Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising*

Weinan Zhang[1], Tianxiong Zhou[2,4], Jun Wang[2,3], Jian Xu[5]
[1]Shanghai Jiao Tong University, [2]University College London,
[3]MediaGamma Limited, [4]TukMob Inc., [5]TouchPal Inc.
{w.zhang, j.wang}@cs.ucl.ac.uk, tianxiong.zhou@tukmob.com, jian.xu@cootek.cn

ABSTRACT

In real-time display advertising, ad slots are sold per impression via an auction mechanism. For an advertiser, the campaign information is incomplete — the user responses (e.g, clicks or conversions) and the market price of each ad impression are observed only if the advertiser's bid had won the corresponding ad auction. The predictions, such as bid landscape forecasting, click-through rate (CTR) estimation, and bid optimisation, are all operated in the pre-bid stage with full-volume bid request data. However, the training data is gathered in the post-bid stage with a strong bias towards the winning impressions. A common solution for learning over such censored data is to reweight data instances to correct the discrepancy between training and prediction. However, little study has been done on how to obtain the weights independent of previous bidding strategies and consequently integrate them into the final CTR prediction and bid generation steps. In this paper, we formulate CTR estimation and bid optimisation under such censored auction data. Derived from a survival model, we show that historic bid information is naturally incorporated to produce Bid-aware Gradient Descents (BGD) which controls both the importance and the direction of the gradient to achieve unbiased learning. The empirical study based on two large-scale real-world datasets demonstrates remarkable performance gains from our solution. The learning framework has been deployed on Yahoo!'s real-time bidding platform and provided 2.97% AUC lift for CTR estimation and 9.30% eCPC drop for bid optimisation in an online A/B test.

Keywords

Unbiased Learning, Censored Data, Real-Time Bidding, Display Advertising

1. INTRODUCTION

The rise of real-time bidding (RTB) based display advertising and behavioural targeting provides one of the most significant cases for machine learning applied to big data. The major supervised learning tasks range from predicting the market price distribution and volume of a given ad impression type [4], estimating the click-through rate (CTR) [23] and conversion rate [17], to the optimisation of a bid [27, 38]. These data driven prediction and optimisation techniques enable ads to be more relevant and targeted to the underlying audience [38].

A challenging yet largely neglected problem in the aforementioned learning tasks is that common supervised learning requires the training and prediction data to follow the same distribution, but in the online display advertising case, the training data is heavily censored by the ad auction selection process [30]. For advertisers, specifically, the above prediction algorithms, e.g., CTR estimation and bid optimisation, are operated over the full volume bid request stream in order to evaluate each potential impression and automatically generate bid [39]. However, the auction selects the ad with the highest bid and displays it to the user. Only in this situation the corresponding user feedback, i.e., click and conversion, to this ad impression, along with the second price (or market price [1]) for this auction, are received by the advertisers as the labels of this data instance. Thus, as illustrated in Figure 1, the obtaining of a training instance is heavily influenced by its bid value; data instances with higher bid price (than the expected market price) would generate a higher probability of winning and thus higher chance to be in the training data. A consequence is that the learning will be overly focused on the instances with a high winning probability (high bid), while neglecting the cases where the probability is small. Such a bias is problematic as intuitively conversions or clicks from those low market-valued impressions are more crucial than those from high market-valued impressions in order to obtain a more economic solution. Ultimately advertisers not only need to identify the impressions that have high chance to be clicks/converted, but also (and equally importantly) require the cost of winning those impressions is relatively small. Thus, we need to have an unbiased learning framework that can take the final optimisation objective into account.

Typically, the bias problem is a missing data problem, which has been well-studied in the machine learning literature [8]. A direct solution would be to identify or assume the missing process and correct the discrepancy (e.g, [25, 22]) during the training. However, the data missing in RTB display advertising depends on both the advertiser's previous bidding strategy and the market competition, neither of them are known as a priori. There are some indirect solutions of alleviating the data bias such as by adding random ad selection probability in the bidding strategy [9], but a better solution would be to decouple the solution with the

*Work done while WZ, TZ were at UCL and JX was at Yahoo! US.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2939672.2939713

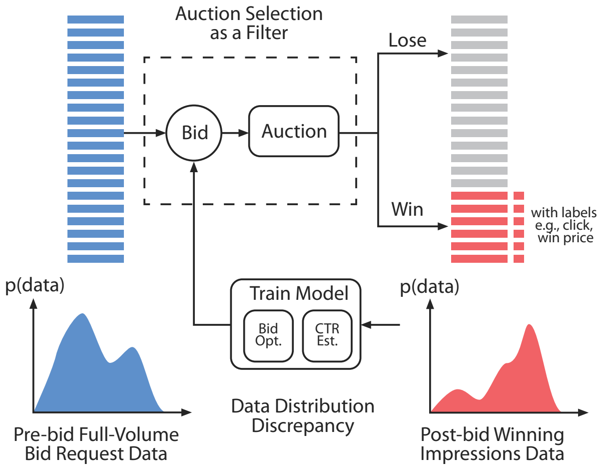

Figure 1: From an advertiser's perspective, the ad auction selection acts as a dynamic data filter based on bid value, which leads to distribution discrepancy between the post-bid training data (red) and the pre-bid prediction data (blue). The y-axis $p(\text{data})$ means the data p.d.f. while the x-axis is a 1-dimension abstraction of the data feature space.

previously employed bidding strategy (when acquiring the training data) and build a link to the final optimisation process.

In this paper, we consider both CTR estimation [23, 17] and bid optimisation [27, 38] together and propose a flexible learning framework that eliminates such auction-generated data bias towards a better learning and optimisation performance. According to the RTB auction mechanism, the labelled training data instance is observed only when the bid is higher than the market price. Inspired by the censored learning work in [1], we explicitly model the auction winning probability with a bid landscape based on a non-parametric survival model, i.e., [14], which is then estimated from the advertiser's historic bid. By importance sampling with the auction winning probability as propensity score [12], we naturally incorporate it into gradient derivation to produce a Bid-aware Gradient Descent (BGD) training scheme for both CTR prediction and bid optimisation tasks. Intuitively, our BGD shows that (i) the higher bid price the impression was won with, the lower valued gradient such data should generate; (ii) to generate a bid, historic bids will further adjust the gradient direction and provide a lower average budget for lower-bidden training instance when learning the bidding function. It is worth noticing that the proposed learning framework is generally applicable to various supervised learning and optimisation tasks mentioned above.

Besides the theoretical derivations, we also conduct empirical studies with the tasks of CTR estimation and bid optimisation on two large-scale real-world datasets. The results demonstrate large improvements brought from our solution over the start-of-the-art models. Moreover, the learning framework was also deployed on Yahoo! DSP in Sep. 2015 and brought 2.97% AUC lift for CTR estimation and 9.30% eCPC drop for bid optimisation over 9 campaigns in an online A/B test.

The rest of this paper is organised as follows. In Section 2 we discuss related work and compare it with ours. We then formulate the problem and propose our solutions for unbiased CTR estimation and bid optimisation under censored auction data in Section 3. Extensive offline empirical study based on two real-world datasets and online A/B test are provided in Section 4. Finally we conclude this paper and discuss future work in Section 5.

2. RELATED WORK

User Response Prediction. Click-through rate (CTR) estimation and conversion rate (CVR) estimation are critical in data driven targeted advertising as these techniques provide a quantification of the user's interest on a specific displayed ad, which in turn help advertisers better allocate budget across audiences [28, 33]. Essentially, CTR/CVR estimation is a probability regression problem where the positive instances are extremely sparse [11]. Various machine learning models with probability-related loss, such as cross entropy and log-likelihood, are used for user response estimation, including linear models such as logistic regression [17], Bayesian probit regression [9], FTRL regression [23], and non-linear ones such as factorisation machines [24] and gradient boosting tree models [11]. Nevertheless, to our best knowledge, none of the existing work considered the bias coming from the ad auction selection for the user response prediction purpose.

