Biding Algorithm - Problem Definition

November 21, 2020

0.1 General problem description

Our goal is to maximize our profit margin M_p over all n elements in the set T of all t_i auctions. Where $r(t_i)$ is the revenue from an auction ($r(t_i) = 0$ of if the auction is lost) and $c(t_i)$ is the cost that carrying out an auctioned contract ($r(t_i) = 0$ of if the auction is lost). In an adversarial setting where the contract is awarded to the lowest bidding agent a (*closed-bid first-price reverse auction*). Where $a \in A$ the set of all agents in the game. Our own agent is denoted as $\alpha \in A$.

$$M_p = \sum_{t_i \in T} r(t_i) - \sum_{t_i \in T} c(t_i) \tag{1}$$

At each step i our profit margin varies by a values $\Delta M_p(i)$:

$$\Delta M_p(i) = r(t_i) - c(t_i) \tag{2}$$

For a given *task* our decision algorithm has the following informations :

Cost of adding a given task to the plan :	$c(t_i)$	$\forall t_i$
Price offered by every agent for previous bids :	$p(t_i, a)$	$\forall t_i, a$
Cost of carrying out auction for previous bids :	$c(i)$	$\forall i$
Winner of a given bid :	$w(i)$	$\forall i$

The problem can essentially be thought of as a *multi-armed bandit* problem with an exploration/exploitation tradeoff, we will use a variation on *Thomson Sampling*. The general idea is to fit a probability distribution to estimate our winning probability at each bid. And then pick a bidding amount from that probability distribution in such a way that our expected winning value $\mathbb{E}(\Delta M_p(i))$ is maximal (greedy algorithm). Or to pick a value at random out of a distribution constructed in such a way that our expected winning value is maximized over time but we don't give up on exploration, which actually tends to yield better results than the greedy solution (*Thomson Sampling*).

1 Defining error

We define the error made for a given auction as follows :

$$e(i) = \min_{\forall a \in A/\ \alpha} \{p(t_i, a)\} - p(t_i, \alpha) \tag{3}$$

For auctions that our agent won we have $e(i_{won}) \geq 0$, for auctions lost we have $e(i_{won}) \leq 0$.

Algorithm 1: Thompson Sampling For Automatic Bidding

repeat
    Compute
until *no task remains to be auctioned*;