Online Learning for Bidding Agent in First Price Auction

Gota Morishita[*] and **Kenshi Abe**[*] and **Kazuhisa Ogawa** and **Yusuke Kaneko**
CyberAgent.Inc.
Shibuya
Tokyo

Abstract

In display advertising, most advertisement inventories are sold via real-time auction. Advertisers typically employ automatic bidding agents to make a bid on their behalf. Traditionally, the ad auction mechanism has been a second-price (SP) auction. However, a giant platform has changed the mechanism to the first-price (FP) auction. Therefore, we study the design of a bidding agent in the FP auction environment. The agent attempts to achieve the target impressions and the target spend. We consider two realistic cases: a fully observable situation, where the winning price is observed after each auction and a partially observable situation, where you only know whether you win or lose an auction. For each case, we provide a bidding agent to learn to submit a bid to meet these targets online. We prove the theoretical performance guarantee using statistical tools. Finally, we conduct experiments to assess the performance of the proposed algorithms. Surprisingly, we experimentally show that the algorithm for the partial information setting performs nearly as well as that for the full information setting.

Introduction

In the past decade, a means to sell and buy display advertisements (ads) on the Internet via real-time bidding (RTB) has been dominant. In RTB, when a user visits a website that has ad inventories, there is an auction on a platform called *Ad Exchange*. Advertisers send their bid to Ad Exchange, who decides a winner of the auction. The winner's ad is placed on the website. It is counted as one *impression*. The above auction process happens within a fraction of a second and there are billions of auctions in one day. Hence, auction-based marketplaces for display ads have necessitated the design of a bidding agent to automatically compute a bid price and make the bid on behalf of advertisers.

Advertisers typically have some goals such as a target quantity of acquired impressions and a target spend. We assume that advertisers would rather use up the total budget than stay within budget. This is because different advertisers have different pieces of information on the value of a specific user, and thus the bid price for a valuable user tends to be high (Ghosh et al. 2009a). Exhausting the budget means acquiring valuable impressions. Therefore, we focus on bidding agents that achieve these two goals: the target number of impressions and the target spend.

We would like to design a bidding agent to win d impressions and spend t per impression. We assume that the bidding agent knows the total supply, or the number of coming ad opportunities n. Define $f = d/n$ to be the target fraction of the supply the agent requires to win. We model the highest bid among other participants as being drawn i.i.d. from a distribution $\mathcal{P}$. Through this paper, we assume that $\mathcal{P}$ is continuous and the support is $[a, b]$. It depends on the distribution $\mathcal{P}$ for a bidding agent to simultaneously meet the target number of impressions and the target spend. If only one of the two targets is feasible, we ignore the target spend and focus on the target impressions. If the distribution $\mathcal{P}$ is known, it is easy to design a bidding agent (Ghosh et al. 2009b; Lang, Moseley, and Vassilvitskii 2012). In reality, however, it is not true that the distribution of the highest bid is known. Therefore, in order to generate a bidding agent to meet the goals, it is crucial to learn the unknown distribution $\mathcal{P}$ from the bidding log data.

The information you obtain after each auction differs among the auction types. In the second-price (SP) auction, if you make the highest bid, you win an impression but pay the second-highest bid, which is known as the *winning price*. In this case, you will know the winning price. If you lose, you will only know that your bid is less than the highest bid. On the other hand, in the first-price (FP) auction, even if you win an auction, you will not know the winning price because you pay your own bid. Hence, it is more difficult to learn the bidding distribution under the FP auction environment. As far as we know, most of the literature has focused on the SP auction because the auction mechanism that is used in most ad exchanges is the SP auction (Ren et al. 2019; Wu, Yeh, and Chen 2015; Zhang, Yuan, and Wang 2014; Lang, Moseley, and Vassilvitskii 2012; Ghosh et al. 2009b). However, Google announces that Google Ad Manager starts the transition to an FP auction to reduce complexity and

[*]Corresponding Author

transparency[1]. Therefore, developing algorithms focused on the design of bidding agents in the FP auction is an urgent task.

We propose online learning algorithms to create a bidding agent for the FP auction setting. In all the suggested algorithms, there are an exploration phase and an exploitation phase. In the exploration phase, they learn the distribution $\mathcal{P}$ from results of auctions. In the exploitation phase, based on the estimated distribution, the algorithms compute a bid price that meets the target fraction and spend to automatically make a bid. As with (Ghosh et al. 2009b), we consider two situations. The first is a fully observable setting where the winning price is revealed in each auction. The second is an intractable setting. It is a partially observable setting where the winning price is not revealed to anyone and you only know if you win or lose an auction.

First, in the full information setting, we extend the Learn-Then-Bid (LTB) proposed in (Ghosh et al. 2009b) to the FP auction environment. Second, we provide two algorithms: Divide-Then-Bid (DTB) and Binary-Search-Then-Bid (BSTB) for the partial information setting. We bound the error of the actual fraction of impressions won by all the algorithms using statistical techniques. Also, we provide the proof that the actual fraction of impressions won converges in probability to the desired fraction. Finally, we conduct experiments to evaluate our algorithms on synthetic data derived from real-world data. Surprisingly, we empirically show that BSTB performs as well as LTB for the fully observable setting despite the difficulty of learning.

