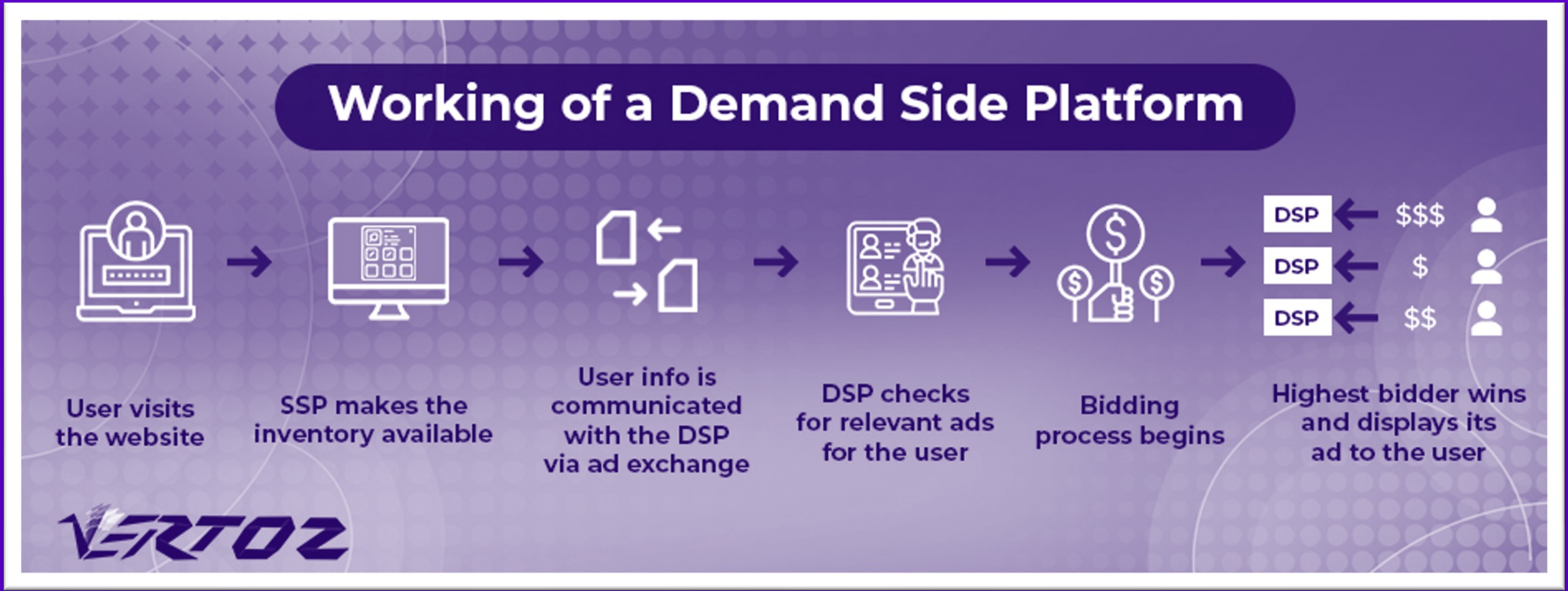
Double Distributionally Robust Bid Shading for First Price Auctions

Yanlin Qu
Columbia Business School
qu.yanlin@columbia.edu

Joint work with Ravi Kant, Yan Chen, Brendan Kitts, San Gultekin, Aaron Flores, Jose Blanchet

yahoo! dsp



yahoo!

