Real-Time Bidding Benchmarking with iPinYou Dataset

Weinan Zhang, Shuai Yuan, Jun Wang
University College London
{w.zhang, s.yuan, j.wang}@cs.ucl.ac.uk

Xuehua Shen
iPinYou Inc.
x@ipinyou.com

ABSTRACT

Being an emerging paradigm for display advertising, Real-Time Bidding (RTB) drives the focus of the bidding strategy from context to users’ interest by computing a bid for each impression in real time. The data mining work and particularly the bidding strategy development becomes crucial in this performance-driven business. However, researchers in computational advertising area have been suffering from lack of publicly available benchmark datasets, which are essential to compare different algorithms and systems. Fortunately, a leading Chinese advertising technology company iPinYou decided to release the dataset used in its global RTB algorithm competition in 2013. The dataset includes logs of ad auctions, bids, impressions, clicks, and final conversions. These logs reflect the market environment as well as form a complete path of users’ responses from advertisers’ perspective. This dataset directly supports the experiments of some important research problems such as bid optimisation and CTR estimation. To the best of our knowledge, this is the first publicly available dataset on RTB display advertising. Thus, they are valuable for reproducible research and understanding the whole RTB ecosystem. In this paper, we first provide the detailed statistical analysis of this dataset. Then we introduce the research problem of bid optimisation in RTB and the simple yet comprehensive evaluation protocol. Besides, a series of benchmark experiments are also conducted, including both click-through rate (CTR) estimation and bid optimisation.

Keywords
Benchmark Dataset, Real-Time Bidding, Demand-Side Platform, CTR Prediction

1. INTRODUCTION

Emerged in 2009 [18], Real-Time Bidding (RTB) has become an important new paradigm in display advertising [13, 4, 9]. For example, eMarketer estimates a 73% spending growth on RTB in United States during 2013, which accounts for 19% of the total spending in display advertising [7]. Different from the conventional negotiation or pre-setting a fixed bid for each campaign or keyword, RTB enables the advertisers to give a bid for every individual impression. A concise interaction process between the main components of RTB ecosystem is shown in Figure 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). Bid requests will be sent via the ad exchange to the advertisers’ buying systems, usually referred to as Demand-Side Platforms (DSPs). Upon receiving a bid request, a DSP will calculate a bid as the 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. Finally, the winner’s ad will be shown to the visitor along with the regular content of the website. It is commonly known that a long time page-loading would greatly reduce users’ satisfaction [13], thus, DSPs are usually required to return a bid in a very short time frame (e.g. 100 ms). More detailed introductions to RTB could be found in [17, 18].

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 impression, given the contextual and behaviour data (usually referred to as user segments). This is apparently also an engineering challenge considering the billion-level bid requests that a DSP could normally see in a day.

Despite its popularity, the majority research activities on RTB have been limited in advertising technology companies [13, 12, 15] so far. It is nearly impossible for researchers from academia to access the sensitive thus highly protected data. Fortunately, a three-season global competition of RTB algorithms was held by iPinYou’ in 2013. As will be discussed in Section 3.1, the competition task focuses on the bidding strategies from the DSP’s perspective: it aims to maximise the campaign’s Key-Performance-Indicator (KPI) with the budget and lifetime constraint by developing the bidding strategy. We refer such task as DSP Bid Optimisation problem. In March 2014, the dataset used in the three seasons of the competition (about 35 GB) was released for the purpose of research. To the best of our knowledge, this is the first large-scale real-world RTB dataset. We believe it will stimulate the interest of RTB research and development of DSP bidding algorithms in the whole data science research community, and further speed up the growth of RTB display advertising ecosystem. The dataset can be directly downloaded from our website of computational advertising research2.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

1http://www.ipinyou.com.cn
2http://data.computational-advertising.org
In this paper, we first report a detailed statistical analysis of this dataset. Then, we formally present the research problem of DSP bid optimisation and its simple yet comprehensive evaluation protocol. Finally, we show the experimental results of some benchmark bidding strategies as well as the click-through rate (CTR) estimation models.

2. THE IPINYOU RTB DATASET

2.1 iPinYou Demand-Side Platform

iPinYou Information Technologies Co., Ltd (iPinYou) was founded in 2008 and is currently the largest DSP in China. iPinYou is headquartered in Beijing and has offices in Shanghai, Guangzhou and Silicon Valley. iPinYou has built world class RTB technology and algorithm, proprietary cloud computing platform and patented audience profiling technology. It has served over 1000 brands in IT, financial service, auto, consumer packaged goods, travel, electric commerce, gaming and more. It has also significantly improved the advertising effectiveness and fostered extensive partnerships with domestic mainstream media and private exchanges. It is established as a leading provider of audience based programmatic advertising technology.