Fully Observable Situation

First, we consider a fully observable setting where the winning bid price is publicly revealed after every auction. Namely, whether you win or lose an auction, you will know the highest bid among other participants. This situation seems hypothetical, but you can know the highest bid price among the opponents in Google Ad Manager because Google sends you a log with the highest bid except for your own bid after an auction you participate.

We propose an algorithm named *Learn-Then-Bid* (LTB) to create a bidding agent under the full information setting. The procedure of LTB is shown in Algorithm 1. This algorithm is almost the same as the one proposed in (Ghosh et al. 2009b). The difference is that there is no uncertainty as to the payment under the FP auction environment. This is because when you win an auction, you pay what you bid. Therefore, the bid price that achieves the target spend is the total budget divided by the required number of impressions. Inputs for the LTB algorithm are:

- f: the original target fraction,
- t: the target spend per impressions,
- n: the number of supply,
- m: the exploration length.

In the exploration phase, the algorithm attempts to learn the bid distribution by bidding 0 for the first m opportunities to obtain observations of the highest bid $\{E_1, \ldots, E_m\}$.

[1]https://support.google.com/admanager/answer/9298211

Algorithm 1 First price Learn-Then-Bid(f, t, n, m)

```
Bid 0 for the first m opportunities to obtain {E_i}_{i=1}^m.
P̂_m(x) ← m^{-1} Σ_{i=1}^m 1[E_i ≤ x].
A_m ← fn / ((n−m) P̂_m(t)).
Z*_m ← inf{z | P̂_m(z) ≥ fn/(n−m)}.
if t ≥ Z*_m then
    for j ∈ {m+1, ··· , n} do
        Bid t with probability A_m, and 0 otherwise.
    end for
else
    for j ∈ {m+1, ··· , n} do
        Bid Z*_m.
    end for
end if
```

From the observations, it estimates the empirical cumulative distribution function (CDF) $\widehat{\mathcal{P}}_m(x) = m^{-1}\sum_{i=1}^m \mathbf{1}[E_i \leq x]$. Based on the estimated CDF, it computes a bid value $Z^*_m = \inf\{z|\widehat{\mathcal{P}}_m(z) \geq \frac{fn}{n-m}\}$ that would achieve the target fraction after exploration, which we refer to as $\gamma = \frac{fn}{n-m}$. Then, the algorithm moves on to the exploitation phase. It checks whether Z^*_m is less than or equal to the target spend t. If this is the case, letting $A_m = \frac{fn}{(n-m)\widehat{\mathcal{P}}(t)}$, it bids t with probability A_m. This stochastic bidding strategy prevents the algorithm from winning more opportunities than expected. If $Z^*_m > t$, it ignores the budget constraint and bids Z^*_m.

We prove that the expected fraction of impressions won by the LTB algorithm converges in probability with rates that are exponential in the length of the exploration phase m. To this end, we introduce a statistical tool that bounds the closeness of an empirically estimated distribution to a true distribution from which empirical samples are drawn. It is termed *Dvoretzky-Kiefer-Wolfowitz* (DKW) inequality (Dvoretzky, Kiefer, and Wolfowitz 1956). We state it in the context of the bidding distribution.

Theorem 1 (DKW inequality)**.** *Given $\epsilon > 0$, for all $x \in [a, b]$,*

$$\Pr\left(|\mathcal{P}(x) - \widehat{\mathcal{P}}_m(x)| < \epsilon\right) \geq 1 - 2\exp(-2m\epsilon^2).$$

We define the properties of an algorithm regarding the accuracy.

Definition 1. *Given $\epsilon > 0$, the algorithm has **point-wise** ϵ-accurate observations at a point $x \in [a, b]$ if*

$$|\mathcal{P}(x) - \widehat{\mathcal{P}}(x)| < \epsilon.$$

Definition 2. *Given $\epsilon > 0$, the algorithm has **uniformly** ϵ-accurate observations if, for all $x \in [a, b]$, it has point-wise accurate observations at x, that is:*

$$\forall x \in [a, b] \quad |\mathcal{P}(x) - \widehat{\mathcal{P}}(x)| \leq \epsilon.$$

We omit *uniformly* if it is not confusing.

Using these definitions, we restate DKW inequality as follows.

Corollary 1. *Given $\epsilon > 0$, the probability that the LTB algorithm has ϵ-accurate observations is greater than $1 - 2\exp(-2m\epsilon^2)$.*

Next, we associate the accuracy of the estimation with the expected fraction of the supply won when bidding Z_m^*.

Lemma 1. *Given $\epsilon > 0$, if the LTB algorithm has ϵ-accurate observations, then*

$$|\mathcal{P}(Z_m^*) - \gamma| \leq \epsilon.$$

We omit the proof because it is identical to the proof of Lemma 4 in (Ghosh et al. 2009b).

The above lemma leads us to conclude that LTB nearly achieves the target fraction if it has uniformly ϵ-accurate observations. Before the main result, we state another lemma which is used to show it.