https://vertoz.com/what-is-a-demand-side-platform-dsp/

Bigabid

First-Price Auction

Profit from shading

Second-Price Auction

Always bid truthfully

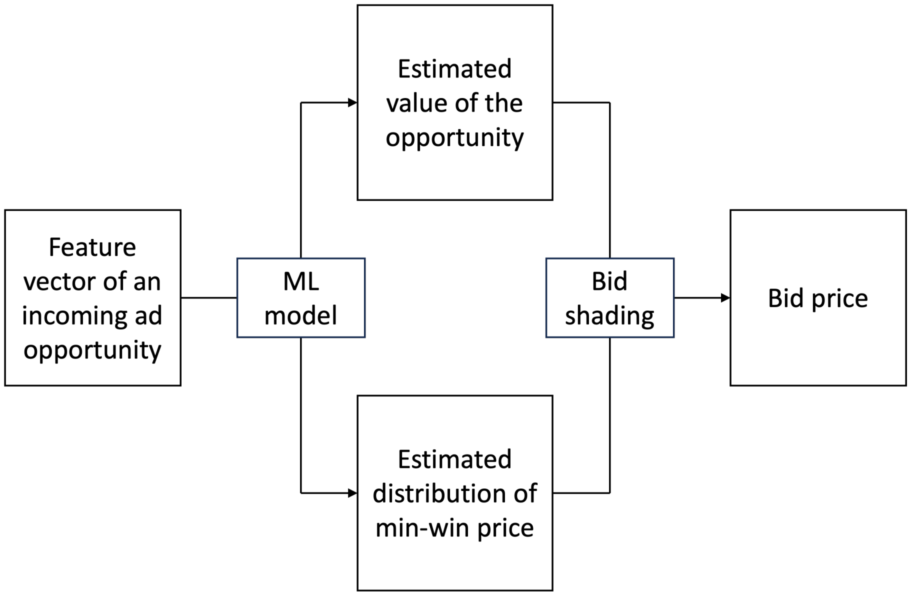
http://www.bigabid.com/first-price-vs-second-price-auctions/

Overview

1. Motivation
    - Noisy real-time bidding system
    - Distributionally robust optimization
2. Distributionally Robust Bid Shading
    - Problem formulation and computable formula
    - Theoretical insight and efficient implementation
3. Experiment
    - Dataset, metric, and spend equating
    - Outperformance and interpretation

Motivation

- Prediction phase (left)
    - Value & min-win price as output
    - Complex nature + latency constraint = very noisy estimate
- Bid shading phase (right)
    - Value & min-win price as input
    - Robustness against relatively significant estimation errors
    - Distributionally robust optimization (DRO)



yahoo!

Distributionally Robust Bid Shading – Problem Formulation

- Realized value V with distribution $\overline{P}$
- Min-win price X with distribution $\overline{Q}$
- Choose bid price b to maximize the expected surplus (baseline)
- Estimates $\overline{P}$ and $\overline{Q}$ tend to be noisy
- Introduce ambiguity sets for them
- Choose bid price b to maximize the worst-case expected surplus (DRBS)

$$\max_{b} \mathbb{E}_{\bar{P},\bar{Q}}(V-b)I(X \le b)$$

$$\Downarrow$$

$$\max_{b\ge 0} \min_{\substack{P\in\mathcal{P}(\delta_V)\\ Q\in\mathcal{Q}(\delta_X)}} \mathbb{E}_{P,Q}(V-b)I(X \le b)$$

$$\mathcal{P}(\delta_V) = \{P : D(P||\bar{P}) \le \delta_V\}$$

$$\mathcal{Q}(\delta_X) = \{Q : D(Q||\bar{Q}) \le \delta_X\}$$

$$D(p_1||p_2) = \mathbb{E}_{p_1}\log(p_1(Z)/p_2(X))$$

yahoo!

Distributionally Robust Bid Shading – Computable Formula

The double DRO problem turns out to be almost analytically solvable.

Notations: click probability $\bar{p} = P_{\bar{p}}(V > 0)$, click reward a, value $v = \mathbb{E}V = a\bar{p}$, worst-case value $\bar{v} = ar^{-1}(\delta_V)$, $r(p) = p\log(p/\bar{p}) + (1-p)\log((1-p)/(1-\bar{p}))$, worst-case baseline policy $\underline{v}$, CDF and PDF of X under $\bar{Q}$: F and f.

Assumptions: V and X are independent; $\delta_V < r(0)$ and $\delta_X < -\log(1-F(\bar{v}))$; F is log-concave; $F(\bar{v}) < 1/2$ and $F(0) = 0$.