2.2 Data Format

There are four different types of logs in the iPinYou dataset: bids, impressions, clicks, and conversions. The logs are organised on a row-per-record basis.

The feature description and example of each column of the ad log data are presented in Table 1. Generally, each record contains three kinds of information: (i) The auction and ad features (all columns except 3, 20 and 21). These features are sent to the bidding engine to make a bid response. (ii) The auction winning price (column 21), i.e. the highest bid from the competitors. If the bidding engine responds a bid higher than the auction winning price, the DSP will win this auction and get the ad impression. (iii) The user feedback (click and conversion) on the ad impression (column 3). If the DSP wins the auction, the user feedback on this ad impression can be checked to update the DSP performance.

Note that all numbers related to money (e.g., bid price, paying price, floor price) use the currency of RMB and are multiplied by 1000, corresponding to the commonly adopted cost-per-mille (CPM) pricing model. However, in our analysis the calculated numbers (e.g., cost, average CPM, and effective cost-per-click) are not multiplied by 1000.

Along with Table 1 we want to give more detailed description for some of the columns here:
(c01) The bid ID serves as the unique identifier of all event logs and could be used to join bids, impressions, clicks, and conversions together.
(c02) The column uses the format of yyyyMMddHHmmssSSS,
(c03) The possible values include: 1 (impression), 2 (click), and 3 (conversion).
(c04) The internal user ID set by iPinYou.
(c05) The column describes the device, operation system, and browser of the user.
(c06) The domain of the hosting webpage of the ad slot.
(c07) The values were hashed.
(c08) The URL is not directly available to the DSP (e.g. masked by ad exchanges) this column will be used. The values are provided by ad exchanges. For one record, either URL or Anonymous URL ID is meaningful.
(c09) The possible values include “Fixed” (fixed size and position), “Pop” (the pop-up window), “Background”, “Float”, “FirstView” or not (“SecondView” to “TenthView”), or unknown (“Na”).
(c10) Possible values include “Fixed” (fixed size and position), “Pop” (the pop-up window), “Background”, “Float”, and “Na” which presents unknown cases.
(c11) Floor (or reserve) price of the ad slot. No bid lower than this price can be accepted. The values were hashed.
(c12) When URL is not directly available to the DSP (e.g. masked by ad exchanges) this column will be used. The values are provided by ad exchanges. For one record, either URL or Anonymous URL ID is meaningful.
(c13) The possible values include “Fixed” (fixed size and position), “Pop” (the pop-up window), “Background”, “Float”, and “Na” which presents unknown cases.
(c14) User tags (segments) in iPinYou’s proprietary audi-

Table 1: The log data format. Columns with * are hashed or modified before releasing; columns with † are only available in impression/click/conversion logs but not in bid logs.

| Col #  | Description                                      | Example   |
|--------|--------------------------------------------------|-----------|
| *1     | Bid ID                                           | 015300008...35a4f1521 |
| 2      | Timestamp                                        | 2013021800120368 |
| †3     | Log type                                         | 1         |
| *4     | iPinYou ID                                       | 35605620124122340227135 |
| 5      | User-Agent                                       | Mozilla/5.0 (compatible; \ MSIE 9.0; Windows NT \ 6.1; WOW64; Trident/5.0) |
| *6     | IP                                               | 118.81.189.* |
| 7      | Region                                           | 15        |
| 8      | City                                             | 16        |
| 9      | Ad exchange                                      | 2         |
| *10    | Domain                                           | e06f4ec7...c01cd1a049 |
| *11    | URL                                              | hz55b00000...3d6275121 |
| 12     | Anonymous URL ID                                 | Null      |
| 13     | Ad slot ID                                       | 2147689_8764813 |
| 14     | Ad slot width                                    | 300       |
| 15     | Ad slot height                                   | 250       |
| 16     | Ad slot visibility                               | SecondView |
| 17     | Ad slot format                                   | Fixed     |
| *18    | Ad slot floor price                              | 0         |
| 19     | Creative ID                                       | e39e78fdd...1ee56bcd |
| *20    | Bidding price                                    | 753       |
| *21    | Paying price                                     | 15        |
| *22    | Key page URL                                     | a8be178fdd...1ee56bcd |
| 23     | Advertiser ID                                    | 2345      |
| 24     | User Tags                                        | 123,5678,3456 |

3 http://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html#patterns
The advertisers\(^4\) and their industrial categories are summarised in Table 2. Note that in the 1st season no advertiser ID was given. The diversity of advertisers makes the dataset more interesting. As we show later in the paper, ads from different fields have greatly different user response behaviour.

