COMPGW02 Web Economics Group Report (Group 05)

Min Ying Ong
University College London
School of Computer Science
min.ong.16@ucl.ac.uk

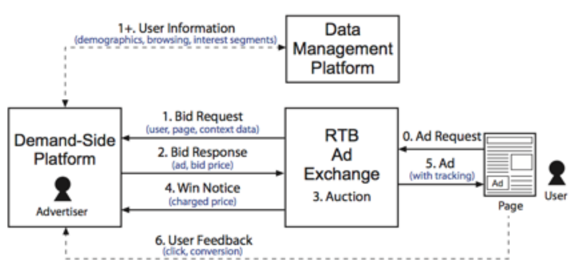
Chun Siong Poh
University College London
School of Computer Science
chun.poh.16@ucl.ac.uk

Kah Siong Tan
University College London
School of Computer Science
kah.tan.16@ucl.ac.uk

1. INTRODUCTION

Online advertising allows advertisers to bid and place their advertisements onto the web pages that are being visited by users. The process of user landing on a web page to the advertiser bidding and successfully displaying their advertisements to the users takes place within a fraction of a second. Such real-time bidding based advertisements requires bidding strategies that would bring in the maximum utility over costs. Figure 1 [12] illustrates the interaction described above. This report proposes a bidding strategy consisting of two components to allocate a bid for each impression:

Figure 1: Real-time advertisements bidding process [12]



- **CTR estimation component**: *based on logistic regression, gradient boosted trees, convolutional neural networks (CNN) and factorisation machines.*

- **Bid price estimation strategy component**: *based on variations of the linear bid model.*

The CTR estimation component attempts to learn the chances of a click based on features collected from bid requests of the online advertising system, while the bid price estimation strategy component formulates an approach to offer the most suitable bid for an impression. The models were implemented and evaluated against other baseline bidding strategies. Lastly, an ensemble model was adopted for the CTR estimation component to improve the prediction accuracy by combining the result of the CTR estimation models.

The results showed that Gradient Boosted Tree outperformed both CNN and Factorisation Machines in terms of Area Under Curve (AUC) of ROC.The top two models were ensembled to produce a better performing AUC for click prediction. The proposed bid estimation models did not outperform the linear bidding strategy when paired with the ensemble click prediction model. As such the final complete bidding strategy for the group consisted of the Gradient Boosted Tree + CNN Click Prediction ensemble and the Linear Bid model for bid estimation.

The source codes of this assignment are available in [7].

2. LITERATURE REVIEW

In Real-Time Bidding ecosystem, an advertiser or demand-side platform (DSP) aims to buy the best-matched impressions that suits their objective via real-time auction. To determine the best-matched impressions, models analyse the contextual and historical features for a given impression to determine the click through probability and estimate its worth before submitting a bid. This whole process completes within 10-100ms.

For online advertisement, prediction of Click-Through-Rate (CTR), or CTR estimation is a widely studied subject. There have been different models and variants developed over it. Some notable examples are Linear Regression (LR) [3], Recurrent Neural Networks (RNN) [4], Convolutional Neural Networks (CNN) [8], Boosted Trees [11] and Factorisation Machines (FM) [9]. The most popular technique would LR for its simplicity and efficiency. However neural networks tend to provide surprisingly good results despite their difficulty in interpretation. Boosted trees provide very clear interpretation compared to neural networks and can determine its own feature importance. Factorisation Machines performs well on sparse data and offers higher levels of feature interactions.

3. APPROACH AND RESULTS

3.1 Data Exploration

The data exploration of the datasets are detailed in the individual reports.