RTB Optimisation. Based on user response prediction, advertisers can estimate the value of a specific ad impression, which is the value of a response (click or conversion) multiplied by the predicted response rate (CTR or CVR) [17]. According to auction theory [7], the truth-telling bidding is the optimal strategy in second price auctions. However, when considering repeated auctions with volume and budget constraints, the optimal bidding strategy is not necessarily truth-telling [38, 37]. In RTB display advertising, with user response prediction and bid landscape forecasting [4], the bidding strategy determines how much to bid on a certain ad inventory. The authors in [27] proposed a linear bidding function w.r.t. the predicted CTR and the scaling parameter is tuned based on the market competition. In [3] the authors proposed to set the bid price as the truth-telling bid minus a value, which is dynamically tuned according to the current performance. In [38], the authors proposed a functional optimisation framework to induce the optimal bidding functions that maximises the target key performance indicator (KPI). Recently, a lift-based bidding strategy was proposed [32], where the bid price was set proportional to the user's CVR lift after seeing the ad impression. The authors claimed that such lift-based bidding strategy could substantially bring more customers to the advertisers. Again, none of the investigated work discussed the data bias problem which causes the data distribution discrepancy between the training and prediction stages.

Unbiased Offline Evaluation. As pointed out in [18], direct online evaluation and optimisation for a new solution are expensive and risky, which is also a dilemma in online advertising [2]. However, it is cheap and risk-free if the model can be optimised and evaluated using offline historic data that was previously collected using another (usually unknown) model. The authors in [19] proposed to use historic data for unbiased offline evaluation of news article recommendation models by replay and rejection sampling. Prerequisites of this approach are that the previous model generating the training data (called exploration model) is known, and that the evaluated policy has sufficiently explored all possible actions [15]. For cases where historic data is collected using a biased or non-stationary policy, the authors in [6] suggested an adaptive rejection sampling approach.

The authors in [36] further built a reinforcement learning framework which directly optimised the lower bound of inverse propensity score based policy value to reduce the training data bias from the historic policy. For cases where the exploration model is unknown, an evaluation scheme with estimated propensity scores and a lower bound of the data observation probability was proposed in [29]. In our case, the exploration model is known as we know the historic bid price for each bid request.

Learning with Missing Data. Handling missing data is a well-studied problem in machine learning [8]. A classic application is item recommendation with implicit feedback [25, 22]. The authors in [25] proposed uniform sampling of negative items for each user's positive-feedback item. The authors in [22] further proposed user response models to learn the missing data distribution instead of regarding it as completely random observations. With the idea that the popular but unrated items were more possible to be the true negative items for a user, the authors in [21, 26] proposed to sample the negative items more from the popular items and obtained significant recommendation improvement. More generally, the authors in [35] hypothesised that the unrated items with high predicted interest could actually be the negative samples to the user, and proposed dynamic negative item sampling which substantially improved the recommendation performance on implicit feedback data.

In online advertising, our work is closely related to [1, 31]. Similar to [1], we also employ a survival model [13] to estimate the market price. However, our purpose and setup are significantly different. The work in [31] specifically focused on forecasting and employing censored regression, while we aim at CTR estimation and bid optimisation. The authors in [1] considered bidding as a Markov decision process and formulated an online learning algorithm under the censored data. The underlying data bias was not considered in the bid optimisation and another potential drawback of this work is that a large amount of existing historic bidding data would not be utilised. Instead, we consider two distinctive training and prediction stages and develop models that can make use of any existing historic bidding data independent of previous bidding strategies. The main novelty of our work lies in deriving bid-aware gradient descent that directly incorporates the auction bias into the CTR prediction and bid generation processes to learn unbiased models.

3. METHODOLOGY

In online RTB display advertising, a bid request can be represented as a high dimensional feature vector [17]. Let us denote the vector as $\boldsymbol{x}$. Without loss of generality, we regard the bid requests as generated from an i.i.d. $\boldsymbol{x} \sim p_x(\boldsymbol{x})$ within a short period [38]. Based on the bid request $\boldsymbol{x}$, the ad agent (or demand-side platform, a.k.a. DSP) will then provide a bid $b_{\boldsymbol{x}}$ following a bidding strategy. If such bid wins the auction, the corresponding labels, i.e., user response y (either click or conversion) and market price z, are observed. Thus, the probability of a data instance $(\boldsymbol{x}, y, z)$ being observed relies on whether the bid b_x would win or not and we denote it as $P(\text{win}|\boldsymbol{x}, b_{\boldsymbol{x}})$. Formally, with the p.d.f. $q_x(\boldsymbol{x})$ denoting how the feature vector $\boldsymbol{x}$ is distributed within the observed training data $D = \{(\boldsymbol{x}, y, z)\}$, the generative process of creating the training data is summarised as:

$$\underbrace{q_x(\boldsymbol{x})}_{\text{impression}} = \underbrace{P(\text{win}|\boldsymbol{x}, b_{\boldsymbol{x}})}_{\text{auction selection}} \cdot \underbrace{p_x(\boldsymbol{x})}_{\text{bid request}}, \qquad (1)$$

where the normaliser of $q_x(\boldsymbol{x})$ has been omitted for formula simplicity. Eq. (1) indicates the relationship (bias) between the p.d.f. of the pre-bid full-volume bid request data (prediction) and the post-bid winning impression data (training); in other words, the predictive models would be trained on D, where $\boldsymbol{x} \sim q_x(\boldsymbol{x})$, and be finally operated on prediction data $\boldsymbol{x} \sim p_x(\boldsymbol{x})$. In the following sections, we shall focus on the estimation of the winning probability $P(\text{win}|\boldsymbol{x}, b_{\boldsymbol{x}})$ and then introduce our solutions of using it for creating bid-aware gradients to solve CTR estimation and bid optimisation problems.

3.1 Auction Winning by Survival Models

The RTB display advertising uses the second price auction [34]. In the auction, the market price z is defined as the second highest bid from the competitors for an auction. In other words, it is the lowest bid value one should have in order to win the auction. Following [1], we take a stochastic approach rather than game theoretical, and assume the market price z is a random variable generated from a fixed yet unknown p.d.f. $p_z^{\boldsymbol{x}}(z)$; then the auction winning probability is the probability when the market price z is lower than the bid $b_{\boldsymbol{x}}$:

$$w(b_{\boldsymbol{x}}) \equiv P(\text{win}|\boldsymbol{x}, b_{\boldsymbol{x}}) = \int_0^{b_{\boldsymbol{x}}} p_z^{\boldsymbol{x}}(z)dz, \qquad (2)$$

where to simplify the solution and reduce the sparsity of the estimation, the market price distribution is estimated on a campaign level rather than per impression $\boldsymbol{x}$ [4, 38]. Thus for each campaign, there is a $p_z(z)$ to estimate, resulting the simplified winning function $w(b_{\boldsymbol{x}})$, similar to [1, 38].

If we assume there is no data censorship, i.e., the ad agent wins all the bid requests and observes all the market prices, the winning probability $w_o(b_{\boldsymbol{x}})$ can directly come from the observation counting:

$$w_o(b_{\boldsymbol{x}}) = \frac{\sum_{(\boldsymbol{x}', y, z) \in D} \delta(z < b_{\boldsymbol{x}})}{|D|}, \qquad (3)$$

where z is the historic market price of the bid request $\boldsymbol{x}'$, the indicator function $\delta(z < b_{\boldsymbol{x}}) = 1$ if $z < b_{\boldsymbol{x}}$ and 0 otherwise. We use it as a baseline of $w(b_{\boldsymbol{x}})$ modelling.

However, the above treatment is rather problematic as it does not take into account that in practice there are always a large portion of the auctions the advertiser loses $(z \geq b_{\boldsymbol{x}})$[1], in which the market price is not observed in the training data. Thus, the observations of the market price are *right-censored*: when we lose, we only know that the market price is higher than our bid, but do not know its exact value. In fact, $w_o(b_{\boldsymbol{x}})$ is a biased model and over-estimates the winning probability. One way to look at this is that it ignores the counts for lost auctions where the historic bid price is higher than $b_{\boldsymbol{x}}$ in the denominator of Eq. (3). In this situation, the market price should have been higher than the historic bid price and thus higher than $b_{\boldsymbol{x}}$. As we will show in our experiment such estimator consistently over-estimate the actual winning probability.