The basic statistical information is given in Table 3. Specifically, the “Win Ratio” column is about the ad auction winning ratio with the default bidding strategy from iPinYou platform. Conversion rate (CVR) is with respect to the number of clicks (instead of impressions). Note that in the original record, there would be multiple clicks on the same impression. However, duplications are removed in our analysis to allow focus on the events themselves (whether users would click or convert, or not).

From Table 3 we can see that (i) all the advertisers has CTR less than 0.1% except for advertiser 2997 (0.444%). Note that 0.1% is usually around the average CTR for desktop display advertising in practice. The high CTR for advertiser 2997 confirms the difference of mobile environment where clicks are more easily generated possibly due to “fat finger” effect; (ii) Although the nine advertisers have similar CPM, their effective cost-per-click (eCPC), i.e. the expected cost for achieving one click, are fairly different. This could be caused by the target rule setting (i.e., the target user demographic information, location and time) and the market of each specific advertiser; (iii) Some advertisers do not record conversions. Even for the ones who did report conversions, their CVRs differ a lot, which could also be due to different market and conversion setting. In the table of test data, there is a conversion weight factor for each advertiser, denoted by \(N\). It shows the relative importance of a conversion against a click for each advertiser. For example, the weight factor for advertiser 3476 (tire, \(N = 10\)) is much higher than that for advertiser 2259 (milk powder, \(N = 1\)).

### 2.3 Basic Statistics

The ad CTR from two advertisers are both volatile to click the ads from Advertiser 1458 while PC users (on weekends) do so more easily generated possibly due to “fat finger” effect; (ii) Although the nine advertisers have similar CPM, their effective cost-per-click (eCPC), i.e. the expected cost for achieving one click, are fairly different. This could be caused by the target rule setting (i.e., the target user demographic information, location and time) and the market of each specific advertiser; (iii) Some advertisers do not record conversions. Even for the ones who did report conversions, their CVRs differ a lot, which could also be due to different market and conversion setting. In the table of test data, there is a conversion weight factor for each advertiser, denoted by \(N\). It shows the relative importance of a conversion against a click for each advertiser. For example, the weight factor for advertiser 3476 (tire, \(N = 10\)) is much higher than that for advertiser 2259 (milk powder, \(N = 1\)).

### 2.4 User Feedback

Figure 2 depicts some statistics of user feedback on advertiser 1458 and 3358. Specifically, the mean value with the standard error of CTR\(^5\) against some features, such as the time, location, user-agent, publisher’s ad slot size, ad exchanges, and user tags\(^6\).

We can see from Figure 2 that for different advertisers, the same feature could have a different impact on the CTR: (i) Advertiser 1458 has received the highest CTR on Monday and weekends while advertiser 3358 does on Tuesday and Wednesday. (ii) The mobile users (on Android or iOS) are more likely to click the ads from Advertiser 1458 while PC users (on Mac and Windows) prefer the ads from Advertiser 3358. (iii) The ad CTR from two advertisers are both volatile across different region locations, and the trend are different. (iv) Ad slot size is correlated with the slot locations in the webpage and the design of creatives. We can see the banner (1000 x 900) and standard (300 x 250) slots generally have the highest CTR for both advertisers. (v) Ad exchanges call for bids and host the auctions. Different publishers (or their supply-side platforms, i.e. SSPs) connect different exchanges, thus the CTR distribution on

---

\(^4\)Every advertiser of this dataset has only one campaign. Thus, these two terms are equivalent in this scenario.

\(^5\)We do not compare the CVR here because the conversion definitions across different advertisers are very different. For example, advertiser 2821 sells footwear and the conversion is defined as a purchase, while advertiser 3358 sells software and the conversion is a download.

\(^6\)Tags are re-indexed by the descending rank of their frequency.
(vi) The CTR against different user tags is depicted in a log scale because the difference is fairly large. For example, the vertical e-commerce advertiser 1458 could receive 2.5% Bidding Behaviour

In the second price auctions, the highest bids are defined as the market price for the winner. If his/her bid is higher than the market price, the advertiser wins this auction and pays the market price. Market price is always modelled as a stochastic variable because it is almost impossible to analyse the strategy of each of the thousands of auction participants [2]. A higher market price reflects a more competitive the environment.

