Maximizing the Success Probability of Policy Allocations in Online Systems

Artem Betlei[1], Mariia Vladimirova[1], Mehdi Sebbar[2], Nicolas Urien[2], Thibaud Rahier[1], Benjamin Heymann[1]

[1]Criteo AI Lab [2]Criteo Ad Landscape

Motivation

- **Application**: bidding for display advertising auctions
- **Task**: allocate bidding policies to users
- **Challenge with bid level design:** requires attributing generated value to the right touchpoints; multiple touchpoints per timeline

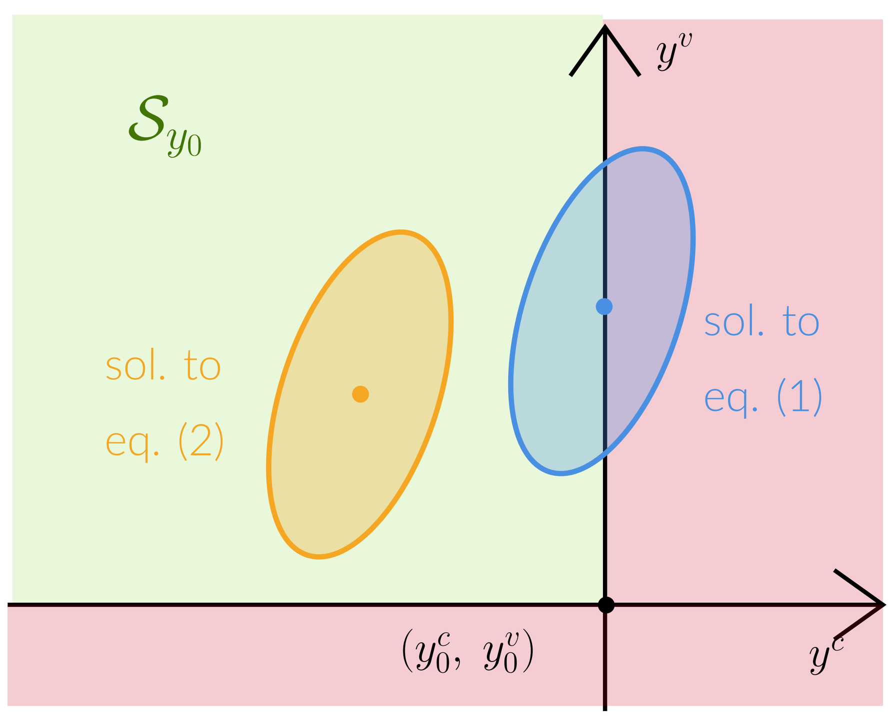
→ bid level algorithms are poorly suited for *causal methods*

- **Challenge with policy allocation:** *cost-constrained expected value maximization* raises intractable *trade-offs* and does not account for uncertainty

→ Policy allocation methods are poorly suited for complex *preferences* over outcome *distribution*

Contribution

- A **user timeline level** formulation, considering entire policies instead of individual bids
- A **success probability** maximization formulation, with a flexible and risk-sensitive criterion



Outcome vector $\mathbf{Y} = (Y^c, Y^v)$ of the (cost,value) distributions. In blue for the solution of Expected Value Maximization (1) and in orange the solution of Success Probability Maximization (2).

Problem Formulation

- $\Pi = \{\pi_0, \pi_1, \dots, \pi_{K-1}\}$ a set of K *candidate policies*, each being *bidding strategy* applied to users,
- $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^d$ contains *features* of user u captured at time t_0,
- $\mathbf{Y} = (Y^v, Y^c) \in \mathcal{Y} \subset \mathbb{R}^2_+$ are the *value* generated by u of period τ and the *cost* spent to advertize to u respectively,
- $\{\mathbf{Y}(\pi)\}_{\pi\in\Pi}$ are realizations of the potential outcomes variables - $\mathbf{y}_u = \mathbf{y}_u(\pi_u)$ is observed *factual outcome* and $\{\mathbf{y}_u(\pi_u)\}_{\pi\in\Pi\setminus\{\pi_u\}}$ are unobserved *counterfactual outcomes*

In **Expected Value Maximization** we are looking for a solution $\psi^* : \mathcal{G} \to \Pi$ to the allocation problem:

$$\max_{\psi\in\Pi^{\mathcal{G}}} \mathbb{E}\left[\sum_{u\in\mathcal{U}} Y_u^v(\psi(G_u))\right] \text{ s.t. } \mathbb{E}\left[\sum_{u\in\mathcal{U}} Y_u^c(\psi(G_u))\right] \le C. \quad (1)$$

where $G_u = \gamma(\mathbf{X}_u)$ for all $u \in \mathcal{U}$, $\gamma : \mathcal{X} \to \mathcal{G}$ is partition function and $\mathcal{G} = \{1, \dots, M\}$ contains partitions indexes.