In this paper, we use survival models [13] to handle the biased auction data. Survival models were originally proposed to predict patients' survival rate for a given time after certain treatment. As some patients might leave the investigation, researchers do not know their exact final survival

[1] In the iPinYou dataset [39] we tested, the overall auction winning rate of 9 campaigns is 23.8%, which is already a very high rate in practice.

period but only know the period is longer than the investigation period. Thus the data is right-censored. The auction scenario is quite similar: the integer market price[2] is regarded as the patient's underlying survival period from low to high and the bid price as the investigation period from low to high. If the bid b wins the auction, the market price z is observed, which is analogous to the observation of the patient's death on day z. If the bid b loses the auction, one only knows the market price z is higher than b, which is analogous to the patient's left from investigation on day b.

Specifically, we follow [1] by leveraging the non-parametric Kaplan-Meier Product-Limit method [14] to estimate the market price distribution $p_z(z)$ based on the observed impressions and the lost bid requests.

Suppose there is a campaign that has participated in N RTB display ad auctions. Its bidding log is a list of N tuples $\langle b_i, w_i, z_i \rangle_{i=1...N}$, where b_i is the bid price of this campaign in the auction i, w_i is the boolean value of whether this campaign won the auction i, and z_i is the corresponding market price if $w_i = 1$. The problem is to model the probability of winning an ad auction $w(b_{\boldsymbol{x}})$ with bid price $b_{\boldsymbol{x}}$.

If we transform our data into the form of $\langle b_j, d_j, n_j \rangle_{j=1...M}$, where the bid price $b_j < b_{j+1}$. d_j denotes the number of ad auction winning cases with the market price exactly valued $b_j - 1$ (in analogy to patients die on day b_j). n_j is the number of ad auction cases which cannot be won with bid price $b_j - 1$ (in analogy to patients survive to day b_j), i.e., the number of winning cases with the observed market price no lower than $b_j - 1$[3] plus the number of lost cases when the bid is no lower than $b_j - 1$. Then with bid price $b_{\boldsymbol{x}}$, the probability of losing an ad auction is

$$l(b_{\boldsymbol{x}}) = \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}, \tag{4}$$

which just corresponds to the probability a patient survives from day 1 to day $b_{\boldsymbol{x}}$. Thus the winning probability will be

$$w(b_{\boldsymbol{x}}) = 1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}. \tag{5}$$

Note the calculation is Eq. (5) is highly efficient, i.e., $O(N)$. Table 1 gives an example of transforming the historic $\langle b_i, w_i, z_i \rangle$ data into the survival model data $\langle b_j, d_j, n_j \rangle$ and the corresponding winning probabilities calculated by Eqs. (5) and (3). We see that the Kaplan-Meier Product-Limit model, which is a non-parametric maximum likelihood estimator of the data [5], makes use of all winning and lost data to estimate the winning probability of each bid, whereas the observation-only counting model $w_o(b_{\boldsymbol{x}})$ does not. As we can see in the table $w_o(b_{\boldsymbol{x}})$ is consistently higher than $w(b_{\boldsymbol{x}})$. Later in experiment, we will further demonstrate such comparisons with real-world data in Figure 5.

3.2 Task 1: CTR Estimation

Generally, given a training dataset $D = \{(\boldsymbol{x}, y, z)\}$, where the data instance $\boldsymbol{x}$ follows the training data distribution $q_x(\boldsymbol{x})$, (the red data distribution in Figure 1), an *unbiased* supervised learning problem can be formalised into a loss-minimisation problem on prediction data distribution $p_x(\boldsymbol{x})$ (the blue data distribution in Figure 1):

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{\boldsymbol{x} \sim p_x(\boldsymbol{x})}[\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))] + \lambda \Phi(\boldsymbol{\theta}), \tag{6}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is $\boldsymbol{\theta}$-parametrised prediction model to be learned; $\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ is the loss function based on the ground truth y and the prediction $f_{\boldsymbol{\theta}}(\boldsymbol{x})$; $\Phi(\boldsymbol{\theta})$ is the regularisation term that penalises the model complexity; λ is the regularisation weight. With Eqs. (1) and (2), one can use importance sampling to reduce the bias of the training data:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x} \sim p_x(\boldsymbol{x})}[\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))] = \int_{\boldsymbol{x}} p_x(\boldsymbol{x}) \mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x})) d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}} q_x(\boldsymbol{x}) \frac{\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x} \sim q_x(\boldsymbol{x})} \Big[\frac{\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{w(b_{\boldsymbol{x}})} \Big] \\ &= \frac{1}{|D|} \sum_{(\boldsymbol{x}, y, z) \in D} \frac{\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{w(b_{\boldsymbol{x}})} = \frac{1}{|D|} \sum_{(\boldsymbol{x}, y, z) \in D} \frac{\mathcal{L}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}}, \end{aligned} \tag{7}$$

where the last equation is our empirical estimation. Based on this framework, if we obtain the auction winning probability $w(b_{\boldsymbol{x}})$, e.g., Eq. (5), we can eliminate the bias for each observed training data instance. Let us look at the case of CTR estimation with logistic regression [28]. With the logistic loss between the binary click label $\{-1, +1\}$ and the predicted probability and L2 regularisation, the framework of Eq. (7) is written as

$$\min_{\boldsymbol{\theta}} \frac{1}{|D|} \sum_{(\boldsymbol{x}, y, z) \in D} \frac{\log(1 + e^{-y \boldsymbol{\theta}^T \boldsymbol{x}})}{w(b_{\boldsymbol{x}})} + \frac{\lambda}{2} ||\boldsymbol{\theta}||_2^2, \tag{8}$$

where the winning probability $w(b_{\boldsymbol{x}})$ is estimated for each observation instance, which is independent from the CTR estimation parameter $\boldsymbol{\theta}$; the update rule of $\boldsymbol{\theta}$ is routine using stochastic gradient descent with the learning rate η. The derived *Bid-aware Gradient Descent* (BGD) of Eq. (8) is

$$\boldsymbol{\theta} \leftarrow (1 - \eta \cdot \lambda) \boldsymbol{\theta} + \frac{\eta \cdot y \cdot e^{-y \boldsymbol{\theta}^T \boldsymbol{x}} \cdot \boldsymbol{x}}{(1 + e^{-y \boldsymbol{\theta}^T \boldsymbol{x}})(1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j})}. \tag{9}$$

Discussion. From the equation above, we observe that with a lower winning bid $b_{\boldsymbol{x}}$, the probability $1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}$ of seeing the instance in the training set is lower. However, the corresponding gradient from the data instance is higher and vice versa as it is in the denominator.

This is intuitively correct as when a data instance $\boldsymbol{x}$ is observed with low probability, e.g., 10%, we can infer there are 9 more such kind of data instances missed because of auction losing. Thus the training weight of $\boldsymbol{x}$ should be

Table 1: An example of data transformation of 8 instances with bid price between 1 and 4. Left: tuples of bid, win and cost $\langle b_i, w_i, z_i \rangle_{i=1...8}$. Right: transformed survival model tuples $\langle b_j, d_j, n_j \rangle_{j=1...4}$ and the calculated winning probabilities. Here we also provide a calculation example of $n_3 = 4$ shown as blue in the right table. The counted cases of n_3 in the left table are 2 winning cases with $z \geq 3 - 1$ and the 2 lost cases with $b \geq 3$, shown highlighted in blue color.

b_i	w_i	z_i
2	win	1
3	win	2
2	lose	×
3	win	1
3	lose	×
4	lose	×
4	win	3
1	lose	×

b_j	n_j	d_j	$\frac{n_j - d_j}{n_j}$	$w(b_j)$	$w_o(b_j)$
1	8	0	1	$1 - 1 = 0$	0
2	7	2	$\frac{5}{7}$	$1 - \frac{5}{7} = \frac{2}{7}$	$\frac{2}{4}$
3	4	1	$\frac{3}{4}$	$1 - \frac{5}{7}\frac{3}{4} = \frac{13}{28}$	$\frac{3}{4}$
4	2	1	$\frac{1}{2}$	$1 - \frac{5}{7}\frac{3}{4}\frac{1}{2} = \frac{41}{56}$	$\frac{4}{4}$

[2]The mainstream ad exchange auctions require integer bid prices. Without a fractional component, it is reasonable to analogise bid price to survival days.