3.2 Evaluation Metrics

3.2.1 CTR estimation model

CTR estimation is essentially a binary classification problem, there are several ways to evaluate its performance. Some of the popular evaluation metrics are Accuracy, Precision, Recall, F1 and Area Under Curve (AUC) of Receiver Operating Characteristic (ROC). Typically, accuracy would be the main metric to use in a classification task but it does not serve well in our extremely imbalanced data sets, such as in our case where click=0 significantly outnumbers click=1 examples. Precision, recall and its associated F1 score serves well in classification problems but again do not give a good interpretation in extremely imbalanced datasets. AUC is not sensitive to class imbalances and as a single metric offers a good interpretation on the model's capability to classify binary classes independent of thresholds. An AUC score of 1.0 implies that all true positive are ranked before true negative, while an AUC score of 0.5 indicates a random ranking. Any score below 0.5 is not typical and should be taken that a bug is responsible for such reading. As such, AUC was chosen to be used as the main metric for CTR estimation model comparison. The validation set is used to measure this metric as it is held-out during training time.

3.2.2 Bid price estimation model

Performance of the bid price estimation model and the overall bid strategy was evaluated based on the following metrics, with the specified limited budget of 6250 CNY fen.

- **Click-Through-Rate (CTR)**: *Ratio of clicks compared to total number of impressions won*
- **Clicks**: *Number of clicked impressions won*
- **Spend**: *Total amount of money paid for all impressions won*
- **Average CPM**: *Cost per mille or cost per thousand impressions*
- **Average CPC**: *Average cost per click based on total amount of money paid for all impressions won*

The team introduced and implemented an importance weighting score on the different bid price estimation metrics. The rationale of this weighting system is to allow the system to adjust according to the situation that demands the appropriate metric. In this assignment, the team chose to optimise for CTR and number of clicks with a bias of 0.6 weighting towards CTR and 0.4 weighted towards number of clicks as shown in equation 1. All other metrics has zero weights. The rationale behind this is to maximise both CTR and number of clicks jointly.

$$Score = \frac{Num\ of\ clicks\ won}{Total\ num\ of\ gold\ clicks} \times 0.6 + CTR \times 0.4 \quad (1)$$

A high CTR could be obtained if very few impressions were bid on and won, however that is counter-intuitive as in the advertising market as you would also want to reach the maximum number of users. Similarly a high number of clicks can be obtained if there was no budget constraints, however the CTR would be very low and the budget high.

Also, due to the already constrained budget of 6250 CNY fen, we chose to maximise that budget usage instead of minimising it as in the case of an unconstrained budget.

In real-world scenarios, different advertisers may have different goals to achieve in advertising. The importance weighting score is flexible in that the weighting could be modified to maximise the utility of the advertiser based on their goals.

3.3 Bidding Strategies

3.3.1 Basic Bidding Strategies

3.3.1.1 Constant Bidding.

Constant bidding is a primitive bidding strategy that bids all impression using a predefined price. Using training dataset, an optimal bid price of 77 (in cost-per-mille (CPM) pricing model) is obtained using grid search that maximised the number of clicks. The budget of 6250 CNY fen is proportionally increased by 9 times as the training set is much bigger than validation set to set an equal ground.

3.3.1.2 Random Bidding.

Two random bidding strategies, Uniform Random and Gaussian Random are implemented. The Uniform Random model submits a random bid between 0 to 300 for every impression. The Gaussian Random model submits a random bid based on a Gaussian distribution with mean 80 and stdev 6. The mean is derived from training dataset average pay price for impression won.

3.3.1.3 Results.

Table 1 tabulates the scores for constant and random bidding models on the validation set using a budget of 6250 CNY fen.

Interestingly, the three models have similar performance. They obtain around 75 clicks. Constant bidding and Gaussian Random use a lower bid price of about 80. Due to the lower bid price, the two models won 23% more impression than uniform random which bids randomly between 0-300 (Average 150 per bid). The three models bid for all impression as they have no mechanism to select which impression to buy. This limitation quickly exhausts the budget limit of 6250 CNY fen. Uniform Random bidding exhausted the budget the earliest as its average bidding price is much higher than the other two models.

