COMPGW02/M041 Web Economics Coursework - DSP Real-time Bidding Strategy for Online Advertising

Group 10 - Group Report

Han Diep
MSc Business Analytics
University College London
ucabhhd@ucl.ac.uk
SID:16121098

Lorenz Luboldt
MSc Computer Science
University College London
lorenz.luboldt.17@ucl.ac.uk
SID:17129593

Federico Ziviani
MSc Computer Science
University College London
federico.ziviani.17@ucl.ac.uk
SID:14060185

1 INTRODUCTION

Unlike traditional sponsored search or contextual advertising, RTB allows an advertiser to submit a bid for each individual impression in less than 100ms [5]. RTB has recently changed the landscape of display advertising and according to Google, it is destined to last. From 2010 and 2011 the growth of spending in RTB touched 68% [1]. This technology requires two components from a Supply Side Platform(SSP) perspective. An API and a bidder (e.g. Google's Invite Mediafis Bid Manager). The first is a server-side pipe that retrieves available impressions, making them available for bidding. The second is the 'brain 'that selects impressions form multiple API's, being responsible to build the best inventory. Each time a user visits a web page, a bid request for the impression is triggered by the publisher and sent to a Demand Side Platform (DSP), via an ad exchange. The process is repeated each time, and the highest bid wins the impression. The RTB brings tangible benefits to buyers. It scales the impression inventory available to buyers, including the most remote ones, who wonfit otherwise be reached. Evidence from Google reports that: 'In April and May 2011, campaigns executed via RTB versus those executed through non-RTB mechanisms, provided a 19% savings on CPM rates and raised CTR performance by .06 percentage points, from .09% to .15%'[1].

'From a Demand side perspective, the three key components to consider in RTB, are the utility estimation, based on response prediction, the cost prediction, centered on the competitive landscape, and the optimization of a bidding strategy [3]'. Before coming up with the appropriate bidding strategies, we want to figure out what exactly we are trying to predict in our model. Our goal is first to maximise the number of clicks just estimating an optimal constant bid price and a bid range, without considering any other selection features. Secondly, we want to set a real-time bid price linearly correlated to pCTR per impression learnt by user response prediction models detailed below. Finally, we want to explore other non-linear bidding models and compare them against the trivial models. We are not trying to predict the types of users and ad types, because our datasets are already targeted and homogeneous in terms of users and ad type. If such simplification was not provided, we would first need to set a target context: figure out the user preferences and match them with our offering, then figure out what kinds of users visit our site and who is competing with us. In our case, we are provided with an uniform ad 'block '. The dependent variable to predict is the probability of click, given a series of features identified as the independent variables. Our value function is maximized under a budget constraint, for the best weighted feature combination. Our non-trivial algorithms aim to learn what is their weighted influence on a click and how they correlate to each other in a click prediction pattern.

2 RELATED WORK

Although bid optimisation is a well studied field, historic research has mainly focussed on classic keyword auctions optimising on CPC, rather than RTB per-impression ecosystems optimising on CPM. In terms of value-based bidding, the 'linear bidding strategy' suggests the bid price to be linearly related to the respective pCTR, which implies large correlation with prior distribution of impression features, as the input parameter is the click prediction [2]. In order to get the bidding engine to work, pre-processes need to be run such as CTR estimator training as explained in [6]. Several user prediction models were studied through the extensive work of [4], e.g. linear models including logistic regression and Bayesian probit regression and non-linear models including factorisation machines and gradient tree models. Please refer to section 4.2.1 for an in-depth explanation of the prediction models that have been used.

3 DATA SET & TOOLS

- Data Description:

- The train dataset contains more than 2.4 millions rows, each row represents a bid for an impression. Each row has 25 columns which represent different features for that impression, these features can be grouped as the following:

- Slot details include: slot-id, slot-height, slot-width, slot-visibility, slot-format, slot-price.

- Users details include: user-id, user-agent which is the devices that user use to browse the Internet, IP, region, city, user-tag

- Bid details include: bid-id, bid-price, pay-price, advertiser, ad-exchange, weekday, hour

- Impression data include: key-page, creative, url, url-id, click.

- Tools:

- iPython notebooks was the environment used for coding and testing. - Preprocessing data: Label Encoder and Standard Scaler from scikit-learn were used to encoded categories data and scale all data prior using them for machine learning - Statistical and calculations were performed using Statistic libraries from Scipy along with numpy and maths Several machine learning libraries from scikit-learn were used, including SVC, Logistic Classifier, Decision Tree Classifier, Random Forest Classifier along with evaluation metrics such as Accuracy Score, Roc Curve, Roc AUC. - For visualising data and analytic result, plotting were done with Matplotlib, and Seaborn

- Protocols for evaluating bidding strategy:

In order to evaluate the different bidding strategy, we developed a simple bidding protocol resembles the real time bidding, where budget is initialised. The protocol will reflect the structure of the original dataset where the given bid price will be replaced with our bid price for that impression. Our bid price will be compared with the original pay price in the dataset. The protocol will evaluate the two bids, and it returns the result of winning bids, along with number of click, CTR and budget spent.