Lemma 2. *For $\gamma, \epsilon > 0$, $\frac{x\gamma}{x+\epsilon}$ strictly increases and $\frac{x\gamma}{x-\epsilon}$ strictly decreases.*

This is easy to show by differentiating the functions with respect to x.

Theorem 2. *Given ϵ and $j > m$, let B_j be the j-th bid of the LTB algorithm. If the algorithm has ϵ-accurate observations, then*

$$\gamma - \epsilon \leq \mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_m] \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon}\gamma.$$

Proof. The bidding strategy depends on whether the estimate Z_m^* of the bid price that meets the target fraction is greater than the target spend per impression t. We would like to show that the inequality holds in either case.

Case 1: $t < Z_m^*$. According to the LTB algorithm, $B_j = Z_m^*$. From Lemma 1, $|\mathcal{P}(Z_m^*) - \gamma| \leq \epsilon$. Hence, $\gamma - \epsilon \leq \mathcal{P}(Z_m^*) \leq \gamma + \epsilon < \frac{\gamma-\epsilon}{\gamma-2\epsilon}\gamma$.

Case 2: $t \geq Z_m^*$. In this case,

$$B_j = \begin{cases} t & w.p. \quad A_m \\ 0 & w.p. \quad 1 - A_m. \end{cases}$$

Therefore, the expected winning rate after the exploration $\mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_m]$ is:

$$\mathcal{P}(t)A_m = \frac{\mathcal{P}(t)}{\widehat{\mathcal{P}}_m(t)}\gamma \geq \frac{\mathcal{P}(t)}{\mathcal{P}(t) + \epsilon}\gamma \geq \frac{\gamma - \epsilon}{\gamma}\gamma = \gamma - \epsilon.$$

This is because $\mathcal{P}(t) + \epsilon \geq \widehat{\mathcal{P}}_m(t)$, $\mathcal{P}(t) \geq \mathcal{P}(Z_m^*) \geq \gamma - \epsilon$, and of Lemma 2.

Next, we show the other side of the inequality similarly:

$$\mathcal{P}(t)A_m = \frac{\mathcal{P}(t)}{\widehat{\mathcal{P}}_m(t)}\gamma \leq \frac{\mathcal{P}(t)}{\mathcal{P}(t) - \epsilon}\gamma \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon}\gamma.$$

□

Lastly, we prove that the expected fraction of impressions won by the LTB algorithm converges in probability.

Theorem 3. *The expected fraction of impressions won by the LTB algorithm $\mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_m]$ converges in probability to γ with the exponential rate in the length of the exploration phase m.*

Proof. Take any $\epsilon > 0$. By DKW inequality and Theorem 2,

$$\Pr\Big(-\epsilon \leq \mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_m] - \gamma \leq \frac{\gamma}{\gamma - 2\epsilon}\epsilon\Big)$$

is greater than or equal to $1 - \exp(-2m\epsilon^2)$. When m tends to ∞, the above probability goes to 1 with the exponential rate. Therefore, the expected fraction of impressions won by the LTB algorithm converges in probability to the target fraction γ. □

Partially Observable Situation

We now proceed with the partial information case, where the information on the winning price is not revealed to anyone, including the auction's winner. This problem is much more difficult for the following two reasons. First, participants only know whether they win or lose an auction unless the auctioneer gives them additional information on bids the others make. Second, as discussed in (Ghosh et al. 2009b), the bidder must pay a cost to obtain information.

As mentioned in the fully observable setting, you can retrieve the information on the winning price in Google Ad Manager. Therefore, it does not seem useful to consider the partial information situation. However, it would be very expensive to develop the infrastructure to retrieve and keep the logs of the highest bid. Besides, currently, other platforms that use the FP auction typically do not provide information on other bidders. In conclusion, the partially observable setting is worth considering.

We propose two methods to learn the winning rates of bids and make a bid to meet the target number of impressions and spend under the partially observable setting. The first is the *Divide-Then-Bid* (DTB) algorithm, where to estimate the winning rate, it places bids in order of the lowest to the highest price in the exploration phase. After the exploration, it makes the best of the estimated winning rate distribution to decide the bid price to satisfy the target impression fraction and spend. The second is the *Binary-Search-Then-Bid* (BSTB) algorithm, which is inspired by a well-known algorithm named *binary search*. DTB tries to estimate the winning rates of all the possible bids. However, we have only to know the bid price to meet the target fraction. Therefore, this algorithm merely tries to search for the bid price that satisfies the target fraction.

Divide-Then-Bid

We describe the DTB algorithm in Algorithm 2. Inputs for DTB are:

- f: the original target fraction,
- t: the original target spend per opportunity,
- n: the supply of opportunities,
- m: the number of samples to estimate the winning rate,
- l: the number of points to divide the bidding range.

