Optimal bidding

A dual approach

Carlos Pita

jampp.com

August 5, 2019

Outline

Section 1

Bidding Problems

Introduction

A typical DSP's day:

- Thousands of advertising *campaigns*, on behalf of. . .
- Hundreds of *clients* (advertisers), served by bidding at. . .
- Billions of *auctions* in real-time *markets*, in order to. . .
- Purchase *conversions* (events of interest) for the clients.

But:

- An auction is usually an opportunity for *more than one* client.
- Each client sets contractual *constraints* on volume, costs, etc.

So, in order to decide *when* and *how much* to bid for *which* client:

- Not only *expected profits* must be compared but also. . .
- *Shadow prices* of each constraint set.

Market and Campaigns

Definition (Campaign)

We have n advertising *campaigns* ($i, j \in [n] = \{1, \cdots, n\}$) competing for the *RTB market*.

Definition (Market)

The RTB market consists of m *auctions* ($k \in [m] = \{1, \cdots, m\}$), each one characterized by:

- $w_k(b)$: *win rate*, the probability of winning by bidding b.
- $c_k(b)$: the *expected cost* of winning by bidding b.
- $e_k(i, a)$: *event rate*, the probability of converting given that we won and displayed *ad* a for campaign i.

Strategies

Definition (Bidding strategy)

A bidding strategy $x\colon [m] \to [n] \times [\overline{a}] \times (0, \overline{b}]$ is a mapping from auctions to vectors (i, a, b), where i is a campaign, a is an ad and b is a bid.

Aggregated by campaign i, strategy x produces (expected) conversions by incurring (expected) costs:

Definition (Aggregate functions)

Given a strategy x and a campaign i:

- $C_i(x) = \sum_{(k,(j,a,b))\in x \,|\, j=i} w_k(b) \cdot c_k(b)$ is the *aggregate cost* function.
- $E_i(x) = \sum_{(k,(j,a,b))\in x \,|\, j=i} w_k(b) \cdot e_k(j, a)$ is the *aggregate event* function.

MBFP Problem: Maximum Budget, Fixed Price

This is the problem we will be mostly dealing with today:

Definition (MBFP Problem)

The client sets an upper bound $\bar{B}_i$ to the amount of money to spend (the *campaign budget*) and pays *price* $\bar{p}_i$ for each conversion.

$$\max_{x \in X} \sum_{i \in [n]} \bar{p}_i E_i(x) - C_i(x) \quad \text{s.t.} \quad \bigwedge_{i \in [n]} \bar{p}_i E_i(x) \leq \bar{B}_i$$

or, equivalently $\max_{x \in X | g(x) \leq \vec{0}} f(x)$ where:

- $f : X \to \mathbb{R} \mid f(x) = \sum_{i \in [n]} \bar{p}_i E_i(x) - C_i(x)$.
- $g : X \to \mathbb{R}^n \mid g(x) = (\bar{p}_1 E_1(x) - \bar{B}_1, \ldots, \bar{p}_n E_n(x) - \bar{B}_n)$.

FBMP: Fixed Budget, Maximum Price

This is another, somewhat harder, problem:

Definition (FBMP: Fixed Budget, Maximum Price)

The client pays $\bar{b}_i$ if we deliver enough events to put the unitary price below $\bar{P}_i$ while keeping our profit margin below $\bar{M}_i$.

$$\max_{x \in X} \sum_{i \in [n]} \bar{b}_i - C_i(x) \quad \text{s.t.} \quad \bigwedge_{i \in [n]} \frac{\bar{b}_i}{E_i(x)} \leq \bar{P}_i \ \wedge \ 1 \leq \frac{\bar{b}_i}{C_i(x)} \leq 1 + \bar{M}_i$$

Since treatment *w.r.t.* duality is analogous to MBFP's we won't dwell on FBMP here. We refer to our paper for further details.

Other bidding problems

In practice, we deal with a handful of different bidding problems/contracts.

Despite having rather different constraints, all problems show the following features:

- Their goal is expected profit. This way we can aggregate different problems company-wise in a way that makes economical sense.
- Goals and constraints are sums of strictly per-auction (*i.e.* unitary) terms.
- Furthermore, each constraint is linear in expected unitary costs $w_k(b) \cdot c_k(b)$ and expected unitary events $w_k(b) \cdot e_k(j, a)$.

Section 2

Continuous Relaxation Solution

Knapsack Problem

Now translate MBFP with a single campaign i according to:

- The campaign is a *knapsack*.
- Its budget is the *weight capacity* of the knapsack $\bar{W} = \bar{B}_i = \bar{B}$.
- Each auction is an *item* to pack with:
  - *Weight* equal to its cost $\omega_k = p_i w_k(b) e_k(i, a) = p w_k e_k$.
  - *Value* equal to its profit $\nu_k = w_k(b)(p_i e_k(i, a) - c_k(b)) = w_k(p e_k - c_k)$.

A strategy can then be represented as a $\{0, 1\}^m$ vector indicating which items will be packed into the knapsack.

We have reformulated a simplified version of MBFP as an instance of the *0-1 knapsack problem* $\Rightarrow$ MBFP is hard.

