COMPGW02: Web Economics Coursework Project

Individual Report

Lynray Barends
SN: 17032368
ucakljm@ucl.ac.uk

ABSTRACT

In this report, I explain the approach used to complete the individual components of a group project relating to real-time bidding. The report comprises of two main tasks: (1) providing a brief data exploration of a subset of the iPinYou data, and (2) implement a non-linear bidding strategy to maximise the number of clicks. My best model is an ensemble of an optimal non-linear bidding strategy (ORTB1) and a linear bidding strategy using gradient boosting decision trees (GBDT).

Keywords

real-time advertising, machine learning, bidding, CTR prediction, web economics

1. INTRODUCTION

The code [1] for this project can be found on GitHub.

Real time bidding is currently a popular trend in the online advertising industry, as it allows advertisers to bid in real-time on ad impressions. This is beneficial to them, as it allows them to buy slots based on a set of contextual (e.g. time and day, domain, location, etc.) and behavioural (e.g. browsing, purchasing, search history, etc.) features [7].

The work presented in this paper serves as an individual component of a bigger group project, focusing on predicting optimal bid prices in order to maximise a KPI. In this instance the KPI is the number of clicks received given a restricted budget of 6250 CNY. Each group member was required to individually :

- provide a brief data exploration analysis of the dataset
- implement an individual bidding strategy, that could possibly be used in the combined group bidding strategy.

The strategy chosen is taken from [7], where it not only takes the impression into account but also certain practical constraints, such as, the bid landscape, budget, campaign lifetime.

2. RELATED WORK

In [8], the authors provide a basic statistical analysis of the full iPinYou dataset (note that in this project we only use a smaller subset - due to computational complexity and resource limitations). They also provide some benchmarking algorithms for CTR-prediction (logistic regression & gradient boosted decision trees) and bid optimisation (constant, random, linear, bidding below max eCPC).

The authors of [3], introduce a decision tree + logistic regression model to predict clicks on Facebook adverts. They highlight the importance of good feature selection and describe one way to sensibly down-sample and recalibrate a high dimensional dataset into one that is less computationally intensive to perform inference over.

Other papers focus on the problem of bid-landscape forecasting, which involves predicting the market price distribution of a give impression. Specifically [2] forecasts the bid for each sample using GBDT regression and then fit a GMM to generate a campaign level bid distribution.

Non-linear bid optimisation strategies are described in papers such as [6], [7]. More specifically, in [7], the authors describe a strategy which takes into account the budget and the campaign lifetime, wanting to bid on more impressions instead of a subset of high valued ones. This strategy is cost effective as it bids on lower valued impressions, meaning that they save money and the chances of winning are higher. In [6], the authors challenge the idea of using Value based bidding, as they suggest that this strategy favours users with an already high action rate (e.g. click, or conversion). Instead they suggest a lift-based bidding strategy that is based on the action rate lift i.e. focusing on users that would be more influenced by the ad specifically.

The authors of [1], [5] formulate the real-time bidding problem as a Markov Decision Process (MDP) and introduced model-based and model-free (respectively) reinforcement learning algorithms to develop bidding strategies that are able to additionally consider budget constraints.

The work of [4] is an extension to [1] and formulates the real-time bidding problem with competition as a multi-agent reinforcement learning problem, which they using the actor-critic framework to solve.

3. ASSIGNMENT PROBLEMS

3.1 Data Exploration

As previously stated, we were provided with a subset of the iPinYou dataset, split into training, validation and test datasets within separate CSVs. Our initial task was to individually explore the datasets to extract relevant information that could possibly help us in the later tasks. I approached this question by first assessing the training and validation datasets, and then move on to briefly analysing user feedback and certain bidding patterns.

