Using Mutual Information for Feature Selection in Programmatic Advertising

Michal Ciesielczyk
Poznan University of Technology
Institute of Control and Information Engineering
ul. Piotrowo 3a, 60-965 Poznan, Poland
Email: Michal.Ciesielczyk@put.poznan.pl

Abstract—**Click-through rate estimation, the core task of programmatic display advertising, is associated with typical big data problems. Online algorithms for generalized linear models, such as Logistic Regression, are the most widely used data mining techniques for learning at such a massive scale. Since these models are unable to capture the underlying nonlinear data patterns, conjunction features are often introduced. This paper is focused on the problem of selecting the most informative 2nd and 3rd order conjunction features used in Logistic Regression. The performance of different feature selection methods based on mutual information is compared over a real-world dataset with over 10 million records. The empirical evaluation show the effectiveness of the proposed approach.**

Index Terms—**Big Data, Mutual Information, Feature Selection, Logistic Regression, Programmatic Advertising, Click-through Rate Prediction, Machine Learning**

I. Introduction

Programmatic advertising, due to its audience targeting efficiency [2,16], is a still growing multi-billion dollar industry that is being rapidly adopted across a variety of online channels. It is estimated that programmatic ad spending in United States during 2017 will reach $31.87 billion, representing 78.0% of total digital display ad spending [7]. In contrast to the traditional online advertising, where a fixed rate is preset for each campaign or keyword, programmatic enables the advertisers to bid for every individual impression in real time.

Many advertisers prefer not to pay for an impression unless it will lead the user to the advertiser's website [2]. Payment models such as cost-per-click (CPC) were introduced to respond to this demand. Under the CPC model the expected cost-per-impression will directly depend on the Click-Through Rate (CTR) – i.e. the probability that the impression will lead to a click [16]. As estimating these probabilities accurately and reliably is essential for an efficient marketplace [2], many machine learning techniques have been employed to predict CTR [12]. Even relatively small improvements may have significant impact on the return-on-investment (ROI) [9].

Since predicting CTR in display advertising is associated with typical big data problems such as massive volumes of heterogeneous data streams that has to be processed in real time it is considered an engineering challenge [22]. A typical CTR estimation system has to provide recommendations on billions of events per day, returning every bid in less than 20 ms, for hundreds of millions of unique users and millions of unique pages [2,12]. Such requirements, combined with a correspondingly large feature space due to lack of easily generalizable features (mostly unique identifiers with no content related information), make the pre-existing solutions concerning modeling clicks (e.g. in the context of search and search advertising) at least as inadequate for CTR prediction [2]. For this reason, this problem setting attracts researchers from both academia and industry [2,9,12,19].

978-1-5090-5795-5/17/$31.00 © 2017 IEEE

The goal of this paper is to share the findings of the experiments performed against real-world data while taking into account the requirements of production-scale systems. Because the field addressing the issues of CTR prediction is now well studied, allowing to process data at a scale that was almost inconceivable even a decade ago [12], this paper is focused on the problem of selecting the most important conjunction features. The issue of modeling the relations between features have recently received less attention but is equally important in dynamic allocation of ads tailored to user interests. As confirmed in [2] and [19], the use of additional conjunctions features may offer performance gains compared with a linear model that can only learn independently from each feature.

II. Related Work

Logistic Regression (LR) due to its efficiency, ability to handle large-scale problems, incremental updates, and easy implementation is probably the most widely used model for CTR prediction [2,9,10,12,16,22]. However, a plain LR model is not capable of capturing complex, nonlinear relationships between features which hinders its ability to provide highly contextual and accurate recommendations. Usually, to overcome this issue, new conjunction features are introduced into the model [2]. A conjunction feature is a Cartesian product of two other categorical or conjunction features.

Note that the composition of conjunction features using a Cartesian product is compliant with the tensor-based definition where the tensor product is used to build algebraic representations of feature conjunctions, as introduced in [19]. Since the tensor space is a space formed over a Cartesian product of the constituent vector spaces, the features are represented by vector spaces and the values are mapped to the dimensions of these spaces.

Nonetheless, due to memory, latency and training time restrictions in real-world applications, the dimensionality of the categorical and conjunction features becomes a major concern [2]. Especially if we consider that 2nd-order conjunction features might be insufficient to catch the underlying data patterns, as found in [21]. Moreover, irrelevant attributes introduce noise and may diminish the final CTR prediction quality [15]. In particular, such an effect may be evident when using a hashing trick [10,20] due to significantly higher number of collisions. Thus, feature selection algorithms are becoming a requirement in many real tasks [2].