DTB puts l points equally spaced in the possible range of bid value $[a, b]$. Then, it places a bid of each point m times to

Algorithm 2 Divide-Then-Bid(f, t, n, m, l)

$g_{remain} \leftarrow fn; budget \leftarrow tg_{remain}$
for $k \in \{1, \cdots, l\}$ **do**
  $S_k \leftarrow 0$.
  for $i \in \{1, \cdots, m\}$ **do**
    Bid $\frac{k(b-a)}{l}$.
    if Bid wins **then**
      $g_{remain} \leftarrow g_{remain} - 1$.
      $budget \leftarrow budget - \frac{k(b-a)}{l}$.
      $S_k \leftarrow S_k + 1$.
    end if
  end for
end for
$\widehat{\mathcal{P}}_{l,m}(x) \leftarrow m^{-1} S_{\min\{k | x \geq \frac{k(b-a)}{l}\}}$
$t^* \leftarrow \frac{budget}{g_{remain}}$.
$A_m \leftarrow \frac{g_{remain}}{(n-ml)\widehat{\mathcal{P}}_{l,m}(t^*)}$
$Z_m^* \leftarrow \inf\{z | \widehat{\mathcal{P}}_{l,m}(z) \geq \frac{g_{remain}}{n-ml}\}$
if $t^* \geq Z_m^*$ **then**
  for $i \in \{ml+1, \cdots, n\}$ **do**
    Bid t^* with probability A_m, and 0 otherwise.
  end for
else
  for $i \in \{ml+1, \cdots, n\}$ **do**
    Bid Z_m^*.
  end for
end if

calculate the winning rate at the point. Based on the winning rate of each point, it estimates the CDF $\mathcal{P}$ as follows:

$$\forall x \in \Big[\frac{k(b-a)}{l}, \frac{(k+1)(b-a)}{l}\Big) \quad \widehat{\mathcal{P}}_{l,m}(x) = \frac{S_k}{m},$$

where S_k is the number of impressions won by DTB when it makes a bid of $\frac{k(b-a)}{l}$. The rest of the procedure, including the bidding strategy, is the same as the LTB algorithm.

Similar to the LTB algorithm, we prove that the expected fraction of impressions won by the DTB algorithm converges in probability with the linear rate at the number of samples at each point m. To do so, we introduce a well-known powerful statistical tool that can be used to prove the weak law of large numbers, namely Chebyshev's inequality (Feller 1968). We state it in a general way.

Theorem 4 (Chebyshev's inequality)**.** *Let X be a random variable with finite expected value μ and finite non-zero variance σ^2. Then for any $\epsilon > 0$,*

$$\Pr(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}.$$

At first, we prove that it is possible for the DTB algorithm to accurately estimate the winning rate in a certain region with high probability.

Theorem 5. *Given $\epsilon, \epsilon' > 0$ and $k \in \{1, \ldots, l\}$, if l is large enough, then for any $x \in \big[\frac{k(b-a)}{l}, \frac{(k+1)(b-a)}{l}\big)$, the probability that the DTB algorithm has point-wise $(\epsilon + \epsilon')$-accurate observations at x is greater than or equal to $1 - \frac{p_k(1-p_k)}{m\epsilon^2}$, that is,*

$$\Pr(|\mathcal{P}(x) - \widehat{\mathcal{P}}_{l,m}(x)| \leq \epsilon + \epsilon') \geq 1 - \frac{p_k(1-p_k)}{m\epsilon^2},$$

where $p_k = \mathcal{P}\big(\frac{k(b-a)}{l}\big)$.

Proof. Now, S_k is a random variable that follows the binomial distribution with two parameters: the number of trials m and a winning probability p_k.

Therefore, $\widehat{\mathcal{P}}_{l,m}\big(\frac{k(b-a)}{l}\big) = \frac{S_k}{m}$ is the unbiased estimator of the winning rate p_k. Because the expectation $\mathrm{E}\Big[\widehat{\mathcal{P}}_{l,m}\big(\frac{k(b-a)}{l}\big)\Big] = p_k < \infty$ and the variance $\mathrm{V}\Big[\widehat{\mathcal{P}}_{l,m}\big(\frac{k(b-a)}{l}\big)\Big] = p_k(1-p_k)/m < \infty$, we can apply Chebyshev's inequality and we have

$$\Pr\left(\Big|\mathcal{P}\Big(\frac{k(b-a)}{l}\Big) - \widehat{\mathcal{P}}_{l,m}\Big(\frac{k(b-a)}{l}\Big)\Big| \leq \epsilon\right) \geq 1 - \frac{p_k(1-p_k)}{m\epsilon^2}.$$

Also, because $\mathcal{P}$ is continuous and l is large enough, we have

$$\mathcal{P}\Big(\frac{(k+1)(b-a)}{l}\Big) - \mathcal{P}\Big(\frac{k(b-a)}{l}\Big) < \epsilon'.$$