.

4 ANALYSIS & EVALUATION

4.1 Basic Bidding Strategies

Based on an in-depth data analysis of the 'training' data set, we started our investigation on Demand-Side-Platform (Advertiser) bidding strategies by creating simple strategies which are not based on a CTR-estimator, taking the high-dimensional feature space of a bid request into account, but rather produce a bid price as an input parameter for the bidding function which is experimentally derived from the 'training' data set. We consider these bidding strategies as trivial, as they do not take into account data preprocessing methods for CTR prediction, as described as 'non-trivial' approaches in subsequent sections.

4.1.1 Constant bidding strategy. For this bidding mechanism, it is assumed that for every bid request, the DSP is bidding a constant value for every incoming bid request, until the budget of CN¥6.500 is consumed. The circumstances described imply an optimisation problem in finding a balance between two extremes: High bid prices and fast consumption of budget versus lower bid prices but risk of winning less auctions and consequently lowering the chance for conversions (clicks) on the won impressions. Rather than randomly choosing a constant bid value, a series of values is tested as the constant bid price from a range of values, which we defined between the minimum and maximum pay prices from the 'training' set, hence between CN¥0 and CN¥300. Following from that, a range of bid prices is plotted against the performance metrics previously defined (see figure 1).

When looking at the number of bids placed and bids won in figure 1, one can clearly see that for constant bid prices below CN¥70, bids are placed for all impression requests (303,925) in the 'validation' set as the budget is never fully consumed. This is due to the fact that not many bids are won with bid prices below that threshold. Above CN¥70 as bid price, the number of bids placed asymptotically decrease, as the budget is consumed more quickly on expensive impressions. This is also visible in the number of bids won. The number increases up to the peak point at bid price CN¥70 because bid prices before are too low to win most of the auctions. Bid prices preceding the peak point lead to faster budget consumption and therefore more and more bids cannot be participated in and in turn not being won which also leads to an asymptotic decrease in bids won, all due to the budget constraint. When analysing the number of clicked impressions won, one observes a maximum point of 68 clicks with constant bid prices of CN¥76 and CN¥78. The number of clicks is rapidly increasing with an increasing bid price, as far more impressions are bid on and hence clicks captured, which is also supported by the notion that no budget constraint is reached for bid prices that low. For higher bid prices than CN¥78, the number of clicked bids is decreasing but at a lower rate. As further increases on bid prices on already won clicked impressions are not effective, the decrease is only driven by the budget constraint which comes from winning more expensive impressions before new clicked impressions can potentially be won. The ceiling in budget consumption can be seen in figure 2.

Looking at overall CTR and hence the effectiveness of our bidding in terms of conversion, the CTR is perceived to increase with an increasing bid price, although not monotonically. Still, CTR is maximised by increasing bid prices, which can be explained by the fact that the number of bids (numerator) is declining faster than the number of clicks (denominator) won with increasing bid prices. This leads to a generally growing CTR ratio and in this case should not serve as the only metric in measuring the effectiveness of the bidding strategy.
Finally, cost metrics have been investigated, i.e. CPM and eCPC. Figure 1 reveals that CPM is monotonically growing for higher constant bid prices. The rapid growth until the above mentioned threshold is driven by the fact that more expensive auctions on impressions are won so that the average price per 1000 won impressions is increasing. Furthermore, despite the budget constraint which is kicking in after a bid price around the mean of is reached , the CPM growth is driven by the decreasing number of auctions won, so the average price of a won auction with stable total costs goes up. eCPC for the constant bidding strategy shows a similar behaviour. Rapid growth in eCPC is driven by higher bids which capture more winning bids than clicks can be won. A local minimum eCPC is perceived where budget is spent and maximum number of clicked impressions is captured. For higher bid prices, less clicked impression auctions are won with maximum budget spent, which leads to slowed increase in eCPC, inverse to number of clicked impressions won with increasing bid prices.