Since optimal feature selection is an NP-hard problem in general [15], many approximate solutions have been proposed in the literature. Conditional mutual information and similar filter methods are frequently used in CTR prediction [2] and recommender systems [6] to determine the features most correlated with the target event. Manual feature engineering has also been suggested in many models [19,21,22]. However, such an approach requires domain expertise and reduces automation. Other solutions include methods such Query Expansion Ranking [13], and Sine Cosine algorithm [8]. Nevertheless, these techniques have not been yet adapted to the scale and specifics of the CTR prediction task.

Non-linear models such as deep neural network (DNN) and higher-order tensor factorization has been proposed to enable the exploration of feature relationships [3,16,18,21]. For instance, in [16], a fully coupled interactions tensor factorization model is proposed to model the pair-wise interactions between the user, publisher and advertiser. However, due to relatively complicated data model, such approaches often do not enable to obtain high CTR prediction accuracy results as reported in [18] or do not allow to process large datasets [16].

III. Online Learning

Usually, while learning on large data streams it is feasible to process each training example only once. Online algorithms for generalized linear models such as LR meet this requirement. Although a feature vector may have billions of dimensions, typically it is very sparse and has only hundreds of nonzero values, enabling to stream examples from disk or over the network [1,14].

A. Logistic Regression

Let $(\mathbf{x}_i, y_i)$ denote a training set, where $\mathbf{x}_i$ is a binary feature vector in a d dimensional space, and y_i is a binary target value. The LR model enables to estimate the probability that an example $\mathbf{x}$ belongs to class 1 according to:

$$P(y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^\intercal \mathbf{x}}} \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the vector of model parameters [2]. The $\mathbf{w}$ vector is found by minimizing:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} log(1 + e^{-y_i \mathbf{w}^\intercal \mathbf{x}}) \tag{2}$$

Since (2) is convex, unconstrained and differentiable, it can be solved with any gradient based optimization technique [2]. Herein, Online Gradient Descent (OGD) was used to learn the model parameters, enabling to produce high prediction accuracy with a minimum of computing resources [12]. The OGD algorithm is a variant of Stochastic Gradient Descent (SGD), in which each example (that need not to be i.i.d.) from the data stream is processed sequentially and only once [12].

B. Hashing Trick

In domains such as programmatic display advertising, the majority of feature values are extremely rare or, in some cases, even unique [12]. As the number of values in a production setting can easily exceed a billion [2], and usually it not known in advance which values are rare [12], various techniques to reduce the dimensionality of the model has been proposed.

Commonly, in place of dictionary encoding, hashing with collisions is used to regulate the size of the model [2,10,20]. As a result, the final number of feature values d is constant and equal to 2^b, where b is the number of bits used for hashing. In this paper, similarly as in [2], all features were hashed into the same space using a different hash function for each feature.

Other methods of reducing the dimensionality of the model, such as those discarding infrequent values (e.g. Poisson Inclusion or Bloom Filter Inclusion), require additional data processing or storage, and may turn out to be computationally prohibitive in a real-world environment [2]. In contrast, the hashing trick is straightforward to implement and apply to online data stream processing.

IV. Feature Selection

A. Mutual Information

Let each training example be a set of features and a target value be represented by a sample from an unknown distribution p over discrete random variables $X_i \in X$ and Y correspondingly. The conditional mutual information [15] between X_i and Y can be defined as:

$$MI(X_i; Y) = \sum_{x_i, y} p(x_i, y) log \frac{p(x_i, y)}{p(x_i)p(y)} \tag{3}$$

where $p(x_i, y)$ is the joint probability distribution function of X_i and Y, and $p(x_i)$ along with $p(y)$ are the corresponding marginal probability distribution functions.

The top K features according to (3) – i.e., having the most information content about the target variable Y – are included in the learning system. In this paper, similarly as in [2], as not to introduce confusion this procedure is referred to as the Standard Mutual Information (SMI) method.

However, as indicated in [2], the SMI is biased towards features with high cardinality. The larger the number of unique values a feature has, the higher the risk the SMI does not generalize on the test distribution. As a result, it leads to inappropriate attribute selection results.

