Ad auction bidding strategy

August 8, 2020

Consider the case of participating in $N \gg 1$ online ad auctions with a limited bidding budget. The task is to create such a bidding strategy that you can win some of them, and that the placed ads generate at least N_C clicks. Also, this should be done by spending as little money as possible.

1 Problem description

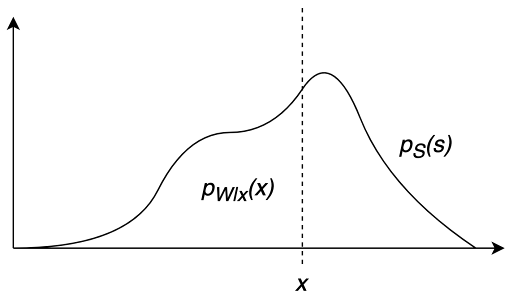


Figure 1: Winning bid distribution of an auction. The probability to win the auction by placing bid x is given by the area under $p_S(s)$ on the left side of x.

For simplicity, we will consider that we only have to create a strategy for a particular ad (for example white sneakers from a specific brand) but the strategy can be easily generalized to multiple ads, each one of them having a different budget and target. The ad exchange generates a huge amount of bid requests which are processed by multiple bidding agents, each of them having the opportunity to make a bid. The user and publisher data contained in every request could be used to predict the probability distribution function of the winning bid price s, and the probability that the user will click on the displayed ad. For every auction $n \in \{1, \dots N\}$ they will be denoted as:

$$p_{C_n} \quad \text{click} - \text{through probability} \tag{1a}$$
$$p_{S_n}(s) \quad \text{probability distribution function of the winning bid price} \tag{1b}$$

For every auction n, we will place a bid price x_n. The probability of winning an auction is given by:

$$p_{W_n|x_n}(x_n) = \int_0^{x_n} p_{S_n}(s)ds. \tag{2}$$

The integral from 0 to x_n takes into account all cases where the winning bid price generated by all other participants except us is smaller than our bid price x_n. Because of the probabilistic nature of our assumptions we can not guarantee which auction we will win or which user will click on the displayed ad. To describe these random events we will use the following Bernoulli random variables:

$$C_n \sim \text{Bernoulli}(p_{C_n}), \tag{3a}$$
$$W_n|x_n \sim \text{Bernoulli}(p_{W_n}(x_n)), \tag{3b}$$

where C_n describes the user ad click events and $W_n|x_n$ - the event of winning an auction by placing a bid price x_n. The total number user clicks on our ad obtained by placing the bids $\{x_n|n = 1, 2 \dots N\}$ is given by:

$$\Upsilon = \sum_{n=1}^{N} C_n \cdot W_n|x_n. \tag{4}$$

This is a random variable, as well. For simplicity, we will look only at its expected value:

$$\begin{aligned}
\mathbb{E}(\Upsilon) &= \sum_{n=1}^{N} \mathbb{E}(C_n \cdot W_n | x_n) \\
&= \sum_{n=1}^{N} \mathbb{E}(C_n) \cdot \mathbb{E}(W_n | x_n) \\
&= \sum_{n=1}^{N} p_{C_n} \cdot p_{W_n|x_n}(x_n). \qquad (5)
\end{aligned}$$

The amount of money spent on the auctions that we have won can be described by the following random variable:

$$M = \sum_{n=1}^{N} x_n \cdot W_n | x_n. \qquad (6)$$

As in the equation for the total number of click events, we will look only at the expectation value of this variable:

$$\begin{aligned}
\mathbb{E}(M) &= \sum_{n=1}^{N} x_n \cdot \mathbb{E}(W_n | x_n) \\
&= \sum_{n=1}^{N} x_n \cdot p_{W_n|x_n}(x_n) \qquad (7)
\end{aligned}$$

The problem of placing N bids $x_1, \ldots x_n$ such that the expected number of user clicks $\mathbb{E}(\Upsilon) = N_C$ and that the spent amount of money on winning bids is minimized can be solved with the method of Lagrange multipliers:

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x), \qquad (8a)$$

$$f(x) = \sum_{n=1}^{N} x_n \cdot p_{W_n|x_n}(x_n), \qquad (8b)$$

$$g(x) = \sum_{n=1}^{N} p_{C_n} \cdot p_{W_n|x_n}(x_n) - N_C, \qquad (8c)$$

where $f(x)$ has to be minimized under the condition that $g(x) = 0$.

