COMP0124 Multi-Agent Artificial Intelligence Group Project (Group 1)

Tianshan Li
University College London
Mathematical Computation
tianshan.li.15@ucl.ac.uk

Xin Chen
University College London
Mathematical Computation
xin.chen.15@ucl.ac.uk

Xiaofeng Fu
University College London
Software Systems Engineering
xiaofeng.fu.15@ucl.ac.uk

1. INTRODUCTION

Online advertising becomes one of the most important way for advertisers to market their products, and Real-Time Bidding (RTB) is a popular paradigm to display advertisements on a website. The advantage of this paradigm is that it allows the advertisers to pay a different price for every impression, so advertisers can bid a high price for the impression that is predicted to be clicked and bid a relatively low price or give up for other impressions.There are multiple different RTB auction systems, for instance, first pay price and second pay price auction system. The second pay price system dedicates that the winning advertiser pays the second highest price in the auction.Moreover, as one of the core parts of the RTB, the using of different bidding strategies will result in significantly different cost and utility. For example, linear bidding strategy is one of the basic strategy in RTB. For finding a optimal bidding strategy we will analyze basic bidding strategies and evaluate the behaviors of each of them, and then we will train different machine learning models based on the information in the dataset and evaluate the performance of the model.

In this report, we will explore the related work of RTB industry briefly in section 2. In section 3, we will analyse the data set and extract basic statistical information from the data, basic bidding strategies will be evaluated as well. Then we will apply Click Through Rate (CTR) estimation to create a linear bidding strategy and experiment further to develop a more optimal non-linear bidding strategy. Finally, a game theory based bidding strategy will be researched. The related code can be found on https://github.com/xiaofengF/multi-agent-ai.

2. LITERATURE REVIEW

In real-time bidding system, the Demand-Side Platform(DSP) or an advertiser aims to buy the impressions which is more likely to be clicked in real-time auction. In order to determine these kinds of best-matched impressions, advertisers should use models to estimate the click through probability for a specific impression via its contextual and behavioural data and then quickly decide whether and how much the bid worth. Besides, avoiding reducing user's satisfactory [4], the whole process returns a bid within 100ms.

We got some inspiration from following papers to make our own bidding strategies. The Weinan Zhang et.al [6] statistically analysed the full dataset of iPinYou and provided some basic bidding strategies such as constant bidding, random bidding, bidding below max eCPC. They also introduced some benchmark algorithms of CTR estimation, i.e., logistic regression (LR) and gradient boosted decision trees (GBDT). As we know, the CTR estimation is a widely studied object, the notable examples are random forest (RF) method [2], neural network (NN) [5] and Factorisation machines (FM) [7]. LR is most the typical models for using estimated CTR value because of its simplicity and efficiency. NN models could get very precise results but is limited by its complicated interpretation. GBDT and RF model can build multiple decision trees and merges them together to get a more accurate and stable prediction. [1] FM performs good result and it is more suitable for sparse data with high level features iteration.

Finally, in the multi-agent real-time bidding environment, actor-critic framework is used to solve the multi-agent reinforcement learning problem in "*Real-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising* "[5].

3. DATA EXPLORATION

3.1 Basic Statistics Analysis of Data Sets

Before we start to analyse the data, the basic definition we used are as follow: Click-Through-Rate (CTR):

$$CTR = \frac{total\ number\ of\ clicks}{total\ number\ of\ impressions}$$

Average cost-per-mille (CPM):

$$agvCPM = \frac{cost \times 1000}{total\ number\ of\ impressions}$$

Effective cost-per-click(eCPC):

$$eCPC = \frac{cost}{total\ number\ of\ clicks}$$

In particular, the unit of all the number related to money (Cost, CPM, eCPC) is Chinese fen.

3.2 Overall Analysis of Data Sets

According to the training data set, we have analysed the data and calculated the related value (CPM, CTR, eCPC), which is given in Table 1.