4.1.2 Random bidding strategy. The principal decision parameter to take into account in the random bidding strategy is the bid price range, since the model imposes to bid on every impression without making discriminations between factors influencing the click probability. The full price range between CN¥0 and CN¥300 was initially used to test for an optimal range (c.f. figure3).
As one can see from the figure, the tendencies and relations discovered for the constant bidding strategy are inherited in the random bidding model, as well. Furthermore, its large variance increased the chances of overpaying for non-clicked impressions, as well as underpaying and missing clicked impressions, and drastically reducing average CTR. To address this issue we looked at the distribution of pay prices for the clicked impressions (c.f. figure 4) in the 'training' dataset, since those are the impressions we aim to maximise bidding for, given the budget constraint. Looking at clicks, we identified the mode pay price at CN¥70 with an high density of clicks in its -20 to +20 price range. We set it as lower and upper boundaries for the random bidding. Since we can not choose when to bid, but only the price range, setting it between

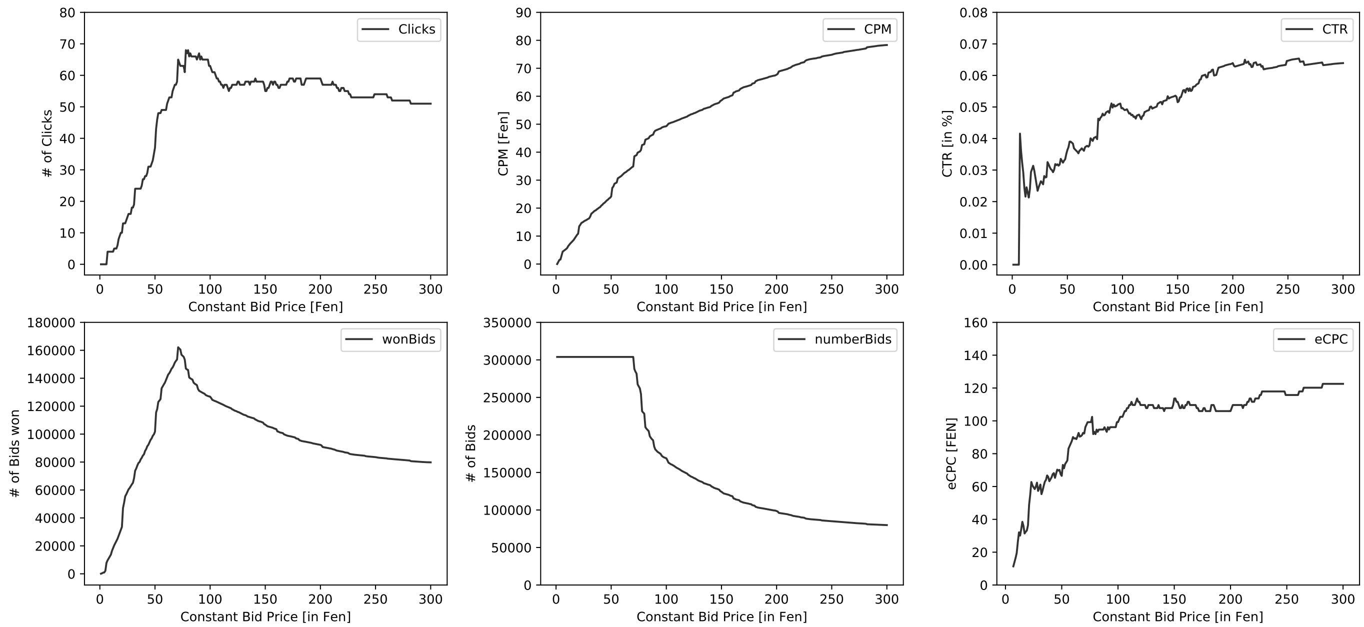

Figure 1: Summary statistics - Constant bidding strategy

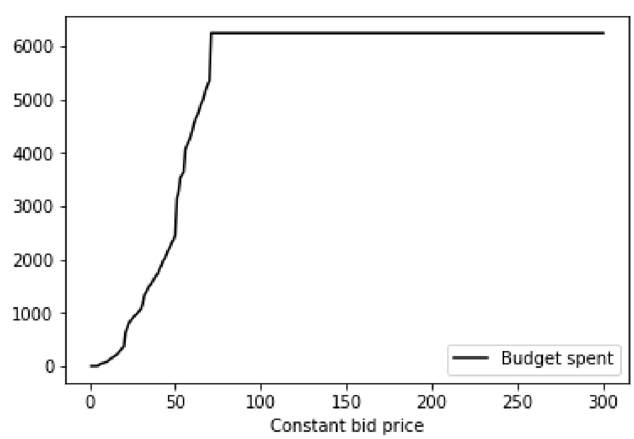

Figure 2: Budget consumption with increasing constant bid price

CN¥50 and CN¥90 represents the optimal compromise for budget saving, yet maintaining above 0.5 the chances of winning the bid for a click, when encountered. To decrease the likelihood of losing a bid given due to random bidding, we narrowed the random price boundaries to 5 price points when validating, e.g. CN¥50-55 , CN¥51- 56, ... , CN¥90-95. We acknowledge one limitation of our strategy might be missing the expensive clicks. As