Received November 25, 2019, accepted December 16, 2019, date of publication December 20, 2019,
date of current version February 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961155

Bid-Aware Active Learning in Real-Time Bidding for Display Advertising

SHUHAO LIU AND YONG YU
Department of Computer Science, Shanghai Jiaotong University, Shanghai 200240, China
Corresponding author: Yong Yu (yyu@apex.sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702327, Grant 61772333, and Grant 61632017, and in part by the Shanghai Sailing Program under Grant 17YF1428200.

ABSTRACT In Real-time Bidding (RTB) based display advertising, demand side platforms (DSPs) estimate the click-through rate (CTR) of each advertisement impression, and then decide whether and how much to bid based on the information of the user and the advertiser. Typically, when a new campaign is launched, the CTR estimation module of the DSP needs to collect data to train an accurate estimator. The advertiser is charged for each ad impression in display advertising, therefore there is some cost for obtaining each training instance. Thus one crucial task is to actively train an accurate CTR estimator within the constraint of the budget. Traditional active learning algorithms fail to deal with such scenario because (i) acquiring training instances is implemented via performing real-time bidding for the corresponding auctions; (ii) RTB requires the bidding agent to make real-time decisions for sequentially coming bid requests; (iii) cost for each ad impression will be unveiled only after giving the bid price and winning the auction; (iv) training data gathered in post-bid stage has a strong bias towards the won impressions. In this paper, we propose a Bid-aware Active Real-time Bidding (BARB) algorithm to actively choose training instances by setting different bid prices for each ad auction, in order to efficiently train an accurate CTR estimation model within the budget constraint. The empirical study on different campaigns of three real-world datasets with three budget constraints shows the effectiveness of our proposed algorithm.

INDEX TERMS Real-time bidding, active learning, user response prediction.

I. INTRODUCTION

Nowadays, Real-time Bidding (RTB) has become an important paradigm in display advertising [1]. Different from the conventional negotiation or pre-setting a fixed bid price for each campaign or keyword, RTB enables advertisers to give a bid price for every individual impression [2].

A concise interaction process between the main components of the RTB ecosystem is shown in Fig. 1. Each ad placement will trigger an auction when the user visits an ad-supported site (e.g., web page, streaming videos and mobile apps). The ad exchange will send bid requests [1] to the advertisers' buying systems, usually referred to as Demand Side Platforms (DSPs). Upon receiving a bid request, a DSP will calculate a bid price as a response after holding an internal auction among all of its qualifying campaigns. An auction will be held at each intermediary (ad networks, ad exchanges, etc.) and finally in the publishers' system. After that, the winner's ad will be shown to the visitor along with the regular content of the website. It is common that a long time page-loading would greatly reduce users' satisfaction [1], thus, DSPs are usually required to return a bid in a very short time frame like 0.1s. Bidding algorithms employed by DSPs are expected to contribute a much higher return-on-investment (ROI) comparing with the traditional channels. It is crucial that such algorithms can quickly decide whether and how much to bid for a specific ad impression, given the contextual and user behavior information (usually referred to as user segments).

From the perspective of a DSP, there are two important factors to decide the bid price for a coming ad impression: the utility and cost for winning this ad auction. On the one hand, the estimation of the utility normally refers to estimating the click-through rate (CTR) or conversion rate (CVR) for each individual ad impression [3], which means how probably the user will click or convert after seeing the ad. On the

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

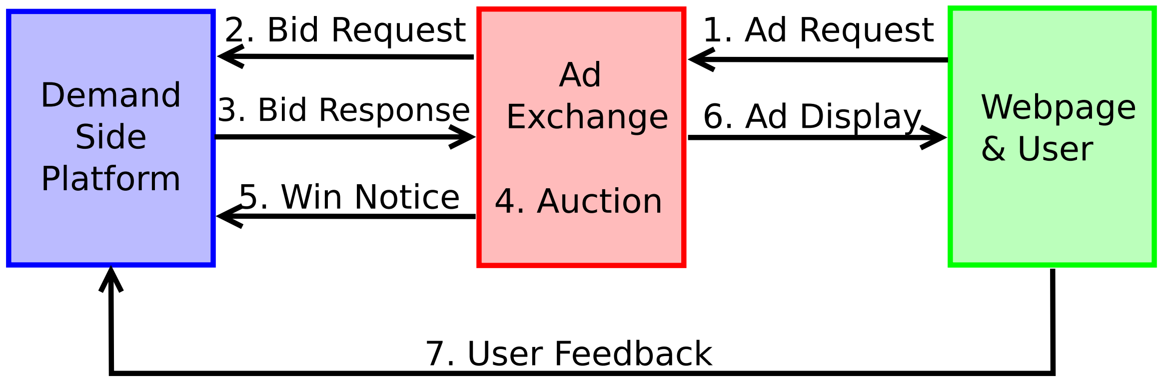

FIGURE 1. A brief illustration of the interactions between user, ad exchange and DSP.

other hand, the estimation of the cost is referred as bid landscape modeling [4] for the popular second price auction. The authors in [5] proposed a DSP bid optimization framework where these two factors are both embedded.

Typically, the CTR estimation model is trained with the previous impression/click data. However, when a new campaign is launched, it will definitely need a launching period to collect training data for the CTR estimation model. Acquiring such data is not cost-free. Display advertising adopts the cost-per-mille (CPM) payment mechanism where the advertiser will be charged for each ad impression [2]. Thus there should be a warming-up budget to train the CTR estimator [5]. As a result, how to efficiently choose training instances to train the model with a limited warming-up budget is an important problem for advertisers and DSPs.

To efficiently learn the CTR estimator, it is natural to employ active learning algorithms that greedily selects the training instances with the highest learning value to the current model. However, traditional active learning algorithms fail to deal with such scenario because (i) acquiring training instances is implemented via performing real-time bidding for the corresponding auctions; (ii) RTB requires the bidding agent to make real-time decisions for sequentially coming bid requests; (iii) cost for each ad impression will be unveiled only after responding a bid price and winning the auction; (iv) training data acquired in post-bid stage has a strong bias towards the won impressions. To the best of our knowledge, there is no such active learning setting based on bidding in previous literature. But it is of much importance in the scenario of RTB display advertising.

Based on the uniqueness of our problem, we propose an active learning framework for user response prediction in RTB, namely Bid-aware Active Real-time Bidding (BARB) to actively decide bid price for each ad impression. The main novelty of this work is:

- We propose an active learning framework to efficiently get the training data in the RTB scenario, where data comes as a stream and bid prices should be decided before checking any instances in the future. It is the first work combining active learning with a bidding strategy.
- DSP will pay for each ad impression to collect user feedback, so the monetized cost and the utility for the estimator are considered for each training instance. The algorithm uses uncertainty as utility and uses a product limit estimator to model market price distribution accurately.
- We embed unbiased learning into the active bidding framework, which can alleviate the data distribution bias caused by non-random instance selection.

II. RELATED WORKS

A. BIDDING STRATEGY

Typically the purpose of bidding algorithms is optimizing advertisers' KPI (Key Performance Indicator) within limited campaign budget. In [6], the authors claim there is a linear relation between predicted CTR (pCTR) and the optimal bid price, whereas the authors of [5] suggest there is a non-linear relation for the optimal bid with the impression level features and they propose an optimal bidding strategy which tries to bid more impressions rather than focus on a small set of high-value impressions. In [2], the authors suggest time-dependent models would be appropriate for capturing the repeated patterns and current bidding strategies are far less optimal, indicating the significant needs for optimization algorithms incorporating the facts such as the temporal behaviors, the frequency and recency of the ad displays. The authors of [7] proposed a bidding strategy managing risk in display advertising, i.e., the randomness of the user behaviors and the cost uncertainty. In recent years, reinforcement learning algorithms are widely applied to bidding strategies [8], [9]. As far as we know, there is no existing work of bidding strategy focused on user response prediction by bidding for data instance actively.