3.3.2 Linear Bidding Strategy

The linear bidding strategy is built on two components. The first is the CTR estimation to predict the probability of click for each impression in the test set. The second is optimal bidding to derive a bid price based on the probability of click and other bid related parameters. Logistic Regression (LR) is employed as the CTR estimation mechanism with a linear bidding mechanism. The LR model is represented as

$$h_\theta(x) = \frac{1}{1 + exp(-\theta^T x)} \quad (2)$$

where $h_\theta(x)$ can be interpreted as

$$h_\theta(x) = P(y = 1|x) \quad (3)$$

Where $P(y = 1|x)$ refers to the probability of the positive label, x refers to input and θ is a weight that would be trained by minimising the following loss function.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \Big(y^i log(h_\theta(x^i) + (1 - y^i) log(1 - h_\theta(x^i)) \Big) \quad (4)$$

where m represents the size of the data set. The implementation of the model was completed with the Scikit-Learn library [10]. Logistic Regression is a supervised learning classification model that predicts discrete variables, in this case a binary variable of an impression with a click or no click. The model is trained using a pruned training set with 20% click through rate and validated using the held out set provided. Using sklearn grid search with 5-fold cross validation, the final parameter used for the model are alpha 0.0005 and penalty L2 with 200 iterations.

Figure 3 shows the prediction histogram for click against the true value on the validation set. The clicks=0 histogram (left red graph) tapers off quickly, demonstrating positive result. However, clicks=1 histogram (right green graph) shows the distribution spreading across the x-axis. The ideal shape of click=1 (Green) would be a mirror image of click=0 (Red) where a 'U' shape histogram is formed when the 2 graph are overlapped together. The model achieved an AUC score of 0.822 on the validation set.

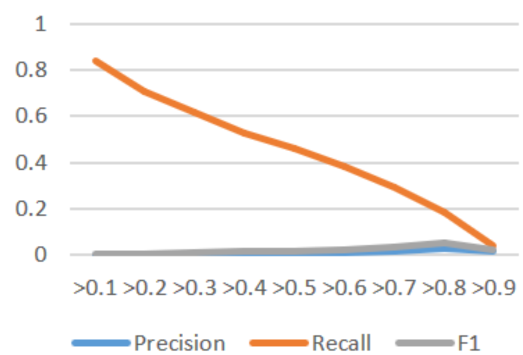


Figure 2: Effect of Probability Threshold on Click=1

Figure 2 shows the F1, Precision and Recall for clicks=1. As observed, a higher probability threshold generally increases the precision of the model marginally to predict a click through while a low probability increases the recall of click through at the expense of precision. Interestingly when probability is >0.9, precision actually dropped suggesting the model did not predict click=1 well. In real world application, the probability threshold could be adjusted for high recall which translates to buying more impressions and achieve more clicks. Or high precision which translates to buying impression with good click through and achieve good CTR.

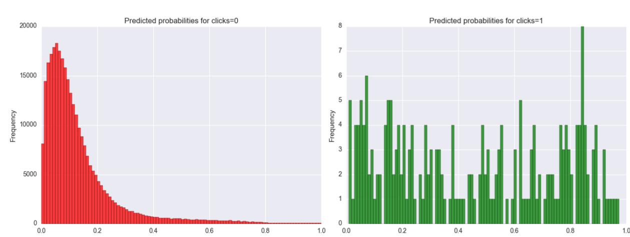


Figure 3: Logistic Regression Trees histogram of click probability

The second component of the linear bidding mechanism is characterised by the following equation.

$$bid = basebid \times \frac{pCTR}{avgCTR} \quad (5)$$

where $pCTR$ is the probability of a click as determined by the CTR estimation model above, and the $avgCTR$ determined by the averaged CTR in the training set. The $basebid$ is a parameter that would be optimised for the optimal set of CTR and clicks. A grid search is conducted and the optimal $basebid$ is found to be 110.

Table 1 shows the comparison of the metrics between Constant Bidding, Random (Uniform) Bidding, Random (Gaussian) Bidding and Linear Bidding. Linear bidding, backed by machine learning has outperformed constant and random bidding significantly. It obtained 163 clicks and has a CTR 2.6 times better the other basic models. This shows that leveraging on historical data and learning patterns in the data through machine learning is beneficial for advertiser to buy the best matched impressions.