We have an investigation of the market price of advertiser 1458 and 3358 from the perspective of a DSP. The market price mean and standard error against different features are depicted in Figure 3, where we can see that just like the CTR plot, the market price has different trends against the same domain features on these two advertisers. For example, for advertiser 1458 the bid competitiveness in the morning is higher than that in the afternoon and evening, while it is inverse for advertiser 3358. In addition, for advertiser 1458 the competitiveness in ad exchange 1 is higher than that in ad exchange 2 and 3, while for advertiser 3358, the ad exchange 2 is the most competitive one.

Comparing the market price and CTR distribution on individual features, we find that the ratio of standard error to its mean of market prices is smaller than that of CTR. This is mainly because the click observations are binary value while market prices are integers.

| Test Data |
|-----------|
| Adv. | Period | Bids | Imps | Clicks | Convs | Cost | Win Ratio | CTR | CVR | CPM | eCPC | N |
| 1458 | 13-15 Jun. | 614,638 | 543 | 0 | 45,216 | 0.088% | 0.000% | 73.57 | 83.27 | 0 |
| 2259 | 22-25 Oct. | 417,197 | 131 | 32 | 43,497 | 0.031% | 24.427% | 104.26 | 332.04 | 1 |
| 2261 | 27-28 Oct. | 343,862 | 97 | 0 | 28,795 | 0.028% | 0.000% | 83.74 | 92.17 | 0 |
| 2281 | 23-26 Oct. | 661,964 | 394 | 217 | 68,257 | 0.060% | 55.076% | 103.11 | 173.24 | 1 |
| 2977 | 26-27 Oct. | 156,063 | 533 | 0 | 8,617 | 0.342% | 0.000% | 55.22 | 16.17 | 0 |
| 3358 | 13-15 Jun. | 300,928 | 339 | 58 | 34,159 | 0.113% | 17.109% | 113.51 | 100.77 | 2 |
| 3386 | 13-15 Jun. | 545,421 | 496 | 0 | 45,715 | 0.091% | 0.000% | 83.82 | 92.17 | 0 |
| 3427 | 13-15 Jun. | 536,795 | 395 | 0 | 46,356 | 0.074% | 0.000% | 86.36 | 117.36 | 0 |
| 3476 | 13-15 Jun. | 523,848 | 302 | 11 | 43,627 | 0.058% | 3.642% | 83.28 | 141.46 | 10 |
| Total | - | 4,100,716 | 3,230 | 318 | 364,243 | 0.079% | 9.845% | 88.82 | 112.77 | - |

For the winner. If his/her bid is higher than the market price, the advertiser wins this auction and pays the market price. Market price is always modelled as a stochastic variable because it is almost impossible to analyse the strategy of each of the thousands of auction participants [2]. A higher market price reflects a more competitive the environment.

Here we have an investigation of the market price of advertiser 1458 and 3358 from the perspective of a DSP. The market price mean and standard error against different features are depicted in Figure 3, where we can see that just like the CTR plot, the market price has different trends against the same domain features on these two advertisers. For example, for advertiser 1458 the bid competitiveness in the morning is higher than that in the afternoon and evening, while it is inverse for advertiser 3358. In addition, for advertiser 1458 the competitiveness in ad exchange 1 is higher than that in ad exchange 2 and 3, while for advertiser 3358, the ad exchange 2 is the most competitive one.

Comparing the market price and CTR distribution on individual features, we find that the ratio of standard error to its mean of market prices is smaller than that of CTR. This is mainly because the click observations are binary value while market prices are integers.

2.6 eCPC

From the joint observation on Figure 2 and 3, if we regard user clicks as the return, we can find some imbalanced return-on-investment (ROI) across the cases with different features. For example, we consider the weekday features for advertiser 3358. It has the lowest CTR but the highest market price on Sunday. Compared with Sunday, it has a lower market price but an around three-time CTR on Thursday. Another example for advertiser 1458 is that it has a quite lower market price on ad exchange 3 than that on ad exchange 2, but a higher CTR there. Such ROI is effectively measured by eCPC, which shows the amount of money that needs to be spent to achieve one click. A lower eCPC suggests a more cost effective algorithm.

