Contextual Reserve Price Optimization in Auctions via Mixed-Integer Programming

Kamil Kropiewnicki

Institute of Theoretical Computer Science
Jagiellonian University

25.02.2021

Overview

Algorithmic Game Theory background

Algorithmic Game Theory background

- Sealed-bid auction
  1. Each bidder i privately communicates a bid b_i to the seller - in a sealed envelope, if you like.
  2. The seller decides who gets the item (if anyone).
  3. The seller decides on a selling price.

Algorithmic Game Theory background

- Sealed-bid auction
  1. Each bidder i privately communicates a bid b_i to the seller - in a sealed envelope, if you like.
  2. The seller decides who gets the item (if anyone).
  3. The seller decides on a selling price.
- First-price auction

Algorithmic Game Theory background

- Sealed-bid auction
  1. Each bidder i privately communicates a bid b_i to the seller - in a sealed envelope, if you like.
  2. The seller decides who gets the item (if anyone).
  3. The seller decides on a selling price.
- First-price auction
- Second-price auction

Algorithmic Game Theory background

- Single-parameter environment

Algorithmic Game Theory background

- Single-parameter environment
- Sponsored Search Auctions
  CTR - click-through rate:
  The CTR α_j of a slot j represents the probability that the end user clicks on this slot. Ordering the slots from top to bottom, we make the reasonable assumption that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$.

Algorithmic Game Theory background

- Single-parameter environment
- Sponsored Search Auctions
  CTR - click-through rate:
  The CTR α_j of a slot j represents the probability that the end user clicks on this slot. Ordering the slots from top to bottom, we make the reasonable assumption that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$.
- Maximizing expected revenue:

$$\mathop{\mathbb{E}}_{v \sim \sigma}[\sum_i p_i(v)] = \mathop{\mathbb{E}}_{v \sim \sigma}[\sum_i \varphi_i(v) x_i(v)]$$

Algorithmic Game Theory background

- Single-parameter environment
- Sponsored Search Auctions
  CTR - click-through rate:
  The CTR α_j of a slot j represents the probability that the end user clicks on this slot. Ordering the slots from top to bottom, we make the reasonable assumption that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$.
- Maximizing expected revenue:

$$\mathop{\mathbb{E}}_{v\sim\sigma}[\sum_i p_i(v)] = \mathop{\mathbb{E}}_{v\sim\sigma}[\sum_i \varphi_i(v)x_i(v)]$$

- Reserve price (in an example below, $r_i \geq \varphi_i^{-1}(0)$)

Applying the mentioned approach to the real world setting

Applying the mentioned approach to the real world setting

- The previous approach utilizes $\varphi_i(v)$ that depends on σ_i . How to discover σ_i then?

Applying the mentioned approach to the real world setting

- The previous approach utilizes $\varphi_i(v)$ that depends on σ_i . How to discover σ_i then?
- Associate a set of features (contextual information) with each auction, such as:
  - publisher data (e.g. ad site and ad size)
  - user data (e.g.device type and various geographic information)
  - time (e.g. date and hour)
  - ...

Applying the mentioned approach to the real world setting

- The previous approach utilizes $\varphi_i(v)$ that depends on σ_i . How to discover σ_i then?
- Associate a set of features (contextual information) with each auction, such as:
  - publisher data (e.g. ad site and ad size)
  - user data (e.g.device type and various geographic information)
  - time (e.g. date and hour)
  - ...
- Use some kind of function $f : \mathcal{F} \rightarrow \Sigma$, where
  $\mathcal{F}$ - a family of sets of features
  Σ - set of probability distributions

Applying the mentioned approach to the real world setting

- The previous approach utilizes $\varphi_i(v)$ that depends on σ_i . How to discover σ_i then?
- Associate a set of features (contextual information) with each auction, such as:
  - publisher data (e.g. ad site and ad size)
  - user data (e.g.device type and various geographic information)
  - time (e.g. date and hour)
  - ...
- Use some kind of function $f : \mathcal{F} \to \Sigma$, where
  $\mathcal{F}$ - a family of sets of features
  Σ - set of probability distributions

Problem with this solution

f is an intermediate step that is potentially error-prone and computationally expensive.

