AdX Agent Design

Enrique Areyan Viqueira and Amy Greenwald

CS1951k Algorithmic Game Theory

April 2, 2020

AdX Game

- A market with multiple copies of heterogeneous goods (users' impressions)
- Your agent's job: procure **enough targeted** impressions at the **lowest** possible cost

Two Sources of uncertainty

- Impressions (Supply)
- Competition (Demand)

Variants of the AdX Game

- One-Day, One-Campaign
- Two-Days, Two-Campaigns

Modeling the game. Example: One-Day Game.

Game Elements:

- Let M be a set of **market segments**, and $\pi = \left\langle \pi_1, \pi_2, \ldots, \pi_{|M|} \right\rangle$ a probability mass function where π_m is the probability of drawing $m \in M$.
- A **user** belongs to a random market segment $m \in M$ (e.g., old, female) according to π.
- A **campaign** $C_j = \langle R_j, B_j, m_j \rangle$
  demands $R_j \sim r(\cdot)$ impressions,
  from a random market segment $m_j \in M$ such that $m_j \sim G(\cdot)$,
  and has budget $B_j \sim b(\cdot)$.
- An **agent** $j \in A$ is characterized by its campaign C_j.

Modeling the game. Example: One-Day Game (cont).

A **One-Day Game**, has N agents, each with a **single** campaign.

Let $\vec{x} = \langle x_1, x_2, \ldots, x_{|M|} \rangle$ be a bundle of impressions.
The **utility** u_j of agent j, as a function of bundle $\vec{x}$, is given by:

$$u_j(\vec{x}, C_j) = \rho(\mu(\vec{x}, C_j)) - p(\vec{x})$$

$p(\vec{x})$ is the total **cost** of bundle $\vec{x}$,

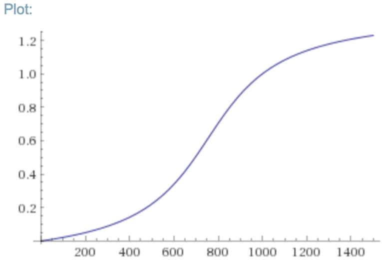
$\mu(\vec{x}, C_j) = \sum_{m \in M} x_m \mathbb{1}_m^j$ where $\mathbb{1}_m^j = 1$ if m matches m_j, 0 o/w.

$\rho(\cdot)$ is a **revenue function** mapping impressions to revenue.

Modeling the game. Example: One-Day Game (cont).

Revenue function

Figure 1: Example of revenue function for campaign with reach 1000.



Modeling the game. Example: One-Day Game (cont).

Game Dynamics

Stage 1: Agent j learns its own type C_j, but not others' types.

Stage 2: All agents compute and submit their bids.

Stage 3: The n users arrive in a random order

$$\langle \vec{m} \rangle = \left\langle m^1, m^2, \ldots, m^n \right\rangle, \text{ where } m^i \sim \pi$$

For each user i that arrives, a second-price auction is held.

The game ends and payoffs are realized.

Modeling the game. Example: One-Day Game (cont).

Strategies

Agent's j **strategy** $s_j(C_j)$ maps a campaign C_j to a tuple $\left\langle \vec{b}, \vec{l} \right\rangle$.

- $\vec{b} = \left\langle b_1, b_2, \cdots, b_{|M|} \right\rangle$ is a bid vector, where $b_m \in \mathbb{R}_+$ is the bid of agent j for market segment $m \in M$.
- $\vec{l} = \left\langle l_1, l_2, \cdots, l_{|M|} \right\rangle$ is a limit vector, where $l_m \in \mathbb{R}_+$ is the total spending limit of agent j in auctions matching market segment $m \in M$.

Notation

Denote by s_{-j} the strategies of agents other than j.

Modeling the game. Example: One-Day Game (cont).

The bundle $\vec{x} = \vec{x}(s_j, s_{-j}, \vec{m})$ procured by agent j depends on:

- its strategy s_j,
- other agents' strategies s_{-j},
- and the realization of the users (types and ordering).

We can now state our goal!

Find s_j^* that maximizes j's interim expected utility:

$$s_j^* \in \arg\max_{s_j} \left\{ \mathop{\mathbb{E}}_{\substack{C_{-j} \\ \vec{m} \sim \pi^n}} \left[u_j(\vec{x}(s_j(C_j), s_{-j}(C_{-j}), \vec{m}), C_j) \right] \right\}$$