We depict the eCPC bars against different features for advertiser 3358 in Figure 4. Just as the first example above, advertiser 3358 suffers the highest eCPC on Sunday, while it is much cost effective on Monday and Wednesday. In fact, the eCPC varies greatly against almost every feature considered here. For example, for advertiser 3358, the advertising to iOS users is about 3 times cost effective than that to Windows users. Its two low ROI creatives have the size of 200 × 200 and 360 × 300. The auctions from ad exchange 3 are much more cost effective than those from ad exchange 2. The users with tag 22, 38, 44, 49, 51, and 65 show significantly higher interest on this advertiser’s ads than other users.

I ideally, if certain kind of features brings a lower eCPC than average, the advertiser (or DSP) should allocate more budget (via bidding higher) in such auctions. The observation from Figure 4 indicates that there is great optimisation...
potential. For example, if the advertiser reallocates part of the budget from Sunday to Monday, more clicks could be achieved with the same budget. But we cannot bid much higher on Monday, because the higher bid price results in higher cost, which always increases the eCPC. Thus, there is a trade-off between the achieved clicks and the eCPC. Due to the page limit, we do not show the eCPC performance for other advertisers. In fact, all advertisers have the inconsistent eCPC across different features, and the changing trends across the features are different, too.

From the above data analysis we can see the same features would have much different impact on the user feedback and market price of different advertisers, which results in different eCPC. Therefore, it is reasonable to independently build the user response models and bidding strategies for each advertiser.

3. TASK AND PROTOCOL

3.1 Task Description

The DSP bid optimisation task for a given advertiser refers to optimising a predefined KPI given a cost budget and the coming bid requests during the lifetime of the budget. For example, a straightforward KPI is the click (conversion) number and the task is to maximise the click (conversion) number with the given budget and coming bid requests. We write the general optimisation formula as:

\[
\begin{align*}
\max_{\text{bidding strategy}} & \quad \text{KPI} \\
\text{subject to} & \quad \text{cost} \leq \text{budget}.
\end{align*}
\]

In the iPinYou bidding algorithm competition, the KPI was a linear combination of the click number and conversion number:

\[
\begin{align*}
\max_{\text{bidding strategy}} & \quad \#\text{click} + N \cdot \#\text{conversion} \\
\text{subject to} & \quad \text{cost} \leq \text{budget},
\end{align*}
\]

where \( N \) is the parameter showing the relative importance of conversions to clicks and it varies across different advertisers, as has been shown in Table 3. This KPI is practically meaningful since conversions are the final measure for advertising but they are usually sparse.

As shown in Figure 1, the input of the bidding strategy is a bid request (with the auction and ad information), the output is the bid response for this bid request. In addition, in the process of making a bid decision, the budget, current cost and achieved performance is accessible to the bidding engine.

Figure 5 presents a flow diagram about the bidding strategy training framework and the following test evaluation protocol. In the latter two subsections, we discuss these two parts respectively.
To pass the next bid request (in the ascending order of achieved click and conversion number) as zero. Defined budget, initialising the cost and performance (e.g., with the information of the budget, current cost and achieved conversion logs). A detailed description of each step is as follow:

1. Bidding Strategy
   - Update the budget, cost and performance
   - Floor price (column 18 in Table 1), the bidding engine wins the auction and gets the ad impression.
   - If test data is available, go to step 4.

2. Bid price for this bid request
   - Auction win price
   - User feedback
   - user feedback notice
   - User feedback notice

3. Auction winning checking
   - bid > winning bid ?
   - Logistic regression and tree models can be used here. We will later show the details in the experiment section.

4. Win notice: the charged price & user feedback if win
   - Data statistics
   - Bid request features distribution
   - Market price model

5. Go to next record

### 3.2 Training Framework

As discussed in Section 2, given the training data, one can investigate the data to learn about the bid request feature distribution and derive its relationship with the user ad response (e.g., CTR) and the market price. As shown in the left part of Figure 5, based on the components of the CTR estimator and the market price model, one can use the training data to perform the bidding function optimisation and finally obtain the bidding strategy.

With the feature engineering on the bid request data and the label extraction from user feedback data, one can train a CTR estimator to predict the probability of the user click on a given ad impression. Standard regression models such as Logistic regression and tree models can be used here. We will later show the details in the experiment section.

Besides the CTR estimation, the market price distribution can also be estimated either by direct data statistics or a regression model, which is called bid landscape forecasting [6]. With the knowledge of the market price distribution, one can estimate the probability of winning a specific ad auction and the corresponding cost given a bid price.