well as keep overpaying non-clicked impressions. Indeed, considering the tight budget constraint, always bidding above CN¥50,leads to quickly spending all money available and missing all the clicks that would come next. On the other side, this strategy gives high probabilities to win the cheapest majority or clicks. This maximizes our value function where: value = clicks / total cost. In this terms, expensive clicks perform worse than cheaper ones, therefore we decided not to remain indifferent at their cost. Further limitation comes from the predominance of non-click in the dataset, the probability of missing one click is much lower than that of overpaying a non-click. For this reason, our performance is improved when we only set the upper boundary of the random range and take all its values starting from zero. In conclusion, we recognise this model is just a local maximum, yet the probabilistically better response for our value function.

4.2 Advanced bidding strategies

In this section, we present different bidding strategies, each based on different CTR estimation sets that were produced using three machine learning frameworks.

4.2.1 Linear Bidding Strategy .

- Calculating the predicted the click through rate (pCTR):

We decided to set the pCTR on the probability of whether an impression may get clicked or not. The machine Learning models used to calculate the probabilities of click are Logistic Regression, Decision Tree and Random Forest. All the features were used to train the machine learning models except bid_id, urlid, bidprice, payprice and click.

-Preprocessing data:

Categories data such user_agent, domain etc. were pre-processed using Label Encoding, as One Hot Encoding was not suitable giving too many categories was chosen. Data was the scaled by removing the mean and scaling to unit variance. The data only has 0.07