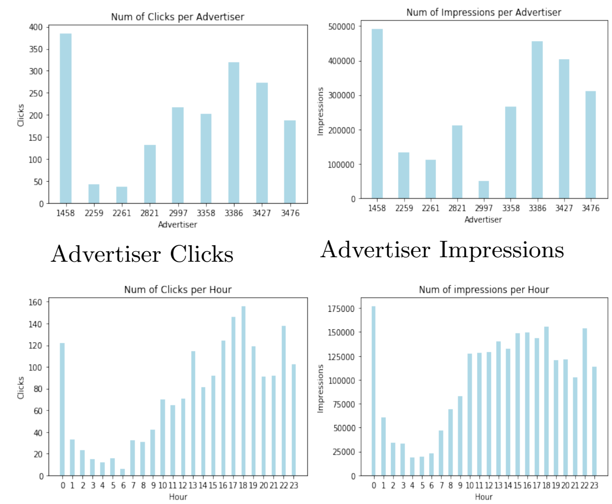
[1] `https://github.com/uclwe/rtb`

Table 1: Dataset Summary

Dataset	Impressions	Clicks	CTR	Cost	CPM	CPC	AvgPrice	AvgBidPrice
Training	2430981	1793	0.00738	189984.608	78.15	105	78.15	272.96
Validation	303925	202	0.000665	23777.27	78.23	117	78.23	273.05

Table 2: Dataset Columns

Field	Distinct Values
click	2
weekday	7
hour	24
bidid	2430981
userid	2342677
useragent	38
IP	503975
region	35
city	370
adexchange	5
domain	23013
url	763961
urlid	1
slotid	52283
slotwidth	21
slotheight	14
slotvisibility	11
slotformat	4
slotprice	286
creative	131
bidprice	8
payprice	301
keypage	19
advertiser	9
usertag	744036



Advertiser Clicks   Advertiser Impressions

Clicks per hour   Impressions per hour

Figure 1: Clicks and Impressions per hour and advertiser from the training dataset

3.1.1 Basic statistics

A summary of the training and validation set can be seen in Table 1. It should be noted that the dataset is quite unbalanced, having only 0.0737% of the clicks in the training data, and even less in the validation set. The validation set seems to be representative of the training data, as similar metrics have been achieved.

A breakdown of the columns and their unique values can be seen in Table 2. There are certain features that could be excluded when training the data, but most obviously the *bidid* (as it is unique to all impressions), and the *urlid* (as it is the same for all impressions).

Both the *slotid* and *domainid* could be useful when training, however, there were too many unique values, so it was decided by our group members to only keep the ones with a frequency over 5000 and set the rest to *null*.

Lastly, *usertags* seem to have 744036 unique values, however, these are in fact just lists of different *usertag* combinations. There are in fact only 69 unique tags (including *null*).

3.1.2 User Feedback

In the dataset there are 9 campaigns. Advertiser **1458**, and **3386** have the most number of impressions, as well as, a high number of clicks (as it can be seen in Figure 1). Advertiser **2997** may have the least number of impressions, but has the fourth highest number of clicks and the best click-through rate (CTR).

Figure 1 strongly suggests that time of day has an impact on the number of clicks and impressions. There is a noticeable drop in the number of clicks from *1a.m.* until about *6a.m.*, which makes sense, as that is the time that most people are asleep. After *6a.m.* it gradually increases reaching a local maximum between *16* and *18*.

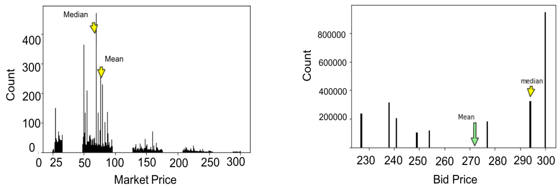
Most of the impressions came from users using *Windows* as an operating system (95%), and over 60% used Internet Explorer as their chosen browser.

According to the training data, there are only 78620 users (by *userid*) that have multiple bid impressions, and out of these users only 100 of them had a history of clicking on an advert. Furthermore, only 3 have clicked on more than 1 advert.