Table 1: Basic Bidding Strategies Performance Metrics

	Constant Bidding	Uniform Random Bidding	Gaussian Random Bidding	Linear Bidding
Imp	161,635	131,476	160,418	124,301
Clicks	76	73	75	163
CTR	0.00047	0.00055	0.00047	0.00124
eCPC	82.24	85.61	83.33	37.46
CPM	38.67	47.54	38.96	46.55
Spend	6,250	6,250	6,250	6,107

3.3.3 *Combined Strategy*

The team developed a total of four CTR estimation models (i.e. Boosted Trees, Convolutional Neural Network (CNN), Factorization Machines and Logistic Regression) and 4 bid estimators. Details of each CTR estimation model is found in the individual reports. Table 2 shows the respective AUC for the CTR estimation model.

3.3.3.1 *Combined CTR Estimation Model.*

Table 2: AUC score for Individual CTR Estimation Model

CTR estimation model	AUC
Gradient Boosted Trees	0.878
Convolutional Neural Network	0.868
Factorization Machines	0.844
Logistics Regression	0.822

The gradient boosted trees is implemented using XGBoost [5]. Figure 4 shows the prediction histogram for clicks against the true value. From clicks=0 histogram (left red graph), it is observed that the model performed well in predicting click=0 as the graph quickly tapers off at higher probabilities. However, click=1 (right green graph) is not as good, with a small number of click=1 spreading across the x-axis. Fortunately, the bulk of clicks=1 still lies on the right of x-axis.

The CNN model was implemented using Keras and Tensorflow [6] [1]. Figure 5 shows the prediction histogram for clicks against the true value. From click=1 histogram (right green graph), it is observed that the model performed well in predicting click=1 as the majority of click=1 is predicted strongly at more than 0.8 probability, and more consistently than the gradient boosting model. However, click=0 (left red graph) is not as good as the gradient boosted trees

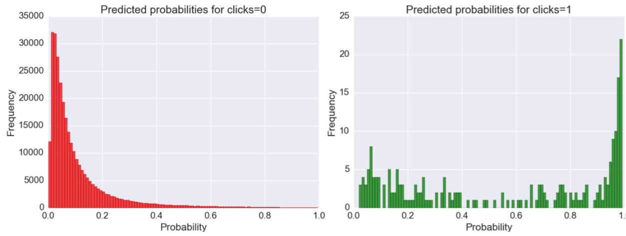


Figure 4: Gradient Boosted Trees histogram of click probabilities

model, with the predictions for click=0 not decaying as strongly as the gradient boosted trees model as it approaches probability of 1.

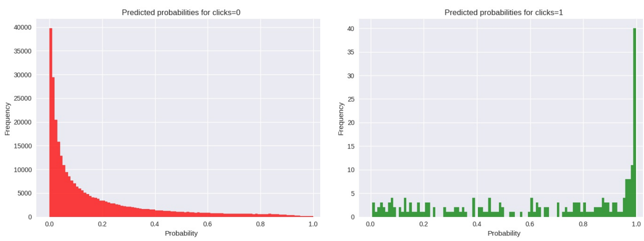


Figure 5: CNN histogram of click probabilities

The Factorisation Machine model was implemented using FastFM[2] and optimised with Scikit-Learn's GridSearch[10]. Figure 6 shows an analysis of predicted probabilities against the actual clicks and non-clicks for the Factorisation Machine model. Although the majority of clicks is predicted strongly above 0.9 probability, there were some clicks predicted across the rest of the spectrum, with it consolidated near probability 0.0. This could be the reason why the overall AUC performed below the other two models.