B. Mutual Information Using a Reference Distribution

To address the issues related with SMI, the mutual information may be computed on a validation set [2]. In particular, the original training set is divided into a new training set (on which the $p(x_i, y)$ is estimated) and a validation set. Then, the information content is calculated with respect to expectations on the reference distribution.

Specifically, the Mutual Information using a Reference distribution (RMI) between X_i and Y is defined as:

$$RMI(X_i;Y) = \sum_{x_i,y} p_r(x_i,y) log \frac{p(x_i,y)}{p(x_i)p(y)} \qquad (4)$$

where $p_r(x_i, y)$ is the joint probability of X_i and Y in a reference distribution.

C. Adapted RMI

RMI is biased towards higher order feature conjunctions. Specifically, 3rd order conjunctions are preferred over 2nd order conjunctions and over base features. Such a property significantly affects the attribute selection results, usually reducing the generalization capabilities of the measure.

To address this issue the RMI measure may be adapted in the following way:

$$aRMI(X_i;Y) = \sum_{x_i,y} \Big(\frac{p_r(x_i,y)}{RMI(X_i;Y)} log \frac{p(x_i,y)}{p(x_i)p(y)}\Big)^{ord} \qquad (5)$$

where ord indicates the conjunction order of feature X. By this means, the lower-order conjunctions are considered more important and their scores are boosted. The additional normalization is needed to compensate for high-cardinality features. In the experimental results, presented in section VI, we show that at least in the evaluated scenarios the adapted RMI (aRMI) enables to achieve significantly higher CTR prediction accuracy.

V. Evaluation Methodology

A. Datasets

For the following experiments, a real dataset containing two seasons of Real-Time Bidding (RTB) records from the iPinYou RTB Dataset[1] was used. The dataset was released by iPinYou Information Technologies Co., Ltd for a global RTB bidding algorithm competition [22]. It contains impression, click, and conversion logs collected from several advertisers, organized on a row-per-record basis.

Each record contains information on a specific user, who visited a publisher website and was given an ad impression, along with other features regarding the RTB auction. In the presented experiments, same feature pre-processing as in [22] and [19] was performed. For the purposes of the LR algorithm, the categorical and numerical features were converted into binary. The result features, along with example values, extracted from the iPinYou dataset are shown in Table I.

TABLE I
iPinYou dataset features with example values.

Feature name	Example value	# of unique values
OS	Android	6
Browser	Firefox	9
IP	124.164.238.*	744,552
City	20	370
Region	15	35
User tag	10006,10110	75
Domain	trqRTJk…wTK4wJB	61,421
Slot ID	mm_1299…0344354	226,420
Slot size	336x280	29
Slot visibility	FirstView	11
Slot format	5	4
Floor price	[11,50]	5
Creative ID	7321	131
Advertiser ID	1458	9
Weekday	3	7
Hour	15	24
AdExchange	2	5

TABLE II
iPinYou dataset statistics.

Season	Dataset	Impressions	Unique values	Clicks
2	training set	12,190,438	801,884	8,838
	test set	2,521,627	543,705	1,873
3	training set	3,147,801	589,866	2,700
	test set	1,579,071	482,202	1,135

Each season dataset is divided – according to the impression date [22] – into two parts, a training set and test set. General dataset statistics are presented in Table II.

More details on the iPinYou dataset may be found in [22].

B. Measures

In this paper, similarly as in [16] and [19], the CTR estimation problem is considered as a recommendation system task, where ads must be recommended for appropriate users depending on the current context. In such a setting, one of the most widely used metrics are area under the ROC curve (AuROC) [2,12,16,19,20,22], area under the precision-recall curve (AuPRC) [2,19], and Root Mean Squared Error (RMSE) [12,16,22]. Because different performance metrics respond differently to model changes, it has been found to be useful to evaluate the system's performance from various perspectives [12].

According to [4,17], to represent the overall performance of a recommendation system, one should analyze the precision-recall or ROC curves based on the properties of the domain and the goal of the application. While curves of the both types measure the proportion of preferred items that are actually recommended, precision-recall curves emphasize the proportion of recommended items that are preferred while ROC curves emphasize the proportion of items that are not preferred

[1] http://data.computational-advertising.org/

that end up being recommended [17]. Such a difference is especially evident in cases of heavy imbalance of positive and negative examples (i.e. *click* and *non-click* events), which is typical in the display advertising domain [18,21]. An algorithm that optimizes AuROC does not guarantee the optimization of AuPRC [5]. Therefore, along with to the AuROC results, the AuPRC results are shown.

