

Bidder Selection Problem in Position Auctions:

A Fast and Simple Algorithm via Poisson Approximation



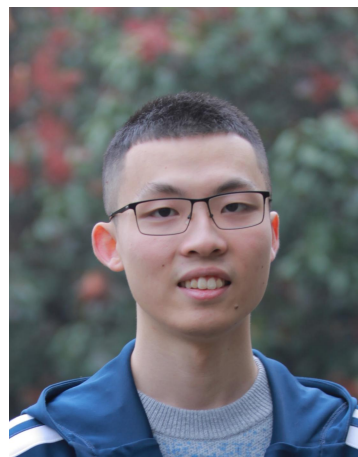
Nick Gravin

ITCS, Shanghai University of
Finance and Economics



Yixuan Even Xu

IIS, Tsinghua University

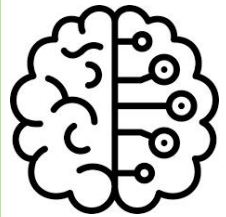


Renfei Zhou

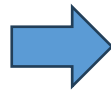
IIS, Tsinghua University

Bidder Selection Problem

ticket discount 



Large ML model:
accurate & **slow**



auction scores:

5.1

4.56

⋮

1.2

impression context,
ad relevance,
CTR estimates, bids



k ads (e.g. 50)

Filter out: $k < n$

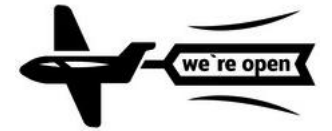


n ads

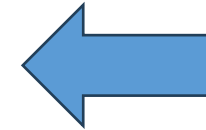


Small ML model:
rough & **fast**

n Advertisers



Bids




Theoretical Model

k -max problem (single-item auction):

- INPUT: n independent random variables X_1, \dots, X_n and $k < n$
- OUTPUT: find k r.v. $S \subset [n]$ to $\max_{|S|=k} \mathbf{E} \left[\max_{i \in S} X_i \right]$

k -max for position auctions:

Advertiser I	\$2.00		10 	20
Advertiser II	\$4.00		4 	16
Advertiser III	\$6.00		2 	12
Advertiser IV	\$8.00		1 	8



- generalizes single-item auction
- INPUT: r.v. X_1, \dots, X_n and $k < n$
- OUTPUT: k r.v. $\max_{|S|=k} \mathbf{E} [Welfare(S)]$

Our Results: Theory

New relaxation of k -max: $\max_{\|x\|_1 \leq k} f(x)$

1. $f(x)$ – nicely structured concave function \Rightarrow

Easy to find

$$x^* = \operatorname{argmax}_{\|x\|_1 \leq k} f(x)$$

2. $f(x^*) \rightarrow \max_{|S|=k} \mathbf{E} \left[\max_{i \in S} X_i \right]$ as k grows

3. Works for general position auctions.

Position auctions:
Implementable PTAS

Our Results: Practice

- Homebrew implementation (python + standard convex libs):

- Benchmarks: Greedy (Submodular Opt.) + Local Search

Runtime: 1 **day**

- Instances as large as $n = 1000$, $k = 200$.

Runtime: 45 **sec.**

- Good approximation on all instances.

Approx. to benchmarks: >99%

- Previous 2 EPTAS algorithms on k -max:

- **Unimplementable**

Runtime: years $\left(2^{O(1/\varepsilon)^{O(1/\varepsilon)}}\right)$

$n=3, k=2, \varepsilon=0.2$