Instead, **Success Probability Maximization** problem is:

$$\max_{\psi\in\Delta^{\mathcal{G}}} \mathbb{P}\Big(\sum_{u\in\mathcal{U}} \mathbf{Y}_u(\psi) \in \mathcal{S}\Big) = \max_{\psi\in\Delta^{\mathcal{G}}} \mathbb{E}\left[\mathbb{I}_{\mathcal{S}}\Big(\sum_{u\in\mathcal{U}} \mathbf{Y}_u(\psi)\Big)\right], \quad (2)$$

Example of $\mathcal{S}$: $\mathcal{S}_{\mathbf{y}_0} = \{(y^v, y^c) \in \mathcal{Y} \text{ s.t. } y^v > y_0^v \text{ and } y^c \le y_0^c\}$.

Algorithm

Let $\mathcal{C}(\psi) = \mathbb{E}\left[\mathbb{I}_{\mathcal{S}}\left(\sum_{u\in\mathcal{U}} \mathbf{Y}_u(\psi)\right)\right]$ be the optimization criterion. We use a Gaussian approximation: $\mathbf{Y}(\psi) \sim \mathcal{N}(\boldsymbol{\mu}(\psi), \boldsymbol{\Sigma}(\psi))$. With $\boldsymbol{\mu}(\psi), \boldsymbol{\Sigma}(\psi)$ mean and covariance of the bivariate Gaussian.

Lemma: The gradient of $\mathcal{C}$ at ψ satisfies

$$[\nabla\mathcal{C}(\psi)]_{g,k} = \mathbb{E}\Big[\mathbb{I}_{\mathcal{S}}(\mathbf{Y})\Big((\mathbf{Y} - \boldsymbol{\mu}(\psi))^{\intercal}\boldsymbol{\Sigma}(\psi)^{-1}\cdot\boldsymbol{\mu}_{g,k} - \frac{1}{2}\big(\boldsymbol{\Sigma}(\psi) - (\mathbf{Y} - \boldsymbol{\mu}(\psi))(\mathbf{Y} - \boldsymbol{\mu}(\psi))^{\intercal}\big)\cdot\boldsymbol{\Sigma}(\psi)^{-1}\boldsymbol{\Sigma}_{g,k}\boldsymbol{\Sigma}(\psi)^{-1}\Big)\Big]$$

```
Input: S, {μ̂_{g,k}}, {Σ̂_{g,k}}, ψ_0, n_st > 0, η > 0
ψ ← ψ_0
for t = 0 to n_st do
    μ̂ ← Σ_{k,g} ψ(g,k) μ̂_{g,k}, Σ̂ ← Σ_{k,g} ψ(g,k) Σ̂_{g,k}
    ∇ ← ∇̂C(ψ)
    ψ ← ψ + η∇
    Project ψ onto Δ^M
end
Return ψ
```

