Web Economics Individual Report

Boyang Liu
University College London

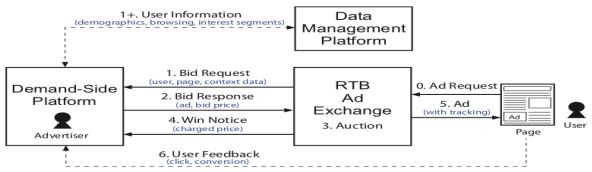

Figure 1: Procedure of RTB

KEYWORDS

ACM proceedings, LaTeX, text tagging

1 INTRODUCTION

In the domain of computational advertising, RTB(Real Time Bidding) performs an important role. Compared with traditional bidding methods, this bidding system is designed to sell the specific chance of advertising to the advertiser with highest requirement, ensuring the maximum profit of both advertiser and the information provider(IP). The web users can also benefit in this way as their opportunities to view ads will be sold to the most suitable advertisers providing the most fitted ads.

Figure1 clearly shows the working procedure of RTB. When a web user requesting a web page the information related to the user and the page will be broadcast to advertisers. The advertisers need to claim a price based on the information they captured about this bidding, no matter from the IP or its own database. What need to be mentioned is that in RTB generally second price auction is applied, which means that the advertiser claiming the highest price will win the auction, while the second highest price will be payed. If the advertiser wins the auction, it will be informed and its advertisement will be shown to the user. The advertiser can also obtain the user feedback such as whether the ad is clicked by the user, which can be used to optimize their bidding algorithm further.

To obtain more profit in advertising, bidding strategies are the most important. Ideal bidding strategy can help the advertiser to maximize the click-through-rate(CTR), which means that their product linked to their ads will be viewed most.

In this project we are required to explore a best bidding strategy based on the given dataset, for the advertisers in it.Here the bidding strategy can be divided into two parts. One is the learning algorithm to better predict the CTR of different impressions. The other is the bidding algorithm calculating the bidding price based on the predicted CTR(pCTR). As an individual I chose to concentrate on the first part. To get benefit from the most state-of-the-art learning models, a deep learning model was implemented at first, but it didn't come with ideal results as imagined. Then I turned to algorithms based on Gradient Boosting Decision Tree(GBDT). The algorithm was implemented using multiple packages, and finally LightGBM was chosen to be well tuned and tested, which brought the best results among all the methods our group tested.

Table 1: Statistical information of Training Set

Adv.	Imps	Clicks	Cost	CTR	CPM	eCPC
3427	402806	272	30458.71	0.000675	75.62	111.98
2821	211366	131	18828.04	0.000620	89.08	143.73
1458	492353	385	33968.74	0.000782	68.99	88.23
2259	133673	43	12428.24	0.000322	92.97	289.03
3386	455041	320	34931.82	0.000703	76.77	109.16
3358	264956	202	22447.23	0.000762	84.72	111.13
3476	310835	187	23918.78	0.000602	76.95	127.91
2261	110122	36	9873.78	0.000327	89.66	274.27
2997	49829	217	3129.27	0.004355	62.80	14.42
total	2430981	1793	189984.61	0.000737562	78.15	105.96

2 RELATED WORK

A number of literatures are found useful for this individual part.
For the data exploration part, 5 is found to be a good reference and inspiration.
The implementation of CNN is inspired by 1 and 3 a lot. Both of them are discussing applying Neural Networks in this field, together with the open-sourced code implemented by Theano and TensorFlow.
The idea to turn to GDBT is inspired by 2 and 4.2 obtains good results with a variation of GDBT, and 4 gives a detail comparison of different packages based on GDBT.

3 APPROACH AND RESULTS

3.1 Data exploration

Table 2 shows the basic statistical information of the validation set.

Table 1 shows the basic statistical information of the training set.

From the table we can see that there is a large difference among the advertisers, while the train set and validation set keep consistent on the metrics.