DRBS policy b^*: the unique solution of $g(b) = \delta_X$ in $[\underline{v}, \bar{v}]$ where

$$g(b) = \log\eta(b) - \log J(b) - \frac{F(b)\log\eta(b)}{J(b)},$$

$$J(b) = F(b) + \eta(b) - F(b)\eta(b), \quad \eta(b) = h^{-1}(L(b))$$

$$L(b) = \frac{F(b)}{(\bar{v} - b)f(b)}, \quad h(x) = \frac{x-1}{\log x}, \quad x \geq 0.$$

Distributionally Robust Bid Shading – Theoretical Insight

$$\textit{DRBS is increasing in } \delta_X \textit{ but decreasing in } \delta_V.$$

- When we are uncertain about the competitive landscape ($\delta_X > 0$), the competition is fiercer than expected in the worst case, so we should bid higher than the baseline to maintain our win rate.
- When we are uncertain about the value ($\delta_V > 0$), the ad opportunity is less valuable than expected in the worst case, so we should bid lower than the base line to maintain a positive profit margin.
- One KL-ball is not enough. The reduced DRBS either always bids higher or always bids lower.

$$\textit{DRBS bids higher (or lower) than the baseline when } v \textit{ is large (or small) enough.}$$

- When the value is oddly high, why not bid aggressively to secure the deal?
- When the value is oddly low, why not bid conservatively to avoid "winning a loss"?
- Two KL-balls are essential. DRBS with $\delta_X, \delta_V > 0$ can reasonably decide to bid higher or lower.

yahoo!

Distributionally Robust Bid Shading – Efficient Implementation

DRBS policy b^*: the unique solution of $g(b) = \delta_X$ in $[\underline{v}, \bar{v}]$ where

$$g(b) = \log \eta(b) - \log J(b) - \frac{F(b) \log \eta(b)}{J(b)},$$

$$J(b) = F(b) + \eta(b) - F(b)\eta(b), \quad \eta(b) = h^{-1}(L(b))$$

$$L(b) = \frac{F(b)}{(\bar{v} - b) f(b)}, \quad h(x) = \frac{x-1}{\log x}, \quad x \geq 0.$$

- Both $\underline{v}$ and $\overline{v}$ are easy to compute.
- Since g is strictly increasing in $[\underline{v}, \overline{v}]$, b^* can be computed via bisection.
- For $y > 1$, $h^{-1}(y) = -y \cdot W_{-1}(-e^{-1/y}/y)$, which can be computed via scipy.special.lambertw.
- $j_{2/3}(y) \leq h^{-1}(y) \leq j_1(y)$, $j_c(y) = (1 + \sqrt{2} \cdot l(y) + c \cdot l^2(y))$, $l(y) = \sqrt{1/y + \log y - 1}$.

yahoo!

Experiment - Dataset

- Yahoo DSP private bidding dataset on Google Ad Exchange
- Information of 2M bid requests
- More than 1K lines (campaigns)
- Baseline: log-normal model ($\overline{Q}$)
- No private values in public datasets

Field	Description
line_id	ID of the line corresponding to the ad that the DSP wants to win the opportunity for
ceiling, floor	Range of allowed bid prices
mu, sigma	Two estimated lognormal parameters of the distribution of the minimum winning price
click_prob	Estimated probability of the ad being clicked
click_reward	Reward if the ad is actually clicked
value	Product of the above two
min_win_price	Actual minimum winning price

yahoo!

