Adaptive targeting in online advertisement: models based on relative influence of factors

Andrey Pepelyshev[1], Yuri Staroselskiy[2], Anatoly Zhigljavsky[1,3], and Roman Guchenko[2,4]

[1]Cardiff University, Cardiff, UK
[2]Crimtan, London, UK
[3]Lobachevskii State University of Nizhnii Novgorod, Russia
[4]St.Petersburg State University, Russia
pepelyshevan@cardiff.ac.uk, yuri@crimtan.com, ZhigljavskyAA@cardiff.ac.uk, rguchenko@crimtan.com

Abstract. We consider the problem of adaptive targeting for real-time bidding for internet advertisement. This problem involves making fast decisions on whether to show a given ad to a particular user. For demand partners, these decisions are based on information extracted from big data sets containing records of previous impressions, clicks and subsequent purchases. We discuss several criteria which allow us to assess the significance of different factors on probabilities of clicks and conversions. We then devise simple strategies that are based on the use of the most influential factors and compare their performance with strategies that are much more computationally demanding. To make the numerical comparison, we use real data collected by Crimtan in the process of running several recent ad campaigns.

Keywords: Online advertisement, Real-time bidding, Adaptive targeting, Big data, Conversion rate

1 Introduction

During the last decade online advertisement became a significant part of the total advertisement market. Many companies including Google, Facebook and online news portals provide possibilities for online advertisement of their webpages to generate revenue. With high penetration of internet, online advertisement has gained attraction from marketers due to its specific features like scalability, measurability, ability to target individual users and relatively low cost per ad shown.

There are three main forms of online advertisement: search advertising (occurrs when a user conducts an online search), classified advertising (ads appear on websites of particular types, e.g. jobs and dating websites), and display advertising (banner ads on websites which are not search engines). During the last five years search and display advertising have moved from direct relationship between seller and buyer of ads to an advanced and flexible auction-based

model [12]. In this model, there is a seller of an ad space and several buyers - technology companies, who specialize in efficient ad delivery. Typically, demand partners pay per view, and prices are defined as cost per thousand ad exposures.

Display advertisement via actions has empowered the growth of the so-called programmatic buying, that is buying when decisions are made by machines based on algorithms and big data sets, rather than people. Demand partners typically collect databases with logs of all previous requests from auctions, impressions, clicks, conversions and users who visited a website which is currently advertised. These logs usually contain an anonimized user id, a browser name, an OS name, a geographical information derived from the IP address and a webpage link where an auction is run. Merging these datasets with third party data sources provides possibilities for contextual, geographical and behavioural targeting.

We consider the problem of online advertisement via auctions holding by independent ad exchanges from the position of a demand partner which wants to optimise the conversion rate. The demand partner has to decide how reasonable is it showing an ad in regard to a request from an auction and then possibly suggest a bid.

Demand partners put a special code on an advertised site to record users who made conversions. After few weeks of monitoring and running an ad campaign, demand partners collect a database with several thousands of conversions with just few records from this database occurring due to impressions. Also demand partners collect another database with requests on possibility to show an ad. By comparing these databases, demand partners have to develop procedures for estimating the conversion rate for new requests and subsequent bidding. Since demand partners should suggest a bid in few milliseconds, these procedures must be fast.

The demand partner has to solve the problem of maximizing either the click through rate (CTR) or the conversion rate by targeting a set of requests under several constraints: (a) budget (total amount of money available for advertising), (b) number of impressions N_{total} (the total amount of ad exposures), and (c) time (any ad campaign is restricted to a certain time period).

The problem of adaptive targeting for ad campaigns was recently addressed in quite a few papers, see e.g. [4,5,7,13]. In 2014 two contests were organized in Kaggle portal, see [14] and [15] on algorithms for predicting the CTR using a dataset with subsampled non-click records so that the CTR for the dataset is about 20% while for a typical advertising campaign the CTR is about 0.4% or less. The algorithms, which were proposed are publicly available and give approximately the same performance with respect to the logarithmic loss criterion

$$\text{logloss} = -1/N \sum_{i=1}^{N} (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)), \tag{1}$$

where N is the size of the data set, p_i is the predicted probability of click for the i-th request, and $y_i = 1$ if the i-th leads to click and $y_i = 0$ otherwise. This criterion, however, does not look very sensible when the probabilities p_i are very small as it pays equal weights to type I and type II error probabilities.