3.1.1 Feature Exploration, Selection and Engineering.

There are 24 features for each impression in the dataset. however, not all of the features are equally significant for our prediction. Some of them are irrelevant attributes or unique identifiers for each bid, and such meaningless information could negatively affect our modelling power and cripple the predictive accuracy. Therefore, we just simply eliminate them from our training set. These features include *bid ID*, *user ID*, *user IP*, *url*, etc. The rest features comprises both numerical variables as well as categorical variables. Features

Table 2: Statistical information of Validation Set

Adv.	Imps	Clicks	Cost	CTR	CPM	eCPC
3427	50183	37	3776.74	0.000737	75.26	102.07
2821	26503	23	2394.90	0.000868	90.36	104.13
1458	62353	49	4294.60	0.000786	68.88	87.64
2259	16715	2	1568.81	0.000120	93.86	784.40
3386	56665	28	4350.79	0.000494	76.78	155.39
3358	32939	23	2794.02	0.000698	84.82	121.48
3476	38841	11	2993.75	0.000283	77.08	272.16
2261	13550	3	1214.88	0.000221	89.66	404.96
2997	6176	26	388.78	0.004210	62.95	14.95
total	303925	202	23777.27	0.000665	78.23	117.71

like *slot width* and *height*, or the *floor price* are numerical variables which could be directly feed into our model without any processing.

For features like *hours* of a day, it is difficult to interpret it as a numerical variable or a categorical variable. On the one hand, 7pm is closer to 6pm or 8pm than it is to 2pm or 10pm, but on the other, there is discontinuity between 23pm and 0am. This attribute is actually an ordinal cyclic variable, so we would like to adopt trigonometric approach to transform time information by using the following equations so that the beginning hour is the same as the end.

$$xhr = sin(2\pi \times \frac{hr}{24}) \tag{1}$$

$$yhr = cos(2\pi \times \frac{hr}{24}) \tag{2}$$

For indicator features like city and region, we compute its frequency proportion of the whole training data. For example, there are 370 different cities in our dataset, and the frequency occurrence of city 2 is 1.55%, so each of the city id will be substituted by its occurrence proportion. The similar approach is also applied to feature region.

Dealing with other categorial variables could be more straightforward. What we did is to encode every category as a one hot encoding vector. This binary vector contains |i| (number of values in this category) elements where all columns are equal to zero except for the category column. Features like user agents, advertiser, user tag, advertiser will be processed in this way.

In sum, we have 127 features which for our training our model and click prediction.

3.1.2 Down-sampling.

Class imbalance is a common problem in online realtime bidding. Based on our data exploration, there are more than 2.4 million winning impressions in our dataset, however, the clicked ones only account for 0.073%. As most conventional classification algorithms are often biased towards the majority class, not taking the data distribution into consideration. The low number of clicks (1793) could make the training process extremely difficult. In order to deal with the problem and improve the bidding performance, our group decide to balance the proportion of class labels by under-sampling the majority class with 0 clicks. To explain in more detail, we empirically choose a set of negative sampling rate, ranging from 0.001, 0.01, 0.025, 0.05, 0.075, 0.1 and test the prediction accuracy of the learning model on the validation set. However, the performance of different models on different training size also varies. Our findings are summarised in Tablexx. The best performance for XGBoost achieved with 0.025 sampling rate, FM is at 0.1 and for DNN, it works better at 0.075 sampling ratio. Therefore, we train our individual model with different sized sample data. Ofparticularnoteis that after a model is trained in a negative sampled dataset, the output probability from a classifier should be calibrated. For example, if we use 0.1 sample rate to construct the training set, the empirical CTR will increase for about 10%. In order to re-calibrate the result to get back to the upsampled one, we re-calibrate the prediction according to the following equation:

$$q = \frac{p}{p + (1-p)/w} \tag{3}$$

Where q is the calibrated probability, p is the prediction in under-sampling space (predicted CTR) and w is the under-sampling rate.

3.1.3 Evaluation metric.

The whole bidding system is composed of two parts: a CTR estimator and a market price model. Different metrics will be applied to evaluate these two models.

CTR estimation model.