Experiment - Metric

- Value v, bid price b, min-win price X, effective value per dollar spent

$$R = \frac{\sum_i v_i I(X_i \leq b_i)}{\sum_i b_i I(X_i \leq b_i)}$$

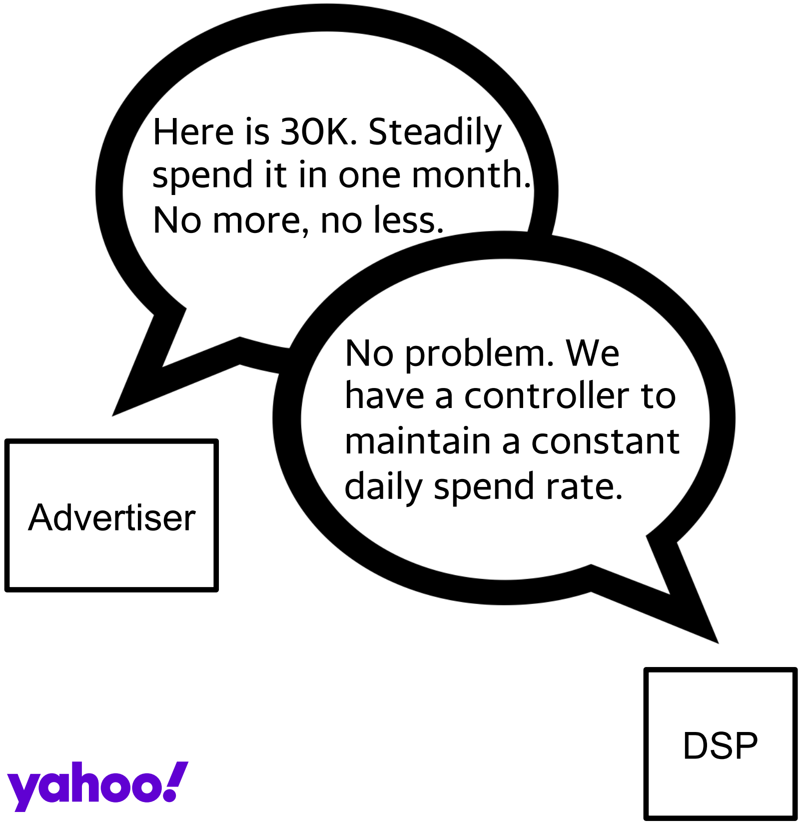
- Line k, DRBS R_k^D, baseline R_k^B, percentage improvement

$$\Delta R_k = (R_k^D / R_k^B - 1) \times 100\%$$

- Spend s_k, spend-weighted average over all lines

$$\Delta R = \frac{\sum_k s_k \Delta R_k}{\sum_k s_k}$$

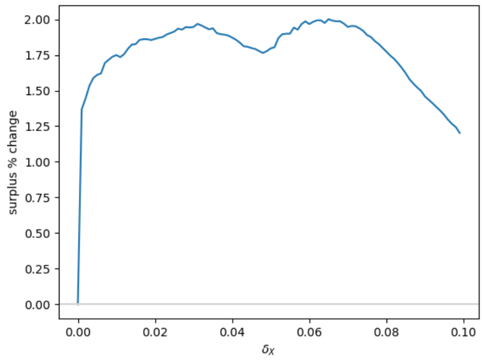
Experiment - Spend Equating

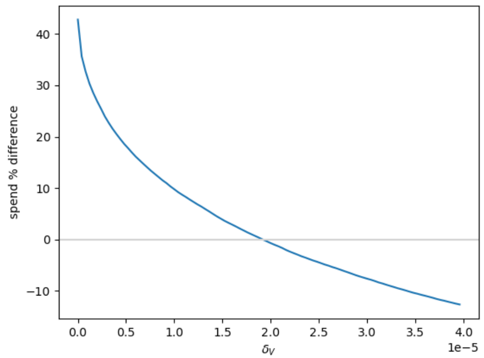


- Spend equating $s_k^D = s_k^B$ is crucial to make offline comparison meaningful in online sense.
- Regulated by the controller (v-modifier), different policies spend the same on average.
- To mimic the controller offline, a standard practice is uniformly modifying v's for the new policy to make it spend the same as the old one.
- This modification violates the problem formulation where DRBS and baseline are facing the same set of v's.

Experiment - Spend Equating via Delta Balancing

- $b_i^*(\delta_X, 0) > b_i^*(0{,}0)$, $b_i^*(0, \delta_V) < b_i^*(0{,}0)$
- Can we make $b_i^*(\delta_X, \delta_V) = b_i^*(0{,}0)$ on average?
- First, choose δ_X to maximize the total surplus $\sum_i \big(v_i - b_i^*(\delta_X, 0)\big) I(X_i \le b_i^*(\delta_X, 0))$.
- Second, choose δ_V to equate the spend $\sum_i b_i^*(\delta_X, \delta_V) = \sum_i b_i^*(0{,}0)$.
- The resulting DRBS policy handles two sources of uncertainty while maintaining the same spend rate as the baseline policy.
- Two KL-balls are essential.





yahoo!

