^U(u)$$

and so $\hat{q}_h(u)$ behaves a lot like $\hat{f}_h^U(u)$ – the KDE with uniform data – which can be efficiently simulated once per kernel and bandwidth h.

Step 3': more inference

Furthermore,

$$\hat{q}_h^U(u) - 1 \approx \frac{\hat{q}_h(u)}{q(u)} - 1 \approx 1 - \hat{f}_h^U(u)$$

and so $\hat{q}_h(u)$ behaves a lot like $\hat{q}_h^U(u)$ – the QDE with uniform data.

In practice, this means that, in order to assess the distribution of the estimator, you can use the exact same code that you were using for that estimator, but with uniform data.

This sounds similar to Bootstrap, but unlike Bootstrap, these uniform simulations can be almost tabulated (stored on a flash drive or the cloud) and reused robustly across various applications.

Step 4: towards functionals

It turns out that many functionals are linear in $v(u)$, thus they are linear in $q(u)$, thus we can push our asymptotic theory of $q(u)$ straight into those functionals, without any need for a "functional delta method".

$$TS = \int_{r^*}^{\bar{v}} x \, dG^M(x), \quad BS = \int_{r^*}^{\bar{v}} (1 - G(x)) G^{M-1}(x) \, dx$$

$$TS = \int_{u^*}^{1} v(u) \, du^M, \quad BS = \int_{u^*}^{1} (1 - u) u^{M-1} \, dv(u)$$

This is an alternative to bootstrap. Bootstrap was also shown to work in Zincenko, F, 2021, albeit only for revenue and only for constant bidders.

P.S. Not every functional is linear, and so we do not know how to simulate those - it is an open question.

Literature

Literature

Notable contributions to the literature:

- Guerre-Perrigne-Vuong, 2000, and Li-Perrigne-Vuong, 2000
- Ingraham 2005, Loertscher-Marx, 2020
- Luo-Wan, 2018, Enache-Florens, 2012
- Marmer et al 2012, 2019*, 2021
- Gimenes-Guerre 2022**
- Zincenko, 2021***

* has uniformity over covariates and counterfactual exclusion levels but only for the density itself (not targets such as revenue)

** has uniformity over covariates but not counterfactual exclusion levels.

*** has uniformity over exclusion levels but only for revenue

Thank you!