As the goal of a CTR Estimator is to predict whether a specific ad impression will be clicked under a given context, it can be simply treated as a binary classifier, in which the system is required to determine the click result. Therefore, percentage correct makes intuitive sense. However, although accuracy is a important to measure most classification model, it does not hold well against our highly imbalanced dataset. As more than 99.9% impressions are associated with 0 clicks, even if the model choose to always predict âĂIJlabel-0âĂİ, we can get an extremely high accuracy. Therefore, to avoid getting misleading result, we will first compute the confusion matrix for the true positive, true negative, false positive and false negative class for two types of predictions. For example, the true positive class exists when the actual click of is 1 and the prediction is 1 as well. Furthermore, based on the confusion matrix, we could calculate and discuss the precision, recall, F1 score and more importantly, plot the Receiver Operation Characteristic (ROC) curve and the Area Under the Curve (AUC) on a chart. The ROC curve plots the false positive rate (X-axis) against the true positive rate (Y-axis) (see equation 4) and 5)). If the AUC score is equal to 1, it implies all clicks are correctly predicted. Any values of AUC higher than 0.5 represents the modelsâĂŹ prediction is better than a random predictor.

$$False\ Positive\ Rate = \frac{FP}{FP + FN} \tag{4}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{5}$$

Bid Price Estimation Model.

The performance of our bid price estimation model will be evaluated based on the following metrics, within a specified limited budget of 6250 CNY fen.

Clicks: This is the main metric of our bidding strategy, indicating the click number of our impressions won.

Figure 2: Constant Bidding and Random Bidding Strategy

Click-Through Rate (CTR): CTR describes the the ratio of clicks to impressions won. A higher CTR means more visitors and better advertising performance.

Spend: This is the total amount of money we spent for all impression won. In our case, we have to spend all 6250 budget to win the impressions.

Average Cost Per Millie (CPM) and **Average Cost Per Click (CPC)**: This is the cost per 1,000 impressions and the average cost for every click. The lower CPM and eCPC are, the more cost efficiency an algorithm provides.

Our main aim is to enhance the total number of clicks from the winning impressions under the limited budget. However, instead of solely based on that, we would also manage to obtain a higher CTR, and reduce the CPC as well as CPC to get the biggest economic benefit.

3.2 Basic Bidding Strategies

3.2.1 Constant biding strategy.

The constant bidding strategy is a simple strategy which bids all impressions using a specific constant bid price. As the market price in the validation set ranges from 0 to 300, we decide to set the constant bidding price from 1 to 301 and count the total number of click. After testing all the 301 values on the validation set, we obtained the optimal bid price as 80 fen, which will generate 68 clicks (See Figure 2).

3.2.2 Random biding strategy.

The random bidding strategy is optimized on the validation set with 6250 CNY fen budget, whose bid-price is generated from the uniform distribution. The upper bound and lower bound of the bidding range are tuned as parameters. The performance of this random bidding strategy is maximized with the lower bound equals to 35 and upper bound equals to 100 which can acquire 74 clicks on the validation set in average. The highest number of click achieved by this model could be 80.

When tuning the parameters, first, we fixed the lower bound and try different upper bounds in the range of lower bound plus 10 to 330 with an interval of 10. Then, different lower bounds in the range of 0 to 100 are tried. Since the performance of the random bidding strategy could vary from time to time, we run the model ten times for each pair of parameters and calculate the average clicks. The average clicks of each pair of parameters is plotted to help us to find the best pair of parameters.

Figure 2 shows the performance of random bidding strategy when the lower bound is fixed as 35. The x axis shows the value of upper bound and y axis shows the number of average clicks. We can see that the random bidding strategy obtains the highest average clicks, which equals to 74, with the lower bound equals to 35 and upper bound equals to 100.

Table 3: CTR Estimation Performance

Model	AUC	Accuracy	Precision	Recall	F1 Score
LR	0.794	99.93%	20.00%	0.50%	0.97%
FM	0.876	99.93%	46.94%	11.39%	18.33%
XGB	0.885	99.94%	81.25%	12.87%	22.22%
LGBM	0.892	99.85%	18.05%	36.63%	24.18%

3.3 Linear Bidding strategies

Besides using constant or random bidding strategies, the normal solution is to estimate or âĂIJlearnâĂİ a bidding price from the information about the specific bidding. Among varieties of bidding strategies, linear bidding strategies are widely used, usually as a baseline due to its easy implementation, high efficiency and ability of interpretation. The linear bidding strategy consist of 2 parts, namely a linear CTR estimator and a linear bidding formula, of which the former is usually a linear machine learning algorithm to predict the CTR, and the later is a function linear to the predicted CTR to provide the bidding price.