B. CLICK-THROUGH RATE ESTIMATION

In the cost-per-click (CPC) model, advertisers will be charged by publishers only if their advertisements are clicked by users [10]. In this mechanism, it is necessary for advertisers and publishers to estimate the click-through rate (CTR), i.e., the probability of an impression to be clicked by a user. When receiving a bid request, the DSP always evaluates the impression through the CTR estimation [5]. The authors of [10] suggest CTR prediction is absolutely crucial to sponsored search advertising because it impacts user experience, profitability of advertising and search engine revenue. The authors from [1], [11] claim that a CTR model should employ a fine-grained and real-time predictor. Recent works for CTR estimation mainly use different kinds of deep neural networks or combine them with factorization machine algorithm [12], [13].

There are a bunch of other works for user response prediction, most of which work on prediction task given a pool of instances and labels rather than actively choose instances from a stream. Our work designs a bidding strategy that can acquire data actively by giving different bid prices and predict user response with limited budgets.

C. ACTIVE LEARNING

Active learning (AL) is one of machine learning techniques for reducing annotation costs of acquiring training data used for prediction models [14]. Particularly, some AL algorithms focus on choosing instances with high uncertainty, which are near to the decision boundary of current prediction model and can improve the performance efficiently.

The most useful strategies of selecting uncertain instances includes query-by-committee [15]–[18], uncertainty-sampling, variance reduction, expected-error-reduction, expected-model-change and so on. In some AL applications, learning algorithms are not able to choose instances as much as they want within the limited budget for annotation. In [19], the authors describe a cost-sensitive AL algorithm which can effectively classify examples and consider misclassification cost. Some works [20]–[22] show that cost-sensitive decision-tree learning is a method to minimize total cost during classification.

Traditional AL mainly focuses pool-based instances [23], which it is able to cache training instances for selection and re-sampling. Another type is online AL [24], which chooses instances in real time. Stream-based AL is like online AL, but re-accessing instances is forbidden. Stream AL must make the decision immediately when an instance comes. To apply AL into RTB scenario, stream-based AL is naturally required, however since the market price is unknown unless the auction is won, it is hard to give a proper bid price to acquire instances. Thus the algorithm must take market price distribution modeling into consideration, which will be discussed in the next subsection.

D. MARKET PRICE DISTRIBUTION

The market price distribution often helps advertisers to decide bid price given the evaluated utility of the current bid request. Many works model the market price as a probabilistic variable because it is impossible to model the strategy of each individual bidder in the auction [25]. In [5], the authors proposed a simple but empirically effective fraction model. With the idea borrowed from financial market modeling, the market price is also always modeled using a log-normal distribution [26]–[29]. While some works model the market price estimation for a specific auction, works focusing on campaign-level [29] or publisher-level [11] are also common.

TABLE 1. Notations and descriptions.

Notation	Description
D, D'	Training data, test data.
$\boldsymbol{x}$	Bid request represented by its features.
$p_x(\boldsymbol{x})$	Probability density function of $\boldsymbol{x}$.
$q_x(\boldsymbol{x})$	*pdf of observed distribution of* $\boldsymbol{x}$.
$p_w(b)$	Winning probability of bid price b.
$p_l(b)$	Losing probability of bid price b.
$p_m(b)$	Probability mass function of market price b.
f	Online user response estimator.
$u(f, \boldsymbol{x})$	Utility of $\boldsymbol{x}$ for estimator f.
$\mathbb{E}[c\|b]$	Expected cost of bid price b.
$\mathbb{E}[c\|b, \text{win}]$	Expected cost of bid price b if the auction is won.

In previous works [4], [29], only if the campaign wins an auction, the observed market price is logged. However, for lost auctions, although we have not seen the exact market price, we know it is higher than the advertiser's bid price. Such censored data could be used to improve the accuracy. There is a non-parametric maximized likelihood estimation algorithm called *product limit estimation* which can be used to estimate the distribution of the market price given the bidding data from an advertiser's (DSP's) perspective [25], [30]. Some complicated recent works [31], [32] also deal with the right-censoring problem. Assuming market price only depends on the campaign, in our work a product limit estimator is employed to model right censored RTB market price data.

III. TASK FORMULATION

A. PROBLEM DEFINITION

To launch a campaign in display advertising, the advertisers upload ad creatives and set targeting rules (e.g. the user segmentation, time, location) and corresponding budget. After the target rules are set, the advertiser would first spend a small amount of budget to bid random impressions in order to learn some statistics before optimizing the bid. For example, as studied in [4], [33] the auction volume forecast (i.e., bid landscape prediction) module is usually employed to estimate auction statistics. Each bid request is represented by a high dimensional feature vector. For each campaign, previous records of bidding and feedback can be used to predict the probability of click or conversion of an impression.

The task of our work is to design a bidding agent that trains an optimal CTR estimator with limited budgets. Specifically, the bidding agent determines bid prices for bid requests and when it wins an auction, click information will be acquired at the cost of the corresponding market price, then the bid requests and the click information are both used for training the estimator.

B. NOTATIONS

The notation table is Table 1. There are some explanations. The **training data** is a sequence of records $D = \{D_t\}$,

$t = 1, \ldots, M$. Each record of data $D_t = (\boldsymbol{x}_t, z_t, y_t)$ consists of three fields:

- $\boldsymbol{x}_t \in \mathcal{X}$ is the bid request and ad creative features, which is the input for a bidding agent;
- $z_t \in \mathbb{R}^+$ is the winning price (usually called market price) of the corresponding auction, which is the cost for winning this impression;
- $y_t \in \{0, 1\}$ is the user feedback, i.e., whether the user clicks this ad impression.

The **test data** $D' = \{D'_t\}$, $t = 1, \ldots, M'$ is in the same form of the training data.

Note that in our work it is assumed that $p_w(b, \boldsymbol{x}) \equiv p_w(b)$. The assumption is reasonable as in the same campaign the winning probability depends on the bid price much more than on the bid request features. Previous bid optimization works [34], [35] for sponsored search also make such assumption on the winning keyword ad slots. With this assumption, it can be easily proved that expected cost $\mathbb{E}[c|b]$ also only depends on b.

Online **CTR estimator** $f : \mathcal{X} \mapsto [0, 1]$, which returns the predicted CTR (pCTR) $\hat{y}_t$ given the input features $\boldsymbol{x}_t$. After observing $(\boldsymbol{x}_t, y_t)$, f can be directly updated. Taking advantage of online learning, each instance of training data will be only used once, which meets the strict time requirement of RTB.

IV. ACTIVE REAL-TIME BIDDING

Defined in III-A, the problem of finding an optimal bidding agent is intractable. The obstacles and corresponding solutions are discussed below.

Firstly, it is hard to evaluate how any single instance in the training data affects the performance of the final estimator. Thus the utility function is used to tackle this problem, which helps measure the utility of the current auction to the estimator. Hence the task becomes finding a bidding agent that maximizes the sum of utility. A straightforward strategy of active learning is to select the instances with the highest utility-cost ratio from all the training data for cost-effective training. However, such a strategy is unsuitable for the streaming scenario in RTB. Another strategy is making a sequential bidding rule by taking a feedback loop and employing a dynamical optimization model such as partially observable Markov decision processes (POMDPs) [36]. However, such models are computationally expensive thus not feasible in RTB scenario where decisions need to be made in real time. Therefore an online active learning algorithm is crucial instead of traditional pool-based active learning algorithms.

Secondly, it is essential to take cost into consideration for cost-effective learning due to limited budget. RTB normally applies the second price auction, so the cost of an impression is closely related to market price distribution. Moreover, the winning probability of the bid price also depends on the market price. In our work, a product-limit estimator and a market price table are used to model the distribution of market prices, which will be explained in details in Sec. IV-B.

Thirdly, training data gathered in post-bid stage has a strong bias towards the won impressions. In our work unbiasedness factor is introduced to alleviate this problem.

In summary, the diagram for the algorithm is Fig. 2. Each component will be explained in next several subsections.

A. UTILITY AND COST-EFFECTIVENESS

In our work $u(f, \boldsymbol{x}_t)$ denotes the utility of input instance $\boldsymbol{x}_t$ for the estimator f if the label y_t is unveiled. Note that only when an auction is won, cost of the market price is spent and the label and utility is gained. Thus the task becomes to design a bidding agent that maximizes the sum of utility of training instances, which are acquired by winning auctions within the budget constraint.