Algorithm: `SuccessProbaMax`

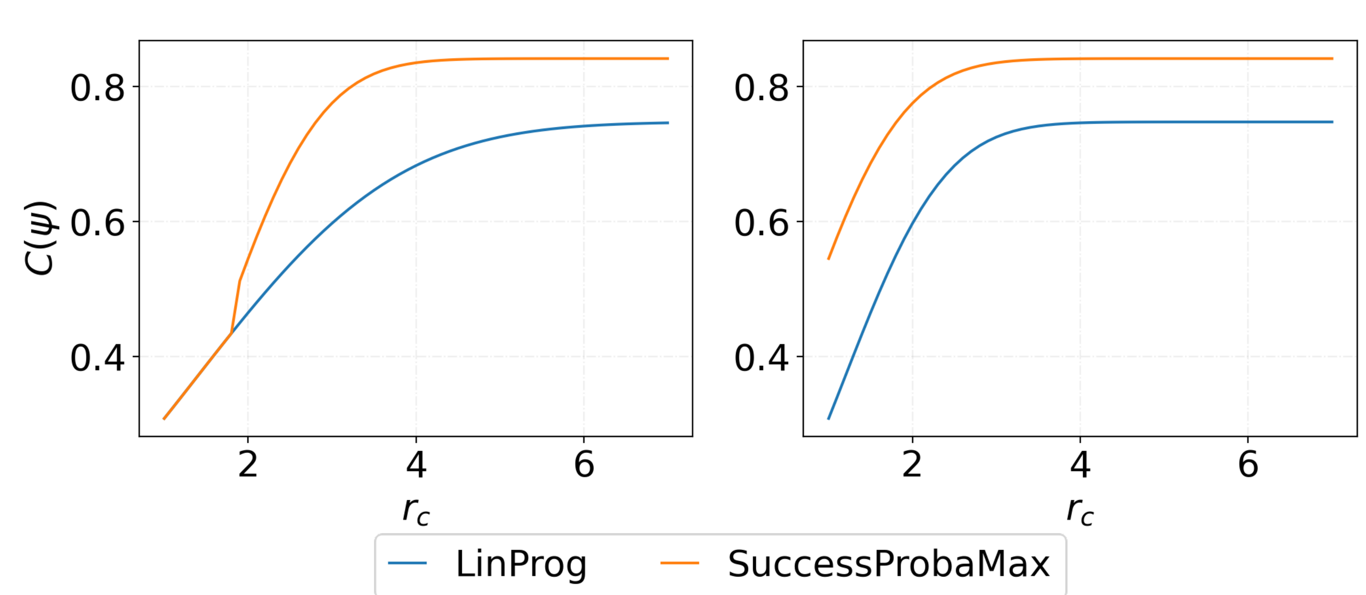
Experiments

$\mathbf{Y} = (Y^v, Y^c)$ is a 2D outcome. Problem is parameterized by 2D difficulty level $\mathbf{r} = (r_v, r_c)$ s.t. $\mathcal{S} = \{(r_v, +\infty), (-\infty, r_c]\}$.

Synthetic data

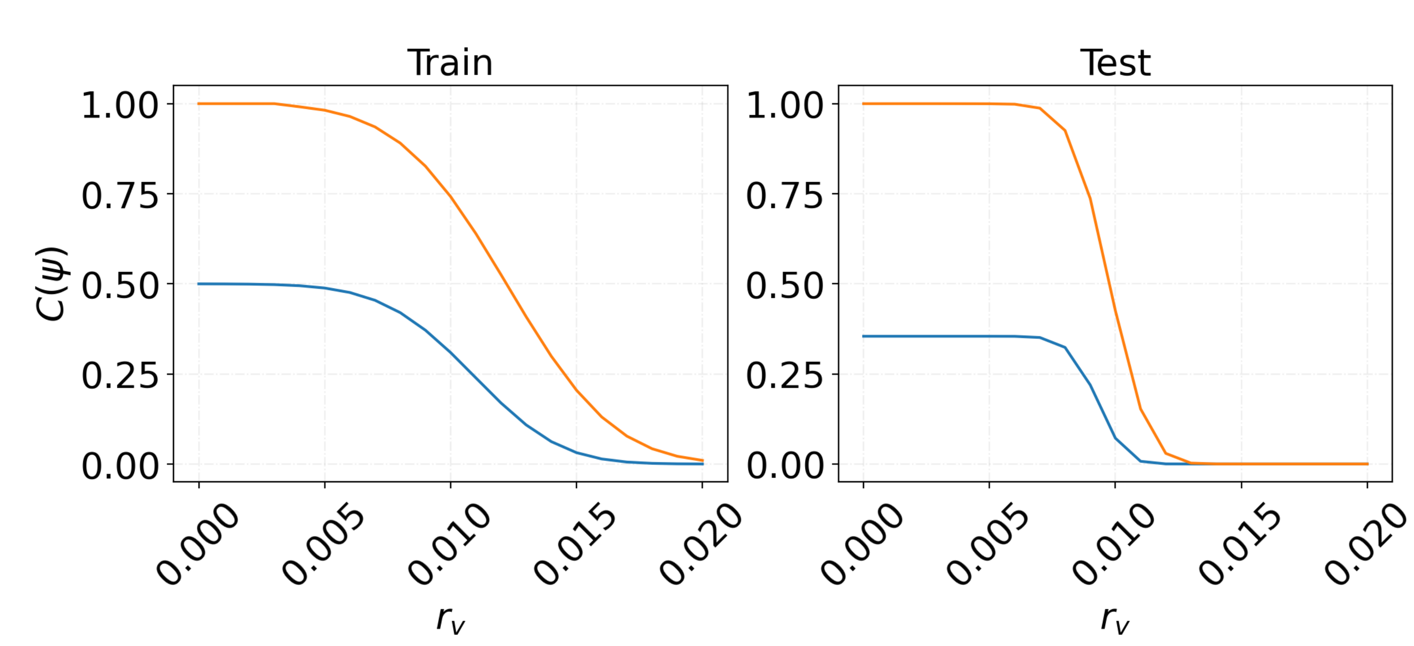
Example	$[\mu^v_{g,k}]$	$[\Sigma^v_{g,k}]$	$[\mu^c_{g,k}]$	$[\Sigma^c_{g,k}]$	ρ
μ_2^c and Σ_1^c larger	$[2,1]$	$[9,1]$	$[1,1.5]$	$[4,1]$	0.5
μ_2^c and Σ_1^c smaller	$[2,1]$	$[9,1]$	$[1,0.5]$	$[1,1]$	0.5

Bivariate Gaussian distribution parameters for generation with one bucket ($M = 1$), two policies ($K = 2$) and 2D outcome $\mathbf{Y}$.



Result for different r_c with fixed $r_v = 0$ on synthetic setup with two-dimensional outcome for cases (i) (left) and (ii) (right).

Real data: Criteo dataset



Results for different Gain r_v while $r_c = 0$ on real data with two-dimensional outcome for train (left) and test (right) splits.