From the Table 1, we could see the average CTR value of nine advertisers is around 0.001 (0.1%). Specifically, most advertisers have CTR value lower than 0.1%, apart from the Mobile e-commerce app install industry (advertiser 2997),

Adv	Imps	Clicks	Cost	CTR	CPM	eCPC
1458	492353	385	33968.736	0.078%	68.99	88.23
3386	455041	320	34931.823	0.070%	76.77	109.16
3427	402806	272	30458.711	0.068%	75.62	111.98
3476	310835	187	23918.779	0.060%	76.95	127.91
3358	264956	202	22447.231	0.076%	84.72	111.12
2821	211366	131	18828.044	0.062%	89.08	143.73
2259	133673	43	12428.238	0.032%	92.97	289.03
2261	110122	36	9873.779	0.033%	89.66	274.27
2997	49829	217	3129.267	0.435%	62.80	14.42

Table 1: Dataset statistics: Training Data

whose CTR value is about 5.5 times better than the next highest advertiser 2458. This result is very similar to what [6] has been mentioned in the paper that that the advertiser 2997 is in the mobile environment which is more clickable than others due to the "fat finger" effect. In addition, the cost and impression value of advertiser 2997 are also lower others, which causes the extremely low value of eCPC and more effective clicks comparing with other company.

Although the average CPM for the nine companies are fairly similar in the range of 60 to 93, their eCPC values are extremely different due to the wide variation in clicks. And the deeper reason for this result is caused by the rule setting for choosing target customer.

3.3 User Feedback

In this section, we made a statistical summary of user feedback of advertiser 1458 and 3358 as examples by comparing their mean value of CTR against some features.

According to Figure 1 we have been analysed, we could observe that the CTR value of each advertiser in the same feature was totally different caused by different parameters. For example:

1. We find that the CTR value of advertiser 3358 drops in bottom of 0.00025 but achieves an extremely high level around 0.0010 on Wednesday. On the contrary, the CTR value of advertiser 1458 was relatively stable during the week.

2. Advertiser 3358 has a noticeable peak in CTR at 11PM and the CTR values approach to zero at 5 and 7 AM. Besides, the CTR values of advertiser 1458 fluctuate between 0.0015 and 0.0005 across the day.

3. IOS as a user platform shows the entirely different CTR values of advertiser 1458 and 3358, one reaches a peak but one stays at bottom. For the browser, Safari appears to have high CTR value for both advertisers.

4. The slot size suggests that both advertiser 1458 and 3358 predominantly obtains better CTR value on size 300x250 and 1000x90.

5. The CTR values from both advertiser are undulate across different region location. Especially, the advertiser 1458 appears a comparable high CTR value in region 393.

4. BASIC BIDDING STRATEGIES

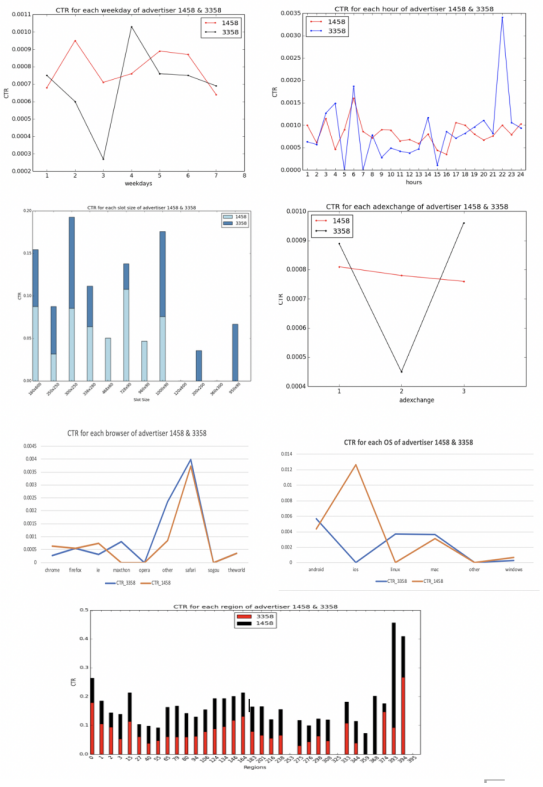


Figure 1: CTR distribution against different features for advertiser 1458 and 3358.

In this section, we will evaluate three different basic bidding strategies (constant bidding, random bidding and multi-agent random bidding) based on the validation set. Constant bidding strategy is that the advertiser bid for every requests with a constant value. For the random bidding strategy, advertisers randomly choose a bid value that in the bounds of the bidding range.