Motivation

Motivation

- A form of auction commonly used in practice by Ad Exchanges is a **second-price auction with reserve price**.

Motivation

- A form of auction commonly used in practice by Ad Exchanges is a **second-price auction with reserve price**.
- In **real time bidding** (RTB) for display ads, a user visiting a webpage instantaneously triggers an auction held by an Ad Exchange, wherein the winner of the auction earns the ad slot and pays the publisher a certain price.

Motivation

- A form of auction commonly used in practice by Ad Exchanges is a **second-price auction with reserve price**.
- In **real time bidding** (RTB) for display ads, a user visiting a webpage instantaneously triggers an auction held by an Ad Exchange, wherein the winner of the auction earns the ad slot and pays the publisher a certain price.
- Digital advertising is a tremendously fast growing industry: the worldwide digital advertising expenditure was $283 billion in 2018, and it is estimated to further grow to $517 billion in 2023. (For comparison, according to Google Answer Box, GDP in Poland in 2018 was $587 billion).

The paper

Contextual Reserve Price Optimization in Auctions via Mixed-Integer Programming

Joey Huchette
Rice University

Haihao Lu
University of Chicago

Hossein Esfandiari
Google Research

Vahab Mirrokni
Google Research

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada

Reserve Price Optimization problem

Reserve Price Optimization problem

$$\max_{\beta \in X} \mathcal{R}(\beta) := \frac{1}{n} \sum_{i=1}^{n} r(\mathbf{w}^i \cdot \beta; b_i^{(1)}; b_i^{(2)})$$

- $b_i^{(1)}; b_i^{(2)}$ - the (nonnegative) highest and second highest bidding price of impression i
- $\mathbf{w}^i \in \mathbb{R}^d$ is the contextual feature vector of impression i
- $X = [L, U] \subset \mathbb{R}^d$ is a bounded hypercube which serves as a feasible region for the model parameters β

Reward function

$$r(v; b^{(1)}; b^{(2)}) := \begin{cases} b^{(2)} & v \leq b^{(2)} \\ v & b^{(2)} < v \leq b^{(1)} \\ 0 & v > b^{(1)} \end{cases}$$

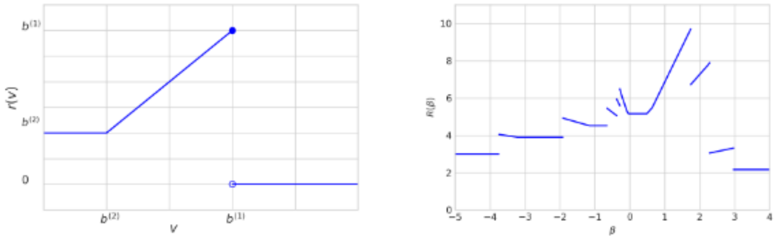

Figure 1: **(Left)** The revenue function $r(v; b^{(1)}, b^{(2)})$. **(Right)** Average revenue function $R(\boldsymbol{\beta})$ with $d = 1$ features and $n = 8$ samples.

Main results

1. Hardness:
   There is no algorithm that solves this optimization problem in polynomial time unless ETH fails.
   Reduction to *k-densest subgraph* problem.
2. Mixed-Integer Programming (MIP) formulation:
   MIP that exactly models the underlying discontinuous reward function.
   Linear Programming (LP) relaxation that yields feasible solutions.
3. Computational validation:
   Both on syntetic and real data. Experiments show that MIP formulation is state-of-the-art.

Hardness - Reduction to *k-densest subgraph* problem

k-densest subgraph problem

Take a graph $G = (V_G, E_G), k \in \mathbb{N}$. The goal is to find a subgraph $H = (V_H, E_H) \subseteq G$ with $|V_E| = k$ that maximizes $\frac{|E_H|}{|V_H|}$.