A bidding function generally takes the predicted CTR (pCTR) and the market price distribution as input and outputs the bid price. As a framework description, here we do not specify the process of optimising the bidding function, which could be different for different models. In fact, the market price model has not been formally considered in previous bid optimisation work [15, 12]. In addition, other factors can also be added into this training framework.

### 3.3 Evaluation Protocol

The evaluation protocol is illustrated as in the middle and right part of Figure 5. Given the bidding strategy and a budget for the test period for a particular advertiser, we can perform a simulation via going through its bid logs and comparing the bidding results with impression, click, and conversion events. A detailed description of each step is as follow:

1. Initialise the bidding engine, such as setting a pre-defined budget, initialising the cost and performance (e.g., achieved click and conversion number) as zero.
2. To pass the next bid request (in the ascending order of the timestamp) to the bidding engine. A bid request contains both contextual and behavioural data of the auction as well as the ad data as shown in Table 1.
3. The bidding strategy computes a bid for this request (with the information of the budget, current cost and achieved performance). This step is highlighted in Figure 5 as it is what bid optimisation focuses on. Note that if the cost has been higher than the budget (i.e., the budget is run out), all the bid responses should be set to zero (i.e., to skip all the left bid requests).
4. To simulate the auction by referencing the impression logs: if the bid price is higher than the logged auction winning price (i.e., paying price on column 21 in Table 1) and floor price (column 18 in Table 1), the bidding engine wins the auction and gets the ad impression.
5. To match the click and conversion events in the logs for this impression if winning the auction. The performance of the bidding strategy is then updated and saved. The cost is also added by the paying price.
6. To check whether there is any bid request left in test data to determine if the evaluation needs to terminate. Note that a tiny amount of over-spend from the last impression is possible but it is neglected in our evaluation.

Compared the procedures in Figure 1 and 5, we can see that step 1 and 3 are the same. The step 4 in evaluation flow actually merges the step 4 (win notice), 5 (ad delivery and tracking) and 6 (user feedback notice) in Figure 1, which is reasonable for offline evaluation since the log data has already collected the user feedback information.

It is worth noting that there are limitations of replaying logs for offline evaluation. Click and conversion events are only available for winning auctions (having impressions); there is no data for the lost auctions. Thus, there is no way to check if the performance could be improved if the algorithm bids higher to win those lost auctions. However, this evaluation protocol do follow the convention of the offline evaluations from sponsored search [10], recommender systems [19] and Web search [5] where the objects (auctions) with unseen user feedback have to be ignored.

### 4. BENCHMARK EXPERIMENTS

As discussed before, it is routine to perform the real-time bidding with a two stage process [15, 12]: (i) estimating the CTR/CVR of the ad impression being auctioned; (ii) making the bid decision based on the evaluation and other information. For iPinYou dataset, different advertisers have largely different setting on their conversions and other information. For iPinYou dataset, different advertisers have largely different setting on their conversions and more than half of them has no conversion record. Therefore, in our benchmark experiment, we will focus on the clicks. We first test the performance of two standard CTR estimators.

Then we compared several bidding strategies...
based on the pCTR. Besides the total achieved clicks, the considered KPIs include conversions and the iPinYou KPI in Eq. (3).

4.1 CTR Estimation

4.1.1 Compared Models and Feature Engineering

In our benchmarking, we consider the following two CTR estimation models. Besides the model setting, the feature engineering details are also discussed respectively.

**Logistic Regression (LR)** is a widely used linear model to estimate CTR in computational advertising [16]. The loss is the cross entropy between the predicted click probability and the ground-truth result. In addition, L2 regularisation is used.

In our experiment, all the features for LR are binary. Specifically, we extract the weekday and hour feature from timestamps. User agent text is processed to extract the ordered combination features, which can hardly be achieved by the feature engineering of LR. However, once finishing training, GBRT will get rid of most features and only keep a small part of features for prediction. In our experiment, we leverage the open-sourced xgboost for implementation. Specifically, we set the max tree depth as 5 and train 50 trees with 0.05 learning rate.

Different with the binary features for LR, here every feature for GBRT is continuous. Specifically, for the indicator features (e.g., city=152) of a domain (e.g., city) we calculate the frequency and CTR based on a subset of the training file and let these two numbers be the values of two GBRT features (e.g., the frequency that city=152 is \(438,432\), the empirical CTR for the cases whose city=152 is 0.1%). We do not make the frequency value for tag features. For the continuous feature (e.g., slot floor price), we directly use the specific value as the feature value (e.g., slot floor price=21). In sum, we have 937,748 binary features for LR training and prediction.