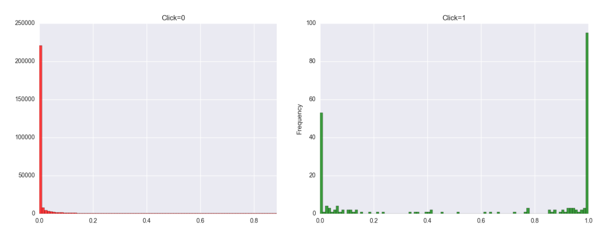


Figure 6: Factorisation Machine histogram of click probabilities

The Logistic Regression implementation was covered in section 3.3.2.

Using the four CTR estimation models, ensembles using a combination of models were implemented. Ensemble methods are commonly used to combined predictions of weak classifiers to obtain higher accuracy and is distinguished by two key family namely averaging and boosting. In averaging, the intuition is to build several estimator and average their output to reduce their variance and thus improve the performance.

Our best ensemble as measured by AUC is a weighted combination of the CNN and gradient boosted trees models listed in Table 2. The ensemble weighted the probabilities of the individual models (0.7 to gradient boosted trees model and 0.3 to CNN model) to produce a final prediction. The ensemble achieved an AUC of 0.884 (0.6% better than gradient boosted trees alone). Figure 7 shows the probabilities histogram of click. As visible from figure 7 the ensemble blends the two models to achieve a better result than one model alone. The click=0 (left red graph) decays quicker towards probability of 1.0 compared to CNN model, and the click=1 histogram (right green graph) is improved compared to gradient boosted trees model alone.

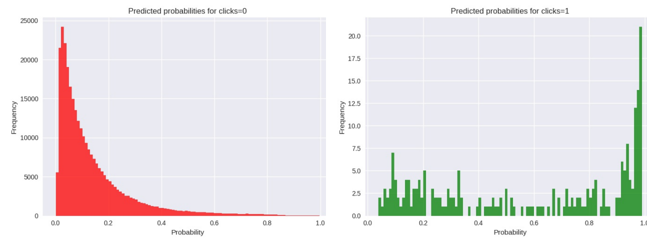


Figure 7: Ensemble model (Gradient Boosted Trees + CNN) histogram of click probabilities

3.3.3.2 Best Bid Estimation Model.

The rest of this section discuss the bid price estimation strategy component, with the limited budget of 6250 CNY fen. A simple linear bidding strategy was discussed in 3.3.2. Three other estimation models that work in hand with the CTR estimation models were conceived. Details of each strategy is found in the individual report.

The linear bid variation bid price estimation strategy is a variation of the linear bid model discussed in section 3.3.2. The linear bid obtained is used as an initial bid which is increased to meet the minimum slot price + 10%. This is to ensure minimum price is met. A bid prune threshold determines which bids are removed and set to -1 if the predicted click probability from the click prediction model is below the threshold. This is to improve CTR by not bidding on any ad slots where we are confident will not get a click.

The linear bid M confidence bid price estimation strategy only bid when the confidence of a click is higher than a M value. The price to bid is dependant on the confidence of a click.

The Sigmoid-Threshold bid price estimation strategy capitalise on higher chances of clicks, bidding from a base and scaling the bid along the probabilities of clicks in a non-linear manner. This means that lower confidence bid requests would not be bid, and there would be a cap on the highest bid per bid request to ensure high confidence bids are treated similarly and doesn't exhaust the budget quickly.

The bid price from each of the above strategies were computed using the output of the best ensemble of (Gradient Boosted Trees and CNN) for CTR estimation. The optimum parameters were found using grid search with average CTR on the full training set taken as 0.00075. Table 3 summarises the optimum parameters for all the models (monetary units in CPM). Table 4 shows the results on the validation set for each strategy, with the importance weighting score as defined in section 3.2.2.