[3]We assume that if there is tie in the auction, the campaign will not get winning.

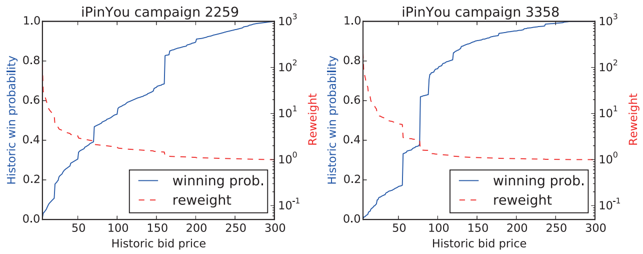

Figure 2: Winning probability and reweighting term in Eq. (9) against historic bid price.

multiplied by 10 in order to recover statistics from the full-volume data. By contrast, if the winning bid is extremely high, which leads 100% auction winning probability, then such data is observed from the true data distribution. Thus there will be no gradient reweighting on this data. Such nonlinear relationship has been well captured in our model in the gradient updates, as illustrated in Figure 2.

3.3 Task 2: Bid Optimisation

Another important problem in online advertising is bid optimisation, i.e. to find the optimal bidding strategy to maximise a campaign KPI, restricted by the campaign budget. Essentially, the bidding function is abstracted as a function mapping from the estimated CTR $f(\boldsymbol{x})$ to the bid price $b(f(\boldsymbol{x}))$.[4] According to [38], with the auction volume T and campaign budget B, it is a functional optimisation problem:

$$\underset{b()}{\arg\max} \quad T \int_{\boldsymbol{x}} f(\boldsymbol{x}) w(b(f(\boldsymbol{x}))) p_x(\boldsymbol{x}) d\boldsymbol{x} \tag{10}$$
$$\text{subject to} \quad T \int_{\boldsymbol{x}} b(f(\boldsymbol{x})) w(b(f(\boldsymbol{x}))) p_x(\boldsymbol{x}) d\boldsymbol{x} = B.$$

With the auction selection, the observed data distribution is actually $q_x(\boldsymbol{x})$. By Eq. (1), Eq. (10) is written as

$$\underset{b()}{\arg\max} \quad T \int_{\boldsymbol{x}} f(\boldsymbol{x}) w(b(f(\boldsymbol{x}))) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} \tag{11}$$
$$\text{subject to} \quad T \int_{\boldsymbol{x}} b(f(\boldsymbol{x})) w(b(f(\boldsymbol{x}))) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} = B.$$

Note that $w(b_{\boldsymbol{x}})$ is different from $w(b(f(\boldsymbol{x})))$, where $b_{\boldsymbol{x}}$ is the historic bid price for the bid request $\boldsymbol{x}$ while $b(f(\boldsymbol{x}))$ is the bid price we want to optimise.

The Lagrangian is

$$\mathcal{L}(b(f), \lambda) = \int_{\boldsymbol{x}} f(\boldsymbol{x}) w(b(f(\boldsymbol{x}))) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} \tag{12}$$
$$- \lambda \int_{\boldsymbol{x}} b(f(\boldsymbol{x})) w(b(f(\boldsymbol{x}))) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} + \frac{\lambda B}{T},$$

According to the derivation of [38], the Euler-Lagrangian condition of Eq. (11) is

$$f(\boldsymbol{x}) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} \frac{\partial w(b(f(\boldsymbol{x})))}{\partial b(f(\boldsymbol{x}))} - \lambda \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} \Big[w(b(f(\boldsymbol{x}))) + b(f(\boldsymbol{x})) \frac{\partial w(b(f(\boldsymbol{x})))}{\partial b(f(\boldsymbol{x}))} \Big] = 0, \tag{13}$$

$$\Rightarrow \lambda w(b(f(\boldsymbol{x}))) = \Big[f(\boldsymbol{x}) - \lambda b(f(\boldsymbol{x})) \Big] \frac{\partial w(b(f(\boldsymbol{x})))}{\partial b(f(\boldsymbol{x}))}, \tag{14}$$

[4]We drop the CTR estimation parameter $\boldsymbol{\theta}$ here as it is not the parameter to optimise in this task.

where we see that the optimal bidding function $b(f(\boldsymbol{x}))$ depends on the winning function $w(b)$. For example, if

$$w(b(f(\boldsymbol{x}))) = \frac{b(f(\boldsymbol{x}))}{c + b(f(\boldsymbol{x}))}, \tag{15}$$

where c is a constant, then the corresponding optimal bidding function is

$$b_{\text{ORTB}}(f(\boldsymbol{x})) = \sqrt{\frac{c}{\lambda} f(\boldsymbol{x}) + c^2} - c. \tag{16}$$

For the solution of λ, the Euler-Lagrangian condition w.r.t. λ is

$$\partial \mathcal{L}(b(f(\boldsymbol{x})), \lambda) / \partial \lambda = 0 \tag{17}$$
$$\Rightarrow \int_{\boldsymbol{x}} b(f(\boldsymbol{x}), \lambda) w(b(f(\boldsymbol{x}), \lambda)) \frac{q_x(\boldsymbol{x})}{w(b_{\boldsymbol{x}})} d\boldsymbol{x} = \frac{B}{T} \tag{18}$$
$$\Rightarrow \frac{1}{|D|} \sum_{(\boldsymbol{x}, y, z) \in D} b(f(\boldsymbol{x}), \lambda) \frac{w(b(f(\boldsymbol{x}), \lambda))}{w(b_{\boldsymbol{x}})} = \frac{B}{|D|}. \tag{19}$$

The numeric solution of λ is highly efficient. A feasible solution of Eq. (19) is

$$\min_{\lambda} \sum_{(\boldsymbol{x}, y, z) \in D} \frac{1}{2} \Big(\frac{b(f(\boldsymbol{x}), \lambda) w(b(f(\boldsymbol{x}), \lambda))}{1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}} - \frac{B}{|D|} \Big)^2. \tag{20}$$

As $b(f(\boldsymbol{x}), \lambda)$ always monotonically decreases w.r.t. λ and $w(b_{\boldsymbol{x}})$ monotonically increases w.r.t. $b(f(\boldsymbol{x}), \lambda)$, the objective of Eq. (20) is convex w.r.t. λ, which makes the solution of λ easy to obtain. The BGD to solve λ is via updating

$$\lambda \leftarrow \lambda - \eta \overbrace{\frac{1}{1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}}}^{\text{instance reweighting}} \overbrace{\Big(\frac{b(f(\boldsymbol{x}), \lambda) w(b(f(\boldsymbol{x}), \lambda))}{1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j}} - \frac{B}{|D|} \Big)}^{\text{gradient direction}} \cdot \underbrace{\Big(\frac{\partial b(f(\boldsymbol{x}), \lambda)}{\partial \lambda} w(b(f(\boldsymbol{x}), \lambda)) + b(f(\boldsymbol{x}), \lambda) \frac{\partial w(b(f(\boldsymbol{x}), \lambda))}{\partial \lambda} \Big)}_{\text{bidding function gradient}}. \tag{21}$$

Discussion. Highlighted in Eq. (21), there are two factors related with the historic bid for updating λ: (i) the instance reweighting, similar with Eq. (9): a small historic bid $b_{\boldsymbol{x}}$ would generate a large weight, amplifying the importance of the training instance. (ii) The historic bid of the training instance also has an impact on the gradient direction, evidenced by the second factor of the update in Eq. (21).