Hardness - Reduction to *k-densest subgraph* problem

k-densest subgraph problem

Take a graph $G = (V_G, E_G), k \in \mathbb{N}$. The goal is to find a subgraph $H = (V_H, E_H) \subseteq G$ with $|V_E| = k$ that maximizes $\frac{|E_H|}{|V_H|}$.

- It is known that it is impossible to give a polynomial $\frac{1}{8}$-approximation algorithm for the *k-densest subgraph* problem unless ETH fails.

Hardness - Reduction to *k-densest subgraph* problem

k-densest subgraph problem

Take a graph $G = (V_G, E_G), k \in \mathbb{N}$. The goal is to find a subgraph $H = (V_H, E_H) \subseteq G$ with $|V_E| = k$ that maximizes $\frac{|E_H|}{|V_H|}$.

- It is known that it is impossible to give a polynomial $\frac{1}{8}$-approximation algorithm for the *k-densest subgraph* problem unless ETH fails.
- Goal: show that, for any G and k, we can construct such *Reserve Price Optimization* problem that if it were possible to solve it in polynomial time, this would imply it is possible to find a $\frac{1}{8}$-approximate solution to a *k-densest subgraph* problem for G.

Hardness - proof

Have G, k as an input. Set $X = [0, 1]^d$. We have two types of impressions:

- $|V_G|^2$ impressions $(w_1, k, 0)$, where $w_1 =< 1, 1, \ldots, 1 >$.
- $\forall_{e=(u,v)\in E_G}$: exists one impression $(w_e, 2, 1.5)$, where w_e has 1 on indices corresponding to v and u and 0 otherwise.

Hardness - proof

Have G, k as an input. Set $X = [0,1]^d$. We have two types of impressions:

- $|V_G|^2$ impressions $(w_1, k, 0)$, where $w_1 =< 1, 1, \ldots, 1 >$.
- $\forall_{e=(u,v)\in E_G}$: exists one impression $(w_e, 2, 1.5)$, where w_e has 1 on indices corresponding to v and u and 0 otherwise.
- Let $H = (V_H, E_H)$ be a k-densest subgraph of G. Define β_H as a vector with 1 on indices correspondng to $v \in V_H$ and 0 otherwise. $\mathcal{R}(\beta_H)$ is a lower bound of the optimum solution.

Hardness - proof

Have G, k as an input. Set $X = [0,1]^d$. We have two types of impressions:

- $|V_G|^2$ impressions $(w_1, k, 0)$, where $w_1 =< 1, 1, \ldots, 1 >$.
- $\forall_{e=(u,v)\in E_G}$: exists one impression $(w_e, 2, 1.5)$, where w_e has 1 on indices corresponding to v and u and 0 otherwise.
- Let $H = (V_H, E_H)$ be a k-densest subgraph of G. Define β_H as a vector with 1 on indices correspondng to $v \in V_H$ and 0 otherwise. $\mathcal{R}(\beta_H)$ is a lower bound of the optimum solution.
- $\mathcal{R}(\beta_H) = \frac{1}{n}(k|V_G|^2 + 2|E_H| + 1.5|E_{G\setminus H}|) = \frac{1}{n}(k|V_G|^2 + 1.5|E_G| + 0.5|E_H|)$

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)
- Let $V^\beta := \{v \in V_G : \beta_v \geq 0.5\}; G^\beta = (V^\beta, E^\beta)$

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)
- Let $V^\beta := \{v \in V_G \colon \beta_v \geq 0.5\}; G^\beta = (V^\beta, E^\beta)$
- $\mathcal{R}(\beta_{OPT}) \leq \frac{1}{n}(k|V_G|^2 + 1.5|E_G| + 0.5|E^\beta|)$