The utility $u(f, \boldsymbol{x}_i)$ for estimation model has been well studied. In our work, we employ entropy function [37] representing the uncertainty of instance as the utility function, which is widely used in uncertainty-sampling strategy AL algorithms:

$$u_{\text{entropy}}(f, \boldsymbol{x}_t) = \sum_y -p(y|\boldsymbol{x}_t; f) \log(p(y|\boldsymbol{x}_t; f)) \qquad (1)$$

where $p(y|\boldsymbol{x}_t; f) = f(\boldsymbol{x}_t)^y (1 - f(\boldsymbol{x}_t))^{(1-y)}$ for $y = \{0, 1\}$ in binary classification. For notation simplicity, $u_{\text{entropy}}(f, \boldsymbol{x}_i)$ will be denoted as u without confusing readers.

In CTR estimation scenario, utility monotonically increases along with pCTR in most cases because pCTR is always less than 0.5. However, this doesn't mean the algorithm acts like a "greedy" algorithm seeking to obtain as much positive instances as possible (which could be a strategy though). The algorithm consider the winning probability, instance utility and expected cost integrally.

As discussed above, the bidding agent should determine bid prices instantly for streaming instances. Thus the optimal bid price for current instance comes from maximizing $\mathbb{E}[u_{\text{total}}|b]$. However computing this optimal bid price is also impossible because utility of instances in the future are unknown. To solve the problem we firstly define cost-effectiveness as the ratio of utility to cost. In particular, let ξ_{i-} be the previous overall cost-effectiveness before the instance $\boldsymbol{x}_t$.

$$\xi_{t-} = \frac{\sum_{i=1}^{t-1} u(f, \boldsymbol{x}_t)}{\sum_{i=1}^{t-1} c(b, \boldsymbol{x}_t)} \qquad (2)$$

Normally, during the training process of an estimator, the later training instances bring the model less utility. Hence we make a hypothesis that the future expected cost-effectiveness ξ_{t+} equals to α times of the previous overall cost-effectiveness. Typically the cost-effectiveness decay factor $\alpha \in [0, 1]$. The effect of α and how to choose it will be discussed in Section V.

$$\xi_{t+} = \alpha \xi_{t-} \,. \qquad (3)$$

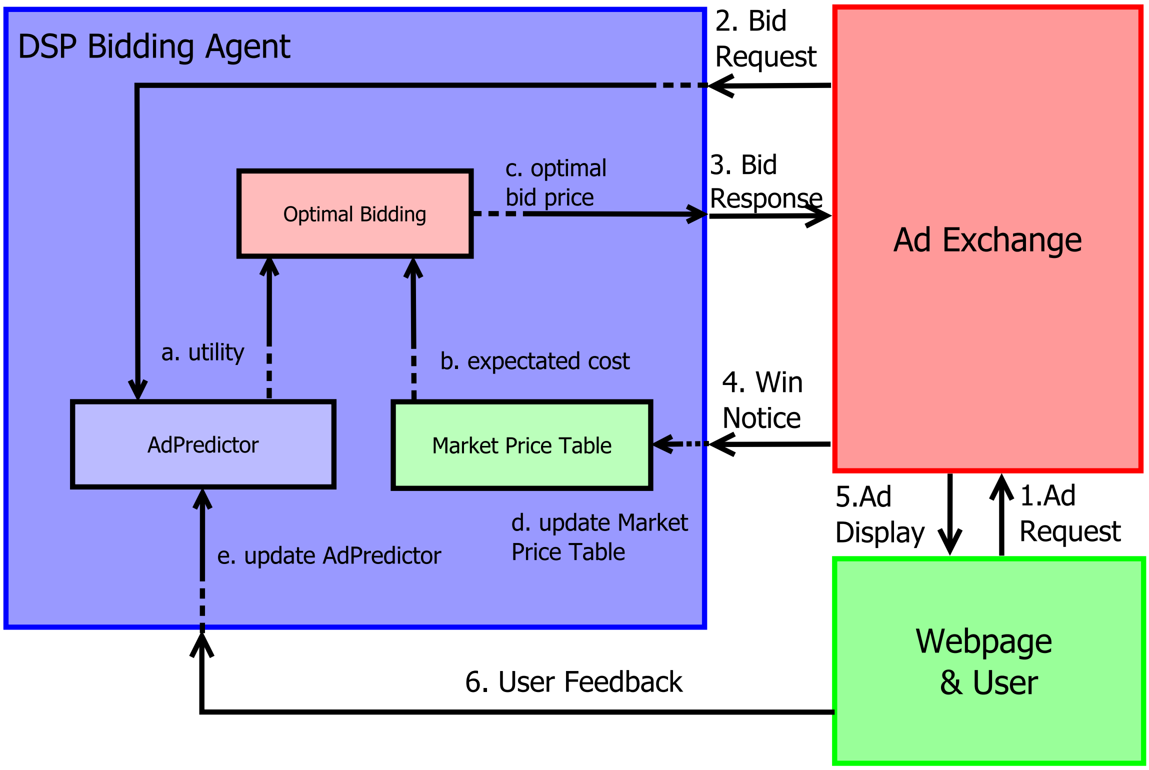

FIGURE 2. Diagram of BARB algorithm.

Denoting B as budget and C as total cost before observing $\boldsymbol{x}_t$, we can compute the expectation of total utility if price b is bid for x_t as below:

$$\begin{aligned}\mathbb{E}[u_{\text{total}}|b] &= p_w(b)[C\xi_{t-} + u + (B - C - \mathbb{E}[c|b, \text{win}])\xi_{t+}] \\ &\quad + (1 - p_w(b))[C\xi_{t-} + (B - C)\xi_{t+}] \\ &= p_w(b)(u - \mathbb{E}[c|b, \text{win}]\xi_{t+}) + \text{Const}\,. \qquad (4)\end{aligned}$$

where $\text{Const} = C\xi_{t-} + (B - C)\xi_{t+}$. The equation calculates expectation of total utility if the auction is won (the utility is added and the cost is subtracted) and the value if the auction is lost, multiplies winning probability and losing probability respectively and sum them up. Note that finding the optimal bid price b^* that maximizes $\mathbb{E}[u_{\text{total}}|b]$ involves calculating $p_w(b)$ and $\mathbb{E}[c|b, \text{win}]$, which will be discussed in the next subsection.

B. MARKET PRICE ESTIMATION

In previous works [4], [29], only observed market prices of the won auctions are used by the estimator. However, although exact market prices of lost auctions are unknown, in second price auction it is obvious that real market prices are higher than bid prices. Such censored data could be used to model market price distribution more precisely and improve the estimator through unbiased learning [30], [32]. There is a non-parametric maximized likelihood estimation algorithm called *product limit estimation* [38] which can be used to estimate the market price distribution given bidding data from a DSP's perspective [25], [30].

The bidding log is a list of N tuples $\langle b_t, w_t, c_t\rangle_{t=1\ldots M}$, where b_t is the bid price for auction t, w_t represents whether auction t is won, and c_t is the cost if $w_t = 1$. Now we transform the data into the form of $\langle b_i, d_i, n_i\rangle_{i=1\ldots N}$, where bid price $b_i < b_{i+1}$, d_i means the number of auctions won with market price b_i, n_i is the number of auctions which cannot be won with any bid price $b < b_i$, i.e., the auctions that are lost with a bid price not less than b and the auctions that are won with a market price not less than b. Note that in our work, we assume that the campaign will win the auction if there is a tie in the auction for mathematical convenience. In the case where an auction is lost when a tie occurs, the bidding agent could simply add a small constant to bid prices. Then the probability of losing / winning an auction with bid price b are:

$$p_l(b) = \prod_{b_i \le b} \frac{n_i - d_i}{n_i} \qquad (5)$$

$$p_w(b) = 1 - p_l(b) \qquad (6)$$

The list of bid prices in the transformed bidding log is denoted as BP, consisting of prices $b_1, b_2, \ldots, b_i, \ldots$ in an ascending order. For $b \in [b_k, b_{k+1})$, the relation of $p_m(b)$ and $p_w(b)$ is obvious:

$$p_w(b) = \sum_{b' \le b} p_m(b) = \sum_{i \le k} p_m(b_i) \qquad (7)$$

$$p_m(b_k) = p_w(b_k) - p_w(b_{k-1}) \qquad (8)$$

TABLE 2. An example of market price table, including 200 randomly generated instances with bid prices between 1 and 9.