The parameter λ converges when the second factor becomes zero. The ratio $B/|D|$ would ensure the budget to be allocated evenly across the new bids. The ratio between the winning rate of the new bid price $w(b(f(\boldsymbol{x}), \lambda))$ and that of the historic bid $1 - \prod_{b_j < b_{\boldsymbol{x}}} (1 - d_j / n_j)$ would adjust the discrepancy of the probability of seeing the impression in the training and that in the prediction.

To further understand this, Figure 3 illustrates the second factor (the gradient direction term) in Eq. (21) against historic bid price $b_{\boldsymbol{x}}$ on two sample campaigns with two new bids ($b(f(\boldsymbol{x}), \lambda) = 50$ and 100). We observe that when the historic bid is small, the gradient direction is more likely to stay positive and leads to higher λ value (as bidding function gradient term in Eq. (21) is always negative) in order to decrease the bid. For example, for a data instance that its historic bid $b_{\boldsymbol{x}}$ is low, the probability of observing the data instance is low, which means there are more similar or the same data instances that are missing in the training. If the

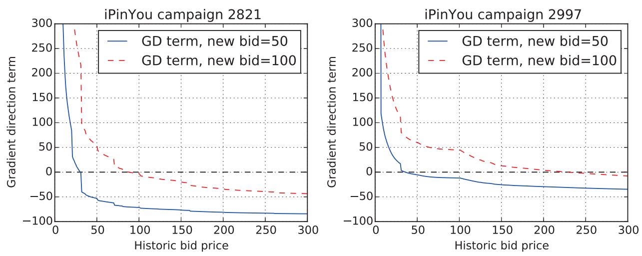

Figure 3: The gradient direction term in Eq. (21) against historic bid price $b_{\boldsymbol{x}}$ with two new bids $b(f(\boldsymbol{x}), \lambda)$.

new bid price $b(f(\boldsymbol{x}), \lambda)$ is high, then the optimal bid price $b(f(\boldsymbol{x}), \lambda)$ should be lower to avoid budget overspending in full-volume data, which is reflected on the positive value of the gradient direction factor to make λ higher and $b(f(\boldsymbol{x}), \lambda)$ lower.

Please note that with the pre-calculated reweighting factor $1/w(b_{\boldsymbol{x}}) = 1/(1 - \prod_{b_j < b_{\boldsymbol{x}}} \frac{n_j - d_j}{n_j})$, it is highly efficient to calculate the above BGD updating and solve λ.

4. EXPERIMENT

4.1 Datasets

Two real-world datasets are used in our repeatable offline empirical study[5]: iPinYou and TukMob.

iPinYou runs the largest DSP in China. The publicly available[6] iPinYou dataset consists of 64.75M bid records, 19.50M impressions, 14.79K clicks and 16K CNY expense on 9 conventional display ad campaigns from different advertisers during 10 days in 2013. According to iPinYou [20], the last 3-day data for each campaign is set as test data while the rest is training data.

TukMob is a major DSP focusing on mobile game and video display ads in China. TukMob dataset is our proprietary dataset which consists of 3.00M impressions, 96.45K clicks and 2.51K CNY expense on 63 campaigns in a video display ad market from Feb. to Aug. 2015. The first 5/6 data in the time sequence is set as training data while the rest is test data.

Each data instance of both datasets can be represented as a triple $(\boldsymbol{x}, y, z)$, where y is the user click binary feedback, z is the historic winning price of the auction, and $\boldsymbol{x}$ is the bid request and ad features of that auction. The auction features contain the information of the user (e.g. the user interest segments, IP address, browser, operation system, location), advertiser (e.g. the creative format and size), publisher (e.g. the auction reserve price, ad slot size, page domain and URL).

We mainly report the experimental results on iPinYou dataset for experiment reproducibility while the study on TukMob acts as an auxiliary part particularly for the high-CTR video ad marketplace to make our experiment more comprehensive.

The online A/B testing experiment is conducted based on Yahoo! DSP, a mainstream DSP in United States ad market. The training dataset comes from its ad log in Aug. and Sep. 2015 while the online A/B testing is performed on 9 campaigns during 7 days of Sep. 2015, which involves 117.1M impressions, 95.4K clicks and 68.6K USD expense.

[5]Experiment code link: https://github.com/wnzhang/rtb-unbiased-learning.

[6]Dataset link: http://data.computational-advertising.org

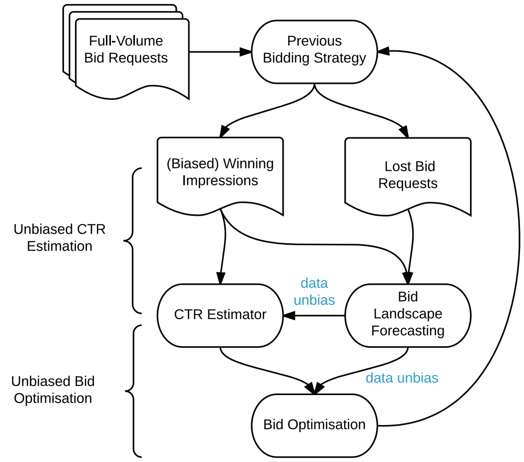

Figure 4: Experiment flow chart.

4.2 Experiment Flow

The experiment flow chart is shown in Figure 4. The original impression log data is reasonably assumed as full-volume bid request data in our experiment[7]. A truth-telling bidding strategy [17] is performed to simulate the historic bidding process and produce the winning (labelled but biased) impression data and lost (unlabelled) bid request data. Based on these two datasets, the bid landscape forecasting module as in Eq. (5) estimates the market price distribution which acts as the winning function in Eq. (1). Thus the observation bias of each data instance from the impression log is estimated. With Eq. (8), the unbiased CTR estimation is performed. Furthermore, with the unbiased CTR estimator and the winning function, the unbiased bid optimisation is performed via Eq. (11) to get the new bidding function, which is in turn operated in the next prediction stage.

4.3 Compared Settings

CTR estimation and bid optimisation are the two tasks we investigate in this work. For each of these tasks, we compare the following four training schemes:

- BIAS - The CTR estimation and bid optimisation are performed based on the impression data without considering any data bias, i.e., all $w(b_{\boldsymbol{x}})$ in Eqs. (8) and (11) are equal to 1. This is the routine training procedure used in most previous work [17, 27, 38].

- UOMP - The bias of each training data instance is estimated by the bid landscape forecastor purely based on the observed market prices from impression log, without using the lost bid request data, i.e., all $w(b_{\boldsymbol{x}})$ in Eqs. (8) and (11) are estimated by Eq. (3).

- KMMP - The bias of each training data instance is estimated by the bid landscape forecastor based on both observed market prices from impression log and the lost bid request data using Kaplan-Meier estimation, i.e., all $w(b_{\boldsymbol{x}})$ in Eqs. (8) and (11) are estimated by Eq. (5).

[7]This assumption is reasonable as this dataset is collected with fixed large bid to reduce the auction-selection bias [20].

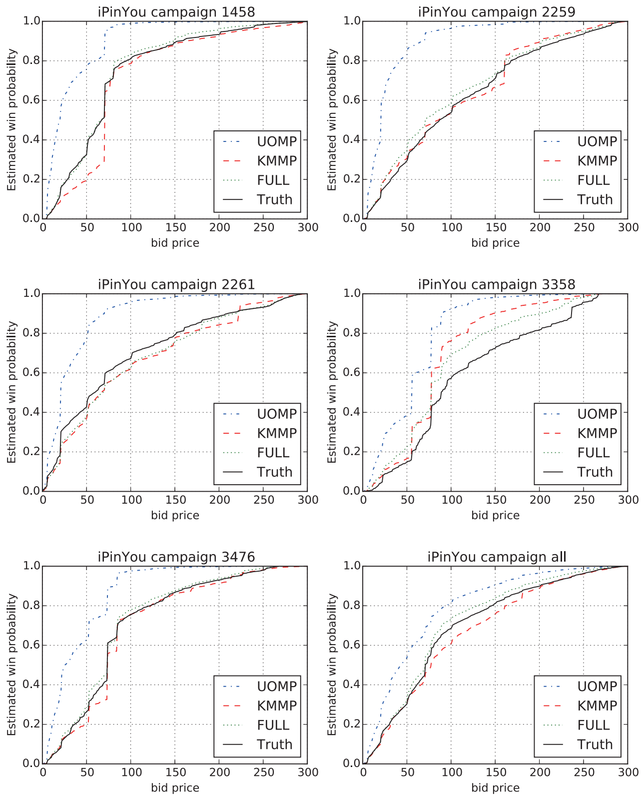

Figure 5: Winning probability against bid price (iPinYou).

