Advancing Ad Auction Realism: Practical Insights and Modeling Implications

Ming Chen (Outreach Inc)
Sareh Nabi (Amazon Ads)
Marciano Siniscalchi (Amazon Ads & Northwestern U)
Ad Auctions
From Theory to Practice

- Second-price auction
From Theory to Practice

• Second-price auction
• Generalized – click-through rate (CTR)
From Theory to Practice

- Second-price auction
- Generalized – click-through rate (CTR)
- Irrelevance Penalties
From Theory to Practice

• Second-price auction
• Generalized – click-through rate (CTR)
• Irrelevance Penalty
• Hard floor
From Theory to Practice

- Second-price auction
- Generalized – click-through rate (CTR)
- Irrelevance Penalty
- Hard floor
- Soft floor
From Theory to Practice

- Second-price auction
- Generalized – click-through rate (CTR)
- Irrelevance Penalties
- Hard floor
- Soft floor
- Limited feedback to guide bidding
From Theory to Practice

- Second-price auction
- Generalized – click-through rate (CTR)
- Irrelevance Penalty
- Hard floor
- Soft floor
- Limited feedback to guide bidding
  - Targeting clauses
From Theory to Practice

- Second-price auction
- Generalized – click-through rate (CTR)
- Irrelevance Penalty
- Hard floor
- Soft floor
- Limited feedback to guide bidding
- Targeting clauses
- ...

Standard equilibrium analysis is not feasible
The Auction Simulator

Objective:
Build a flexible tool to simulate the strategic behavior of advertisers in complex ad auctions.

Requirements:
- **Flexible**
  Allow arbitrary ranking and pricing rules, heterogeneous bidders, multiple ad slots...

- **Strategic**
  Focus on how the interaction among bidders determines prices, allocations, predicted clicks / conversions...

- **Complex Auctions**
  This is not Auctions 101 anymore... bids target multiple queries, compete in multiple auctions, with different competitors, and only aggregate feedback...
Model: High-level view

- **Inputs:**
  - Distribution $F_i$ of bidders’ “types,” i.e.:
    - willingness to pay (per click) $v_i$
    - Click-through rates $c_i$
  - Pricing rule $P(b_1, \ldots, b_N; \ldots)$
  - Possible shopper queries

- **Simulation:**
  - Draw bids $b_i$ (and targeting clauses later)
  - Compute price $p = P(b_1, \ldots, b_N; \ldots)$
  - Observe rewards: 0 or $c_i(v_i - p_i)$
  - Update bid probabilities

- **Outputs:**
  - Bid Distribution
  - KPIs: revenues, cost per click, conversion rates...

- **A collection of principled learning algorithms**
  - Game Theory: Stochastic Fictitious Play
  - Online / Reinforcement Learning: Hedge, EXP3IX...
Application: Exploring Soft Floors

Soft floors switch auction to first-price if winning bid too low

Zeithammer (2019): BNE analysis, partial results
• with symmetric bidders, soft floors ineffective
  • Equilibrium + continuum of bids/values: Revenue Equivalence
• with asymmetric bidders, some special cases:
  • stochastically stronger bidders: soft floors can lift revenues for some param values
  • deterministically stronger bidders (e.g., major brand):
    • low soft floors do not lift, can depress revenues
    • intermediate / high soft floors: unknown effect
**Soft floors:**
**Keywords and Queries**
Injecting realism, one complication at a time

- Advertisers bid on **keywords** (i.e., targeting clauses)
- User queries are **matched** to relevant keywords
- Ex: keyword **shower curtain** may match with
  - snap on shower curtain with liner
  - blue shower curtains for bathrooms
  - vw van shower curtain for bathroom
  - shower curtain liner mold
- These have **different estimated CTRs**
- And presumably **different values** to the bidder

Our model: targeting clause = set of queries to match
Soft Floors: A New Rationale

- Explore example with 2 queries
- Let $N = 3$, equally likely queries, values and CTRs as follows

<table>
<thead>
<tr>
<th>$F(.)$</th>
<th>$v_{i,1}$</th>
<th>$c_{i,1}$</th>
<th>$v_{i,2}$</th>
<th>$c_{i,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3</td>
<td>0.5</td>
<td>0.3</td>
<td>0.25</td>
<td>0.1</td>
</tr>
<tr>
<td>1/3</td>
<td>0.25</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1/3</td>
<td>0.25</td>
<td>0.1</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Soft Floors: A New Rationale?

Revenue Equivalence does not hold
Different algorithms give different answers

<table>
<thead>
<tr>
<th>Format</th>
<th>Revenues – Hedge</th>
<th>Revenues – EXP3IX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd Price</td>
<td>0.0857 (0.0001)</td>
<td>0.0509 (0.0007)</td>
</tr>
<tr>
<td>1st Price</td>
<td>0.0691 (0.0016)</td>
<td>0.0830 (0.0008)</td>
</tr>
<tr>
<td>2nd Price w/50c soft floor</td>
<td>0.0741 (0.0061)</td>
<td>0.0813 (0.0007)</td>
</tr>
</tbody>
</table>

- Bids: [0,1], step size 0.05
- Learning periods $T = 500,000$ (Hedge) or $T = 1M$ (EXP3IX)
- 5 runs per experiment (stdevs in parens)
- No revenue equivalence: soft floors may beat 2nd-price
- Different implications of learning algorithms (more later…)
- Note: did not optimize “standard” reserve prices (“hard floors”)
Application: Hedge vs. EXP3IX
Second-Price Auction, Hedge
Application: Hedge vs. EXP3IX
Second-Price Auction, EXP3IX
Key takeaways