2 Solutions of the optimization problem

We will consider an analytically solvable case that can be used to check if our numerical solution is implemented correctly, then we will look at a more general case.

2.1 Single click-through probability and winning bid distibution

To continue with our analysis, we will assume that the winning bid distribution for every auction n can be parametrized by an exponential distribution:

$$p_{S_n}(s) = \alpha_n e^{-\alpha_n s} \quad \alpha_n > 0. \qquad (9)$$

It follows that the probability of winning an auction n if our bid is x_n is given by:

$$\begin{aligned}
p_{W_n|x_n}(x_n) &= \int_0^{x_n} p_{S_n}(s) ds \\
&= \int_0^{x_n} \alpha_n e^{-\alpha_n s} ds \\
&= 1 - e^{-\alpha_n x_n}. \qquad (10)
\end{aligned}$$

To make the problem analytically solvable we have assumed that all $p_{W_n|x_n}(x_n)$ are the same, and all p_{C_n} are equal:

$$\alpha_n = \alpha, \qquad n \in \{1, \dots N\} \tag{11a}$$
$$p_{C_n} = p_C \qquad n \in \{1, \dots N\}. \tag{11b}$$

In this case, we obtain the optimal bid price x_n and the expected spent amount of money to be:

$$x_n = \frac{1}{\alpha} \ln\Big(\frac{N \cdot p_C}{N \cdot p_C - N_C}\Big), \quad n \in \{1, \dots N\} \tag{12a}$$
$$\mathbb{E}(M) = \frac{N_C}{p_C} \frac{1}{\alpha} \ln\Big(\frac{N \cdot p_C}{N \cdot p_C - N_C}\Big). \tag{12b}$$

In real situations, we expect that $Np_C \gg N_C$ (i.e., we have to win only a small fraction of all auctions to achieve the goal of getting N_C clicks) which allows us to expand the natural logarithm around 1:

$$x_n \approx \frac{1}{\alpha} \frac{N_C}{N \cdot p_C}, \tag{13a}$$
$$\mathbb{E}(M) \approx \frac{1}{\alpha} \frac{N_C^2}{N \cdot p_C^2}. \tag{13b}$$

Since $\frac{1}{\alpha}$ is the mean value of the exponential distribution function and $\frac{N_C}{N \cdot p_C} \ll 1$, it follows that x is a small value, i.e., we are participating at every auction with a very small bid price. We could argue that a similar result will be achieved if we use different probability distribution functions for the winning bid prices. This also implies that we have to be very precise about the behavior of $p_{W_n|x_n}$ for small x_n which could be a hard task to achieve in practice. We could study this problem in more detail in the future.

2.2 Multiple click-through probabilities and winning bid distibutions

The general case where every auction is described by a unique probability distribution function and where the click-through probabilities p_{C_n} can be different for every n can be solved numerically with help of the python scipy library. This approach becomes quickly unfeasible when N is of the order of 10^3 which is not enough to handle more realistic cases where $N > 10^6$.

To make the problem manageable by the python scipy library we will assume that the winning bid distribution of an auction can be described by one out of I different possible probability distribution functions:

$$\tilde{p}_{S_i}(s) \quad i \in \{1, 2, \dots I\}. \tag{14}$$

The same idea can be applied to the click-through probability which can take only J different values:

$$\tilde{p}_{C_j} \quad j \in \{1, 2, \dots J\}. \tag{15}$$

If we look closely at the solution of the optimization problem (8) we will see that the optimal bidding price for all auctions with the same winning bid distribution $\tilde{p}_{S_i}(s)$ and user click-through probability $\tilde{p}_{C_j}$ is the same. We will denote this optimal price as $\tilde{x}_{ij}$. Under these considerations, the functions f, g from the Lagrange optimization problem can be rewritten as:

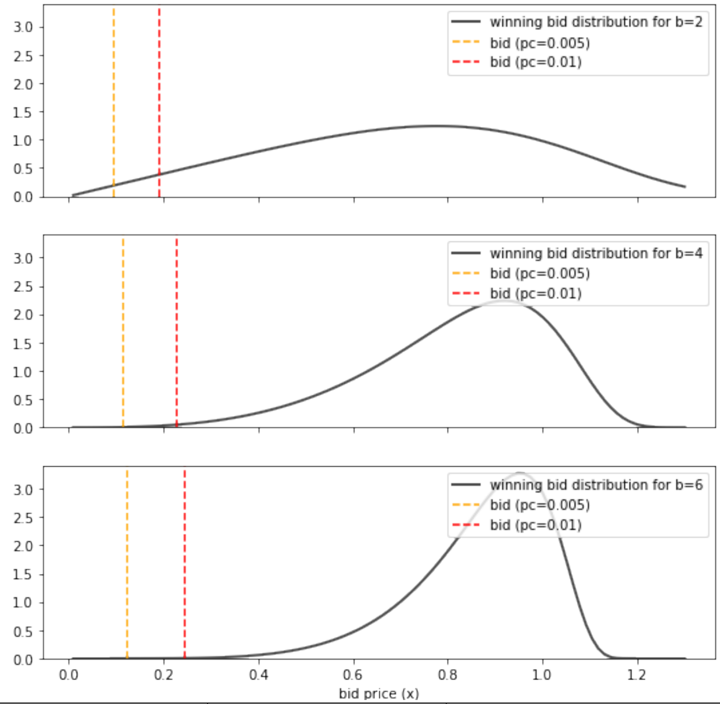
$$f(\tilde{x}) = \sum_{i=1}^{I} \sum_{j=1}^{J} N_{ij} \cdot \tilde{x}_{ij} \cdot \tilde{p}_{W_i|\tilde{x}_{ij}}(\tilde{x}_{ij}), \tag{16a}$$
$$g(\tilde{x}) = \sum_{i=1}^{I} \sum_{j=1}^{J} N_{ij} \cdot \tilde{p}_{C_j} \cdot \tilde{p}_{W_i|\tilde{x}_{ij}}(\tilde{x}_{ij}) - N_C, \tag{16b}$$

where N_{ij} is equal to the number of cases where $p_{S_n} = \tilde{p}_{S_i}$ and $p_{C_n} = \tilde{p}_{C_j}$. With this simplification, we can numerically solve problems where $I \cdot J < 10^3$.

To demonstrate the applicability of this approach, we have considered the case where $I = 3$, $J = 2$:

$$\tilde{p}_{C_1} = 0.005, \tag{17a}$$
$$\tilde{p}_{C_2} = 0.01, \tag{17b}$$
$$\tilde{p}_{S_i}(s) = b_i s^{b_i - 1} \exp\big(1 + s^{b_i} - \exp(s^{b_i})\big) \qquad b_1 = 2,\ b_2 = 4,\ b_3 = 6. \tag{17c}$$

The optimal solution is visualized in Figure 2.



b	p_C	N (elements in subset)	x (analytical pdf)	x (pdf inferred from data)	dx (relative)
2	0.005	1e5	0.0953	0.0949	0.004
4	0.005	2e5	0.1144	0.1171	0.023
6	0.005	8e5	0.1225	0.1207	0.015
2	0.010	2e5	0.1906	0.1908	0.001
4	0.010	8e5	0.2288	0.2270	0.007
6	0.010	3e5	0.2451	0.2421	0.012

Figure 2 & Table 1: Top: Optimal bids for the case of having three types of auctions (described by the winning bid distribution $p_S(s)$) and two types of click-through probabilities p_C. Bottom: used parameters and optimal values. The N (subset) column referes to the number of auctions where the winning bid distribution functions and the user click-through probabilities are the same.

2.3 Use observed data instead of analytical probability distribution functions

Under realistic conditions, we will have to infer the winning bid probability distribution function from the events (winning bid prices) in our data. We can count the event occurrences for a grid of x values and then use spline interpolation as an approximation of the distribution function. We have applied this idea to the previous example where instead of using the analytical form of $\tilde{p}_{S_i}(s)$ the data is generated by sampling values from this distribution. From the table below (missing?) you can see that the differences between both solutions are minimal. We have to take into account that the number of sampled data points per distribution is of the order of 10^6. A smaller number of sampled data points will inevitably lead to a decreased precision of the spline approximation. In addition, we have to keep in mind that the spline approximation creates a function $h(h)$ whose second derivative $d^2h(x)/dx^2$ is equal to zero at the boundaries of the x grid. This restriction can become problematic for probability distribution functions which do not go to 0 for $x \to 0$. Such an example is the exponential probability distribution function whose second derivative at $x = 0$ is equal to:

$$\frac{d^2}{dx^2}\alpha e^{-\alpha x}\Big|_{x=0} = \alpha^3 > 0. \tag{18}$$