- FULL - A progressive bidding strategy is performed to win all the bid requests via bidding extremely high. In such case the full-volume bid requests are collected with labels to train the CTR estimator and bid optimisation. In such setting, the data has no bias and is of full volume, and thus it is regarded as the (unrealistic) upper bound setting of the training.

4.4 Winning Probability Estimation

Before evaluating the practical CTR estimation and bid optimisation tasks, let us first take an analysis of the compared models' performance on winning probability estimation, i.e., $w(b_{\boldsymbol{x}})$ in Eq. (2).

First, Table 2 demonstrates the statistics of the full-volume data and the winning impression data by the 'historic' truth-telling bidding strategy as described in Section 4.2. As can be observed, for both datasets the winning impression data which is fed into BIAS, UOMP and KMMP training schemes is much smaller than the full-volume data which is fed into FULL training scheme.

Figure 5 shows the curves of winning probability w.r.t. the bid price with three compared settings, i.e., UOMP, KMMP and FULL, on iPinYou dataset. As expected, all the curves start from 0 given the bid 0 and then increase as the bid price increases and finally converge to 1 when the bid price surpasses a threshold (300 for iPinYou dataset). The TRUTH curve is built from all the market price observations from the full-volume prediction data, regarded as the ground truth here. We observe that FULL curve is the closest one to TRUTH curve since FULL makes use of the full-volume training data and is naturally unbiased. The only reason of the slight difference between FULL and TRUTH is the data

Table 2: Winning data statistics: the full-volume data is used in FULL training scheme, while the winning data is used in BIAS, UOMP and KMMP training schemes (both datasets).

iPinYou Camp.	Full Vol.	Win Vol.	Win rate
1458	2,055,371	257,077	12.51%
2259	557,038	239,328	42.96%
2261	458,412	213,930	46.67%
2821	881,708	305,134	34.61%
2997	208,292	60,556	29.07%
3358	1,161,403	336,769	29.00%
3386	1,898,535	332,223	17.50%
3427	1,729,177	563,592	32.59%
3476	1,313,574	303,341	23.09%
all	10,263,506	3,973,989	38.72%
TukMob Camp.	Full Vol.	Win Vol.	Win rate
all	2,500,000	962,690	38.51%

Table 3: Winning probability estimation (iPinYou).

	Pearson Correlation			KL-Divergence		
Camp.	UOMP	KMMP	FULL	UOMP	KMMP	FULL
1458	0.9067	0.9903	0.9995	0.4053	0.1204	0.0407
2259	0.7811	0.9959	0.9980	0.7163	0.1870	0.0713
2261	0.9018	0.9947	0.9972	0.3483	0.1057	0.0346
2821	0.8234	0.9947	0.9931	0.5659	0.1421	0.0697
2997	0.8535	0.9285	0.9955	0.3862	0.1761	0.0210
3358	0.9269	0.9772	0.9926	0.5243	0.2652	0.1521
3386	0.9116	0.9821	0.9995	0.3232	0.1391	0.0444
3427	0.9743	0.9977	0.9996	0.1838	0.0762	0.0525
3476	0.9303	0.9979	0.9993	0.3807	0.1147	0.0451
all	0.9795	0.9958	0.9988	0.0893	0.0385	0.0237

distribution shift between the training and prediction period. UOMP always over-estimates the winning probability, as pointed out in Section 3.1. Compared to UOMP, KMMP curve is much closer to TRUTH, which shows its advantage of making use of the lost bid request data to improve the winning probability estimation.

Table 3 presents the detailed Pearson correlation and KL-divergence between each of the three compared settings and TRUTH on iPinYou dataset. We observe that for all investigated campaigns, KMMP provides a much better estimation, i.e., higher Pearson correlation and lower KL-divergence, than UOMP, and it is even highly comparable with FULL on Pearson correlation. These results demonstrate the surprisingly large improvement that the lost and free bid request data brings to the estimation of winning probability (market price distribution).

4.5 CTR Estimation Results

With different biased or unbiased settings, we train the logistic regression model and evaluate its performance. Table 4 presents the detailed AUC and cross entropy performance of these 4 compared training schemes for each campaign in iPinYou dataset. Table 5 presents the AUC performance comparison on TukMob dataset. We can observe that (i) the proposed unbiased training schemes UOMP and KMMP always outperform the biased but widely adopted BIAS training scheme on all the test campaigns (except for 3476). Such consistent outperformance shows the effectiveness of our models in eliminating the training data instance bias which makes the prediction model generalise better on prediction data. (ii) Comparing the unbiased settings UOMP and KMMP and the upper bound oracle setting FULL, we can see KMMP outperforms UOMP for all the campaigns (except for 3476). For some campaigns, e.g., 1458 and 2997, KMMP

Table 4: CTR performance on iPinYou dataset.

Camp.	AUC (%) BIAS	UOMP	KMMP	FULL	Cross Entropy (‰) BIAS	UOMP	KMMP	FULL
1458	98.26	98.56	99.13	98.57	2.42	2.39	2.39	2.32
2259	60.27	60.94	62.00	67.37	4.04	4.03	4.02	4.00
2261	57.49	58.86	59.05	60.91	3.75	3.74	3.74	3.72
2821	59.25	59.69	60.28	62.36	7.07	7.06	7.04	6.92
2997	59.35	60.50	60.79	59.28	32.89	32.84	32.81	32.38
3358	96.59	96.78	97.01	97.32	4.48	4.47	4.38	4.36
3386	73.74	74.01	74.16	78.23	8.84	8.83	8.83	8.64
3427	96.04	96.42	96.78	97.02	3.37	3.37	3.33	3.31
3476	93.66	93.55	92.19	95.93	4.35	4.34	4.34	4.08
all	71.76	73.84	74.80	78.38	7.71	7.61	7.55	7.31

Table 5: CTR performance on TukMob dataset.

Camp.	AUC (%) BIAS	UOMP	KMMP	FULL
all	60.49	60.51	60.67	60.96

even slightly outperforms FULL[8] which again shows the advantages of making use of the lost auction information for better estimating the instance bias.

Figure 6 shows the AUC and cross entropy on prediction data of all iPinYou campaigns for each training round. We can observe the unbiased UOMP and KMMP models learn stably and consistently outperform BIAS. FULL substantially outperforms other compared training schemes, which is not surprising as FULL obtains much more training data instances (as shown in Table 2) and the data distribution is unbiased.

Note that we do not compare calibration techniques [11] in our experiment because it is another dimension of reducing the model bias. If the training data is auction-biased, then the calibration based on that is still biased.

4.6 Bid Optimisation Results

For bid optimisation experiment, we mainly focus on the click performance improvement from bidding strategy parameter optimisation via Eqs. (16) and (19) instead of the difference of CTR estimation. Thus in our training/prediction environment, the logistic regression CTR estimator is trained based on a separate unbiased training data and is shared in all 4 compared training schemes of bid optimisation. For each training scheme, we train the optimal parameter λ in Eq. (19) via the biased or unbiased training data, then apply the corresponding bidding strategy Eq. (16) on prediction data to observe its performance.

We follow [38] to set the budget proportions to perform offline bid optimisation, where the train/test budget is set as 1/64, 1/32, 1/16, 1/8, 1/4 and 1/2 of the total expense of the train/test dataset. We cannot set the proportion as 1 because in such case one may simply bid infinity to win all the impressions and clicks in the data and just spend all the budget.