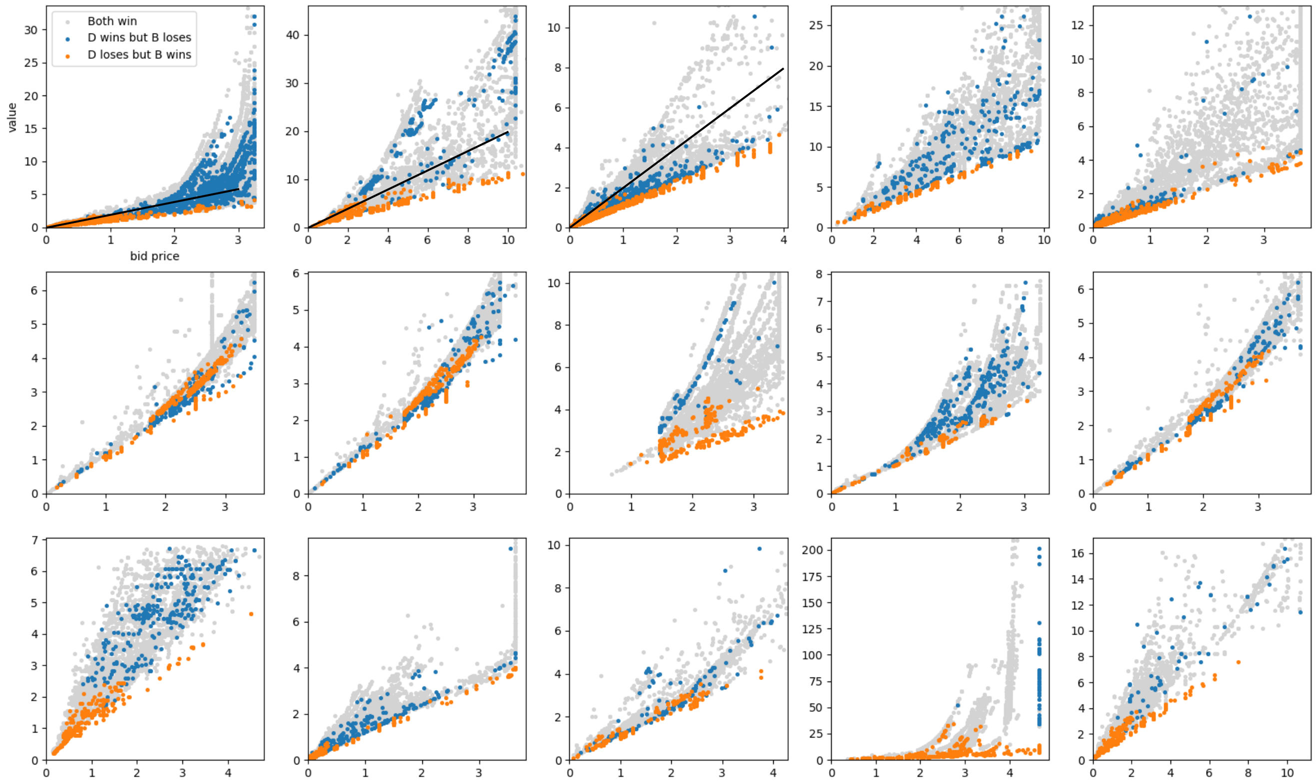
Experiment - Outperformance and Interpretation

- For each line, we use 25% of the data to compute δ_X and δ_V while the rest 75% is used for testing.
- For the 3 largest lines with spend weights 8.6%, 8.6%, 6.6%, $\Delta R_1 = 1.0\%$, $\Delta R_2 = 2.3\%$, $\Delta R_3 = 0.0\%$.
- For all lines, the spend-weighted average $\Delta R = 0.65\%$. Where does the gain come from?
- Exchange low-v/b wins for high-v/b wins.

X	b^B	b^D	v	v/b
2.47	2.29	2.49*	6.64	2.67
0.59	0.58	0.60*	0.89	1.48
2.09	2.00	2.20*	6.83	3.10
2.96	2.93	3.11*	10.25	3.30
1.78	1.77	1.86*	4.22	2.27

X	b^B	b^D	v	v/b
0.13	0.13*	0.00	0.25	1.92
0.13	0.13*	0.00	0.25	1.92
0.34	0.34*	0.00	0.40	1.18
0.32	0.33*	0.00	0.43	1.30
0.24	0.24*	0.24	0.39	1.63

Experiment - Outperformance and Interpretation



yahoo!

Takeaway

- Real-time bidding algorithm needs to be robust.
- Double distributionally robust optimization works.
- Two KL-balls are essential (for spend equating).
- DRBS policy has computable formula and interpretable outperformance.

Thank You

https://quyanlin.github.io