3.1.3 Bidding

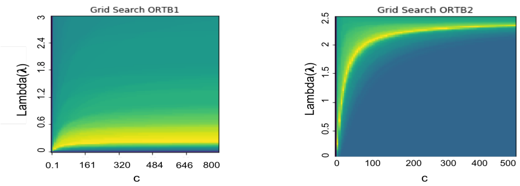
As it may be seen in Figure 2, the market price ranges from 0 to 300 in the training dataset, however ,it is skewed to the right. Majority of the market prices are between 50 and 100. It also seems as though there are about 5 different groups of pricing, possibly 5 normally distributed groups. The bid prices in the training data range between 227 and 300. Using Figure 2 and Table 2, you may notice that there are only 8 distinct bid prices in the training data. These bids are fairly high compared to the market prices of majority of the ads.

Given the budget we were allocated, we would only be able to participate in 3% of the the auctions in the training set, and 26% of the auctions in the validation set, as we would win every impression, resulting in us running out of budget fairly quickly.



Market Price   Bid Price

Figure 2: Distribution of Market price and Bid Price in the training dataset



ORTB1   ORTB2

Figure 3: Grid search for optimal c and λ paramaters

3.2 Individual Best Bidding Strategy

For our second task, we were asked to find a more optimal (non-linear) strategy. Our group's best linear bidding strategy received 163 clicks and is discussed further in our group report.

My approach for this task was to briefly experiment with different CTR predictors. Following that, I implemented a variation of the ORTB strategy found in [7]. Lastly, I experiment with ensembling different models in order to find the optimal performance. In the end, my best model was an ensemble of the ORTB Bidding model (**XGB75-ORTB2**) using an GBDT CTR predictor, and my best linear bidding model (**LinXGB120**).

3.2.1 Variation of CTR predictors

In section 3.1.1, I stated that we only took the *domainid* of those that had a frequency greater than 5000, and set the rest to *null*. I chose to explore whether separating these *null* values from impressions that actually were set to *null* made a any difference. This gave me a dataset with one more column. I retrained a logistic regression model, and a gradient boosting tree model on this dataset, and received an improved performance from the one trained within our group's linear bidding strategy. I then performed a grid search to find more optimal parameters for our original dataset on GBDT. My non-linear bidding strategy made use of these new CTR predictors.

Table 3: Parameter Settings for CTR predictors

Model	*n_feat*	*n_est*	*max_depth*	η	*roc_auc*
LinLogReg	824	-	-	0.01	0.845
LinXGB75	824	75	10	0.3	0.899
LinXGB120	823	120	10	0.1	0.8955

3.2.2 ORTB

As previously stated, I implemented a non-linear bidding strategy by referring to the work done in [7]. This strategy encourages allocating more budget to bidding on lower-valued impressions. This is beneficial to the advertiser, as it maximises the opportunity of winning the bid, and it is more cost effective. They propose a functional optimisation framework which results in a non-linear bidding strategy. The functional optimisation problem can be stated as:

$$\begin{aligned} b()_{ORTB} &= \underset{b()}{\operatorname{argmax}}\ N_T \int_\theta \theta w(b(\theta)) p_\theta(\theta) d\theta \\ \textit{subject to } N_T &\int_\theta b(\theta) w(b(\theta)) p_\theta(\theta) d\theta \leq B \end{aligned} \tag{1}$$

where b refers to the bidding function, w refers to the winning probability, θ refers to the predicted KPI, $p_\theta(\theta)$, refers to the probability density function the KPI, B refers to the Budget, and N_T refers to estimated number of bid requests during the lifetime T The winning rate function can be generically formulated as :

$$w(b(\theta)) = \frac{b^\alpha(\theta)}{c^\alpha + b^\alpha(\theta)} \tag{2}$$

where $\alpha = 1, 2$, and $\theta =$ predicted KPI (predicted CTR).

The derivative of the winning functions, with respect to the bid, are inserted into the Lagrangian of the objective function in order to derive 2 different bidding strategies:

$$b_{ORTB1}(\theta) = \sqrt{\frac{c}{\lambda}(\theta) + c^2} - c \tag{3}$$

$$b_{ORTB2}(\theta) = c\left[\left(\frac{\theta + \sqrt{c^2\lambda^2 + \theta^2}}{c\lambda}\right)^{\frac{1}{3}} - \left(\frac{c\lambda}{\theta + \sqrt{c^2\lambda^2 + \theta^2}}\right)^{\frac{1}{3}}\right] \tag{4}$$

ORTB1 allocates more budget on low cost cases, where decreasing high bid does not significantly (if at all) decrease the winning probability. However, increasing a low bid could drastically increase the winning probability. **ORTB2**, compared to **ORTB1** only starts increasing rapidly when bid prices are larger, as it is more suited to competitive targets or publishers having a higher reserve price [7]

These bidding strategies take our predicted CTR(KPI) as a parameter, as well as c and λ, which can both be tuned.