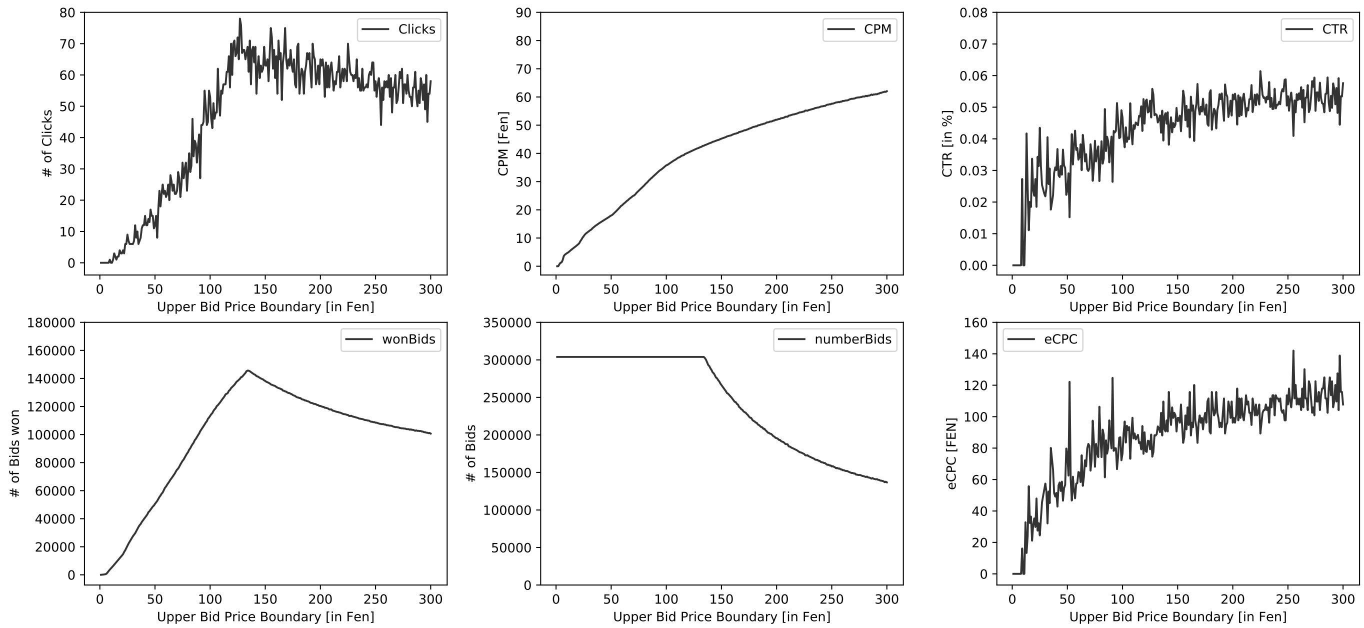

Figure 3: Summary statistics - Random bidding strategy upper price boundary

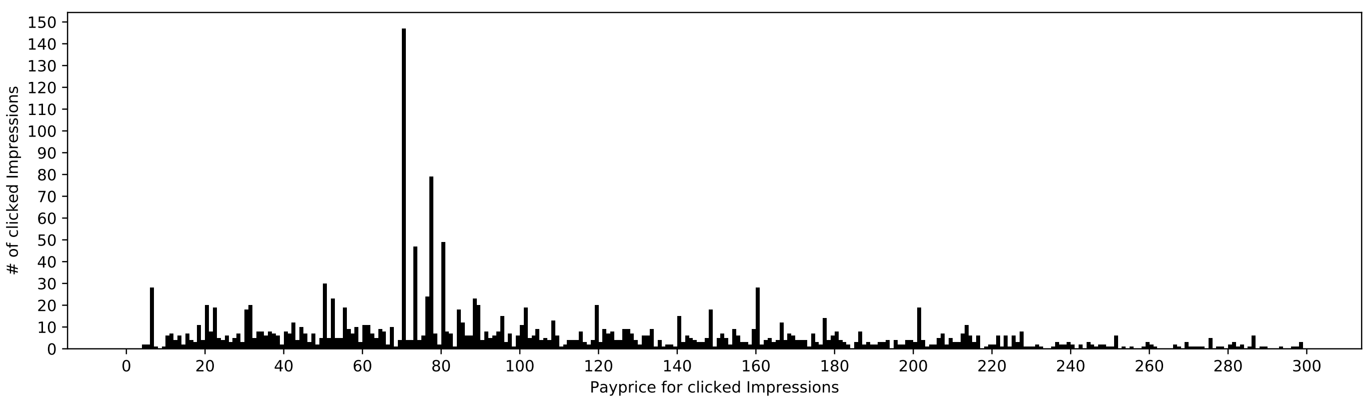

Figure 4: Distribution of pay prices for clicked Impressions

percent impressions that had clicks. This resulted in highly imbalanced classed problem for classification. Several approaches were tested to correct this issue. Both SMOTE and Under Sampling were evaluated. We finally chose the latter as it performed faster results. SMOTE(Synthetic Minority Over Sampling): where synthetics data that were generated from the minority class, then adding back to the original data to create a balanced data set. Under sampling: approach to balance the data by eliminating the majority class. Both approaches were used to balance the data then trained with logistic regression. The accuracy did not vary much, but under sampling was much faster to train.

- Evaluate the two classifiers predictions:

Decision Tree with depth of 2 was also trained with the same features as logistic model. RSME was calculated to evaluate the two models.The result showed Logistic has slightly better prediction rate.

Our first approach for choosing the a constant price for base price is value iterations. Base price = 0.111 proved to give the best CTR. The linear price strategy was then tested along with constant price strategy and random price strategy though our own bidding protocol. This base price resulted in 100 clicks and CTR equals to 0.06 percent CTR.The formula for our linear bidding strategy is:

$$BidPrice = (baseBidPrice) * 3 * pCTR$$

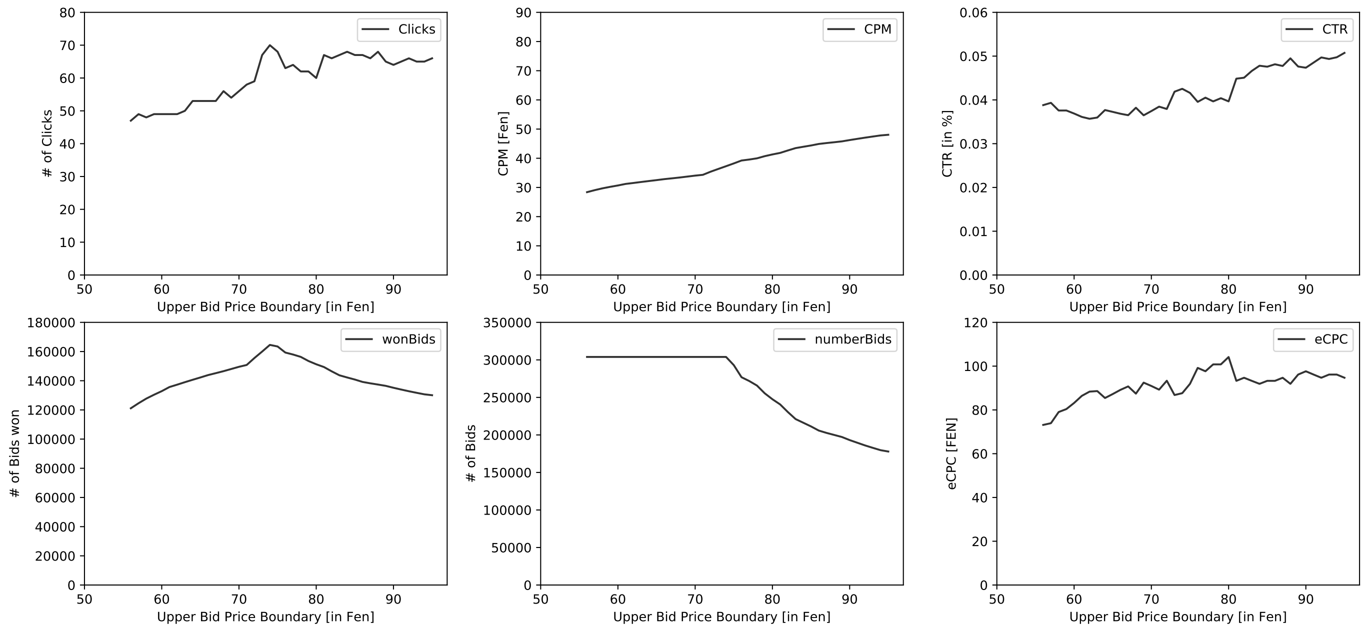

Figure 5: Summary statistics - Random bidding strategy with narrow boundaries

Table 1: Evaluation for Decision Tree and Logistic Classifiers for linear bidding strategy

Models	RMSE	Correct Classified	Misclassified
Logistic Regression	0.51	79897	224028
Decision Tree	0.49	73985	229940

The constant value of 3 was found through interpolation setting the clicks maximization as the target value to maximize. This formula is applied only for those bids that our classifiers expect to be clicks. When the click is not expected a floor price is bid.

- Optimise linear bidding strategy

Our second approach is to optimise not just base bid price but the other parameters too, which are: predicted CTR, base bid and premium. The weighted combination of these three parameters yields the bid price for every impression that is the objective of our algorithm.

Such an optimization algorithm was done under a given budget constraint. We followed an iterative process to find the most efficient combination of the three parameters. At each iteration each we kept two parameters constant, as independent variables, and played around with the third to find which version of it gave the highest number of clicks. We repeated the process for each of the three parameters. 1. The CTR per impression was predicted using three different models: Linear Regression, Random Forest and Decision Tree. Only the last three were tested for the algorithm. We didnﬁt use the CTR predicted by the linear regression because it proved to be the weakest in both accuracy and ROC AUC classification scores. The best performance, recorded at 115 clicks was reached by Decision Tree model. While the Random Forest only reached 103 clicks, even once optimized for premium and bid price.

Table 2: Summary statistics - Linear Bidding using Random Forest for CTR prediction

Key Metrics	Performance
# of Bids	285,501
# Bids won	95,273
# of Clicks	103
Total budget consumed (in CN¥)	6,249.899
CTR (in %)	0.1081%
Average CPM (in CN¥)	65.60
Average CPC (in CN¥)	60.68

We are not surprised by such a performance. It is in line with our accuracy scores for the three deep learning models.

2. We set the base bid at about 82 Yen, its value does not change across the three models. That figure comes from the average pay price per single impression from the validation dataset. It is adjusted for an error factor that is summarized in two times its standard deviation. However, in our strategy the base bid is not a floor bid. Indeed, the floor bid is arbitrary set to be very low and is targeting those impressions which click probability is forecast below 0.5. The optimal floor bid value is set at 19 Yen for the Random Forest, while it is 21 Yen for the Decision Tree. These values was found with interpolation. First the upper and lower value boundaries were found, with trial and error, to both yield the immediately sub-optimal clicks result. Second the value within the boundaries was set to be the optimal floor bid for that model. The floor bid

Table 3: Summary statistics - Linear Bidding using Decision Tree for CTR prediction

Key Metrics	Performance
# of Bids	300,327
# Bids won	111,497
# of Clicks	115
Total budget consumed (in CN¥)	6,249.844
CTR (in %)	0.1031%
Average CPM (in CN¥)	56.05
Average CPC (in CN¥)	54.34

value, used in unlikely clicks, is less than one third of the base bid employed for the likely clicks. This empirical evidence supports our choice not to use the base bid for unlikely clicks. Moreover, our floor bid performs better than just deciding not to bid when the probability of clicks is >0.5. Indeed, this choice reduces the probability of missing clicks that is a significant error in our models. Such an error is not driven by a lower than the market bid, yet by the pCTR estimation error. We can say it since more than 500 Yen are left unspent if we set the floor price to zero. In this case only 108 clicks are achieved using the optimal bid price and premium. Even when we increase the bid price to always bid the maximum pay price of the dataset, the clicks increase at 110 before drastically decreasing. The price-to-click trade-off analyzed above supports our choice to basically set two base bids one for the likely and one other, much lower, for the unlikely clicks.