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)
- Let $V^\beta := \{v \in V_G : \beta_v \geq 0.5\}$; $G^\beta = (V^\beta, E^\beta)$
- $\mathcal{R}(\beta_{OPT}) \leq \frac{1}{n}(k|V_G|^2 + 1.5|E_G| + 0.5|E^\beta|)$
- $\mathcal{R}(\beta_H) \leq \mathcal{R}(\beta_{OPT}) \implies |E_H| \leq |E^\beta|$

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \dots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)
- Let $V^\beta := \{v \in V_G \colon \beta_v \geq 0.5\}$; $G^\beta = (V^\beta, E^\beta)$
- $\mathcal{R}(\beta_{OPT}) \leq \frac{1}{n}(k|V_G|^2 + 1.5|E_G| + 0.5|E^\beta|)$
- $\mathcal{R}(\beta_H) \leq \mathcal{R}(\beta_{OPT}) \implies |E_H| \leq |E^\beta|$
- $V^\beta := \{v \in V_G \colon \beta_v \geq 0.5\} \wedge \sum_v \beta_v \leq k \implies |V^\beta| \leq 2k$

Hardness - proof

Let $\beta_{OPT} =< \beta_1, \beta_2, \ldots, \beta_{|V_G|} >$ be the vector that maximizes $\mathcal{R}(\beta)$.

- $\sum_v \beta_v \leq k$ (otherwise we wouldn't get anything from impressions of the first type, which would lead to β_H being a superior solution)
- Let $V^\beta := \{v \in V_G : \beta_v \geq 0.5\}$; $G^\beta = (V^\beta, E^\beta)$
- $\mathcal{R}(\beta_{OPT}) \leq \frac{1}{n}(k|V_G|^2 + 1.5|E_G| + 0.5|E^\beta|)$
- $\mathcal{R}(\beta_H) \leq \mathcal{R}(\beta_{OPT}) \implies |E_H| \leq |E^\beta|$
- $V^\beta := \{v \in V_G : \beta_v \geq 0.5\} \wedge \sum_v \beta_v \leq k \implies |V^\beta| \leq 2k$

Given a graph with $2k$ vertices, one can easily cover the edges with 8 subgraphs of size k. One of these subgraphs has at least $\frac{|E^\beta|}{8}$ edges. Since $|E^\beta| \geq |E_H|$, we found $\frac{1}{8}$-approximate solution to the k-densest subgraph problem.

Mixed-Integer Programming formulation

We want to model: $\max_{\beta \in X} \mathcal{R}(\beta) := \frac{1}{n} \sum_{i=1}^{n} r(\mathbf{w}^i \cdot \beta; b_i^{(1)}; b_i^{(2)})$.
Let $gr(r(\cdot; b^{(1)}; b^{(2)}); D) := \{(v, y) \colon v \in D, y = r(v; b^{(1)}; b^{(2)})\}$

$$\max_{\beta, v, y} \quad \frac{1}{n} \sum_{i=1}^{n} y_i \tag{1a}$$

$$\text{s.t.} \quad v_i = \mathbf{w}^i \cdot \beta \qquad \forall_i \in \|n\| \tag{1b}$$

$$(v_i, y_i) \in cl(gr(r(\cdot; b^{(1)}; b^{(2)}); [l_i, u_i]) \qquad \forall_i \in \|n\| \tag{1c}$$

$$\beta \in X \tag{1d}$$

Mixed-Integer Programming formulation

A valid MIP formulation for the constraint

$$(v_i, y_i) \in cl(gr(r(\cdot; b^{(1)}; b^{(2)}); [l_i, u_i])$$

is:

$$y \leq b^{(2)} z_1 + b^{(1)} z_2, \quad y \geq b^{(2)}(z_1 + z_2) \tag{2a}$$

$$y \leq v + (b^{(2)} - l) z_1 - b^{(1)} z_3, \quad y \geq v - u z_3 \tag{2b}$$

$$l \leq v \leq u \tag{2c}$$

$$z_1 + z_2 + z_3 = 1, \quad \mathbf{z} \in [0,1]^3 \tag{2d}$$

$$\mathbf{z} \in \mathbb{Z}^3 \tag{2e}$$

Linear programming relaxation

Linear programming relaxation (LP) is the formulation of MIP with integrality constraint (in our case: $\mathbf{z} \in \mathbb{Z}^3$) omitted.