3.3.1 CTR estimation model. As a basic CTR estimator, Logistic Regression algorithm is applied. In this algorithm the prediction can be expressed as:

3.3.2 Bid price estimation. linear

3.4 Combined Bidding strategies

3.4.1 Separate Bidding strategy Analysis. Our group have developed a total of four different models for pCTR estimation. The first one is a simple Logistic Regression Model, the second one is based on Factorisation Machine (FM), the last two applies scalable boosting methods XGBoost and LightGBM. Details of the model implementation can be found in individual report and their performance results have been summarised in Table 3.

Compared with the baseline LR model, all three improved models have displayed a far better performance in most aspects. Even though the LGBM does not achieve an accuracy higher than LR, this can be attribute to the characteristics of highly imbalanced dataset. However, if we focus on AUC, we could find the three models have generated an 10% higher scores than LR. As the AUC scores can be seen as useful metric for datasets with highly unbalanced classes, this reflect the overall performance of the models. Additionally, XGBoost has achieved the best precision, which means it has an advantage of identifying negative labels where as LGBM performs best in terms of finding all the positive samples (click=1).

To generate the final bidding price, five bid estimators have been applied and combined with these CTR predictors. The click number results of different strategy combinations has been shown in Table 4. Each row in 4 represents a single bidding price generation strategy and the columns list different CTR prediction models. The Values

Table 4: Click Performance of different Bidding Strategies

Strategy&Mode	LR	FM	XGB	LGBM
Constant			68	
Random			74	
Linear	142	169	167	167
ORTB1	-	-	166	-
ORTB2	-	-	166	-
PRUD	-	159	-	-

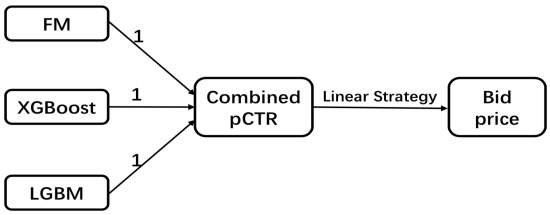

Figure 3: Pipeline of Bidding Strategy Combination

area displays the number of clicks generated by a particular strategy combination, which is the result tested on the validation set.

By comparing the click performance of different combinations, we could find the Linear, ORTB and PRUD algorithm generally works much better than Constant and Random bidding strategy. As the latter strategies takes the predicted CTR into consideration, this situation verifies the significance of our impression value evaluation. Furthermore, even though Linear bidding strategy is relatively straightforward, it can always obtain a higher click number than others by simply tunning the base_bid and to adapt different bidding circumstances.

3.4.2 Combined Bidding Strategy.

In order to further enhance the bidding performance, our team decide to combine different pCTR estimation models into one predictive model. With an aim of decreasing variance, reducing bias and improving the predictions, such ensemble method is commonly used in machine learning. In our CTR prediction case, each model can be seen as a weak classifier. By analysing their pCTR estimation performance, we find FM, XGBoost and LGBM share approximately the same prediction result. Therefore, we average together multiple estimations from these models (average the pCTR outputs from three models) and use this new pCTR to represent the final click-through rate of the impressions in the validation set. The pCTR of the impressions in the test set will be predicted in the same way.