Moreover, the criterion (1) and other loss functions are not much often used in the industry as the advertisers are not interested in approximating click (conversion) probabilities at the entire range of admissible values of these probabilities; they are interested in making a decision (whether to show an ad) and hence they are only interested in making a correct decision whether some p_i is smaller or larger than some threshold value p^* (so that if $p_i \geq p^*$ then the demand partner will propose a bid for the i-the user). The threshold value p^* should be small enough to ensure that we will get the total number of impressions in time. On the other hand, we cannot let p^* to be too small as otherwise the final CTR (or conversion rate) will be too small. Rather than reporting values of the logloss or other criteria for different strategies, we present graphs which display the conversion rate as a function of the size of the sample with largest predicted values of the conversion probabilities. These types of figures are very common in the industry for assessing performances of different strategies.

In previous two papers [8,9] we have made a critical analysis of several procedures for on-line advertisement, provided a unified point of view on these procedures and have had a close look at the so-called 'look-alike' strategies. In the present paper we study relative influence of different factors on the conversion rate and hence develop simple procedures which are very computationally light but achieve the same accuracy as computationally demanding algorithms like Gradient Boosting Machines (see Section 2.5) or Field-Aware Factorization Machines (FFM), see e.g. [10]. Note that the number of parameters in the simplest FFM models is the sum of all factor levels plus perhaps interactions between factor levels. It counts to millions and if at least some interactions are taken into account then the count takes to much larger numbers.

Our models proposed in Section 2.4 are entirely different, they have a relatively small number of parameters. In particular, we propose a sparse model where only a few most significant factors are used for predicting the conversion rate.

2 Relative influence of factors

2.1 Notation and statement of the problem

Databases of logs contain records with many factors. Therefore, it is important to find the relative influence of all available factors and then build a prediction model using only the most important factors (and perhaps their interactions) in order to keep the computational time of evaluating the model for a new request small. Let us start with a formal statement of the problem.

Suppose that we have a database with records $x_1, \ldots, x_N$ and a vector $y_1, \ldots, y_N$ of binary outputs such that $y_j = 1$ if the j-th record has led to a conversion and $y_j = 0$ otherwise.

Each record x_j is described by m factors, $x_j = (x_{j,1}, \ldots, x_{j,m})$. The list of factors typically includes a browser name, an OS name, a device type, a country, a region, a visited webpage and behaviour categories. Let $p(x)$ be an idealistic

conversation rate for a request x; that is, $p(x) = \Pr\{y(x) = 1\}$. The knowledge of function $p(\cdot)$ would help us to construct an effective strategy of adaptive targeting for online advertisement. In practice, the function $p(\cdot)$ is unknown and even its existence is a mathematical model.

Suppose that the i-th factor has L_i levels $l_{i,1}, \ldots, l_{i,L_i}$. A relationship between the i-th factor and the binary output can be described by the contingency table. Specifically, we define

$$n_{i,k,s} = \#\{j : y_j = s, x_{j,i} = l_{i,k}\}$$

as the number of records for which the output y_j equals s and the value $x_{j,i}$ takes the value at the k-th level for the i-th factor. Here $s \in \{0, 1\}$, $i = 1, \ldots, m$ and $k = 1, \ldots, L_i$; note that $k = k_i$ depends on i.

For fixed i, the frequency table $(p_{i,k,s})_{k=1,\ldots,L_i}^{s=0,1}$ with

$$p_{i,k,s} = n_{i,k,s}/N$$

provides the joint empirical distribution for the pair of the i-th factor and the binary output, where N is the total number of records.

The row-sums for these frequency tables are $p_{i,k,*} = p_{i,k,0} + p_{i,k,1}$, so that the vector with frequencies $p_{i,k,*}$, $k = 1, \ldots, L_i$ gives the empirical distribution of levels for the i-th factor.

The column-sums for the frequency tables are

$$p_{i,*,s} = p_{i,1,s} + \ldots + p_{i,L_i,s}\,.$$

These values clearly do not depend on i so that

$$P = \frac{\sum_{j=1}^{N} y_j}{N} = p_{i,*,1} \quad \text{and } p_{i,*,0} = 1 - P$$

for all i where P is the overall frequency of 1 for the binary output; that is, the overall conversion rate for the database. Note, however, that P is not the conversion rate of an ad campaign because the database contains records of non-converted requests and converters which are not related to the active ad campaign (the converters recorded directly by the demand partners who put a special code on an advertised site to record users who made conversions).