Using these above inequalities, if $\mathcal{P}(x) \geq \widehat{\mathcal{P}}_{l,m}(x)$, at least with probability $1 - \frac{p_k(1-p_k)}{m\epsilon^2}$ we have

$$\begin{aligned} 0 \leq \mathcal{P}(x) - \widehat{\mathcal{P}}_{l,m}(x) &\leq \mathcal{P}\Big(\frac{(k+1)(b-a)}{l}\Big) - \widehat{\mathcal{P}}_{l,m}\Big(\frac{k(b-a)}{l}\Big) \\ &\leq \mathcal{P}\Big(\frac{(k+1)(b-a)}{l}\Big) - \mathcal{P}\Big(\frac{k(b-a)}{l}\Big) + \epsilon \\ &\leq \epsilon' + \epsilon. \end{aligned}$$

The first inequality follows because $\mathcal{P}$ and $\widehat{\mathcal{P}}_{l,m}$ increase. Similarly, if $\mathcal{P}(x) < \widehat{\mathcal{P}}_{l,m}(x)$, $-(\epsilon' + \epsilon) \leq \mathcal{P}(x) - \widehat{\mathcal{P}}_{l,m}(x) < 0$.

□

The next corollary extends it to show that DTB can accurately estimate the winning rate of any bid price with high probability. This statement follows immediately from the previous theorem.

Corollary 2. *Given $\epsilon, \epsilon' > 0$, if l is large enough, then the probability that the DTB algorithm has uniformly $(\epsilon + \epsilon')$-accurate observations is greater than or equal to $\Pi_{k=1}^{l}\big(1 - \frac{p_k(1-p_k)}{m\epsilon^2}\big)$, that is,*

$$\Pr(\forall x |\mathcal{P}(x) - \widehat{\mathcal{P}}_{l,m}(x)| \leq \epsilon + \epsilon') \geq \Pi_{k=1}^{l}\Big(1 - \frac{p_k(1-p_k)}{m\epsilon^2}\Big).$$

Next, similar to the full information, we link the accuracy of the estimation with the expected fraction of supply won by the DTB algorithm after the exploration. Let $\beta = \frac{fn-w}{n-lm}$, where w is the number of impressions won by the DTB algorithm in the exploration. The proof is omitted because it is identical to the previous ones.

Lemma 3. *Given $\epsilon > 0$, if the DTB algorithm has ϵ-accurate observations, then*

$$|\mathcal{P}(Z_m^*) - \beta| \leq \epsilon,$$

where Z_m^* is an estimate of the bid that meets the target fraction after the exploration.

Theorem 6. *Given ϵ and $j > m$, let B_j be the j-th bid of the DTB algorithm. If the algorithm has ϵ-accurate observations, then*

$$\beta - \epsilon \leq \mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_{ml}] \leq \frac{\beta - \epsilon}{\beta - 2\epsilon}\beta.$$

Combining Theorem 6 and Corollary 2, we have the theorem that the expected fraction of impressions won by the DTB algorithm after the exploration converges in probability with the linear rate in the number of samples at each point m.

Theorem 7. *If l is large enough, the expected fraction of impressions won by the DTB algorithm $\mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_{lm}]$ converges in probability to β with the linear rate in the number of samples at each point m.*

Proof. Take any $\delta > 0$. Letting ϵ and ϵ' be $\frac{\delta}{2}$ in Corollary 2, we have

$$\Pr(|\mathcal{P}(x) - \widehat{\mathcal{P}}_{l,m}(x)| \leq \delta) \geq \Pi_{k=1}^{l}\Big(1 - \frac{4p_k(1-p_k)}{m\delta^2}\Big).$$

Therefore, by Theorem 6,

$$\Pr\Big(-\delta \leq \mathrm{E}[\mathcal{P}(B_j)|E_1, \ldots, E_{ml}] - \beta \leq \frac{\beta}{\beta - 2\epsilon}\delta\Big)$$

is greater than or equal to $\Pi_{k=1}^{l}\big(1 - \frac{4p_k(1-p_k)}{m\delta^2}\big)$. When m tends to ∞, the probability goes to 1. $\square$

Binary Search-Then-Bid

DTB tries to learn about the winning rate of every single bid. However, it is enough to know the winning rate of the bid that meets the target fraction. Hence, the *Binary-Search-Then-Bid* (BSTB) searches for the bid price that satisfies the target fraction of impressions.

The algorithm procedure is shown in Algorithm 3. As always, we explain the algorithm in detail. Inputs for BSTB are:

- f: the initial target fraction of impressions,
- t: the initial target spend per required impressions,
- n: the number of impression supply,
- m: the number of samples to estimate the winning rate,
- ϵ: the allowable error in the estimation of the bid that meets the target fraction,
- α: the ratio at which the algorithm divides the range of the bid value.

Here are the notations in the algorithm.

- z_{min}: the minimum value of the range that the algorithm searches,
- z_{max}: the maximum value of the range,
- z: the current bid that the algorithm makes,
- j: the number of auctions held already,
- g_{remain}: the remaining number of required impressions.