After generating the pCTR, we need to choose a best bidding strategy to compute the final bidding price. Initially, we would like to apply the linear bidding strategy as it always provides us the best click performance when combined with individual pCTR estimators. The combined pCTR of the validation set will be used to search for an optimal base_bid. Once the base_bid has determined, the final bid price of the test set impressions can be obtained.

lr;xg;fm;lgbm;åŘĎçğĮ)æĺąåđŃ evaluation metric èąĺæăij

åŘĎçğĮ)strategyçŽĎevaluation metricèąĺæăij

Table 5: Original Bidding Model Performance

Impression	Spend	Clicks	CTR	avgCPC
1744	2461.99	12	0.006881	205.17

Table 6: Modified Model Performance (add constant)

Constant	Impression	Spend	Clicks	CTR	avgCPC
0	2040	2331.46	19	0.009314	122.71
10	2409	2448.19	19	0.007887	128.85
30	4115	2936.99	19	0.004617	154.58

Table 7: Modified Model Performance (Multiply Factor)

Factor	Impression	Spend	Clicks	CTR	avgCPC
1	1683	2489.67	22	0.013072	113.17
1.3	5956	6229.20	43	0.007220	144.87
1.5	5885	6250.23	39	0.006627	160.26

pctr threshold åŻ¿

combined biddingïijŽ åęĆä¡Ţcombineå Ż¿ïijŽcombine pctrïijŇ combine price+pctr threshold

combineåŘŐèąĺçŐřclick;CTR;spend;avg CPM; avg CPC

page6 end

3.5 Multi-agent Bidding strategies

Based on our best single-agent bidding model, a multi-agent bidding model is proposed. The single-agent model is improved to be more adaptable to the multi-agent situation.

Since our test environment is shared with other research groups, the test results could vary due to the different opponentsâĂŹ bid-price. Therefore, we always test new multi-agent models and the original single-agent model at the same time and compare their results to evaluate if the new models have better performance in this uncertain environment.

First, the original single-agent bidding model is tested in the multi-agent environment. From table 5 we can see that around 40% of budget is spent and the numbers of impressions and clicks are low. It seems that the multi-agent environment is more competitive, so our initial model cannot win much impressions and the budget is not fully utilized.

We assume that the bid-price distributions of all opponents are fixed. To improve the probability of wining, we can move our bid-price distribution to the right to some extent by adding a constant to our original bid-price. From table 6, we can see that, the impression increases significantly after adding the constants, while the number of clicks remains the same. Although, this method increases our probability of wining, the performance becomes worse regarding CTR and avgCPC. There are much more impressions with no click than impressions are clicked, so if we add the same constant to each price, we could waste much budget on impressions with no click.

Therefore, to maximum the number of clicks with limited budget, the bid-price of impressions are more likely to be clicked should be

raised more. The pCTR shows the probability that an impression being clicked, so the increase of bid-price should be positively related to pCTR. As explained before, the bid-price generated from our single-agent model is positively related to the pCTR, so we multiply the original bid-price by a factor to generate the new bid-price. The table 7 could suggest that the model performance can be maximum with the factor equals to 1.3 when the number of clicks doubles, but the CTR and avgCPC become slightly worse than the original model. This 1.3 factor model also achieved the first rank in the test environment. When the factor is set as 1.5, the number of clicks begins to decrease due to the lack of budget.

Other factors like 1.1 and 1.2 are also tried, these results are not presented due to the space limitations. We cannot compare these results directly due to the variation of the test environment, but by comparing with the original model, the experimental results suggest that a factor in the range of 1.2 to 1.3 should be able to fully optimize our model in the multi-agent situation.

4 CONCLUSION

To start with, basic data exploration is done, including the tables of the required metrics, further analysis on user feedback, and some extra analysis. Based on such preliminary analysis, basic bidding strategies including constant bidding, random bidding, and linear bidding are implemented as described in detail in the group report. Taking the linear strategy as the baseline to compared with, mainly two non linear bidding algorithms are implemented. The CNN model is attempted to be a novel state-of-the-art solution by applying minor modifications such as convolutional layer. Although the results turns out to be disappointed, the reason is also analyzed in this report. After that, a much more robust model based on GBDT is implemented and tuned using LightGBM package. The results are fully analyzed comparing with linear strategy, proving that the model can bring best results with 167 clicks on validation set.

However, although no efforts is spared to obtain the best bidding model with best performance, there are still potential improvements due to the limited time. For example, the deep learning method doesn't bring good results described in many papers. This may be tested on more complex dataset.

REFERENCES