

This directory contains the code for the paper "Estimation of Beauty Contest Auctions" using synthetic data.

(1) Please ensure that the following tools are installed on your system:

- * g++ compiler

- * Stata: the code assumes a Mac OS X installation at the following location:
/Applications/Stata/StataMP.app/Contents/MacOS/stata-mp; if it is installed in a different location, modify line 204 in driver.cc with the appropriate command; you would also need to modify line 1 of synthdata.do to point to the directory where you unpacked the files.

- * Matlab: the code assumes that it is installed in /usr/bin/matlab; if it is installed in a different location, modify lines 114, 120 in sortkde3.cc with the appropriate command; you would also need to change lines 1, 2, 3, 4, and 6 of files cmds.m and cdfcmds.m to point to the directory where you unpacked the files.

(2) Compile the C++ code files using g++ by invoking the following commands.

```
g++ -o simulate3-clean -g -O3 simulate3-clean.cc
g++ -o sortkde3 -g -O3 sortkde3.cc
g++ -o driver -g -O3 driver.cc
```

(3) The synthetic data is stored in "synthdata.txt" and has the following attributes.

- each auction has exactly 10 bids (numbids)
- auctions are drawn from three different types at equal probability of about 0.33
- the buyer attributes are the following:
 - * number of ratings for the buyer (NumReps)
 - * sum of ratings for the buyer (SumRep)
 - * number of prior auctions cancelled by the buyer (PreviousCancels)
- the seller attributes are the following:
 - * the bid value (BidAmount)
 - * the average reputation which is either 8, 9, or 10 (SellRep)

"synthdata.txt" contains 4002 auctions with 11 lines per each auction: the first line provides just the auction and buyer specific attributes (used for the top-level nest choice of no

bid being chosen) and 10 other lines for the 10 bids associated with the auction. The file contains 14 fields per line, which are tab separated. The ordering of the fields in a line is as follows:

BidRequestId: id for the auction

DecisionNo: 1 through 11; 1 is for no bids being selected, 2–11 for bids

Decision: 0 or 1 corresponding to the choice made for the auction

OverallDecision: 1 if no bids were selected, 2 if some bid was selected

SumRep: sum of all reputation values for the buyer

NumReps: number of reputations for the buyer

PreviousBids: number of prior auctions made by the buyer

PreviousCancels: number of prior auctions canceled for the buyer

numbids: number of bids for the auction (10)

ttype: true type used for the data generation (0...2)

avbid: average of all bids for the auction

cost: true cost associated with the seller

SellRep: seller's reputation value (8 or 9 or 10)

BidAmount: bid made by the seller

The synthetic data consists of auctions of the three different types with roughly equal proportion. The costs for the bidders for the three types are distributed as follows:

- for low type auctions, the costs for the bidders are uniformly distributed over [100:200]
- for medium type auctions, the costs for the bidders are uniformly distributed over [150:250]
- for high type auctions, the costs for the bidders are uniformly distributed over [150:250]

The SellRep value is randomly and uniformly distributed across the three values [8,9,10].

(4) Invoke the driver code using the following command line:
`"./driver synthdata.txt"`

The code `driver.cc` is the top-level file that performs the following:

- * In each step, it generates a sample of the auctions with replacement which will be used for a single instance of the Monte Carlo simulation. The generated file is stored in `"sampled-aucs.txt"`. This file contains the same set of fields as `"synthdata.txt"`.
- * It then runs `"./sortkde3 sampled-aucs.txt synthdata-weights2.txt"` to separate the `bidAmount` mixture distribution into the constituent distributions. The output is `synthdata-weights2.txt` that contains two posterior probabilities corresponding to the probability of an auction being high type (`aucWt`) or medium type (`aucWt2`). These values are stored as two additional fields along with each line in addition to the fields that were in `sampled-aucs.txt`
- * It then invokes a Stata do file `"synthdata.do"` to obtain the estimates from a nested logit. The output of the estimation is stored in `coeff.txt`
- * It then invokes `"simulated3-clean"` to estimate the underlying cost for a seller given the observed bid for each of the three different possible types for the auction. The output file `"synthdata-costs*.txt"` contains six additional fields, which are the derived costs and margins for each of the three different types. More specifically, fields 17, 19, and 21 are the derived costs for the three different auction types.
- * It then accumulates the cost distributions taking into account the posteriors into its local data structure

The driver code is currently setup to run 100 iterations, which takes about 6 to 8 hours on a modestly provisioned 4-core machine.

It then outputs three files `"costs-dist*.txt"` which contains the percentiles for the three cost distributions from each iteration of the Monte Carlo simulation. At the tail of each file are summary statistics for each percentile of the cost distribution with the `kth` line containing the following five values:

- * mean of the `kth` percentile across all runs
- * 5th percentile of the `kth` percentile across all runs
- * 95th percentile of the `kth` percentile across all runs

- * 50th percentile of the kth percentile across all runs
- * variance of the kth percentile across all runs