To conduct the strategy evaluation, the evaluation protocol created by Zhang is used to set up the experiment and simulate the auction system. The experiment involves 4 steps [6]:

1. Initialise the bidding system (Set up budget, initialising the cost and performance).

2. Input the bid requests to the bidding system based on the default rank.

3. Use different bidding strategies to give a bid for this request.

4. If the bid price is higher than the pay price and the floor price in the dataset, the bidding system wins the auction (Winning criterion #1).

In our experimental setup process, we decide to set the budget as 6250 Chinese fen, which is same as our final evaluation criterion.

4.1 Constant Bidding

This bidding strategy has only one parameter: the specific constant bid price. To find the best performance, we need

Advertiser	highest pay price	average pay price	lowest pay price	clicked: highest pay price	clicked: average pay price	clicked: lowest pay price
1458	300	68.87562748	3	300	86.63265306	15
2259	294	93.85629674	1	201	177.5	154
2261	294	89.65874539	0	133	84.66666667	60
2821	294	90.36335509	1	253	141.0869565	23
2997	277	62.9507772	4	270	116.7692308	6
3358	238	84.82409909	1	231	118.5652174	28
3386	300	76.78095826	0	282	104.3928571	20
3427	238	75.25925114	1	207	93.83783784	12
3476	252	77.07708349	0	226	99.81818182	42

Table 2

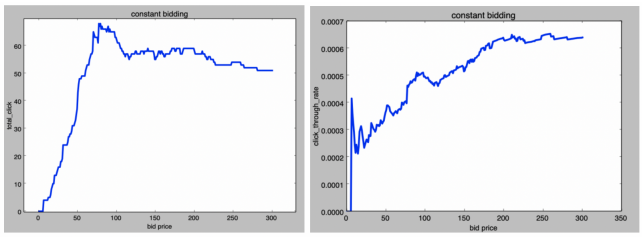


Figure 2

a relatively high bid price to win as many impressions as possible, but we also need the price to be low enough to avoid running out the budget early (Only 6250 Chinese fen). Therefore, we have discussed different methods and finally we decide to conduct statistical analysis of the data set first to evaluate the data.

According to the Table 2, we list all the important stats of the data set and we found that there exists lowest pay price equals to 0, this situation means that no one compete with the advertiser for this impression. After discussion, we decide to use two methods to analyse the optimal constant value. Firstly, we set the constant bid price as different significant values: the average pay price, highest clicked pay price and average clicked pay price separately in the experiment to analyse the result. In the experiment, we use python to simulate the bid request and bidding process, and the experiment result is:

1. Average pay price: total clicks: 68, CTR: 0.00047
2. Highest clicked pay price: total clicks: 51, CTR: 0.00064
3. Average clicked pay price: total clicks: 57, CTR: 0.00047

Secondly, we enumerate the bid price from 1 to 300, the highest pay price in the validation data set, to find the optimal value. In Figure 2, the click number and CTR are shown with the increasing bid price. The optimal constant bids is between 77 and 79 with 68 clicks in the left picture of Figure 2 while the best CTR is in an increasing trend with the rising of the bid price. It is unexpected that the optimal constant bid is the same as the average pay price.

4.2 Random Bidding

The parameters of this strategy is the upper bound and lower bound of the bid price range. As the same method above, we firstly set three bounds manually to analyse the data. Since the random bid price will lead to random results, we simulate each pair of bounds for 10 times to get a relatively accurate result. The left value is upper bound and the right value is lower bound:

1. Highest pay price | lowest pay price: average clicks: 54.3, avgCTR: 0.00054
2. Highest clicked pay price | clicked average pay price: average clicks: 54.9, avgCTR: 0.00062
3. Highest pay price | average pay price: average clicks:56.9, avgCTR: 0.00058

Moreover, we briefly enumerate the bounds in the bidding system. The lower bound is set from 1 to 100, and the upper bound is from (lower bound + 10) to 300. Each time the upper bound is increased by 10 until it reaches 300. To avoid the randomness of the results, we take the average bid price of 5 random values for each pair of bounds in the enumeration. Finally, the optimal lower bound is equal to 44 and the upper bound is equal to 94. We get 77 total clicks for this optimal bounds, which is higher than the number in constant bidding, and the CTR is equal to 0.00048. It is clear that the optimal bounds have a much better performance than the bounds set manually.