b	$n(b)$	$d(b)$	$f(b)$	$p_l(b)$	$p_w(b)$	$\mathbb{E}[c\|b]$	$\mathbb{E}[c\|b, win]$
1	200	39	0.805	0.805	0.195	0.195	1.000
2	151	17	0.887	0.714	0.286	0.376	1.317
3	125	10	0.920	0.657	0.343	0.548	1.598
4	109	17	0.844	0.555	0.445	0.958	2.151
5	83	10	0.880	0.488	0.512	1.292	2.523
6	64	5	0.922	0.450	0.550	1.521	2.764
7	44	3	0.932	0.419	0.581	1.735	2.987
8	29	4	0.862	0.361	0.639	2.198	3.441
9	12	3	0.750	0.271	0.729	3.011	4.130

Then expected cost of bid price $b \in [b_k, b_{k+1})$ is:

$$\mathbb{E}[c|b] = \sum_{b' \le b} b' p_m(b') = \sum_{i \le k} b_i[p_w(b_i) - p_w(b_{i-1})] \quad (9)$$

Typically, DSPs are not charged for lost auctions, so expected cost of bid price b if the auction is won is:

$$\mathbb{E}[c|b, \text{win}] = \frac{\mathbb{E}[c|b]}{p_w(b)} = \frac{\sum_{i \le k} b_i[p_w(b_i) - p_w(b_{i-1})]}{p_w(b_k)} \quad (10)$$

In our work, we maintain useful information for modeling market price distribution in a table like Table 2, namely the market price table. To build the market price table, firstly the bidding log is transformed from $\langle b_t, w_t, c_t\rangle_{t=1...M}$ into $\langle b_i, d_i, n_i\rangle_{i=1...N}$ then we have $n(b_i)$ and $d(b_i)$. Secondly we define $f(b)$ as Eq. (11) thus we can simplify the calculation of $p_l(b)$ into a recursive style as Eq. (12). Thirdly $\mathbb{E}[c|b]$ in Eq. (9) can also be calculated recursively:

$$f(b_i) = \frac{n(b_i) - d(b_i)}{n(b_i)} \quad (11)$$

$$p_l(b_i) = p_l(b_{i-1}) \cdot f(b_i) \quad (12)$$

$$\mathbb{E}[c|b_i] = \mathbb{E}[c|b_{i-1}] + b_i[p_w(b_i) - p_w(b_{i-1})] \quad (13)$$

where $p_l(b_0) = 1$ and $\mathbb{E}[c|b_0] = 0$. Moreover, $p_w(b_i)$ and $\mathbb{E}[c|b_i, \text{win}]$ can be calculated easily using Eq. (6) and Eq. (10).

In RTB scenario, bid requests will come as a stream, so the market price table should be updated incrementally. Whenever a bidding log entry comes, the table should be updated in two steps. In the first step n, d and f are updated. If $b \notin BP$ we insert b into the proper position as b_k and set $d(b_k) = 0$, $n(b_k) = n(b_{k-1})$. Note that here b denotes the bid price if the auction is lost or the market price otherwise. Then add one to $n(b_i)$ for all $i \le k$ and if the auction is won, add one to $d(b_k)$. At last, $f(b_i)$ for each $i \le k$ is re-calculated as Eq. (11). In the second step $p_l(b_i)$, $p_w(b_i)$, $\mathbb{E}[c|b_i]$ and $\mathbb{E}[c|b_i, \text{win}]$ are updated recursively as Eq. (12)(6)(13) and (10) for each $b_i \in BP$ in ascending order.

C. OPTIMAL BIDDING

From Sec. IV-A we know that the optimal price for a bid request maximizes $p_w(b)(u - \mathbb{E}[c|b, \text{win}]\xi_{i+})$, in which $p_w(b)$ and $E[c|b, \text{win}]$ can be calculated using the method in Sec. IV-B. As bid price b increases, $p_w(b)$ increases and $u - \mathbb{E}[c|b, \text{win}]\xi_{i+}$ decreases, thus their product firstly increases to a peak value and then decreases as b increases. The relations are shown in Fig. 4 and discussed in Sec. V-C.

From Eq. (5) and (6), we can see that for all $b \in [b_k, b_{k+1})$, $p_w(b)$ are the same and so are $\mathbb{E}[c|b, \text{win}]$. Therefore for all $b \in [b_k, b_{k+1})$, $p_w(b)(u - \mathbb{E}[c|b, \text{win}]\xi_{i+})$ are also the same. Thus to find the optimal bid price we can just calculate $p_w(b_k)(u - \mathbb{E}[c|b_k, \text{win}]\xi_{i+})$ for all $b_k \in BP$ and find the price b_{k^*} that maximizes it, then the optimal bid price lies in $[b_{k^*}, b_{k^*+1})$. Thus the bidding agent randomly gives a bid price $b \in [b_{k^*}, b_{k^*+1})$.

$$b_{k^*} = \arg\max_{b_k \in BP}[p_w(b_k)(u - \mathbb{E}[c|b_k, \text{win}]\xi_{t+})] \quad (14)$$

$$b^*_{\boldsymbol{x}_t} \in [b_{k^*}, b_{k^*+1}) \quad (15)$$

Till now, the algorithm can calculate optimal the bid price $b_{\boldsymbol{x}_t}$ as Eq. (14)(15) for x_t given the estimator f. In the next subsection, an online CTR estimator AdPredictor is introduced and a modification is employed for unbiased learning.

D. CTR ESTIMATION MODEL

1) AdPredictor

In order to actively and quickly update the model f, an online learning based CTR estimation model is necessary in our task. A popular online regression model is Bayesian probit regression [10], [39]. In our work, we implement f using the widely-used AdPredictor in [10]. Different from conventional models like logistic regression which periodically retrains the model, Bayesian probit regression updates the posterior distribution of its parameters whenever new data is observed. Such property enables Bayesian probit regression to perform online learning and avoid retraining. Basically the probit regression function is

$$P(y_t|\boldsymbol{x}_t, \boldsymbol{w}) = \Phi(y_t \boldsymbol{x}_t^T \boldsymbol{w}) \quad (16)$$

where $\Phi(\theta) = \int_{-\infty}^{\theta} N(s; 0, 1)ds$ is the cumulative standard Gaussian distribution. The model parameter w is assumed to be drawn from a Gaussian distribution, which originates from a prior and is updated with the posterior distribution by data observation.

$$p(\boldsymbol{w}|\boldsymbol{x}_t, y_t) \propto P(y_t|\boldsymbol{x}_t, \boldsymbol{w})N(\boldsymbol{w}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}) \quad (17)$$

The posterior is non-Gaussian and it is usually solved via variational methods in practice. Let $N(\boldsymbol{w}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ be the posterior distribution of $\boldsymbol{w}$. Variational inference tries to find the optimal parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ which minimises Kullback-Leibler divergence:

$$(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \underset{(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{arg\,min}\; \mathbf{KL}\Big(\Phi(y_t \boldsymbol{x}_t^T \boldsymbol{w})N(\boldsymbol{w}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})||N(\boldsymbol{w}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)\Big) \quad (18)$$

Consider up to the second-order factors, the close form of this optimization problem is (derivation omitted):

$$\mu_t = \mu_{t-1} + \alpha \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t \quad (19)$$

$$\Sigma_t = \Sigma_{t-1} - \gamma(\boldsymbol{\Sigma}_{t-1}\boldsymbol{x}_t)(\boldsymbol{\Sigma}_{t-1}\boldsymbol{x}_t)^T \quad (20)$$

where

$$\alpha = \frac{y_t}{\sqrt{\boldsymbol{x}_t^T \boldsymbol{\Sigma}_t \boldsymbol{x}_t + \beta^2}} \frac{N(\theta)}{\Phi(\theta)} \tag{21}$$

$$\gamma = \frac{1}{\sqrt{\boldsymbol{x}_t^T \boldsymbol{\Sigma}_t \boldsymbol{x}_t + \beta^2}} \frac{N(\theta)}{\Phi(\theta)} (\frac{N(\theta)}{\Phi(\theta)} + \theta) \tag{22}$$

$$\theta = \frac{y_t \boldsymbol{x}_t^T \boldsymbol{\mu}_{t-1}}{\sqrt{\boldsymbol{x}_t^T \boldsymbol{\Sigma}_{t-1} \boldsymbol{x}_t + \beta^2}}. \tag{23}$$

Here β is a hyper-parameter of AdPredictor algorithm. Following [10] we can assume the independence between each two features and only focus on the diagonal elements in $\boldsymbol{\Sigma}_t$ in practice.