Table 6 shows the click performance of the 4 compared training schemes with 1/64 and 1/4 budget settings respectively for each iPinYou campaign. Table 7 shows the overall click and eCPC performance comparison against different budget settings on TukMob dataset. We can observe that the unbiased UOMP and KMMP consistently outperform the traditional BIAS which were used in the most of the previous bid optimisation work [16, 27, 38]. This shows the great potential of our proposed unbiased training schemes in bid optimisation. Furthermore, KMMP outperforms UOMP and it

[8]This is mainly caused by the local data distribution, which is not significant.

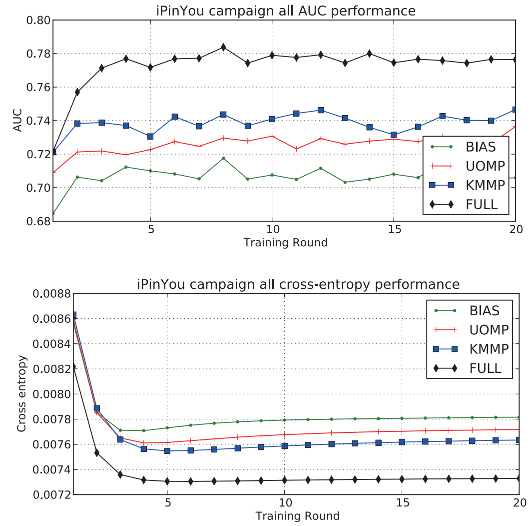

Figure 6: CTR performance training convergence (iPinYou).

Table 6: Bid optimisation click performance (iPinYou).

Camp.	1/64 budget setting BIAS	UOMP	KMMP	FULL	1/4 budget setting BIAS	UOMP	KMMP	FULL
1458	363	400	460	468	469	470	471	482
2259	5	5	5	7	49	51	51	50
2261	5	5	7	7	35	36	35	45
2821	18	18	23	27	65	91	106	134
2997	37	39	42	44	156	188	222	226
3358	86	117	137	140	183	198	220	221
3386	9	9	22	38	69	78	144	165
3427	103	119	154	169	242	262	286	314
3476	6	8	11	13	59	99	108	106
all	268	372	462	521	1,584	1,740	1,871	2,087

Table 7: Bid optimisation click performance (TukMob).

Budget Setting	Click Number BIAS	UOMP	KMMP	FULL	eCPC BIAS	UOMP	KMMP	FULL
1/32	846	848	866	871	9.25	9.23	9.04	8.99
1/16	1,829	1,831	1,863	1,838	8.56	8.55	8.40	8.52
1/8	3,721	3,721	3,774	3,775	8.42	8.42	8.30	8.29
1/4	7,181	7,178	7,226	7,257	8.72	8.72	8.67	8.63
1/2	13,127	13,132	13,163	13,019	9.54	9.54	9.52	9.62

is very close to the theoretic upper bound from FULL, in 17 out of 20 test cases, suggesting it is generally much better to leverage the winning probability obtained from the censored observations of both winning impressions and lost bid requests.

Figure 7 further provides the click, impression improvement percentages and eCPC drop percentage of the unbiased training schemes against BIAS with different budget settings. The improvements for clicks and impressions are positive for all budget settings and the eCPC drops are negative for all budget settings (except FULL on 1/2), which show the robustness of the unbiased training schemes. Also we can observe that KMMP dominates UOMP and heavily approach the upper bound FULL.

4.7 Online A/B Testing

We deployed the unbiased KMMP training scheme on Yahoo! DSP and performed online A/B testing for 9 campaigns during 7 days in Sep. 2015. For each campaign, we created two experiment buckets: control and treatment. Each was allocated 50% of the bid request traffic (based on user ID to avoid attribution conflicts), and 50% of the campaign's budget. The control bucket used gradient boosting decision tree

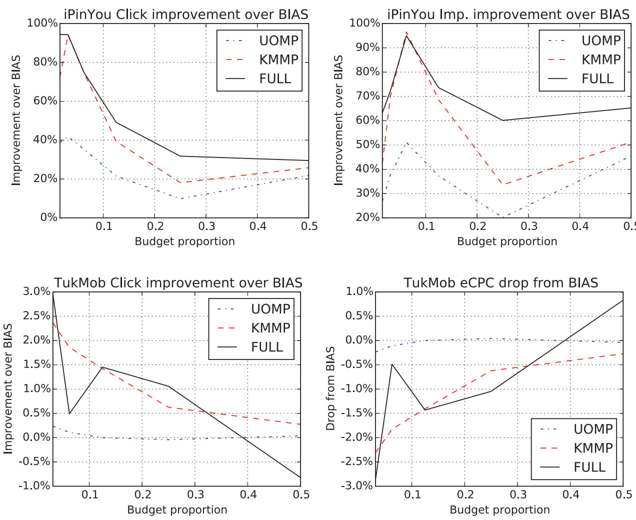

Figure 7: Improvement over BIAS w.r.t. budget proportions.

Table 8: Online A/B testing of CTR estimation (Yahoo!).

Camp.	BIAS AUC.	KMMP AUC	AUC Lift
C1	63.78%	64.12%	0.34%
C2	87.45%	88.58%	1.13%
C3	69.73%	75.52%	5.79%
C4	88.82%	89.55%	0.73%
C5	69.71%	72.29%	2.58%
C6	89.33%	90.70%	1.37%
C7	77.76%	78.92%	1.16%
C8	74.57%	76.98%	2.41%
C9	71.04%	73.12%	2.08%
all	73.48%	76.45%	2.97%

click predictor [11] trained with BIAS, while the click model used in the treatment bucket was trained with KMMP. The deployed bidding strategy is the conventional truth-telling bidding [17].

In order to perform an unbiased evaluation of the CTR estimation, we deployed a bidding agent performing very high constant bid in Sep. 2015 to collect an ad impressions dataset which can be regarded as full-volume unbiased test data. The training data was still the traditional biased ad impression dataset during Aug. and early Sep. 2015. Table 8 provides the detailed CTR estimation performance for each campaign and the overall performance. As can be observed, KMMP provided a consistent AUC improvement over BIAS across all investigated campaigns. The overall AUC was 73.48% for BIAS and 76.45% for KMMP, which was a very large improvement for CTR estimation task in practice.

Table 9 further presents the detailed performance of A/B testing of bid optimisation on the 9 campaigns. Figure 8 depicts the relative difference comparing the performance of KMMP againt BIAS. We found that with the same campaign budget the KMMP-trained model acquired more clicks (most of the time) but fewer impressions than the BIAS-trained one, which made its CTR much higher than BIAS. This is because there was less over-prediction on many cheap cases. In the biased training data, the over-predicted CTR on cheap cases were more likely to be sampled because the historic bidding strategy overbid on these cheap cases, vice versa on expensive cases. With the KMMP training scheme, the bidding strategy to-some-extent got rid of such bias to avoid over-prediction on cheap cases, which provided fewer impressions but more clicks.

Table 9: Online A/B testing of bid optimisation (Yahoo!).

Camp.	Impressions (M)		Clicks (K)		CTR (%)		eCPC ($/click)	
	BIAS	KMMP	BIAS	KMMP	BIAS	KMMP	BIAS	KMMP
C1	1.07	0.89	0.62	0.67	0.06	0.08	4.54	4.16
C2	7.73	6.02	0.94	1.19	0.01	0.02	7.49	5.89
C3	22.18	16.96	27.18	30.06	0.12	0.18	0.26	0.23
C4	0.37	0.12	0.61	0.61	0.16	0.49	2.46	2.48
C5	9.57	7.51	6.42	6.93	0.07	0.09	0.45	0.42
C6	0.32	0.22	0.46	0.46	0.14	0.21	2.17	2.18
C7	10.13	7.31	2.99	3.28	0.03	0.04	0.37	0.34
C8	1.04	0.52	1.04	1.13	0.10	0.22	1.92	1.78
C9	13.67	11.46	5.12	5.71	0.04	0.05	1.76	1.58
all	66.07	51.01	45.37	50.03	0.07	0.10	0.76	0.69