In the exploration phase, the algorithm makes a bid of the initial price t m times to estimate the winning rate[2]. If the estimated winning rate is less than the target fraction $\frac{g_{remain}}{n-j}$, the left half of the interval $[z_{min}, t]$ in which the target bid value is not likely to lie is eliminated and the search continues on the remaining half, again taking the point that divides the interval in the ratio α to $1-\alpha$ as the new z. Note that α can be different in each iteration[3]. Otherwise, the right half of the interval $[z, z_{max}]$ is eliminated and the same procedure is conducted. This process is defined as one iteration. After each iteration, the length of the interval gets α or $1-\alpha$ times smaller. Therefore, if the estimation of the winning rates is correct and n is large enough, the algorithm finds the bid price to achieve the target fraction of impressions at finite times. When the exploration stops at Line 17 in Algorithm 3, it estimates the bid value that achieves the target fraction with accuracy and moves on to the exploitation phase. BSTB does not estimate the winning rate of the bid price to meet the target spend t^*. Hence, unlike the previous algorithms, its bidding strategy is deterministic in such a way that it merely makes a constant bid with probability 1 until it wins the required impressions.

We calculate the probability that BSTB finds the target fraction and prove that the probability converges to 1 when $m \to \infty$.

Theorem 8. *Given $\epsilon > 0$, if n is large enough, the BSTB algorithm finds the bid price whose winning rate is within $[\beta_I - \epsilon, \beta_I + \epsilon]$ at finite times I at least with probability $\Pi_{i=1}^{I}\{1 - (\sigma_{i,m}^2/\min\{|\mathcal{P}(z_i) - \beta_i + \epsilon|, |\mathcal{P}(z_i) - \beta_i - \epsilon|\})\}$, where $\beta_i = \frac{g_{remain}}{n-im}$ and z_i is the bid at the ith iteration.*

Proof. There is a likelihood that the bid that meets the target fraction β_i moves into intervals eliminated already while bidding. It is avoidable by checking $\frac{g_{remain}}{n-j}$ is close to the winning rates at z_{max} and z_{min} after each bid. If so, setting z_{max} or z_{min} as z, the algorithm moves on to the exploitation phase. Hence, we assume that the target fraction bid stays within the searching interval while bidding.

Let S_i be the number of auctions won in the ith iteration. When the true winning rate $\mathcal{P}(z_i)$ is within $[\beta_i - \epsilon, \beta_i + \epsilon]$, the estimated winning rate $\frac{S_i}{m}$ should be within $[\beta_i - \epsilon, \beta_i + \epsilon]$. To achieve this, the error in the estimation has to be less than $\min\{|\mathcal{P}(z_i) - \beta_i + \epsilon|, |\mathcal{P}(z_i) - \beta_i - \epsilon|\}$.

Next, if $\mathcal{P}(z_i)$ is outside $[\beta_i - \epsilon, \beta_i + \epsilon]$, BSTB should continue to explore and does not eliminate the interval in

[2]The initial price can be anything, but in practice t is a good option.

[3]If α gets smaller and smaller in each iteration, there is a possibility that the algorithm does not end (e.g. $\alpha = 10^{-i}$). Therefore, ensure that α and $1-\alpha$ are at least larger than a certain small value.

Algorithm 3 Binary-Search-Then-Bid$(f, t, n, m, \epsilon, \alpha)$

$g_{remain} \leftarrow fn$; $budget \leftarrow tg_{remain}$;
$z \leftarrow t$; $z_{min} \leftarrow a$; $z_{max} \leftarrow b$; $j \leftarrow 0$
do
  $S \leftarrow 0$.
  for $k \in \{1, \cdots, m\}$ **do**
    Bid z.
    $j \leftarrow j + 1$.
    if Bid wins **then**
      $g_{remain} \leftarrow g_{remain} - 1$.
      $budget \leftarrow budget - z$.
      $S \leftarrow S + 1$.
    end if
  end for
  if $\frac{S}{m} < \frac{g_{remain}}{n-j}$ **then**
    $z_{min} \leftarrow z$; $z \leftarrow z + \alpha(z_{max} - z_{min})$
  else
    $z_{max} \leftarrow z$; $z \leftarrow z - \alpha(z_{max} - z_{min})$
  end if
  if $g_{remain} = 0$ or $n - j = 0$ **then** Terminate
while $|\frac{S}{m} - \frac{g_{remain}}{n-j}| > \epsilon$
$t^* \leftarrow \frac{budget}{g_{remain}}$.
if $t^* \geq z$ **then** $B \leftarrow t^*$ **else** $B \leftarrow z$
while $g_{remain} > 0$ and $n > j$ **do**
  Bid B.
  $j \leftarrow j + 1$.
  if Bid win **then**
    $g_{remain} \leftarrow g_{remain} - 1$; $budget \leftarrow budget - t^*$.
  end if
end while

which the desired bid lies. Particularly, when $\mathcal{P}(z_i) > \beta_i + \epsilon$, $\frac{S_i}{m} > \beta_i + \epsilon$ is required to hold. Similarly, when $\mathcal{P}(z_i) < \beta_i - \epsilon$, $\frac{S_i}{m} < \beta_i - \epsilon$ is required to hold. To this end, the error in the estimation has to be smaller than $\min\{|\mathcal{P}(z_i) - \beta_i + \epsilon|, |\mathcal{P}(z_i) - \beta_i - \epsilon|\}$.