To identify how the i-th factor affects the conversion rate $p(x)$, we consider several statistics which measure the dispersion or mutual information.

2.2 Relative influence via dispersion

To find the relative influence of the i-th factor on the conversion rate in the sense of the dispersion of the conversion rate for different levels of the i-th factor, we propose the statistic defined by

$$I_i^{(D)} = \sum_{k=1}^{L_i} p_{i,k,*}(q_{i,k} - P)^2$$

where

$$q_{i,k} = \frac{n_{i,k,1}}{n_{i,k,0} + n_{i,k,1}} = \frac{p_{i,k,1}}{p_{i,k,*}}$$

is the conversion rate for the records with k-th level for the i-th factor.

2.3 Relative influence via mutual information

Mutual information is an information-theoretic measure of divergence between the joint distribution and the product of two marginal distributions, see the classical book [2]. If two random variables are independent, then the mutual information is zero. We apply mutual information to measure a degree of dependence between the i-th factor and the binary output.

To find the relative influence of the i-th factor in the sense the mutual information based on the Shannon entropy, we consider the statistic defined by

$$I_i^{(Sh)} = \sum_{k=1}^{L_i} \sum_{s=0}^{1} p_{i,k,s} \log_2 \frac{p_{i,k,s}}{p_{i,k,*}\, p_{i,*,s}} \,.$$

To find the relative influence of the i-th factor in the sense the mutual information based on the Renyi entropy of order α, we consider the statistic defined by

$$I_i^{(Re,\alpha)} = \log_2 \sum_{k=1}^{L_i} \sum_{s=0}^{1} \frac{p_{i,k,s}^{\alpha}}{p_{i,k,*}^{\alpha-1}\, p_{i,*,s}^{\alpha-1}} \,.$$

It is known that mutual information is not robust in the case in which there are levels with rare occurrence, see [1]. To regularize the above statistics $I_i^{(D)}, I_i^{(Sh)}$ and $I_i^{(Re,\alpha)}$, we perform a pre-processing of the database by replacing rare levels with $n_{i,k,*} \leq 9$ by a dummy level.

Note that the Renyi mutual information $I_i^{(Re,\alpha)}$ was used for factor selection in the literature, see e.g. [6]; however, the range of applications was entirely different. The Shannon mutual information $I_i^{(Sh)}$ is a standard in many areas.

2.4 MI-based model for estimating the conversion rate

Suppose that we are given a new request $X = (X_1, \ldots, X_m)$ and we want to estimate the conversion rate $p(X)$. As an estimator of $p(X)$, we propose

$$\hat{p}(X) = \frac{\sum_{i=1}^{m} I_i q_{i,k_i}}{\sum_{i=1}^{m} I_i} \tag{2}$$

where k_i is such that $X_i = l_{i,k_i}$ and I_i is a relative influence of the i-th factor.

Furthermore, if we want to use a sparse predictive model then we can set the values of I_i such that $I_i \leq \epsilon$ to zero, for some small $\epsilon > 0$.

The expression (2) resembles the form of the multi-factor multi-level ANOVA model. However, the model (2) uses totally different methods of estimating parameters than standard ANOVA regression.

The main advantage of the proposed model (2) is its simplicity and time efficiency. As we demonstrate below, precision of this model is basically identical to the precision of much more complicated models based on the use of the Gradient Boosting Machines (GBM).

2.5 Gradient Boosting Machines

GBM is a method of sequential approximation of the desired function $p(x)$ by a function of the form

$$p^{(k)}(x) = \sum_{i=1}^{k} \alpha_i h(x, \theta_i),$$

where at iteration k the coefficient α_k and the vector θ_k are estimated through minimizing some loss criterion $L(\cdot, \cdot)$; see e.g. [3,11]; the values of α_i and θ_i for $i < k$ being kept from previous iterations. Since many factors are categorical, we consider the special case of the so-called tree-based GBM, where the function $h(x, \theta)$ is called a regression tree and has the form

$$h(x, \theta) = \sum_{j=1}^{J} b_j \mathbf{1}_{R_j}(x)$$

where $R_1, \ldots, R_J$ are disjoint sets whose union is the whole space and these sets correspond to J terminal nodes of the tree. The indicator function $\mathbf{1}_R(x)$ equals 0 if x belongs to a set R and 0 otherwise. The vector θ for the regression tree $h(x, \theta)$ is a collection of $b_1, \ldots, b_J$ and $R_1, \ldots, R_J$, which parameterize the tree. Note that levels of categorical variables are encoded by integer numbers.