2) UNBIASED LEARNING

A common problem for online active learning is the data distribution bias due to non-randomly instance selection. From [30], [37], there is an unbiasedness factor $\delta_t = \frac{p_x(\boldsymbol{x}_t)}{q_x(\boldsymbol{x}_t)}$ introduced, which means the ratio of probability density function of the real distribution to that of the observed distribution for $\boldsymbol{x}_t$. When parameters are updated, the learning rate could be multiplied by δ_t for unbiased learning.

In RTB scenario, instances with higher bid prices will have higher probability to win the auctions and be selected as training data, which can mathematically represented by Eq. (24), thus we can derive bid-aware unbiasedness factor as Eq. (25):

$$q_x(\boldsymbol{x}_t) \propto p_w(b_{\boldsymbol{x}_t}) p_x(\boldsymbol{x}_t) \tag{24}$$

$$\delta_t = \frac{p_x(\boldsymbol{x}_t)}{q_x(\boldsymbol{x}_t)} = \frac{N_{\text{win}}/N_{\text{total}}}{p_w(b_{\boldsymbol{x}_t})} \tag{25}$$

where $p_x(\boldsymbol{x}_t)$ and $q_x(\boldsymbol{x}_t)$ are respectively the probability density function of the real distribution and the observed distribution of $\boldsymbol{x}_t$, N_{total} is the total number of all the auctions, N_{win} is the number of the won auctions, $\frac{N_{win}}{N_{total}}$ is the normalization constant that equals to $\int_{\boldsymbol{x}} p_w(b_{\boldsymbol{x}_t}) p_x(\boldsymbol{x}_t) d\boldsymbol{x}_t$.

Hence we can update the model with less bias using δ_t: whenever one instance is observed with $\delta_t \in (0, 1)$, the parameters of AdPredictor are approximately updated as Eq. (26) and Eq. (27). When one instance with $\delta_t > 1$ is observed, we can update once as Eq. (19) and Eq. (20) and then re-update the parameters recursively with an observation of unbiasedness factor $\delta_{t'} = \delta_t - 1$.

$$\mu_i = \mu_{i-1} + \alpha \delta_i \boldsymbol{\Sigma}_{i-1} \boldsymbol{x}_i \tag{26}$$

$$\Sigma_i = \Sigma_{i-1} - \gamma \delta_i (\boldsymbol{\Sigma}_{i-1} \boldsymbol{x}_i)(\boldsymbol{\Sigma}_{i-1} \boldsymbol{x}_i)^T \tag{27}$$

Note that the unbiasedness factor requires little extra calculation because $p_w(\boldsymbol{x}_t)$ is already known. The unbiased parameter updating can be seemed as a kind of over / under-sampling strategies, which are widely used in active learning for imbalanced classification [40], [41]. In our unbiased learning, the strategy is used to recover the original distribution of feature vectors from censored distribution caused by non-randomly selection rather than break the class distribution. Thus the Bid-aware Active Real-time Bidding (BARB) algorithm is complete, we can train an unbiased online CTR estimator with active bidding strategy which makes use of uncertainty utility and market price estimator. In summary, the pseudo code of BARB is written in Algorithm 1.

Algorithm 1 Active Real-Time Bidding

```
Initialize: market price table, estimator f, budget constraint B and cost-effectiveness ξ
for all input case x_t in sequence do
    Calculate b_{k*} by solving Eq. (14)
    Bid price b*_{x_t} randomly sampled from [b_{k*}, b_{k*+1}) for x_t

    if Win x_t then
        Receive the market price z_t and user click y_t
        Update f using Eq. (19)(20)
        Update ξ as Eq. (2)(3)
        Update Budget B
        if Budget B runs out then
            break
        end if
    end if
    Update the market price table
end for
```

Normally there are only several hundreds of integer prices in the bid price action space [5]. In addition, Eq. (12) and Eq. (13) can both be updated recursively, therefore it is quite efficient to find the optimal price. The time complexity is $O(L)$ where L denotes the size of BP, which gradually becomes steady as auctions proceed. One can increase the granularity of bid prices to have a smaller BP and speed up the algorithm further.

V. EXPERIMENTS

In this section, we present the detailed experiment settings and the corresponding results. We also publish our code for reproductive experiment.[1]

A. ESTIMATOR PERFORMANCE

1) DATASETS

We have experiments on three real-world datasets:

iPinYou [42] dataset comes from a real-world bidding feedback log from a well-known DSP company. It records more than 15M impressions and the user feedback of 9 campaigns (namely 1458, 2259, 2261, 2821, 2997, 3358, 3386, 3427 and 3476) from different advertisers during ten days in 2012 and 2013. Each log entry of the market price and the bid request containing the information of the user, advertiser, publisher and the context.

Criteo [43] is a pioneering company in online advertising research. They have published this dataset

[1] Anonymous code link: http://bit.ly/2RChFN5.

TABLE 3. Performance of different bidding strategies for each campaign with different budget limits.

Budget	Algo.	1458	2259	2261	2821	2997	3358	3386	3427	3476	Criteo	YOYI
1/256	Rand	0.58333	0.61225	0.54623	0.55117	0.55946	0.65779	0.6791	0.58321	0.57483	0.55861	0.79768
	Const	0.5745	0.61678	**0.55612**	0.55176	**0.57706**	0.66101	0.67259	0.57721	0.56984	0.56102	0.79689
	LIN	0.58617	0.62534	0.55599	0.53788	0.57695	0.56582	0.59036	0.54888	0.57073	0.46657	0.73024
	BARB	**0.60881**	**0.62863**	0.55597	**0.56524**	0.57594	**0.67101**	**0.68053**	**0.58928**	**0.57512**	**0.56158**	**0.83107**
1/128	Rand	0.60922	0.62211	0.55683	0.57345	0.57362	0.67247	0.6576	0.62056	0.56269	0.55650	0.82019
	Const	0.60401	0.61332	**0.55979**	0.5519	0.57706	0.67633	0.67228	0.6474	0.56281	0.56073	0.81936
	LIN	0.61101	0.64545	0.55683	0.55661	0.57695	0.62465	0.59036	0.56137	**0.58065**	0.45391	0.73687
	BARB	**0.63005**	**0.64586**	0.55523	**0.57459**	**0.57786**	**0.69899**	**0.67895**	**0.65092**	0.56875	**0.57296**	**0.84895**
1/64	Rand	0.61853	0.63621	0.53171	0.56927	0.56992	0.69439	0.62998	0.60776	0.54391	0.58243	0.84198
	Const	0.62464	0.63885	0.56222	0.57938	**0.57706**	0.67672	0.62755	0.59325	0.54517	0.58378	0.84115
	LIN	0.61831	0.63109	0.56336	0.55222	0.57695	0.71024	0.61692	0.58524	**0.55394**	0.44829	0.74675
	BARB	**0.63061**	**0.64507**	**0.58253**	**0.58434**	0.57588	**0.72014**	**0.63156**	**0.61968**	0.55041	**0.58692**	**0.86034**

for attribution modeling in real-time auction based advertising [43]. The includes more than 16M impressions and 45K conversions over 700 campaigns. Impressions may derive conversions so each instance has a label indicating whether a conversion has occurred. In reality such information of conversion may arrive quite late, but the experiment shows that our algorithm is able to predict user feedback from sparse positive instances.