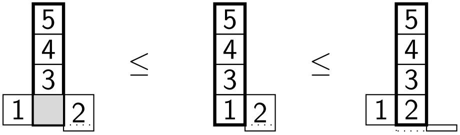
Single Campaign

💡 Here's an idea: sort the items in decreasing "specific value" (*i.e.* value per unit of weight) $\rho_k = \nu_k/\omega_k$ order.

☹ But, in general, there will be a next-to-be-packed item with ρ^* that won't fit the sack, leaving wasted space.

☺ BUT, if items were divisible, item ρ^* could have been split to exactly fill the sack.

That is a *continuous relaxation* solution due to Dantzig. It will be close to our approximate solution in a huge market of tiny transactions.

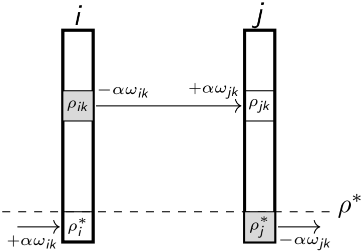


Approximate (easy) Optimal (hard) Relaxed (unfeasible)

Multiple Campaigns

Consider any pair of campaigns i, j and an amount of infinitely divisible auctions assigned to them in decreasing ρ-order so as to exhaust budgets.

Was auction k assigned to the right campaign? If so, a compensated transference of fraction α shouldn't increase expected net value:



So, assuming we can always extend our "buying frontier" a bit, we require $-\alpha\omega_{ik}(\rho_{ik} - \rho_i^*) + \alpha\omega_{jk}(\rho_{jk} - \rho_j^*) \leq 0$.

Bidding rule

From our previous analysis a bidding rule immediately follows:

Definition (ρ-rule)

Pick the campaign i with highest positive $\omega_{ik}(\rho_{ik} - \rho_i^*)$ (if any) for some bid b and ad a.

The "buying frontier" $\rho^* = \rho_1^*, \ldots, \rho_n^*$ sorts of measure how far we go in order to complete budgets.

The focus has shifted to finding an optimal buying frontier ρ_{opt}^* that complete all budgets when following the ρ-rule.

Not the end of the road

We actually implemented an algorithm that follows ρ-rule and daily adjusts ρ^* towards ρ^*_{opt}.

Still we needed to:

- Prove stronger optimality and convergence results.
- Extend it to other contracts and identify general conditions that enable that extension.
- Support noisy and changing real-life market environments.
- Support both first-price and second-price auctions.

For that we developed the more abstract framework that follows, which contains the previous intuitive solution as a special case.

Section 3

Convex Relaxation Solution

Lagrangian

So take two! The Lagrangian of MBFP is:

$$\begin{aligned}\mathcal{L}(x,\theta^*) &= f(x) + \langle \theta^*, g(x) \rangle \\ &= \sum_{i \in [n]} \bar{p}_i E_i(x) - C_i(x) + \theta_i^* (\bar{p}_i E_i(x) - \bar{B}_i)\end{aligned}$$

$\mathcal{L}(x,\theta^*)$ is a sum over a large number of per-auction k terms:

$$\begin{aligned}u_k(i,b,a) &= w_k(b)(\bar{p}_i e_k(i,a) - c_k(b)) + \theta_i^* w_k(b)\bar{p}_i e_k(i,a) \\ &= w_k(\bar{p}_i e_k - c_k) + \theta_i^* w_k \bar{p}_i e_k\end{aligned}$$

☺ The additive structure implies that the contribution of each auction can be computed without considering other auctions.

Bidding Rule

Therefore, in order to maximize the Lagrangian, we just follow:

Definition (MBFP Rule)

Assign auction k to campaign i^* with bid b^* and ad a^* such that $i^*, b^*, a^* = \arg\max_{i,b,a} u_k(i, b, a)$ if and only if $u_k(i^*, b^*, a^*) > 0$.

- Since

$$u_k(i, b, a) = w_k[(1 + \theta_i^*)\bar{p}_i e_k - c_k]$$

it's clear that for 2nd price auctions $b^* = (1 + \theta_i^*)\bar{p}_i e_k$.
(You might think of $\theta_i^* \leq 0$ as a "pacing" parameter).

- The rule is seen to be equivalent to our previous ρ-rule:

$$u_k = w_k \bar{p}_i e_k \left(\frac{w_k(\bar{p}_i e_k - c_k)}{w_k \bar{p}_i e_k} + \theta_i^* \right) \underset{\theta_i^* := -\rho_{ik_i^*}}{=} \omega_{ik}(\rho_{ik} - \rho_{ik_i^*})$$

Dual Problem

The previous rule compute the MBFP dual function q in θ^*:

Definition (MBFP Dual Function)

Given θ^*, the MBFP dual function $q(\theta^*)$ maximizes the Lagrangian $\mathcal{L}(x, \theta^*)$ over the set of strategies X, *i.e.* $q(\theta^*) = \sup_{x \in X} \mathcal{L}(x, \theta^*)$. We call a maximizer $x_{opt}(\theta^*)$.