3. The premium is a third factor we introduced in our linear strategy to take into account for the click probability. Indeed, it increases the bid price, in linear proportion with the pCTR only when the probability of a click is estimated to be above 0.5. It works as a trigger for the linear bidding. Since any bidding made for pCTR greater than 0.5 is not linear rather constant and greater than zero. We choose pCTR greater than 0.5 as a trigger for our constant bidding because, according to our prediction models, those predictions are more likely to be clicked. Considering that the proportion of non-clicks over clicks is extremely high, the risk of running out of budget is higher than that of missing a click, especially when its probability is predicted below 0.5. Ultimately, setting pCTR >0.5 as the minimum condition to bid takes into account the budget constraints. Empirical evidence from values interpolation shows that, if we want to be more confident with pCTR, to place a linear bid, we are missing a lot of clicks and the number of clicks is suboptimal for any percentage greater that 0.5. interpolation represents a great improvement in terms of clicks from the constant and random bidding. We believe this is the case because no money are wasted if our prediction models donfit believe there will be a click. On the other hand, when the likelihood of click is believed to be above 0.5, the premium is triggered. It drastically increases the bid price proportionally to the probability of click. In our strategy the premium is at the foundation of the linear relationship between bid price and predicted likelihood of click. The optimal premium value differed for each of the three models, yet yielding to the same result of 103 clicks. Its optimal values were found, for each model, with trial and error. First the upper and lower boundaries were set to both yield the suboptimal 102 clicks result. The value within them was found to be the optimal premium for that model. In particular: 3*pCTR was the premium for the Random Forrest model. 100*pCTR was the premium for the and Decision Tree models.

In conclusion we went beyond the initial linear bidding requirements introducing the premium element. This works as a trigger to make the algorithm linearly proportional to the probability of click only when this is above 0.5, otherwise the algorithm performs a constant, very low, bid.

4.2.2 Non-Linear Bidding Strategy. We evaluated the results from random, constant and linear bidding. In the linear bidding strategy, pCTR is the probability of getting a click that was generated by the our Classifiers.

The linear formula generates lower bid price for impressions with low probability of a click, however the bid is not able to grow fast enough when the probability increases. To address this problem, and win the highly likely clicks, we had to start bidding high also for those uncertain impressions. With our budget constrained, this led to spending the entire budget quickly or losing many bids if we lowered either the base bid price or the slope gradient. In order to improve our bidding strategy, we decided to improve our prediction for clicks. The reason for this is bidding very low for any impressions that have low probability of a click. Therefore the accuracy of click prediction is important as this will act as the key factor in deciding to bid or not.

- Combining our models for CTR Estimation:

To address the problems associated to linearity, our group implemented and tested four different non-linear models: CTR Estimation with Support Vector Machine (SVM), CTR Estimation with Ensemble Learning, ORTB and Exponential Bidding with Voting Classifier.

Looking at the output of our individual non linear models, the exponential one was the best performing with 136 clicks. However, its main drawback are extremely high prices the most likely clicks. This would make it too risky in the bidding competition with the other teams. Therefore, we decided to adopt a more conservative approach and combine the SVM with Ensemble Learning. The table 4 summarizes our individual strategies and the best one. The latter is indeed the best performing for three metrics over four. Indeed only in number of impressions is not the best. One of the ways to improve classifier predictions is by implementing Ensemble Learning. Ensemble Learning is made up of many classifiers, these are called weak learners. Each of these classifiers is trained and tested separately. Once we have the best version of each classifier, these models are inputs used to predict the validation data. A voting classifier that is called a strong learner is implemented; the strong learner aggregates the predictions from all weak learners. The Voting classifier will output the class that has the majority of votes from all weak learners. The Ensemble Learning we first implemented was combined these four classifiers: Logistic, Random Forest, Decision Tree and AdaBoost. The predictions from the first Ensemble Learning model gave lower false positive rate.