To build the GBM model for a real data, we take the `gbm` package in `R`. We use the function `gbm` which constructs the generalized boosted regression model has the following parameters, see [11]:

- (i) `n.trees`, the total number of trees in the model,
- (ii) `interaction.depth`, the maximal depth of factor interactions,
- (iii) `n.minobsinnode`, the minimal number of records in the terminal nodes of trees,
- (iv) `bag.fraction`, the fraction of records from the training set randomly selected to construct the next tree,
- (v) `shrinkage`, the learning rate ν which is used to define $\alpha_i = \nu\gamma_i$, where

$$\gamma_i = \arg\min_{\gamma} \sum_{j=1}^{N} L(y_j, p^{(i-1)}(x_j) + \gamma h(x_j, \theta_i)).$$

The values used in industry are typically as follows `n.minobsinnode`=100, `n.trees`=500, `shrinkage`=0.1, `interaction.depth`=5, `bag.fraction`=0.5

3 Numerical results

In the present section we analyze several ad campaigns which were executed by Crimtan.

To investigate the performance of different strategies for the database of requests for an ad campaign, we split the database of records into 2 sets: the training set of past records with dates until a certain time T and the test set of future records with dates from the time T. The training set contains 50,000 records but the test sets are much larger (their sizes are in the range of 1 million). We now compare GBM and the model based on the use of (2) by comparing the conversion rate for the samples of most favorable requests with the highest chances of conversion.

To form the sample of most favorable requests for the GBM approach, we construct the GBM model using the training set and then apply this model to predict the probability of conversion for each request from the test set. Now we can sort the predicted probabilities and create samples of requests with highest predicted probabilities of conversion.

In Figure 1 we can see that all four considered versions of the relative influence of factors give somewhat similar orderings. We note that the factor 36 provides significant influence in some ad campaigns and small influence in others. However, factors 33 and 40 have large influence in all four ad campaigns.

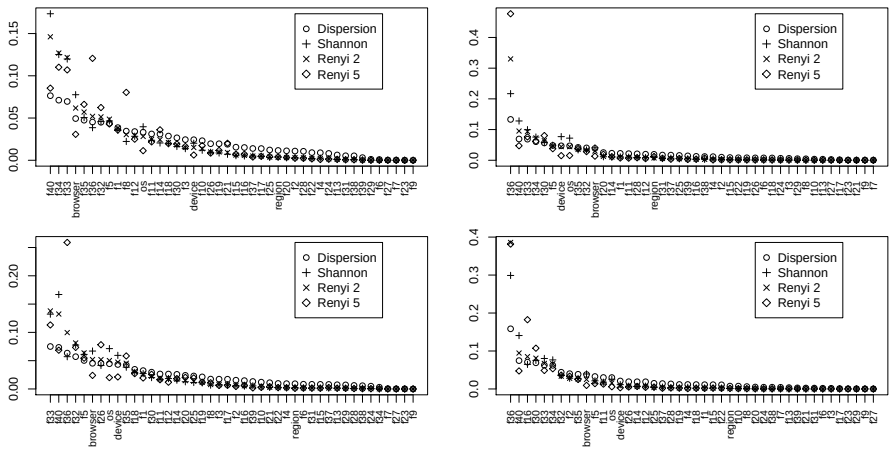

Fig. 1. Relative influence of factors for 4 ad campaigns.

In Figure 2 we can see that the performance of the MI-based model is the same for the four considered versions of the relative influence of factors both for the training set and the test set for a chosen ad campaign. Since the performance for the test set is similar to the performance for the training set, we can conclude that there is no over-fitting in the MI-based model. This is not the case for the

GBM, see Figure 3. In particular, if the depth level is high then the GBM performance for the training set is visibly better than its performance for the test set.