- The choice of algorithm matters
- Bandit (e.g. EXP3IX) algorithms learn *way* more slowly
  - in realistic settings
- Yet they are more principled: better fit with observational reality
- Hedge as compromise?
Application: Inferring Values from Bids

- Scenario: observe **aggregate** bid distribution
- Want to **infer advertisers’ values** (willingness to pay)
- (At this level, can (only) take CTRs to be the same for all)
- With standard auction formats:
  - Second-price: bids = values
  - First price: invert equilibrium bids (Guerre, Perrigne, Vuong, 2000)
- But what about real-world auctions?
  - Cannot solve for equilibrium!
- We propose to: **simulate and iterate**
Low-traffic keyword

5 iterations
Assuming different pricing rules
High-traffic keyword

8 iterations, \( T=800,000 \)
Assuming “realistic” pricing rule
Conclusions

- Simulate Advertisers’ Strategic Behavior
- Principled learning algorithms
- Can be used to
  - Perform “what if” analysis
  - Infer advertisers’ willingness to pay
  - And more!
Questions?
Thank you
Appendix
The Model – single query

(Multiple queries: later)

- $N$ advertisers
- Bidding to show an ad for a given shopper query in a given slot
- Bidder $i$ characterized by value per click $v_i \in [0, \bar{V}]$, CTR $c_i \in [0,1]$
  - $(v_i, c_i)$ is $i$’s type
  - Drawn according to cdf $F_i$
- “Cost per click:” winner is charged only if the ad is clicked
- Hence expected payoff for winner $i$, given charged price $p$, is

$$c_i \cdot (v_i - p)$$
Generalized Second-Price Auction

- Common for ad auctions (often, with tweaks)
- Given bids \(b_1, ..., b_N\) and CTRs \(c_1, ..., c_N\):
  - Compute ranking scores \(r_i = c_i \cdot b_i\)
  - Winner is \(i\) with highest ranking score: \(i \in \text{argmax}_k r_k\)
  - Runner-up is \(j\) with second-highest score: \(j \in \text{argmax}_{k \neq i} r_k\)
  - Price per click is “performance-adjusted”:
    \[
    p = \frac{r_j}{c_i}
    \]
  - Intuition: minimum \(b_i\) such that \(i\) still wins (Vickrey, Myerson)
- In practice, add “floors,” “irrelevance penalty”...
What advertisers really see

Bids compete in many auctions (“campaign”)
Feedback aggregated over all auctions
Learning: Experts/Bandits Approach

e.g. Freund-Schapire (1999); Auer, Cesa-Bianchi, Freund, Schapire (1995); Kocák et al. (2014); Lattimore and Szepesvári (2020)

- $T$ periods: at each $t$,
  - Fresh draw of $(v_i, c_i)$
  - Auction is run, payoffs accrue
- Bidders only observe their own rewards
  - “experts” approach (e.g., Hedge): learn payoff of all bids
  - “bandits” approach (e.g. EXP3IX): learn payoff of bid actually played
- At each $t$, play bid with highest cumulative reward so far, with perturbation
- Not strategically or statistically sophisticated
  - Generic: need not know auction rules, own WTP/CTR!
  - Good fit for online ad auctions
- Finite-sample regret guarantee vs. best action in hindsight
Results: Soft-Floor Reserve Pricing

- (For simplicity, set all CTRs to a constant, e.g., 1)
- Idea: “price support” / “insurance”
  - “the goal is to ‘harvest’ higher bids while not compromising on lower bid opportunities” (Weatherman 2013).
- Fix a soft floor $s \in [0, \bar{V}]$
- Let $b_i$ be the highest bid, $b_j$ the runner-up
- Then price $p$ is as follows:
  - If $b_j \geq s$, then second-price rule: $p = b_j$
  - If $b_i \geq s > b_j$, then $s$ acts as floor: $p = s$
  - If $s > b_i$, then first-price: $p = b_i$
The Model – multiple queries

- $Q$ possible queries
- In each period, probability over queries $G$
- Bidder $i$’s values and CTRs depend on the query: $v_{i,q}, c_{i,q}$
- So now cdf $F_i$ on tuples $(v_{i,1}, c_{i,1}, ..., v_{i,Q}, c_{i,Q})$
- Each bidder now chooses
  - A bid $b_i$
  - A keyword, identified with the queries that it matches: $K_i \subset \{1, ..., Q\}$
- Key restriction: same bid $b_i$ for all queries in $K_i$
- Expected payoff for winner $i$, given prices per query $p_q$
  $$\sum_{q \in K_i} G(q) \cdot 1_{i \text{ wins } q} \cdot c_{i,q} (v_{i,q} - p_q)$$
Inferring Values

- Data: aggregate bid data
  - E-commerce website
  - Two queries: low traffic, high traffic
- Approach:
  1. To initialize, assume values equal observed bids: \( v = b^o \)
  2. Run Auction Simulator, compute predicted bids \( b^p \) for every value \( v \)
  3. Adjust values:
     1. Compute predicted bid shading: \( \sigma = \frac{b^p}{v} \)
     2. Infer value: \( v \leftarrow v + \alpha \left( \frac{b^o}{\sigma} - v \right) \) plus “flattening” for monotonicity
  4. Go to 2 until termination
- Each iteration: run 3x, \( T = 500,000 \) learning periods,