- Optimal reward for LP upper bounds the reward for any feasible solution for the original problem of Reserve Price Optimization problem.
- LP is solvable in polynomial time

Linear programming relaxation

Linear programming relaxation (LP) is the formulation of MIP with integrality constraint (in our case: $\mathbf{z} \in \mathbb{Z}^3$) omitted.

- Optimal reward for LP upper bounds the reward for any feasible solution for the original problem of Reserve Price Optimization problem.
- LP is solvable in polynomial time
- Tightness: All extreme points of this LP relaxation are integral. We call such MIP *ideal*.

Linear programming relaxation

Linear programming relaxation (LP) is the formulation of MIP with integrality constraint (in our case: $\mathbf{z} \in \mathbb{Z}^3$) omitted.

- Optimal reward for LP upper bounds the reward for any feasible solution for the original problem of Reserve Price Optimization problem.
- LP is solvable in polynomial time
- Tightness: All extreme points of this LP relaxation are integral. We call such MIP *ideal*.
- If $n = 1$, the LP relaxation LP is exact, and so exactly represents the convex hull of feasible points for MIP. Unfortunately, the composition of ideal formulations will, in general, fail to be ideal. In fact, the optimal reward from LP can be arbitrarily bad as n grows.

The feasible region

In the optimization problem we look for $\max_{\beta \in X}$. How to choose X?

The feasible region

We cannot bound the magnitude of the components of an optimal solution solely as a function of n, d, and the magnitude of the data.

Proof:
Parametrize the sequence of instances by i. For each i, define $w^{i,1} = (\sqrt{1 - i^{-2}}, i^{-1}), w^{i,2} = (-\sqrt{1 - i^{-2}}, i^{-1}), b_i^{(1)} = 1, b_i^{(2)} = 0$. Note that $\|w^{i,1}\|_2 = \|w^{i,2}\|_2 = 1$, and so all the problem data is bounded by magnitude by one. The unique optimal solution is $\beta^{i,*} = (0, i)$

Computational study

Authors compared 7 methods:

1. CP: $\max_v \frac{1}{n}\sum_{i=1}^{n} r(v; b_i^{(1)}; b_i^{(2)})$ - the optimal constant reserve price policy.
2. LP
3. MIP: MIP terminated after a time limit
4. MIP-R: MIP terminated at the root node (solver-specific)
5. DC: The difference-of-convex algorithm of Mohri and Medina (a previous state-of-the-art)
6. GA: Gradient ascent
7. UB: $\frac{1}{n}\sum_{i=1}^{n} b_i^{(1)}$ - optimum (upperbound)

The metric used is "gap closed" of the improvement of MIP over DC:

$$\frac{MIP - DC}{UB - DC}$$

GA has very poor performance, which is unsuprising given the Figure (1). However, Neural Networks are trained using gradient-based algorithms - which indicate that they might fail in such a setting.

My thoughts: what I liked

- Paper that expands a topic covered on classes.
  A different approach to the problem compared to the one from Algorithmic Game Theory.
- Paper from Machine Learning conference that resembles what we are used to.
  Mathematical approach rather than purely empirical.
- Difficulty: it's inspiring that problems that are both long studied and of great importance still have easy approaches/solutions to discover. (Especially given the fact that the paper got accepted to NeurIPS)

References

Joey Huchette, Haihao Lu, Hossein Esfandiari, Vahab Mirrokni (2020)
Contextual Reserve Price Optimization in Auctions via Mixed-Integer Programming
https://arxiv.org/abs/2002.08841

Tim Roughgarden
Twenty Lectures on Algorithmic Game Theory