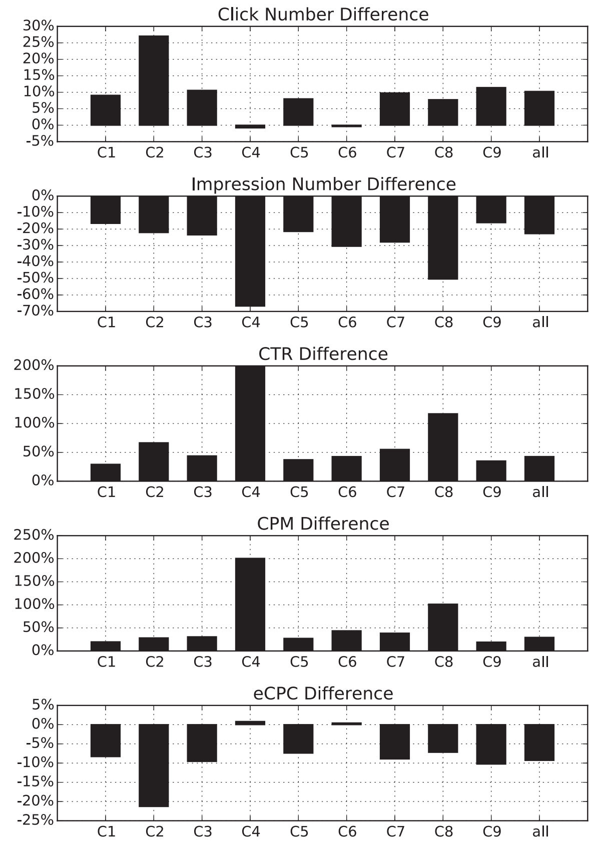

Figure 8: Relative performance difference between KMMP and BIAS in Yahoo! online A/B testing: (KMMP-BIAS)/BIAS.

Overall, with the same budget, the bidding strategy trained with KMMP achieved much better eCPC (9.30% drop) and CTR (42.8% rise) than the conventional one trained with BIAS. The KMMP-trained click model effectively alleviated over-prediction especially in the low-CTR region and thus became more efficient in acquiring clicks. Therefore, with the bidding strategy with unbiased KMMP-trained click model, campaigns could acquire clicks in a more cost-effective way.

5. CONCLUSIONS

In this paper, we studied the data observation bias problem in display advertising generated from the auction selection that would hurt the performance of various supervised learning models. To address this problem, we proposed a model-free learning framework that eliminates the model bias generated from censored auction data. The derived Bid-aware Gradient Descent (BGD) learning scheme naturally incorporates the historic auction and bid information, which is the main novelty of this paper. We found that the historic bid for each instance could influence both BGD learning weight and update direction. Comprehensive empirical study based on iPinYou and TukMob datasets demonstrated the large improvement of our learning framework over strong baselines in both CTR estimation and bid optimisation tasks. With light engineering work, the learning framework was deployed on Yahoo! DSP and brought 2.97% AUC lift in CTR estimation and 9.30% eCPC drop in bid optimisation over 9 campaigns.

It is important to point out that such learning framework is flexible with other supervised learning tasks than the investigated ones in this work, such as budget pacing and frequency capping in online advertising as well as other data science problems, such as interactive recommender systems [40], off-policy reinforcement learning [10], which are our planned future work.

Acknowledgement. We sincerely thank Quan Lu from Yahoo! US for his support of the online experiment. Weinan thanks the CSC funding for supporting the research.

6. REFERENCES

[1] K. Amin, M. Kearns, P. Key, and A. Schwaighofer. Budget optimization for sponsored search: Censored learning in MDPs. *UAI*, 2012.

[2] L. Bottou, J. Peters, J. Quinonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *JMLR*, 14(1):3207–3260, 2013.

[3] Y. Chen, P. Berkhin, B. Anderson, and N. R. Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *KDD*, pages 1307–1315. ACM, 2011.

[4] Y. Cui, R. Zhang, W. Li, and J. Mao. Bid landscape forecasting in online ad exchange marketplace. In *KDD*, pages 265–273. ACM, 2011.

[5] D. M. Dabrowska. Non-parametric regression with censored survival time data. *Scandinavian Journal of Statistics*, pages 181–197, 1987.

[6] M. Dudík, D. Erhan, J. Langford, and L. Li. Sample-efficient nonstationary policy evaluation for contextual bandits. In *Proceedings of the UAI*, 2012.

[7] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *National Bureau of Economic Research*, 2005.

[8] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[9] 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 *ICML*, 2010.

[10] H. Hachiya, T. Akiyama, M. Sugiayma, and J. Peters. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, 22(10):1399–1410, 2009.

[11] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *ADKDD*, pages 1–9. ACM, 2014.

[12] K. Hirano, G. Imbens, and G. Ridder. Efficient estimation of average treatment effects using the estimated propensity score, 2000.

[13] N. L. Johnson. *Survival models and data analysis*, volume 74. John Wiley & Sons, 1999.

[14] E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[15] J. Langford, A. Strehl, and J. Wortman. Exploration scavenging. In *ICML*, pages 528–535. ACM, 2008.

[16] K.-C. Lee, A. Jalali, and A. Dasdan. Real time bid optimization with smooth budget delivery in online advertising. In *ADKDD*, page 1. ACM, 2013.

[17] K.-C. Lee, B. Orten, A. Dasdan, and W. Li. Estimating conversion rate in display advertising from past erformance data. In *KDD*, pages 768–776. ACM, 2012.

[18] L. Li. Offline evaluation and optimization for interactive systems. In *WSDM*, pages 413–414. ACM, 2015.

[19] L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *WSDM*, 2011.

[20] H. Liao, L. Peng, Z. Liu, and X. Shen. iPinYou global rtb bidding algorithm competition dataset. In *ADKDD*, 2014.

[21] Q. Lu, T. Chen, W. Zhang, D. Yang, and Y. Yu. Serendipitous personalized ranking for top-n recommendation. In *Web Intelligence*, 2012.

[22] B. M. Marlin and R. S. Zemel. Collaborative prediction and ranking with non-random missing data. In *RecSys*, 2009.

[23] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *KDD*, pages 1222–1230. ACM, 2013.

[24] R. J. Oentaryo, E.-P. Lim, J.-W. Low, D. Lo, and M. Finegold. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In *WSDM*, pages 123–132. ACM, 2014.

[25] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE, 2008.

[26] U. Paquet and N. Koenigstein. One-class collaborative filtering with random graphs. In *WWW*, 2013.

[27] C. Perlich, B. Dalessandro, R. Hook, O. Stitelman, T. Raeder, and F. Provost. Bid optimizing and inventory scoring in targeted online advertising. In *KDD*, 2012.

[28] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, pages 521–530. ACM, 2007.

[29] A. Strehl, J. Langford, L. Li, and S. M. Kakade. Learning from logged implicit exploration data. In *NIPS*, 2010.

[30] H. R. Varian. Online ad auctions. *The American Economic Review*, pages 430–434, 2009.

[31] W. C.-H. Wu, M.-Y. Yeh, and M.-S. Chen. Predicting winning price in real time bidding with censored data. In *KDD*, pages 1305–1314. ACM, 2015.

[32] J. Xu, X. Shao, J. Ma, K.-c. Lee, H. Qi, and Q. Lu. Lift-based bidding in ad selection. In *AAAI*, 2016.

[33] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *WWW*, pages 261–270. ACM, 2009.

[34] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: measurement and analysis. In *ADKDD*, 2013.

[35] W. Zhang, T. Chen, J. Wang, and Y. Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, pages 785–788. ACM, 2013.

[36] W. Zhang, U. Paquet, and K. Hofmann. Collective noise contrastive estimation for policy transfer learning. In *AAAI*, 2016.

[37] W. Zhang and J. Wang. Statistical arbitrage mining for display advertising. In *KDD*, pages 1465–1474. ACM, 2015.

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

[39] W. Zhang, S. Yuan, and J. Wang. Real-time bidding benchmarking with ipinyou dataset. *arXiv preprint arXiv:1407.7073*, 2014.

[40] X. Zhao, W. Zhang, and J. Wang. Interactive collaborative filtering. In *CIKM*, pages 1411–1420. ACM, 2013.