**Gradient Boosting Regression Tree (GBRT)** [8] is a non-linear model widely used in regression and learning to rank applications [3]. Comparing to the linear model LR, GBRT has the advantage of learning the non-linear features, which can hardly be achieved by the feature engineering of LR. However, once finishing training, GBRT will get rid of most features and only keep a small part of features for prediction. In our experiment, we leverage the open-sourced xgboost for implementation. Specifically, we set the max tree depth as 5 and train 50 trees with 0.05 learning rate.

**Evaluation Measure**

The area under ROC curve (AUC) is a widely used measure for evaluating the ad CTR estimator [10, 14]. Besides AUC, the root mean square error (RMSE) is also chosen as the evaluation measure here as it is widely used in various regression tasks. Because of the huge imbalance of positive/negative cases in ad clicking, the empirically best regression model usually provides the pCTR very close to 0, which results in the RMSE having a quite small value and the improvement on RMSE is much slight, compared with AUC.

### Table 4: CTR estimation performance.

| Season | Adv. | AUC | GBRT | RMSE | AUC | GBRT |
|--------|------|-----|------|------|-----|------|
| 2      | 1458 | 0.9881 | 0.9707 | 0.0191 | 0.0263 |
| 3      | 2259 | 0.6865 | 0.6791 | 0.0177 | 0.0176 |
| 3      | 2261 | 0.6238 | 0.5739 | 0.0168 | 0.0167 |
| 3      | 2821 | 0.6325 | 0.5820 | 0.0239 | 0.0238 |
| 3      | 2907 | 0.6039 | 0.5979 | 0.0582 | 0.0581 |
| 2      | 3358 | 0.9753 | 0.9722 | 0.0246 | 0.0279 |
| 2      | 3386 | 0.9008 | 0.7686 | 0.0284 | 0.0285 |
| 2      | 3427 | 0.9735 | 0.9342 | 0.0214 | 0.0245 |
| 2      | 3476 | 0.9625 | 0.9422 | 0.0230 | 0.0231 |
| 2      | Total | 0.9141 | 0.9200 | 0.0263 | 0.0260 |
| 3      | Total | 0.7615 | 0.7715 | 0.0268 | 0.0268 |
| 2,3    | Total | 0.8307 | 0.8518 | 0.0270 | 0.0263 |

### 4.2 DSP Bid Optimisation

#### 4.2.1 Compared Bidding Strategies

We compare the following bidding strategies in our benchmark experiment for DSP real-time bidding. The parameters of each bidding strategy are tuned using the training data. And the evaluation is performed on the test data.

- **Constant bidding (Const).** Bid a constant value for all the bid requests. The parameter is the specific constant bid price.
- **Random bidding (Rand).** Randomly choose a bid value in a given range. The parameter is the upper bound of the random bidding range.
- **Bidding below max eCPC (Mpc).** The goal of bid optimisation is to reduce the eCPC. In [12], given the advertiser’s goal on max eCPC, which is the upper bound of expected cost per click, the bid price on an impression is obtained by multiplying the max eCPC and the pCTR. Here we calculate the max eCPC for each campaign by dividing its cost and achieved number of clicks in the training data. No parameter for this bidding strategy.
- **Linear-form bidding of pCTR (Lin).** In the previous work [15], the bid value is linearly proportional to the pCTR under the target rules. The formula can be generally written as \(\text{bid} = \text{base_bid} \times \text{pCTR} / \text{avgCTR}\), where the tuning parameter \(\text{base_bid}\) is the bid price for the average CTR cases.

Among the compared bidding strategies, only Mpc is non-parametric and it does not consider the budget limits while the others do by tuning their parameters. In addition, Mpc and Lin need to evaluate CTR for each impression. Thus we denote the two bidding strategies with
LR CTR estimator as Mcpc-L, Lin-L and with GBRT CTR estimator as Mcpc-G and Lin-G.

4.2.2 Experimental Setting

The evaluation follows the protocol in Section 3.3. The only issue discussed here is about the pre-set budget for each advertiser. In our experiment we set the budget for each advertiser as a proportion of the original total cost in the test log. Particularly, in order to check the bidding strategies’ performance under different budget limits, we set the budget as 1/32, 1/8, and 1/2 of the original total cost in the test log. Note that we cannot set the budget higher than the original total cost because in such case, simply bidding as high as possible on each auction will make the DSP win all the auctions without running out of the budget.