Table 3: Bidding Strategies Optimum Parameters

	Linear Bidding	Linear Variation Bidding	Linear M Confidence Bidding	Sigmoid -Thresh Bidding
baseBid	0.4	0.25	200.00	n/a
predThresh	n/a	0.1	0.2	0.2
variableBid	n/a	n/a	100	n/a
minBid	n/a	n/a	n/a	250
bidRange	n/a	n/a	n/a	250
sigmoidDeg	n/a	n/a	n/a	-10

Table 4: Bidding Strategies Performance Metrics

	Linear Bidding	Linear Variation Bidding	Linear M Confidence Bidding	Sigmoid -Thresh Bidding
Imp	113,804	97,704	55,577	58,887
Clicks	183	174	170	173
CTR	0.0016	0.0018	0.0031	0.0029
eCPC	31.13	34.60	26.50	30.70
CPM	50.06	61.62	81.04	90.18
Spend	5,697	6,020	4,504	5,310
Score	0.3337	0.3174	0.3109	0.3163

The results in table 3 show that using the importance weighting score as defined in section 3.2.2, the baseline linear bidding strategy still outperforms the three alternative bidding strategies proposed when combined with the ensemble model for click prediction.

We can see that the linear bid model outperforms the other bid estimation models in terms of number of clicks achieved (81% of number of gold clicks compared to 75%-76% with the other three strategies). This was achieved at the expense of CTR which is at 0.16% for linear bid model, whereas the Linear M Confidence and Threshold Sigmoid models managed to double the CTR at 0.31% and 0.29% respectively. Overall, the CTR rates achieved is 2-4 times the 0.075% average in the original validation set for all the strategies proposed.

For the team's final bid strategy ensemble model, the Linear Bid model was chosen as the bid estimation model. This is because the importance weighting score defined in section 3.2.2 placed emphasis on number of clicks and CTR. In real-world scenarios, if an advertiser wants more emphasis placed on CTR then the Linear M Confidence and Threshold Sigmoid models may fare better.

3.4 Conclusion

This assignment proposed a bidding system consisting of two components to allocate a bid for each impression: CTR estimation and Bid price estimation strategy. Four CTR estimation models were experimented and evaluated against each other.

An ensemble of CTR estimation models was performed to achieve results that is better than any of the models alone. The results demonstrate that the ensemble of solutions is a very good way to capitalise on strengths of individual models, at the same time reduce the impact of individual weaknesses.

The proposed bid estimation models did not outperform the linear bid model, however in the real world the bid estimation model chosen would be dependent on the importance an individual advertiser places on particular metrics.

The final complete bidding strategy for the group consisted of the Gradient Boosted Tree + CNN Click Prediction ensemble and the Linear Bid model for bid estimation.

Future potential direction for this work could include having independent models for CTR estimation and bid price estimation for each advertiser (or clusters of advertisers with similar attributes).

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[2] Immanuel Bayer. 2016. fastFM: A Library for Factorization Machines. In *Journal of Machine Learning Research*, Vol. 17. 1–5.

[3] O. Chapelle, E. Manavoglu, and R. Rosales. 2014. Simple and scalable response prediction for display advertising. In *ACM Trans. Intell. Syst. Technol*, Vol. 5. 61–1–61–34.

[4] Qiao-Hong Chen, Shi-Min Yu, Zi-Xuan Guo, and Yu-Bo Jia. 2016. Estimating AdsâĂŹ Click through Rate with Recurrent Neural Network. In *ITM Web of Conferences*, Vol. 17. 1–5.

[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 785–794. DOI:http://dx.doi.org/10.1145/2939672.2939785

[6] François Chollet. 2015. Keras. https://github.com/fchollet/keras. (2015).

[7] Min Ying Ong Kah Siong Tan, Chun Siong Poh. 2017. Web Economics assignment Group 5. (Apr 2017). https://github.com/jax79sg/webecon2017

[8] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A Convolutional Click Prediction Model. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 1743–1746. DOI: http://dx.doi.org/10.1145/2806416.2806603

[9] Zhen Pan, Enhong Chen, Qi Liu, Tong Xu, Haiping Ma, and Hongjie Lin. 2016. Sparse Factorization Machines for Click-through Rate Prediction. In *2016 IEEE 16th International Conference on Data Mining*.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[11] Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. 17. *Using boosted trees for click-through rate prediction for sponsored search*. Technical Report. Yandex.

[12] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.