YOYI [44] is another mainstream DSP, which mainly focuses on multi-device display advertising. The dataset comprises 441.7M impressions, 416.9K clicks and 319.5K CNY expense during 8 days in Jan. 2016. The first 7 days are set as training data and the last day is set as test data.

2) BASELINE METHODS

Since as far as we know there is no existing work of bidding strategy focused on user response prediction by bidding for data instance actively, we choose several widely used bidding strategies as baseline methods embedded with AdPredictor.

- Random bidding agent: The first baseline strategy randomly gives bid prices lower than the given upper bound.
- Aggressively bidding agent: The second baseline strategy always bids the same constant price for all instances until budget runs out.
- Linear bidding agent: The third baseline strategy is from [6]. It gives bid price $b = b_0 \frac{\theta(x_t)}{\theta_{avg}}$ where $\theta(x_t)$ is the pCTR given by CTR estimator, θ_{avg} is average CTR and b_0 is the parameter of the bidding agent.

3) RESULTS AND DISCUSSION

In this experiment, an AdPredictor is employed as the prediction model by BARB strategy and all three other baseline strategies. In details, β is 0.05, prior probability is 0.01 and unbiasedness factors are clipped to no more than 5. The experiment is conducted with different hyper-parameters: *max bid price* for Rand strategy, *fixed bid price* for Const strategy, *linearfactor* for LIN strategy and *decay factor* α for BARB.

We use AUC as our experimental evaluation measurement, which is a widely used for evaluation of binary classification problems and CTR estimator in computational advertising especially [10], [45]. Table 3 shows the performance of different bidding strategies of for each campaign with different budget limits: 1/256, 1/128 and 1/64 of the total budget. Table 3 shows that our proposed BARB algorithm outperforms other three bidding strategies in most settings.

In the cases where the budget is abundant, the baseline bidding strategies with high parameters are able to acquire more training instances by giving high bid prices and the less censored distribution is more similar to the original distribution. In these cases the strategies could possibly compete or outperform our proposed algorithm. In our future work, we could design a more flexible strategy based on BARB, which will bid higher prices in the case when the warming-up budget is abundant. One possible way of doing so is to take the length of warming-up stage lifetime into consideration, adding statistics such as the ratio of the budget spent and the ratio of the lifetime come through into the calculation of total utility expectation.

Note that LIN strategy which performs quite well in other scenarios didn't have excellent performance in our experiments because the strategy is normally not used in a streaming scenario. The performance is greatly affected by the accuracy of estimator severely in the early stage. An estimator with bad random initialization will slow down the training process or even make itself worse, i.e. on Criteo dataset LIN strategy gets worse performance as the budget increases. Our BARB strategy also suffers from the same problem, but it can get rid of the problem by taking uncertainty utility into consideration. On the contrary, Const strategy and Random strategy will not be affected by an estimator with bad initialization and low accuracy.

We can also see that performance of CTR estimator on the whole becomes better with more budget by acquiring more training data. In addition, such improvement is more obvious when budget is less, which verifies our view that the later training instances bring less utility in the training process of a model. Fig. 3 shows that the average entropy of the prediction model will go lower while the training proceeds (on five campaigns of iPinYou dataset), which could partly

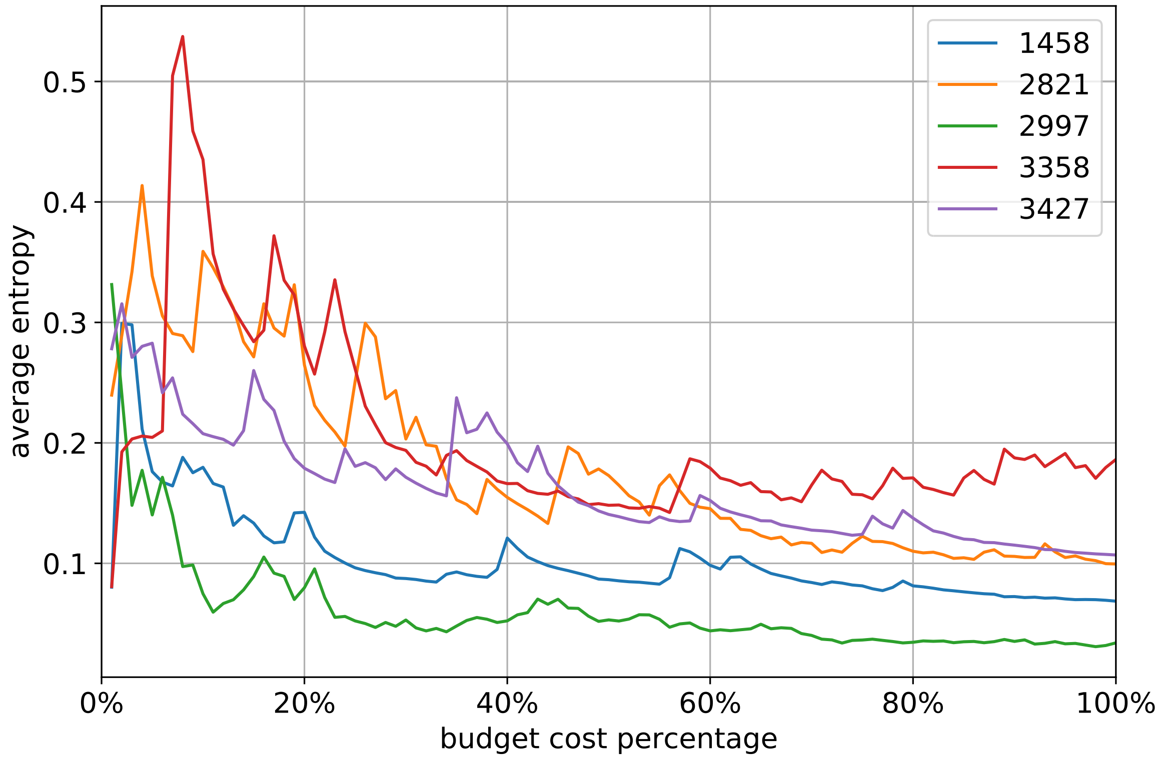

FIGURE 3. Entropy decay during traing process of BARB algorithm on 5 campaigns of iPinYou dataset.

TABLE 4. Example of market price distribution comparison.

b	$n(b)$	$d(b)$	$p_w^r(b)$	$p_w^p(b)$	$p_w^o(b)$	$\mathbb{E}^r[c\|b]$	$\mathbb{E}^p[c\|b]$	$\mathbb{E}^o[c\|b]$	$\mathbb{E}^r[c\|b,\text{win}]$	$\mathbb{E}^p[c\|b,\text{win}]$	$\mathbb{E}^o[c\|b,\text{win}]$
1	10000	1123	0.111	0.112	0.120	0.111	0.112	0.120	1.000	1.000	1.000
2	4459	568	0.222	0.225	0.210	0.333	0.338	0.327	1.500	1.502	1.553
3	1861	278	0.333	0.341	0.337	0.667	0.686	0.670	2.000	2.010	1.988
4	778	116	0.444	0.439	0.434	1.111	1.079	1.049	2.500	2.455	2.414
5	313	52	0.556	0.532	0.556	1.667	1.544	1.609	3.000	2.900	2.893
6	127	29	0.667	0.639	0.596	2.333	2.185	2.057	3.500	3.418	3.452
7	41	9	0.778	0.718	0.805	3.111	2.739	2.948	4.000	3.813	3.661
8	17	7	0.889	0.834	0.912	4.000	3.667	4.324	4.500	4.395	4.742
9	7	7	1.000	1.000	1.000	5.000	5.157	5.409	5.000	5.157	5.409
RMSE	×	×	×	0.0296	0.0272	×	0.1864	0.2061	×	0.0999	0.2013

reflect the point discussed above. Note that the entropy decay cannot fully reflect the decay of utility for estimator, that is the reason why we introduce cost-effectiveness decay factor.

B. MARKET PRICE DISTRIBUTION

To validate the product limit estimator (PLE) of market price distribution in our algorithm, we calculate the values of $p_w(b)$, $\mathbb{E}[c|b]$ and $\mathbb{E}[c|b, \text{win}]$ using PLE and observation statistics model, comparing them with theoretical real values given data distribution. In the experiments of this subsection, the market prices are generated from a uniform distribution from 1 to 9 and the bid prices are generated with a exponential decaying probability, which can evidently reflect the right censoring of RTB data.