4.3 Multi-agent Random Bidding

For multi-agent random bidding, the main difference between this strategy and the strategies above is the competition of every impression from other bidders. To simulate the actions of other bidders, we add 50 agents in our bidding system, and all of them use the optimal bounds we obtained in the random bidding part (2, 127) to bid for a impression. Then we try the same process above manually to compete with these 50 agents:

1. Highest pay price | lowest pay price: average clicks: 17.5, avgCTR: 0.00034
2. Highest clicked pay price | clicked average pay price: average clicks: 29.1, avgCTR: 0.00062
3. Highest pay price | average pay price: total clicks:28.7, avgCTR: 0.00058

Then we enumerate the bounds as the same process above, the lower bound from 1 to 100 and upper bound from ïijĹlower bound + 10ïijĽ to 300. And we get the highest click number 49 for the upper bound 146 and lower bound 16. The average CTR is 0.00069 and it is quite high comparing with all above strategies. Compared with the single agent strategy, the multi one has a higher upper bound and smaller lower bound. However, since the price is selected randomly, so the result will be different every time in our experiment.

In Figure 3, we try to find out the relationship between agents number and the number of clicks. We use the optimal bounds we got - (16, 146) to compete with agents vary from 50 to 100 in our bidding simulation system. It shows that the number of clicks floats significantly through the changing of the agent number, and the highest click number is obtained when competing with 63 agents while the lowest click number appears when competing with 62 agents. In our experiment, all other agents use the same bidding strategy with the same bounds, Therefore, it may be the reason that causes the trend of this diagram.

5. LINEAR BIDDING STRATEGY

The linear bidding strategy consists of two parts. First, build a CTR estimation to predict the probability of click

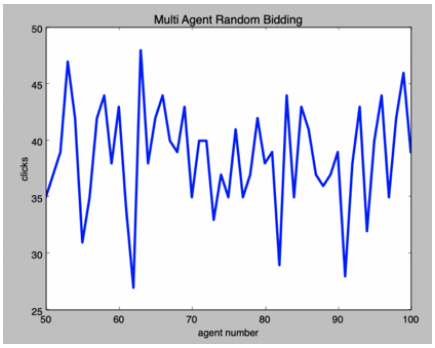


Figure 3

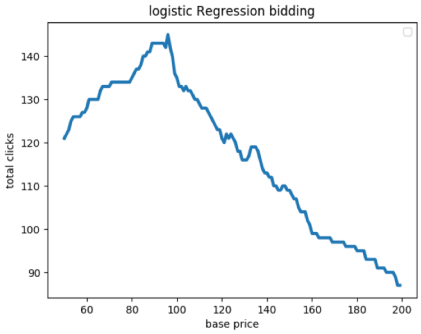


Figure 4

(pCTR value) for each ad impression in the training set. Second, calculate the optimal bid price for each impression in validation set based on the pCTR value and other bid related parameters. And the relation between the bid price and the pCTR value has been defined as:

$$bid = base\_bid \times \frac{pCTR}{avgCTR}$$

Note the bid value is linearly proportional to the pCTR value and the tuning parameter base_bid is the bid price for the average CTR cases.

5.1 CTR Prediction

Logic Regression is used as a pCTR (click-through-rate prediction) estimation mechanism with linearly bidding mechanism in computational advertising and it is a supervised learning classification model that can predicts discrete variables (click: yes or no). The key consideration we choose this model is that logic regression output "well-calibrate "predicted probabilities.

We use a Python machine learning library, Scikit-Learn, to develop the logistic regression model. The model is trained by using the training data set and validated by using the held out set provided. To encode the training feature, one-hot encoding method is used to transform the features into binary data, and the following features were dropped since they were almost unique for each case, or meaningless to be add to LR training:

Dropped features: 'adexchange', 'click', 'bidprice', 'payprice', 'bidid', 'IP', 'userid', 'creative', 'domain', 'url', 'urlid', 'slotid', 'keypage'.