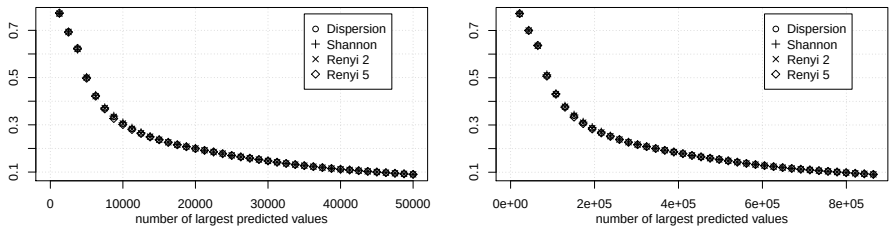

Fig. 2. The performance of the MI-based model with 4 forms of the relative influences of factors for the training set (left) and the test set (right) for an ad campaign. The y-scale is the conversion rate of samples of largest predicted values for various sample sizes.

In Figure 3 we can also see that the performance of the GBM model does not depend on the interaction depth, when the number of trees is 500 and the bag fraction is 0.5. Comparing Figures 2 and 3 we can see that the performance of the simple MI-based model is very close to the performance of the complex, computationally demanding GBM model.

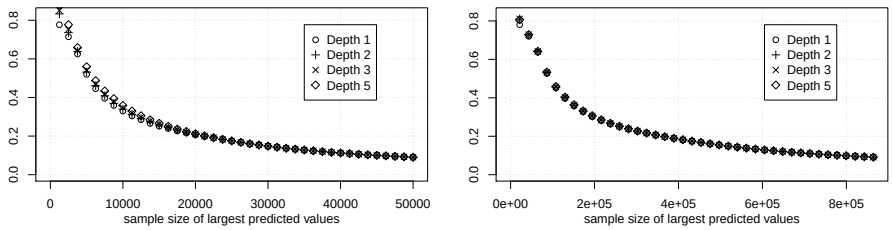

Fig. 3. The performance of the GBM model with different interaction depth for the training set (left) and the test set (right) for an ad campaign, with the number of trees 500.

In Figure 4 we can see that the performance of the GBM model with larger number of trees on the training set is marginally better than with smaller number of trees. However, the performance of the GBM model with different number of trees for the test set is virtually the same.

In Figures 5 and 6 we compare the performance of the proposed MI-based model with $I_i^{(D)}$ to the performance of the GBM model with 500 trees and the interaction depth 5 (which is a very time-consuming algorithm). For both ad campaigns, GBM performance on the training sets is slightly better than the

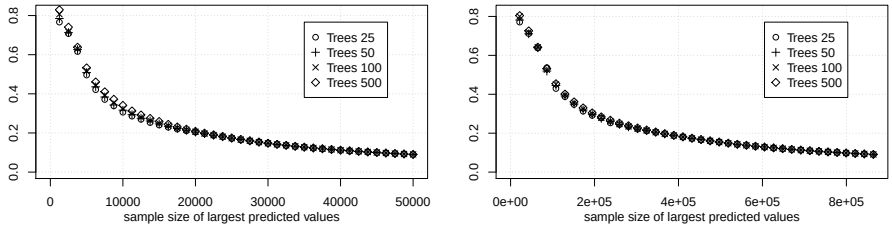

Fig. 4. The performance of the GBM model with different number of trees for the training set (left) and the test set (right) for an ad campaign, with the interaction depth 2.

performance of the proposed algorithm. This can be explained by the fact that GBM has thousands times more degrees of freedom than our model and, by the definition of the method, GBM tries to fit the data as best as it can.

GBM performances on the test sets are slightly worse than that on the training sets and they are very similar to the performance of the proposed algorithm. A slight advantage of GBM over the MI-based method for the records X that have high values of probabilities $p(X)$ is not important for the following two reasons: (a) high probabilities of $p(X)$ can only be observed for the supplementary part of the database containing the records which are not a part of the ad campaign, and (b) as mentioned above, we are interested in a good estimation of $p(X)$ such that $p(X) \simeq p^*$, where p^* is the threshold value, which is quite small.

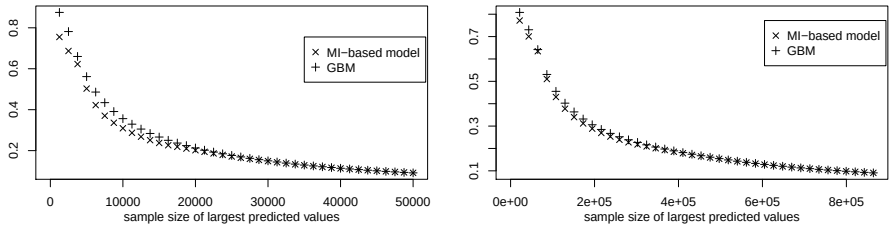

Fig. 5. The performance of the MI-based model with $I_i^{(D)}$ and the GBM model with 500 trees and the interaction depth 5 for the training set (left) and the test set (right) for an ad campaign.