The probability that the error is less than these values is at least $1 - \sigma_{i,m}^2 / \min\{|\beta_i - \mathcal{P}(z_i) + \epsilon|, |\beta_i - \mathcal{P}(z_i) - \epsilon|\}^2$. In the Ith iteration, the probability that BSTB finds the desired bid price is at least $\Pi_{i=1}^{I}\big(1 - (\sigma_{i,m}^2 / \min\{|\beta_i - \mathcal{P}(z_i) + \epsilon|, |\beta_i - \mathcal{P}(z_i) - \epsilon|\}^2)\big)$ by Chebyshev's inequality. □

Theorem 9. *If n is large enough, let B be the bid in the exploitation phase, the expected fraction of impressions won by the BSTB algorithm* $\mathrm{E}[\mathcal{P}(B)|E_1, \ldots, E_{mI}]$ *converges in probability to β_I with the linear rate in the number of samples at each point m.*

Proof. Take any $\epsilon > 0$. According to the algorithm, it stops when it wins the target impressions. Therefore,

$$\mathrm{E}[\mathcal{P}(B)|E_1, \ldots, E_{mI}] \leq \beta_I.$$

By Theorem 8, we have

$$|\mathcal{P}(z_I) - \beta_I| \leq \epsilon$$

with probability $\Pi_{i=1}^{I}\{1 - (\sigma_{i,m}^2 / \min\{|\beta_i - \mathcal{P}(z_i) + \epsilon|, |\beta_i - \mathcal{P}(z_i) - \epsilon|\})\}$. Hence, following the same logic as Theorem 2, we have

$$\beta_I - \epsilon \leq \mathrm{E}[\mathcal{P}(B)|E_1, \ldots, E_{mI}].$$

In conclusion, $\Pr\big(-\epsilon \leq \mathrm{E}[\mathcal{P}(B)|E_1, \ldots, E_{mI}] - \beta_I \leq 0\big)$ is greater than or equal to $\Pi_{i=1}^{I}\{1 - (\sigma_{i,m}^2 / \min\{|\beta_i - \mathcal{P}(z_i) + \epsilon|, |\beta_i - \mathcal{P}(z_i) - \epsilon|\})\}$. When m goes to ∞, $\sigma_{i,m}^2$ goes to 0 in the linear rate, that is, the probability goes to 1 with the linear rate. □

Experiment

In this section, we evaluate the suggested algorithms on synthetic data that simulates the real data from the Right Media Exchange (Ghosh et al. 2009b). This section continues as follows. First, we describe the data and the experiment setting in detail. Second, we show the results of the LTB algorithm for the full information setting. Lastly, we show the results of DTB and BSTB for the partial information setting.

Data and Experimental Setting

The synthetic data are drawn from the log-normal distribution. This is because, according to (Ghosh et al. 2009b), the real log data in Real Media Exchange is well-fitted with a log-normal distribution even though the best parameters vary depending on the publisher. The real data was collected in the SP auction. However, we do not have enough data for the FP auction. We assume that the bid in the FP auction follows the family of the log-normal distributions. We conduct the experiments varying the mean and the variance of the log-normal distribution. We show part of the results, but all the results have the same trend. Some samples drawn from the log-normal distribution are too large for the real data, so we discard samples larger than an upper-limit b which is chosen to be the 99.7th percentile.

The supply n is $50,000$ impressions. To evaluate our algorithms under a wide variety of circumstances, we choose 16 equally spaced values of the target f and t from the interval $(0, 1)$. For each pair of values (f, t), we measure the actual fraction and spend, running each algorithm 500 times to average out the sampling fluctuation.

Results: Full Information Setting

The results of the LTB algorithm for the fully observable setting are shown in Figure 1. In Figure 1 (a) and (b), the dotted line is an ideal one that represents the minimum spending necessary to achieve the target fraction of supply. The line is given by $\max\{\mathcal{P}^{-1}(\gamma), t\}$, that is, when the fraction and spend goals cannot be satisfied at the same time, the algorithm ignores the target spend and focuses on the target fraction only. The statistics of the spending are summarized by a box-and-whisker plot where we draw a box from the 25% to the 75%. A vertical line (the orange one) goes through the box at 50% (the median) and the whiskers go from the minimum to the maximum.

Figure 1 (a) and (b) illustrate that the budget spent by the LTB algorithm is very close to the ideal one. When the bid price that achieves the target fraction is less than the target spend per impression, the algorithm bids the target spend and you pay the target spend. Hence, the target spend is definitely achieved. Therefore, boxes in the diagonal line in Figure 1 (a) are squashed.