For simplification we use superscript r for the real values, p for the values calculated by PLE and o for the values calculated by observation statistics model. One example of such comparison is shown in Table 4. The bottom line of the table of RMSE of two methods respectively for p_w, $\mathbb{E}[c|b]$ and $\mathbb{E}[c|b, \text{win}]$ compared to theoretical real values. We conduct several experiments with six different numbers of instances (from 10 to 10^6). The result shown in Table 5 obviously proves that the PLE model outperforms the observation statistics model.

Note that the experiment is based on the assumption that market prices are only dependent on the campaign (i.e., campaign-level price modeling). Recent complicated works [31], [32] (impression-level price modeling) could perform better in real market environment with enough training instances. However, in the scenario of BARB, non-parametric PLE is capable of reducing error fast with much fewer instances as shown in Table 5.

C. OPTIMAL BIDDING

We calculate the relation between b and (a) $p_w(b)$, (b) $\mathbb{E}[c|b]$ and $\mathbb{E}[c|b, \text{win}]$, (c) $u - \mathbb{E}[c|b, \text{win}]\xi$ and (d) $\mathbb{E}[u|b]$

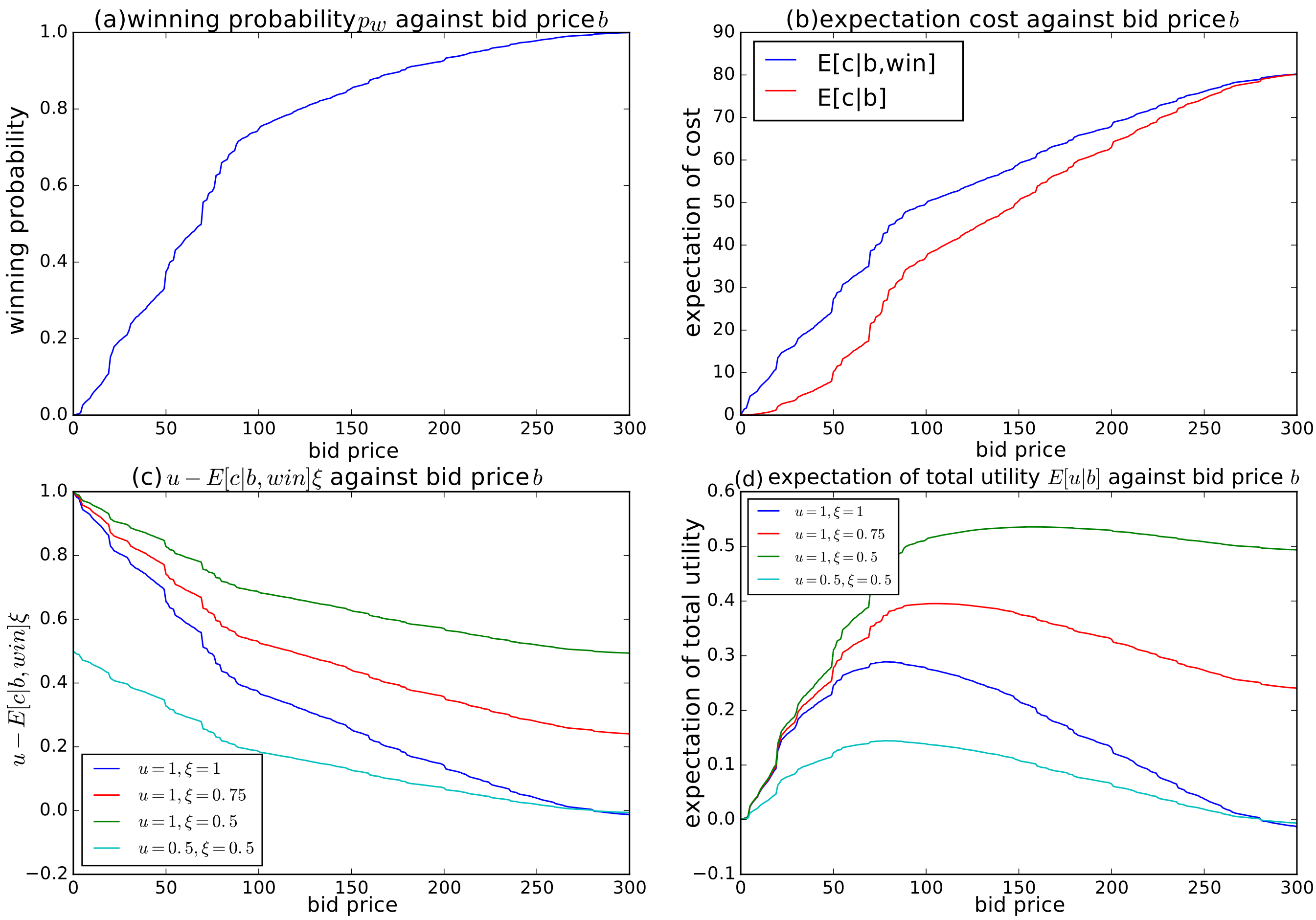

FIGURE 4. **Relation between b and corresponding (a) p_w, (b) $\mathbb{E}[c|b]$ and $\mathbb{E}[c|b, win]$, (c) $u - \mathbb{E}[c|b]\xi$ and (d) $\mathbb{E}[u_{total}|b]$, based on statistics of iPinYou dataset.**

TABLE 5. **RMSE Comparison of two methods for market price distribution modeling, N represents the number of auctions.**

N	p_w^p	p_w^o	$\mathbb{E}^p[c\|b]$	$\mathbb{E}^o[c\|b]$	$\mathbb{E}^p[c\|b, \text{win}]$	$\mathbb{E}^o[c\|b, \text{win}]$
10	**0.200**	0.345	**0.469**	0.983	**0.949**	1.975
100	**0.090**	0.195	**0.477**	0.511	**0.392**	1.126
10^3	**0.049**	0.074	0.432	**0.423**	**0.259**	0.406
10^4	0.030	**0.027**	**0.186**	0.206	**0.100**	0.201
10^5	**0.004**	0.013	**0.024**	0.088	**0.019**	0.063
10^6	**0.002**	0.003	**0.011**	0.024	**0.007**	0.024

respectively using iPinyou dataset for statistics. The results are shown in the Fig. 4. Specifically, it shows the relation between b and (c) $u - \mathbb{E}[c|b, \text{win}]\xi$ and (d) $\mathbb{E}[u|b]$ with different settings of synthetic u and ξ.

As b increases, $p_w(b)$ increases and $u - \mathbb{E}[c|b, \text{win}]\xi$ decreases, thus their product firstly increases to a peak value and then decreases as b increases. From Fig. 4(d) we can see that with the same p_w and u, the smaller ξ is, the higher the optimal bid price is (the optimal bid prices are about 80, 100 and 150 for $\xi = 1, 0.75, 0.5$).

The reason is, when ξ is small, $u - \mathbb{E}[c|b_k, \text{win}]\xi$ is hardly effected by b, then the optimal bid price is more likely to be high. Because in Eq. (14), ξ_+ is computed as Eq. (3), when α is close to 1, ξ decays slowly and bid prices are high, which probably perform well when the budget is abundant. On the contrary, when ξ is big, the optimal bid price is more likely to be low because the lower b is, the less $\mathbb{E}[c|b, \text{win}]$ is. Therefore a small α leads to low bid prices and the algorithm could possibly perform well when the budget is inadequate. Our experiments shows that α around an empirical value 0.9 is likely to lead to a relatively accurate estimator.

VI. CONCLUSION

Our work proposes a Bid-aware Active Real-time Bidding algorithm (BARB), which can efficiently and actively get training data from streaming instances in RTB scenario. Combined with active learning and market price distribution modeling, the strategy gives optimal bid prices, and trains an unbiased online CTR estimator. The empirical study on three real-world datasets with three settings of budget constraints shows our proposed algorithm outperforms three other bidding strategies in common use.