We should notice that the MI-based model $\hat{p}(X)$ is not good for estimating the conversion rate $p(X)$ in view of some bias. We can only use the MI-based models for ranking the requests using predictive values and choosing the most promising ones. If one wishes to enhance the MI-based model and obtain a good estimator of $p(X)$, then we recommend to remove non-influential factors by computing the mutual information and build a logistic model using the most influential factors and possibly their interactions.

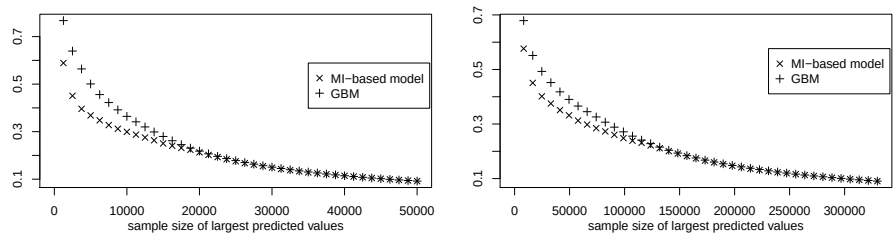

Fig. 6. The performance of the MI-based model with $I_i^{(D)}$ and the GBM model with 500 trees and the interaction depth 5 for the training set (left) and the test set (right) for another ad campaign.

Finally we would like to highlight the time efficiency of computations for the proposed model. Construction of the MI-based model and evaluating of the MI-based model for new requests is at least 10 times faster comparing with the GBM model. In fact, the MI-based model can be used in the regime of real-time learning; that is, the contingency tables and the predictive model can be easily updated as bunches of new requests arrived.

Acknowledgement

The paper is a result of collaboration of Crimtan, a provider of proprietary ad technology platform and University of Cardiff. Research of the third author was supported by the Russian Science Foundation, project No. 15-11-30022 "Global optimization, supercomputing computations, and application".

References

1. Buja, A., Stuetzle, W., Shen, Y.: Loss functions for binary class probability estimation and classification: Structure and applications. Working draft (2005)
2. Cover, T. M., Thomas, J. A.: Elements of information theory. John Wiley & Sons (2012)
3. Friedman, J. H.: Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189–1232 (2001)
4. Jansen, B.J., Mullen, T.: Sponsored search: an overview of the concept, history, and technology. International Journal of Electronic Business 6(2), 114–131 (2008)
5. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Candela, J. Q.: Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (pp. 1-9). ACM (2014)
6. Lima, C. F. L., de Assis, F. M., de Souza, C. P.: An empirical investigation of attribute selection techniques based on shannon, renyi and tsallis entropies for network intrusion detection. American Journal of Intelligent Systems, 2(5), 111-117 (2012)
7. McMahan, H. B., Holt, G., Sculley, D., et al.: Ad click prediction: a view from the trenches. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 1222–1230 (2013)

8. Pepelyshev, A., Staroselskiy, Y., Zhigljavsky, A.: Adaptive Targeting for Online Advertisement. In Machine Learning, Optimization, and Big Data. Springer International Publishing (pp. 240-251). (2015)
9. Pepelyshev, A., Staroselskiy, Y., Zhigljavsky, A.: Adaptive Designs for Optimizing Online Advertisement Campaigns. Statistical Papers, vol. 57 (pp. 199-208). (2016)
10. Rendle, S.: Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 995–1000). IEEE (2010)
11. Ridgeway, G.: Generalized Boosted Models: A guide to the gbm package. Update 1.1 (2007)
12. Wang, J., Yuan, S., Shen, X., Seljan, S.: Real-time bidding: A new frontier of computational advertising research. CIKM Tutorial (2013)
13. Yang, S., Ghose, A.: Analyzing the Relationship between Organic and Sponsored Search Advertising: Positive, Negative or Zero Interdependence?, Marketing Science, 29 (4), 602–623 (2010)
14. `https://www.kaggle.com/c/avazu-ctr-prediction`
15. `https://www.kaggle.com/c/criteo-display-ad-challenge`]