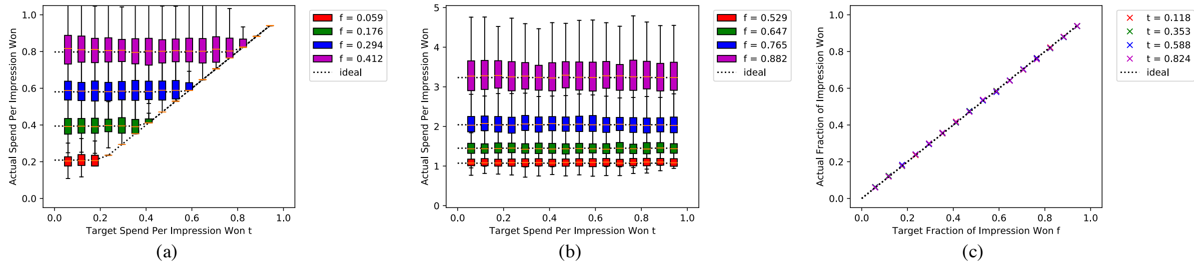

Figure 1: Results of the LTB algorithm with $m = 100$. (a), (b) Actual spend per impression won, (c) Actual fraction of impression won.

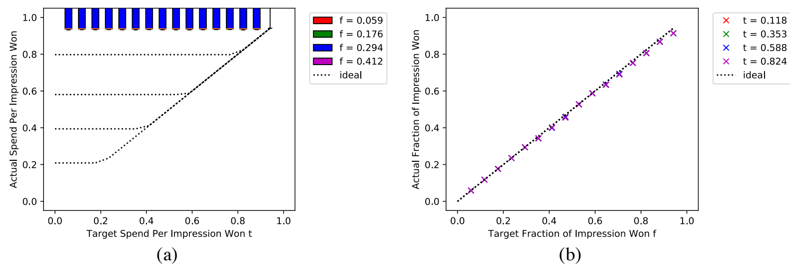

Figure 2: Results of the DTB algorithm with $m = 100, l = 25$. (a) Actual spend per impression won, (b) actual fraction of impression won.

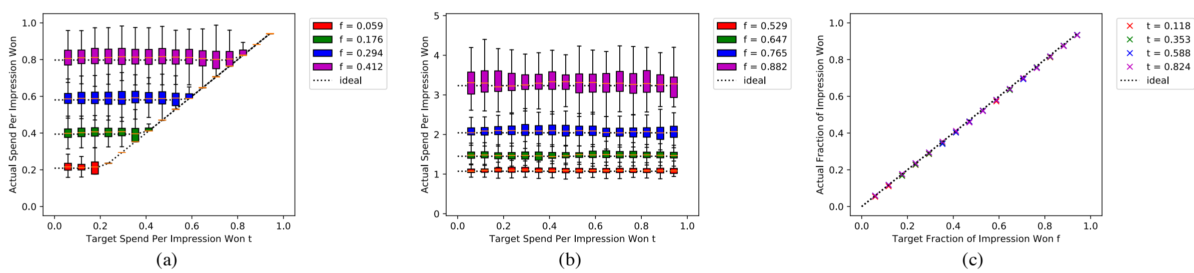

Figure 3: Results of the BSTB algorithm with $m = 100, \epsilon = 0.01$. (a), (b) Actual spend per impression won, (c) Actual fraction of impression won.

Figure 1 (c) depicts the performance of the target quantity. The dotted line is an ideal one, so it is a linear function. The algorithm performs nearly as well as the ideal one for a wide range of the target spend.

Results: Partial Information Setting

The results of the DTB algorithm are shown in Figure 2. Figure 2 (a) plots the actual spend. In all settings, the algorithm overspends the budget. Figure 2 (b) depicts the actual fraction of impressions won by the algorithm. Unlike the target spend, it achieves the ideal target fraction.

Figure 3 shows the actual fraction and spend the BSTB algorithm achieved. The most striking result is that the performance of the BSTB algorithm is very close to that of the LTB algorithm for the fully observable setting. This concurs well with (Ghosh et al. 2009b). The lack of information on others' bids does not burden BSTB even in the FP auction environment.

Conclusion

In this study, we propose algorithms to design a bidding agent to make a bid automatically to meet two criteria: impressions and spend in the FP auction environment. In the full information setting, we propose the *Learn-Then-Bid* algorithm. In the partial information setting, we propose the DTB and BSTB algorithms. We theoretically guarantee their performance in the target fraction of impressions and spend per impression. While DTB does not work well empirically, BSTB and LTB perform very well. It is worth noting that BSTB performs as well as LTB empirically.

References

Dvoretzky, A.; Kiefer, J.; and Wolfowitz, J. 1956. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley.

Ghosh, A.; McAfee, R. P.; Papineni, K.; and Vassilvitskii, S. 2009a. Bidding for representative allocations for display advertising. In *WINE*.

Ghosh, A.; Rubinstein, B. I. P.; Vassilvitskii, S.; and Zinkevich, M. 2009b. Adaptive bidding for display advertising. In *WWW*.

Lang, K. J.; Moseley, B.; and Vassilvitskii, S. 2012. Handling forecast errors while bidding for display advertising. In *WWW*.

Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; and Yu, Y. 2019. Deep landscape forecasting for real-time bidding advertising. In *KDD*.

Wu, W. C.-H.; Yeh, M.-Y.; and Chen, M.-S. 2015. Predicting winning price in real time bidding with censored data. In *KDD*.

Zhang, W.; Yuan, S.; and Wang, J. 2014. Optimal real-time bidding for display advertising. In *KDD*.