4.2.3 Results

We list the achieved click performance for each algorithm under different budget limits in Table 5. We can see from Table 5 that Lin and Mcpc generally work much better than Const and Rand, which verifies the importance of impression-level evaluation and real-time bidding. To some advertisers, such as 2259 and 2261 with 1/32 and 1/8 budget limits, Mcpc achieves fewer clicks than Const and Rand. This is because Mcpc is not adaptive with different budget limits, which results in running out of budget quite soon for the low budget settings. Moreover, Lin works much better than Mcpc. Compared with Mcpc, Lin has the ability to change the bidding scale by tuning its parameter base_bid, which helps Lin adapt different budget limits against the coming ad auction volume. Such adaptivity is essential to DSP bidding strategies because the ad auction volume and market price could vary a lot as time goes by [6]. Generally, if the budget allocated per ad auction is high, then it is encouraged to bid high to achieve more user clicks and conversions, while if the budget allocated per ad auction is low, the bid price should be reduced to remain the high ROI.

Table 6 lists the conversion performance on four advertisers with conversion records. And Table 7 lists the corresponding iPinYou KPI score of click + N * conversion, where N for each advertiser has been shown in Table 7. We can see the significant improvement of Lin against other bidding strategies. Another common important point from the results of these three KPIs is that in the lower budget setting, Lin achieves the higher improvement rate against other strategies. This is because when the budget is quite limited, it is more important to identify which cases are probably valuable and adaptively lower the overall bid price.

5. OTHER RESEARCH TOPICS

Besides the CTR estimation and DSP bid optimisation problem, there are other potential research topics where this dataset can be used for experimentation.

- **Bid landscape modelling.** As previously mentioned, for RTB based display advertising, market price can varying quite a lot [6]. The current DSP work [15, 12] mainly focuses on the bidding strategy based on CTR/CVR estimation. However, another important factor for the bidding decision making is the market price. As a repeated auction game with the budget constraint, the optimal bidding strategy is not truth-telling while it does depend on the market price distribution. Thus the research problem of modelling the market price distribution is much meaningful for a better bidding strategy.

- **Adaptive bid control.** Just like sponsored search, a practically significant problem is to adaptively control the scale of real-time bidding to exactly deliver the budget during the campaign’s lifetime. For some advertisers setting a high bidding scale at the beginning, their budget could be run out just at the beginning of the campaign lifetime. The techniques of pacing [11] could be used here. More generally, in [2], the authors model the bid control problem in sponsored search via Markov decision processes (MDPs) where the left budget and ad auction volume are states while the bid prices are actions. Further research on adaptive bid control can be based on this dataset.

6. CONCLUSIONS

Due to the sensitivity, research on computational advertising, especially from academia, is seriously restricted by the data availability. With the publication of iPinYou dataset, we believe the research on RTB display advertising will be stimulated.
In this paper, we performed a detailed statistical analysis on iPInYou dataset; formally defined the bid optimisation problem; and presented a simple yet comprehensive offline evaluation protocol for bidding strategies. We conducted benchmark experiments on CTR estimation and DSP bid optimisation with several baseline algorithms. The dataset and report are hosted on our website for computational advertising research.

7. REFERENCES

[1] A. Ahmed, A. Das, and A. J. Smola. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In WSDM, 2014.

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

[3] C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. Learning, 11:23–581, 2010.

[4] T. Chakraborty, E. Even-Dar, S. Guha, Y. Mansour, and S. Muthukrishnan. Selective call out and real time bidding. In Internet and Network Economics, pages 145–157. 2010.

[5] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In WSDM, pages 87–94. ACM, 2008.

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

[7] eMarketer. Rtb and second-price auction. http://www.emarketer.com/article/RTB-Ad-Spend-Continues-Robust-Growth/1009783, 2013.

[8] J. H. Friedman. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4):367–378, 2002.

[9] Google. The arrival of real-time bidding. Technical report, 2011.

[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 ICML, pages 13–20, 2010.

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

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

[13] S. Muthukrishnan. Ad exchanges: Research issues. In Internet and network economics, pages 1–12. Springer, 2009.

[14] 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 WSDM, 2014.

[15] C. Perlich, B. Dalessandro, R. Hook, O. Stitelman, T. Raeder, and F. Provost. Bid optimizing and inventory scoring in targeted online advertising. In KDD, pages 804–812, 2012.

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

[17] J. Wang, S. Yuan, X. Shen, and S. Seljan. Real-time bidding: A new frontier of computational advertising research. In CIKM Tutorial, 2013.

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

[19] 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.