The work is the first to apply active learning into RTB. The framework is easy to be deployed and flexible to change the components. For example, different utility functions in active learning may be used instead of the entropy function, the market price distribution can be modeled with more complicated methods and the estimator could be changed to other (online) learning algorithms like FTRL. The future works can work on different components' combination and usefulness, deeper study on the utility for estimator and the decay of cost-effectiveness during training process.

ACKNOWLEDGMENT

The corresponding author Y. Yu and the team thank the help of X. Wu, who is pursuing his Ph.D. at Santa Clara University.

REFERENCES

[1] S. Muthukrishnan, "Ad exchanges: Research issues," in *Internet and Network Economics*. Berlin, Germany: Springer, 2009, pp. 1–12.
[2] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: Measurement and analysis," in *Proc. ADKDD*, 2013.
[3] K.-C. Lee, B. Orten, A. Dasdan, and W. Li, "Estimating conversion rate in display advertising from past performance data," in *Proc. KDD*, 2012, pp. 768–776.
[4] Y. Cui, R. Zhang, W. Li, and J. Mao, "Bid landscape forecasting in online ad exchange marketplace," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, New York, USA. ACM Press, Aug. 2011, p. 265.
[5] W. Zhang, S. Yuan, and J. Wang, "Optimal real-time bidding for display advertising," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2014.
[6] C. Perlich, B. Dalessandro, R. Hook, O. Stitelman, T. Raeder, and F. Provost, "Bid optimizing and inventory scoring in targeted online advertising," in *Proc. KDD*, pp. 804–812, 2012.
[7] H. Zhang, W. Zhang, J. Wang, and Y. Rong, "Managing risk of bidding in display advertising," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2017.
[8] H. Cai, K. Ren, W. Zhang, K. Malialis, and D. Guo, "Real-time bidding by reinforcement learning in display advertising," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., 2017.
[9] W. Di, X. Chen, Y. Xun, W. Hao, Q. Tan, X. Zhang, and K. Gai, "Budget constrained bidding by model-free reinforcement learning in display advertising," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Turin, Italy, 2018.
[10] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 13–20.
[11] Y. Chen, P. Berkhin, B. Anderson, and N. R. Devanur, "Real-time bidding algorithms for performance-based display ad allocation," in *Proc. KDD*, 2011.
[12] Y. Qu, C. Han, R. Kan, W. Zhang, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Barcelona, Spain, 2017.
[13] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, Australia, 2017.
[14] J. Attenberg and S. Ertekin, "Class imbalance and active learning," *Imbalanced Learning: Foundations, Algorithms, and Applications*, IEEE, 2013, pp. 101–149.
[15] Y. Freund, "Sifting informative examples from a random source," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 85–89.
[16] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. ICML*, vol. 95, 1995, pp. 150–157.
[17] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Information, prediction, and query by committee," in *Proc. Adv. Neural Inf. Process. Syst.*, 1993, pp. 483–490.
[18] K. Tomanek and U. Hahn, "Reducing class imbalance during active learning for named entity annotation," in *Proc. 5th Int. Conf. Knowl. Capture*, 2009, pp. 105–112.
[19] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011.
[20] P. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *J. Artif. Intell. Res.*, vol. 2, pp. 369–409, 1995.
[21] V. S. Sheng, C. X. Ling, A. Ni, and S. Zhang, "Cost-sensitive test strategies," in *Proc. Nat. Conf. Artif. Intell.*, vol. 21. Cambridge, MA, USA: MIT Press, 2006, p. 482.
[22] S. Esmeir and S. Markovitch, "Anytime induction of cost-sensitive trees," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 425–432.
[23] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Springer-Verlag, 1994, pp. 3–12.
[24] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, 1994.
[25] K. Amin, M. Kearns, P. Key, and A. Schwaighofer, "Budget optimization for sponsored search: Censored learning in MDPs," in *Proc. UAI*, 2012.
[26] A. Ghosh, P. McAfee, K. Papineni, and S. Vassilvitskii, "Bidding for representative allocations for display advertising," in *Internet and Network Economics*. Berlin, Germany: Springer, 2009, pp. 208–219.
[27] S. Lahaie and R. P. McAfee, "Efficient ranking in sponsored search," in *Internet and Network Economics*. Berlin, Germany: Springer, 2011, pp. 254–265.
[28] M. Ostrovsky and M. Schwarz, "Reserve prices in Internet advertising auctions: A field experiment," in *Proc. 12th ACM Conf. Electron. Commerce*, 2011, pp. 59–60.
[29] K. J. Lang, B. Moseley, and S. Vassilvitskii, "Handling forecast errors while bidding for display advertising," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 371–380.
[30] W. Zhang, T. Zhou, J. Wang, and J. Xu, "Bid-aware gradient descent for unbiased learning with censored data in display advertising," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 665–674.
[31] C. H. Wu, M. Y. Yeh, and M. S. Chen, "Predicting winning price in real time bidding with censored data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015.
[32] Y. Wang, R. Kan, W. Zhang, J. Wang, and Y. Yong, "Functional bid landscape forecasting for display advertising," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases (ECML-PKDD)*, Riva del Garda, Italy, 2016.
[33] S. Cetintas, D. Chen, and L. Si, "Forecasting user visits for online display advertising," *Inf. Retr.*, vol. 16, no. 3, pp. 369–390, 2013.
[34] B. Kitts and B. LeBlanc, "A trading agent and simulator for keyword auctions," in *Proc. Joint Conf. Auto. Agents*, 2004, pp. 228–235.
[35] W. Zhang, Y. Zhang, B. Gao, Y. Yu, X. Yuan, and T.-Y. Liu, "Joint optimization of bid and budget allocation in sponsored search," in *Proc. KDD*, 2012, pp. 1177–1185.
[36] S. Yuan and J. Wang, "Sequential selection of correlated ads by POMDPs," in *Proc. CIKM*, 2012, pp. 515–524.
[37] W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng, "Unbiased online active learning in data streams," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 195–203.
[38] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *J. Amer. Stat. Assoc.*, vol. 53, no. 282, pp. 457–481, Jun. 1958.
[39] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang, "Learning click models via probit Bayesian inference," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 439–448.
[40] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 39, no. 2, pp. 539–550, Apr. 2009.
[41] J. Zhu and E. H. Hovy, "Active learning for word sense disambiguation with methods for addressing the class imbalance problem," in *Proc. EMNLP-CoNLL*, vol. 7, 2007, pp. 783–790.
[42] W. Zhang, S. Yuan, J. Wang, and X. Shen, "Real-time bidding benchmarking with iPinYou dataset," 2014, *arXiv:1407.7073*. [Online]. Available: https://arxiv.org/abs/1407.7073
[43] D. Eustache, M. Julien, P. Galland, and D. Lefortier, "Attribution modeling increases efficiency of bidding in display advertising," in *Proc. AdKDD TargetAd Workshop (KDD)*, Halifax, NS, Canada, Aug. 2017.
[44] K. Ren, W. Zhang, Y. Rong, H. Zhang, Y. Yu, and J. Wang, "User response learning for directly optimizing campaign performance in display advertising," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 679–688.
[45] R. J. Oentaryo, E. P. Lim, D. J. W. Low, D. Lo, and M. Finegold, "Predicting response in mobile advertising with hierarchical importance-aware factorization machine," in *Proc. WSDM*, 2014.

SHUHAO LIU received the B.Eng. degree from the Department of Computer Science, Shanghai Jiao Tong University, where he is currently pursuing the master's degree. He is a member of the Data Mining Group, Apex Data and Knowledge Management Laboratory. His research interests include computational advertisement and recommendation systems.

YONG YU received the M.Sc. degree in computer science from East China Normal University, Shanghai, China, in 1986. He is currently a Ph.D. Candidate Tutor and the Chairman of the E-Generation Technology Research Center, Shanghai Jiao Tong University (SJTU), Shanghai. His research interests include semantic web, web mining, information retrieval, and computer vision. He was the Head Coach of the SJTU ACM-ICPC Team, and his team were the recipients of the top prize of the ACM ICPC Championships in 2002, 2005, and 2010.