I found that the CTR values produced by my CTR predictors were quite small, especially the GBDT models, hence, I chose to multiply them by a certain factor of 10 (CTR_X). I explored with different CTR_X values for each predictor. This may have been the incorrect approach, however, it served the purpose of being able to appropriately differentiate between high and low valued impressions.

I implemented both **ORTB1** and **ORTB2** using the logistic regression model (**LinLogReg**), as well as, the gradient boosting model (**LinXGB75**) as CTR predictors.

I explored the best c and λ parameters using a grid search (an example can be seen in Figure 3). My c and λ may vary from the expected values due to the CTR_X factors. The best ORTB model used **LinXGB75** and **ORTB1**, producing 167 clicks on the validation set.

Figure 4 shows the winning and bidding function of **ORTB1** and **ORTB2**.

As expected, in **ORTB1** when the bid price is low, a small increase in the bid price can result in a large increase in the winning probability. However, in **ORTB2**, this large increase happens at a higher bid price.

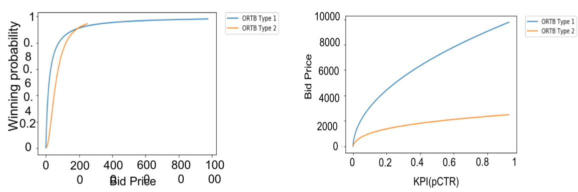
Table 4: Results Summary

Model	Parameters	Clicks	CTR	aCPM	eCPC	Spend	Lifetime	Win%
LogORTB1	C =727.27 $\lambda = 0.019445$ $CTR_X = 10$	148	0.001174	20.42	41.93	99.29%	100%	41.47%
LogORTB2	C = 15.15 $\lambda = 0.0252$ $CTR_X = 100$	108	0.000684	19.59	55.19	95.25%	100%	51.91 %
XGB75-ORTB1	C=194.02 $\lambda = 0.182$ $CTR_X = 100000$	167	0.001358	20.49	37.3	99.65%	100%	40.44%
XGB75-ORTB2	$C = 606.15$ $\lambda = 1.89$ $CTR_X = 100000$	154	0.000943	20.6	40.58	100.00%	99.8%	53.49%
XGB75-ORTB2 +LinXGB120	$XGB75 - ORTB2\% = 45.45\%$ $LinXGB120\% = 52.53\%$	169	0.001344	20.21	36.35	98.29%	100%	41.37%

Table 5: Number of clicks and CTRs on a variation of budgets

Model	Budget	Budget/2	Budget/4	Budget/8	Budget/16	Budget/32
LogORTB1	148 (0.0012)	70 (0.0011)	37 (0.0012)	16 (0.001)	12 (0.0015)	7 (0.0017)
LogORTB2	108 (0.0007)	49 (0.0006)	26 (0.0006)	12 (0.0006)	10 (0.001)	7 (0.00135)
XGB75-ORTB1	167 (0.00135)	81 (0.00131)	44 (0.00142)	18 (0.00116)	12 (0.00156)	8 (0.0021)
XGB75-ORTB2	153 (0.0009)	75 (0.0009)	40 (0.001)	17 (0.0008)	13 (0.00128)	8 (0.00157)
XGB75-ORTB2 +LinXGB120	169 (0.00134)	83 (0.0013)	44 (0.0014)	19 (0.0012)	13 (0.0016)	8 (0.0020)

Likewise, in **ORTB1** when the KPI is low, a small increase in the KPI (predicted number of clicks), can result in a large increase in the bid price.