For our best bidding strategy, we combined the Ensemble Learning and Support Vector Machine to optimise the prediction for click. We did this by re-implementing the Ensemble Learning and

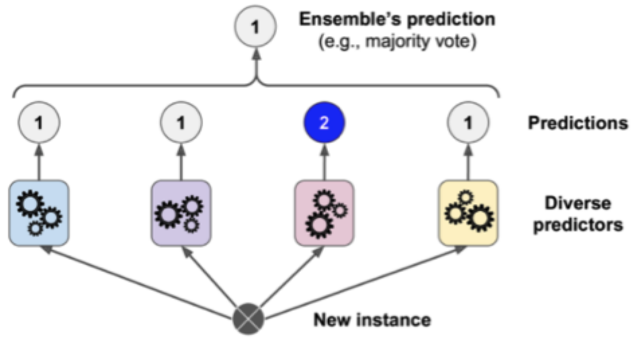

Figure 6: Ensemble learning process

Table 4: Evaluation for all bidding strategies

Models	Imps	Clicks	CTR	CPC
ORTB Bidding Strategy with Decision Tree	303,925	103	0.066	38.53
Linear Bidding Strategy	300,327	115	0.103	54.34
Constant Bidding with SVM	108,555	115	0.106	53.68
Constant Bidding with Ensemble Learning	113,669	132	0.116	46.58
Exponential Bidding with Voting Classifier	303,935	136	0.113	45.27
Best Bidding Strategy	121,914	140	0.115	44.62

incorporating the SVM model as a weak learner along with other weak learners. The second Ensemble Learning with the addition of the SVM as another weak learner did not result in a higher correct click prediction but it reduced the false positive rate. For the budget constraint problem, we designed our bidding price to be based on the probability of an impression that may get a click falling into three groups:

- Impression that have probability of a click less than 0.3, then no bid

- Impression that have probability of a click from 0.35 — 0.5, then bid price will be:

$$MeanPayprice + 0.6 * StandardDeviation$$

- Impressions that have probability of a click from 0.5 to 0.75, then bid price will be:

$$MaximumPayprice + 0.8 * StandardDeviation$$

- Impressions that have probability of a click above 0.75, then the bid price will be:

$$MaximumPayprice + 1.35 * StandardDeviation$$

4.3 Summary of Results

From our constant price bidding strategy, the most effective method to choose a value for the constant was by value iteration. The results were then compared to identifying the value, which generated the highest CTR. Our best bidding strategy we combined the value iteration method to generate the right value to multiply with the standard deviation. This would result in the highest CTR. Standard deviation is being used as an adjustable value that depends on the click probability for the impression, while the base price is kept constant. The base price for impression with click probability from 0.3-0.5 percent, is the mean of the pay price and base price for impression with click probability higher 0.5 percent, is the max of the pay price. With this pricing strategy, we were able to control budget spending better. This was achieved by: - Improved predictions from Ensemble Learning, which reduced false positive rate. This allowed us to omit bids for impressions that have low click probability. - Reduced over bidding by having different bid prices for impressions that have different likelihood of a click - Iteration to find best values for pricing strategy also reduce overbidding The result for our best bidding strategy for test data is 136 clicks from 119883 impressions, CTR equals to 0.0011, CPC equals to 45.16

5 CONCLUSIONS

In conclusion, we combined the best prediction for pCTR with a non-linear bidding price function to express at best the compromise between budget spending and conversion rate. Looking at the non-linear models, there is no one that is performing better for every metric. Indeed, even if Ensemble Learning with Support Vector Machine reached the highest level of clicks, it seems to be very conservative when it comes to placing bids as it has more than half the number of impression than the second best performing strategy. Comparing the best two strategies we see that the second best strategy is placing twice as many bids the best one, yet keeping a very similar CTR and CPC. This brings to evidence the main drawback of our best strategy: missing the cheap clicks. This could be due to tight confidence intervals. Basically placing bids only when the probability of click is extremely high.One attempt to capture the 'cheap' clicked impressions was the ORTB strategy, which indeed resulted in a considerably lower eCPC, but also in overall lower clicks and CTR. Further work needs to be done to explore this issue and eventually build a model that is less conservative in terms of impressions.

6 SOURCE CODE REFERENCE

The entire source code for this project, including the fully detailed data exploration, can be accessed via the following link to the corresponding GitHub repository: GOMPGW02 - Team 10 - DSP RTB Algorithm

REFERENCES

[1] Google Inc. [n. d.]. The Arrival of Real-Time Bidding and What it Means for Media Buyers. ([n. d.]). Retrieved April 06, 2017 from https://static.googleusercontent.com/media/www.google.com/en//doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf

[2] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. 2007. Bid Optimizing and Inventory Scoring in Targeted Online Advertising. *In WWW, pages 521fi530* (2007).

[3] Kan Ren, Jun Wang, Weinan Zhan, Ke Chang, Yifei Rong, and Yong Yu. 2018. Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* (2018).

[4] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. *Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting* (2nd ed.). Now Publishers Inc., Breda, Netherlands.

[5] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time Bidding for Online Advertising: Measurement and Analysis. *ADKDD fi13* (2013).
[6] Weinan Zhang, Jun Wang, and Shuai Yuan. 2014. Optimal Real-Time Bidding for Display Advertising. *KDDfi14* (2014).