Learning to Bid Without Knowing your Value

Zhe Feng, Chara Podimata and Vasilis Syrgkanis

Harvard University, Harvard University, MSR New England

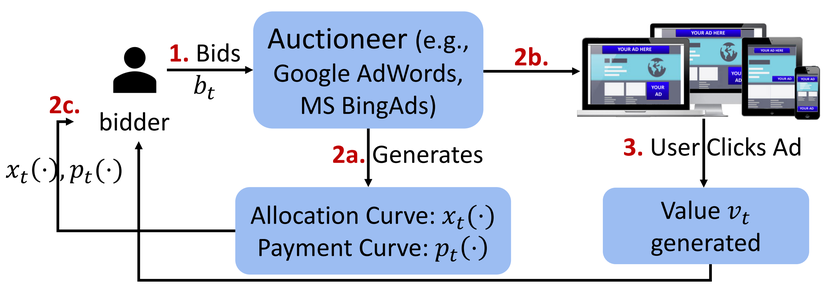
Motivation

Standard **Assumption** in Mechanism Design/ Game Theory:
Bidder **knows** her **value** for an item, **before** the **auction**

✓ Small markets, infrequent transactions

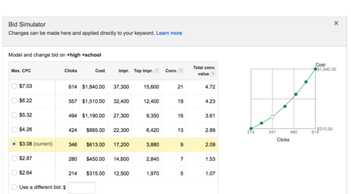
✗ Online ad-auctions

Online ad-auctions Flowchart

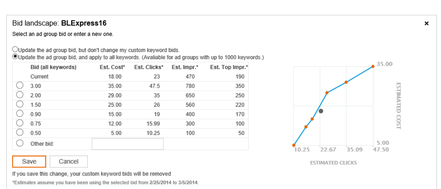


5. Bidder's Utility: $u_t(b_t) = \left(v_t - p_t(b_t)\right) \cdot x_t(b_t)$

Allocation and Payment Curves in Real Life Ad Auctions



Google AdWords



Microsoft BingAds

Main Question Addressed

How can the bidder learn how to bid without knowing their value a priori?

Use No-Regret Learning!

Minimize: $\mathrm{R(T)} = \sup_{b^*} E[\sum_{t=1}^{T} u_t(b^*)] - E\,[\sum_{t=1}^{T} u_t(b_t)]$

Main Result

First Thought:

- Uniform ϵ-Discretization of bidding space: $B = [1/\epsilon]$.
- Apply off-the-shelf EXP3 for the discretized space.
- Regret: $R(T) \leq \mathcal{O}(\sqrt{\mathrm{TB}})$.
- **Careful!** Utility **not** Lipschitz!

We can do better than that! Instead of $R(T) \leq \mathcal{O}(\sqrt{\mathrm{T}|\boldsymbol{B}|})$, achieve regret of the form: $R(T) \leq \mathcal{O}(\sqrt{T\log(|\boldsymbol{B}|)})$.

Intuition: Use extra information provided by allocation and payment curves in real life ad auctions!

WIN-EXP Algorithm (Win-Only Feedback)

- Distribution over bidding space: $\pi_t(b), b \in B = \left[\frac{1}{\epsilon}\right]$
- At each round $t \in [T]$:
  - Draw bid $b_t \sim \pi_t$
  - Observe allocation $x_t(b), \forall b \in B$
  - If you win (e.g., get clicked), observe value, compute reward: $r_t(b) = v_t - p_t(b), \forall b \in B$ and estimate:
  $$\tilde{u}_t(b) = \frac{(r_t(b) - 1)x_t(b)}{\sum_{b'} x_t(b')\,\pi_t(b')}, \forall b \in B$$
  - Else, estimate: $\tilde{u}_t(b) = -\frac{1 - x_t(b)}{1 - \sum_{b'} x_t(b')\pi_t(b')}, \forall b \in B$
- Update: $\pi_{t+1}(b) \propto \pi_t(b)\exp\left(\eta\,\tilde{u}_t(b)\right), \forall b \in B$

Regret

Theorem. Regret of WINEXP algorithm with win-only feedback is: $R(T) \leq \mathcal{O}(\sqrt{T\log(|\boldsymbol{B}|)})$.

WINEXP Algorithm can be extended to **outcome-based** feedback!

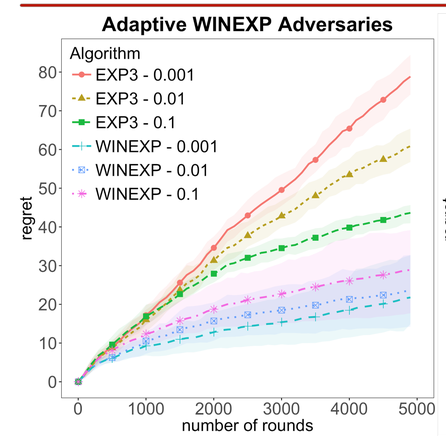
- Set of outcomes: O
- Reward: $r_t: B \times O \to [-1,1]$, allocation: $x_t: B \to \Delta(O)$
- Bidder's Utility: $u_t(b) = \sum_{o \in O} r_t(b,o)x_t(b,o)]$
- Regret: $R(T) \leq \mathcal{O}(\sqrt{T\,|\boldsymbol{O}|\log(|\boldsymbol{B}|)})$

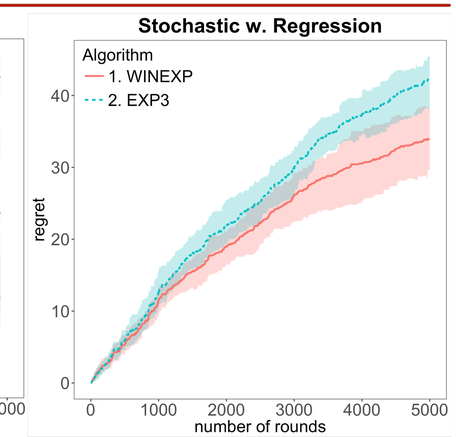
Applications to Learning in Auctions

Auctions have win-only/outcome-based feedback structure!

- Value-per-Click Auctions: $R(T) \leq \mathcal{O}(\sqrt{2T\log(|\boldsymbol{B}|)})$.
- Multi-Slot Auctions & Unit Demand Auctions (K items): $R(T) \leq \mathcal{O}(\sqrt{(K+1)T\log(|\boldsymbol{B}|)})$.

Experiments





Extensions

- **[Continuous Action Spaces]** Proof that Discretization Error of Δ^0-Piecewise Lipschitz utilities is 0 → WINEXP Algo Regret:
$$R(T) \leq \mathcal{O}\left(\sqrt{T d \log\left(\max\left\{\frac{1}{\Delta^0}, T\right\}\right)}\right)$$
- Applications: Sponsored Search, First-Pay, All-Pay auctions.

Open Questions

1) In what other real-life settings (apart from auctions) does win-only feedback appear?
2) Proof that WINEXP estimates are robust to noise up to a certain degree (experimental results suggest so).
3) Proof that WINEXP estimates can be reconstructed in some cases with fully-bandit feedback (experimental results suggest so too, in some cases of interest).