Winning Function   Bidding Function

Figure 4: Winning and Bidding function of my model

3.2.3 *Ensemble*

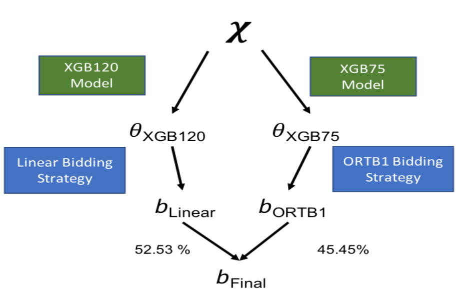


Figure 5: Ensemble of XGB75-ORTB1 and LinXGB120

Once I had found my best performing model, I tested ensembles of the various models developed. Ensembles of multiple predictors is a common approach in recent machine learning solutions, since one model's bias can be offset by another's [3].

Taking my best performing models, I executed a grid search for the best composition of each pairing. I, however, did not want to limit the portions to only equating to 1. In Figure 6, you can see the grid search performed for the **XGB75-ORTB1** and **LinXGB120** models. According to the results, the best values usually equate to 1, but sometimes it may not. The best model was composed of 45.45 % of **XGB75-ORTB1** and 52.53% of **LinXGB120**.

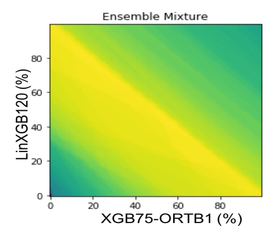


Figure 6: Grid search for best composition of LinXGB120 and XGB75-ORTB1

3.2.4 *Evaluation*

A summary of the results can be found in Table 4. The ORTB models using GBDT CTR predictors produced more clicks than the ones using Logistic Regression, resulting in 19 more clicks using the **ORTB1** strategy and 46 more clicks using **ORTB2**. The estimated cost-per-click (eCPC) is also reduced in both instances. While **ORTB2** may have had a higher win ratio, it produced less clicks than **ORTB1**. This means that it is winning impressions that result in no clicks.

As you can see in Table 4, these models usually last the whole duration of the dataset without using all of their budget. However, they do not spend the full budget. This could be due to certain parameter tuning. What it does say is that even the best performing models still have money left, so there is room for improvement.

The ensemble of LinXGB120 and XGB75-ORTB1 produces the highest number of clicks, and has the lowest eCPC.

Lastly, I had a look at the number of clicks and CTR with various budgets (Table 5). The ensemble consistently provides the highest number of clicks. The ORTB models with the logistic regression CTR predictor are the worst performing.

4. CONCLUSION AND FUTURE WORK

In this report I provided a brief data exploration of the iPinYou dataset in order to show understanding of the data.

Following that, I implemented a non-linear bidding strategy based on work done in [7]. Finally, I evaluated my results and assessed the outcome that different budgets may have on my models. My ORTB model produced a high number of clicks, however, it performed best when combined with another linear GBDT model.

Given more time I would have explored trimming down the features significantly, as about half of the features we used were insignificant. I would also work on experimenting further with ensembles and better parameter tuning of my models.

As a group, we were interested in looking further into using Multi-agent reinforcement methods.

I thoroughly enjoyed working with my team members, as we all worked well together. We all contributed to the group work equally.

ACKNOWLEDGMENTS

Thanks to Jun and the TAs!

5. REFERENCES

[1] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 661–670. ACM, 2017.

[2] Y. Cui, R. Zhang, W. Li, and J. Mao. Bid landscape forecasting in online ad exchange marketplace. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2011.

[3] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.

[4] J. Jin, C. Song, H. Li, K. Gai, J. Wang, and W. Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. *arXiv preprint arXiv:1802.09756*, 2018.

[5] D. Wu, X. Chen, X. Yang, H. Wang, Q. Tan, X. Zhang, and K. Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. *arXiv preprint arXiv:1802.08365*, 2018.

[6] J. Xu, X. Shao, J. Ma, K.-c. Lee, H. Qi, and Q. Lu. Lift-based bidding in ad selection. In *AAAI*, pages 651–657, 2016.

[7] W. Zhang, S. Yuan, and J. Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1077–1086. ACM, 2014.

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