Furthermore, 'useragent' and 'usertag' are different from other features. 'useragent' contains information of both OS and browser type. And 'usertag' includes numerous tags to identify the feature of a user. It is difficult to encode them without further processing. We divide operating system 'useragent' into two different columns- 'os' and 'browser' and 'usertag' is split to multiple unique tag columns. Meanwhile, the feature 'slotprice' needs to be divided into different slot price bucket since it has a continuous price. So 5 buckets are created automatically by using Pandas. For the strategy evaluation part, we considered using the logarithmic loss, ROC curve and AUC as evaluation metric to calibrated probabilities.

5.2 Linear Model

The second part of linear bidding strategy is using the following equation:

$$bid = base\_bid \times \frac{pCTR}{avgCTR}$$

where the avg CTR is the averaged CTR in the training set, and the pCTR is the predicted probability of a click as based on the CTR estimation model. The optimal parameter for pCTR is penalty L2 with 200 iterations, which was found from *sklearn* grid search with five-fold cross validation. In our experiment, the optimal base bid price value which achieved the highest click number is 97 CNY. The Figure 4 shows the representation of the base bid optimisation.

6. OPTIMAL BIDDING STRATEGY

6.1 Non-linear Strategy

For non-linear bidding strategies, we have conducted research on 3 different models - random forests, GBDT and neural network. Moreover, The performances of these three algorithms are also compared. Both random forest and GBDT are ensemble learning algorithms and predict classification by combining results from decision trees. Both of them have better prediction performance than logistic regression, but also have some drawbacks. In terms of GBDT, it is difficult to tune the parameters and GBDT is more sensitive to over-fitting if the data are noisy. In addition, both of GBDT and random forests take a long processing time on real-time prediction when the number of trees becomes large. For the neural network algorithm, it can learn and model non-linear and complex feature relationships. Also, it can generalise - infer unseen relationships on unseen data and makes better model at predicting on unseen data [3].

The experiment consists of 3 steps: data preprocessing, training and predicting,and bidding simulation. In the data processing stage, we used the same method that used in linear bidding strategy that data were one-hot encoded to fit the training model. Then in training and predicting process, we inputted training data set to train the model (x_train and y_train) and the 'click'column of validation data set was used to verify the accuracy of the prediction. Moreover, since different models require different parameters, grid search method was used to get the optimal parameter inputs.

As mentioned before, GBDT and random forests take a long processing time to train a model and data pre-processing

Algorithms	AUC Score	log loss	total click	base_bid
GBDT	0.863	0.0022	156	112
Random Forests	0.848	0.0043	154	138
Neural Network	0.853	0.0039	156	105
Logistic Regression	0.790	0.0047	146	97

Table 3

take a while as well. To speed up the training process, we limited the number of iterations to 100 and edited the one-hot encoding function. In addition, We found that modifying the data frame during pre-processing 'usertag', for example, grouping by a table consumes a long time. Therefore, we used $CountVectorizer$, a sklearn pre-defined function to generate one-hot encoding of 'usertag'.

In the evaluation part, AUC score and log loss were used to evaluate the performances. AUC (Area under the Curve) score is a performance measurement for classification problem at different thresholds, the higher AUC means better the model is at predicting 0s as 0s and 1s as 1s. Log loss (Logarithmic loss) is also a way to measure the performance of our models. It evaluates the uncertainty of our prediction based on the level it varies from the validation label.

After we had the set of pCTRs of different models, bid prices were calculated based on the formula shown in previous section:

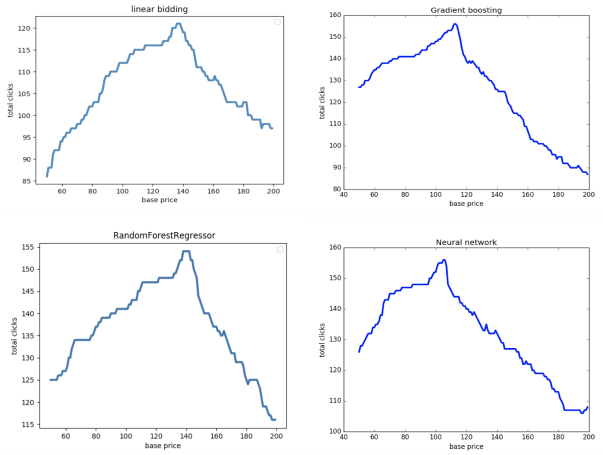
$$bid = base\_bid \times \frac{pCTR}{avg(CTR)}$$