We will see that by finding a θ^*_{opt} that minimizes q we get an approximate solution to MBFP (recall our equivalent open problem of finding ρ^*):

Definition (MBFP Dual Problem)

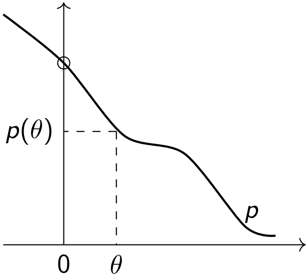
The MBFP dual problem consists in minimizing the dual function $q(\theta^*)$ over the non-positive orthant, *i.e.* $\inf_{\theta^* \leq \vec{0}} q(\theta^*)$. We call a minimizer θ^*_{opt}.

Primal Problem

Now consider this *primal function*:

$$p(\theta) = \min_{x|g(x)\leq\theta} -f(x) = -\max_{x|g(x)\leq\theta} f(x)$$

- Customarily, we are rehashing our original problem as a minimization one.
- By varying θ we can tighten or relax the constraints of our original problem.
- p is clearly non-increasing in θ.
- We want $-p(\vec{0})$ (but we content ourselves with an approximation $-p^{**}(\vec{0})$).

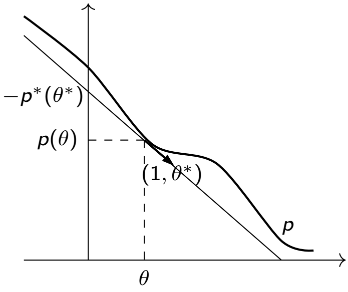


Fenchel Conjugate

We will be taking advantage of some known facts about the Fenchel (*aka* convex) conjugate p^* of p in what follows.

Recall that:

- $-p^*(\theta^*)$ gives the intercept of the supporting hyperplane of the epigraph of p with slope θ^*.
- Thus, p^* can be seen as encoding an alternative representation of p by mapping slopes to intercepts.
- If p is convex the encoding is "loseless".



Subgradient Descent

It is relatively easy to show that $p^* = q$, *i.e.* conjugate and dual functions are the same.

ℹ Now, it's known that p^*:

- Is convex (no matter whether p is also convex or not).
- Has a subgradient $g(x_{opt})$ at point θ^*.

Recall that, given θ^*, we compute the optimal strategy x_{opt} by picking $\arg\max_{i,b,a} u_k(i, b, a)$ for each auction k.

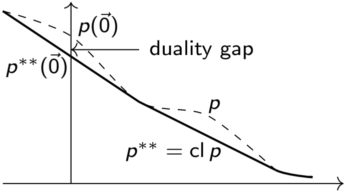
☺ This implies we can solve the dual problem using a simple subgradient descent method. With constant learning rate $\alpha_t = \alpha$ this converges logarithmically near the optimum.

Small Duality Gap

It's also quite easy to show that by solving our dual we get $-p^{**}(\vec{0})$.

ℹ But it's also known that conjugating again recovers the convex closure of p (hence the name "convex relaxation"):

- If p is convex the process is "loseless", *i.e.* $p^{**} = p$.
- But if p is "almost convex" we might still be fine, *i.e.* $p^{**}(\vec{0}) \approx p(\vec{0})$.



☺ We have reasons to believe $-p$ (p) is "almost concave (convex)":

- Constrains limit the sum of many small, quite substitutable, auctions.
- While relaxing θ, the optimizer will pick better opportunities first, yielding mostly decreasing marginal returns.

Section 4

Practical Algorithm

Iterative Algorithm

Our previous analysis suggests an iterative algorithm. Each period:

- Given current multipliers θ_t^*, run the optimal strategy $x_t = x_{opt}(\theta_t^*)$.
- Then descend along $-g(x_t)$ to get new multipliers $\theta_{t+1}^*(x_t)$.

We still need some kind of stationarity/ergodicity assumption:

- We consider a daily period to be a reasonable compromise, since most significative seasonality happens within a day and not between days.
- By keeping the learning rate α small but above some threshold, the optimizer remains adaptive to longer seasonal cycles and trends, and also reactive to structural breaks.

Noisy Environment

💡 We can use a simple *stochastic* subgradient descent algorithm that only relies on having unbiased estimates of the subgradient to logarithmically converge *in expectation* near the optimum with constant learning rate $\alpha_t = \alpha$.

☹ But there is a catch: when daily computing $g(x_t)$ we only have access to per-campaign effective cost and event aggregates for the day, that is *realizations* instead of the *expectations* that $g(x_t)$ depends on.

☺ Nevertheless, since constraints g are ultimately linear in $w_k \cdot c_k$ and $w_k \cdot e_k$ we conclude that these realizations can be used to compute an *unbiased estimate* of $g(x_t)$ (details in paper).

Current Work

Currently at an advanced stage in the implementation and A/B testing of Gloval (as in Capt. Gloval from Robotech), an optimizer module for our bidder based on the previous analysis.

Over the next months, we plan to publish a follow-up paper reporting:

- Bounds for the duality gap.
- Empirically calibrated values for the learning rate and other hyper-parameters.
- Overall economical performance of the algorithm.