Simulating the bidding process was implemented by comparing the bid price with pay price in the validation set. Then the evaluation result is shown in Table 3, we can find that GBDT and neural network models have better AUC scores compared with others. Both of them can reach 156 total clicks in the validation set, and it is interesting that this click number is two times the best click number of basic bidding strategies we evaluated above. Moreover, GBDT has a very small log loss at 0.0022 while others log loss are around 0.004.

6.2 Game Theory and Best Strategy

In this section, we will discuss the combination of multi-agent bidding strategy and game theory. Game theory is a well-known study of strategic interaction between different strategy makers. It can be divided simply into two main directions, repeated games and simultaneous games. For the repeated games, agents play out over and over for a period and the strategy made by a agent is strongly depended on the previous step of the opponent. Yet for simultaneous game, agents make decisions together and they do not know others' strategies.

In the real time bidding system, all agents give their bid prices at the same time and they do not know each other's strategy, which can be considered as a simultaneous game. Especially for this case, it is a classic prisoner's dilemma, which is a game theory example that rational individuals



will not necessarily cooperate in which it has better payoff. Agents can either bid for the real price of each impression or bid for a high price and pay for the second highest price. The best-case scenario is that they both bid for the real price and then they will cost less and get a distributed click number. However, the agent who bid for a high price can gain by just losing (second highest price - real price). For example, assuming a impression is valued $10 for agent 1 and $12 for agent 2, if both of them pay for the real price, agent 2 will win this impression. If agent 1 pays for a high price, $20, he will only pay for $12 and win the impression and only $2 lose ($12 - $10). Furthermore, If both of them pay for a high price, the one who wins the impression will lose much.

The problem here is that there are too many agents within the RTB system (more than 30 groups) and it is impossible for all groups to play together. Therefore, to get a relatively high click number when competing with other groups in the project, the prediction of their strategies is very important.

Two questions are asked by ourselves to analyse the opponents'strategy:

1. What options will the competitor actively consider?
2. Which option will the competitor most likely choose?

To answer these questions, we have cout out with some possible strategies that maybe used by our opponents.

Price: true value < P0 < P1 < P2 < all-in price

Note: P1 is the bid price for other competitors. We assume that other competitors set a higher bid price than true price because they aim to win more impressions.

1. Strategy: Bid P1 for all bid requests.
2. Strategy: Bid a true value for all bid requests.
3. Strategy: Bid P1 for the impression that has a high predicted CTR, and bid a true value for other bid requests.
4. Strategy: Bid P2 for the impression that has a high predicted CTR, and bid P0 for other bid requests.
5. Strategy: Only bid a all-in price for the impression that has a high predicted CTR, and do not bid for other bid requests.

	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5	Strategy 6	Performance
Strategy 1	0	+1	+1	−1	+1	+1	3
Strategy 2	−1	0	−1	−1	−1	−1	-5
Strategy 3	−1	+1	0	−1	+1	−1	-1
Strategy 4	+1	+1	+1	0	+1	+1	5
Strategy 5	−1	+1	−1	−1	0	−1	-3
Strategy 6	−1	+1	+1	−1	+1	0	1

Table 4

6. Strategy: Bid a all-in price for the impression that has a high predicted CTR, and bid a true value for other impression.

In Table 4, we analysed the interaction between each of these strategies with each row represents the strategy we would like to use and each column represents the strategy used by other agents. We mark +1 for advantages, -1 for disadvantages and 0 for no gain no loss, and we use Python to run these strategies in pairs one by one with the same prediction model to identify whether each strategy has advantages against others.

We can find that honesty is not the best policy in case of RTB, and it has the worst performance and bid for all-in price is very risky when other agents also bid for a high price. You may lose all budget when competing for one impression. In conclusion, strategy 4 has the best performance no matter what strategy opponents choose. This strategy means we assume others will bid a price higher than the true value for every impression. To get a better click number, we split the bid price into two cases: first, when we get a comparable high pCTR value, we provide a very high bid price P2 to win the impression. Second, we give a relatively high bid price P0 for the rest of impression whose pCTR value is not incredible. Note P0 is higher than true price but lower than the high price bid by other agents because it can increase the cost of other agents when they compete with us. This method ensures we can keep a stable number of clicks before we run out of budget. Although strategy 4 also has many drawbacks (for example, high cost, difficulty of setting the price level), it can ensure our team to get a stable number of clicks in this competition, meanwhile, increase our opponent's cost.

Strategy 4 is a weakly dominant strategy in game theory. The main difficulty of this strategy is to find the price level of P2 and P0. To find P2, we use

$$Bid\_P2 = highest\_basebid \times \frac{highest\_pCTR}{avg(CTR)}$$

to calculate the highest possible price that we may bid. Specifically, the highest_basebid value is the base bid value for random forest model. Since random forest model has the greatest value of base bid compared with other models, we use this value to prepare the worst if other competitor set a large base bid value. Then we multiply the bid price by 2 as the P2 to achieve weakly dominant. In our experiment, the highest bid price can reach around 350000. For P0, we applied the normal bid price formula:

$$Bid\_P0 = base\_bid \times \frac{pCTR}{avg(CTR)}$$

but slightly increase the base bid to 200 to gain more possible click.

Another difficulty of this strategy is to determine what is a 'high pCTR'. Therefore, we decide to set a baseline of the 'high pCTR'. If a pCTR is higher than the baseline of the 'high pCTR', which means this impression is highly possible to be clicked, we will bid P2 for this impression. After discussion, we use GBDT in validation set to get the mean pCTR of the impressions which has been clicked as our baseline. Actually it is impossible to find a absolute optimal price in our approach because we do not know other's price level and strategy, and the price level can only be evaluated after using in a real environment.

Finally, we choose our best performance model: GBDT to apply our game theory approach, and we gained rank 1 before we submit the report.

7. CONCLUSION

In conclusion, we have shown several optimal bidding strategies which are successful in estimating the probability of click (pCTR) for each impression and deciding whether and how much the bid worth across an unseen data set within a fixed budget. Firstly, we enumerated some basic bidding strategies and simply analysed each of them. Secondly, we provided four CTR estimation models (logistic regression, gradient boosted decision trees, random forest, neural network) to predict the probability of click for each impression. These models had been trained by training data set and used sklearn grid search to find optimal parameters for each model. By comparing their AUC source and log loss, we could find the best model to use in the real-time bidding. In our experiment, GBDT model performed better than the models, which reached 171 clicks in the test. Then, we used the following formula to calculate the optimal bid price:

$$bid = base\_bid \times \frac{pCTR}{avg(CTR)}$$

Different from the single-agent environment, we should also consider what bidding strategies our competitors would use in multi-agent environment. Based on the game theory, we used weakly dominant strategy combined with our best performance model, GBDT, to set a suitable bid price for auction. Furthermore, after exploring the literature, we decided to use the multi-agent reinforcement learning to optimise our multi-agent strategy models to achieve more clicks.

8. REFERENCES

[1] N. Donges. The random forest algorithm https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd. 2018.

[2] J. M. Kim and M. L. S. K. H. . Hojin Jung Corresponding author: School of Economics, Henan University. Predicting bid prices by using machine learning methods. 2014.

[3] J. Mahanta. Introduction to neural networks, advantages and applications https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207. 2017.

[4] S. Muthukrishnan. Ad exchanges: Research issues . in internet and network economics, pages 1 to 12. springer. 2009.

[5] J. Wang. Yong yu, weinan zhang, kleanthis malialis, defeng guo,han cai, kan ren, real-time bidding by reinforcement learning in display advertising,arxiv:1701.02490v2 [cs.lg] 12 jan. 2017.

[6] J. W. Weinan Zhang, Shuai Yuan and X. Shen. Real time bidding benchmarking with ipinyou dataset. arxiv preprint arxiv: 1407.7073, 2014. 2014.

[7] Q. L. T. X. H. M. Zhen Pan, Enhong Chen and H. Lin. Sparse factorization machines for click-through rate prediction. in 2016 ieee 16th international conference on data mining. 2016.