The RMSE metric, due to heavy class imbalance, has been reported to be inappropriate for measuring the CTR estimation system performance [18,21].

C. Experimental Setup

In all experiments, the OGD algorithm – described in Section III – was used to estimate the CTR. Due to essential differences between the advertisers in the iPinYou dataset [22], a separate LR model was build for each advertiser. No regularization was added to the model since in the evaluated scenarios it did not affect the results significantly. The number of bits used for hashing was set to 18, similarly as in the Vowpal Wabbit by default [11]. The learning rate was set to 0.01.

The feature selection algorithms were used to pick the most informative 2nd and 3rd order conjunction features to be introduced into the LR model. In addition to the algorithms based on mutual information and introduced in Section IV – namely SMI, RMI, and aRMI – a random feature selection method has been evaluated. To reduce the variations in the consequent runs, all the experiments were repeated 10 times (30 times in case of random feature selection) and the average values are reported. To compare the prediction quality we present the AuPRC and AuROC values, and the improvement over the base OGD model with no conjunction features used.

All experiments were conducted on a Linux workstation with Intel Core i7-950 processor and 16 GB RAM.

VI. Experimental Results and Discussion

A. Feature Selection Results

All evaluated ranking algorithms based on mutual information were applied to determine the most informative 2nd and 3rd order conjunction features. Table III presents the top features for iPinYou advertiser 3358. Although the SMI scores directly reflect the information content of the attributes, the selected features are too specific. Spuriously correlated features with high cardinality tend to be ranked substantially high. In particular, higher order conjunctions, composed of features such as IP which to a large extent identify the data point in the training set, have the highest SMI scores.

Calculating the conditional mutual information using a reference distribution enables to filter the features that do not generalize across the dataset. Nevertheless, in case of RMI, the 3rd order conjunctions – although they are substantially more specific than the 2nd order conjunctions – are considered more important. Such a tendency towards higher order conjunction features may cause the attributes with lower relevance for prediction to be included into the model. Using an additional scaling factor based on conjunctions order, as shown in (5), enables to provide a more appropriate ranking (i.e. not biased towards higher order conjunctions).

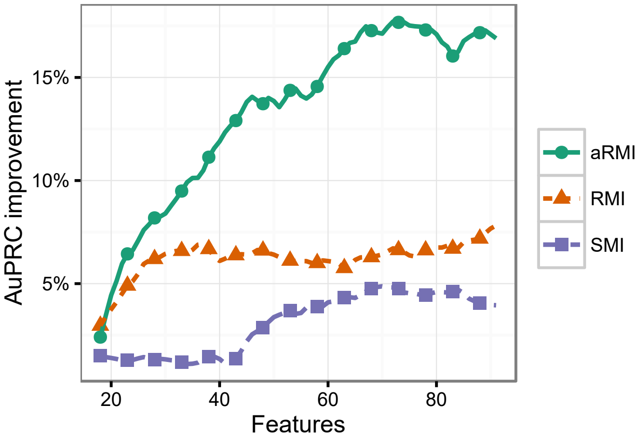

Fig. 1. Total AuPRC improvement over the whole iPinYou dataset as a function of the number of features.

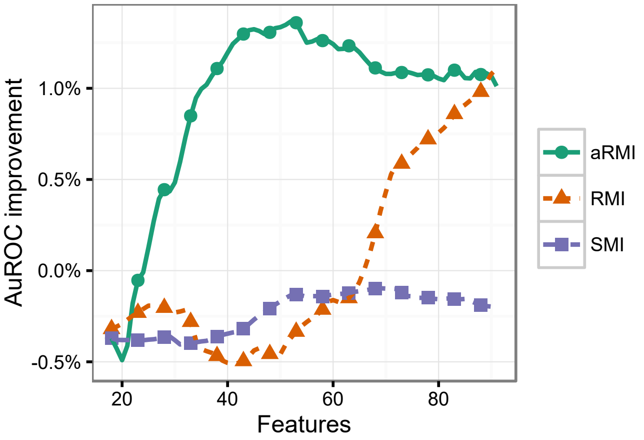

Fig. 2. Total AuROC improvement over the whole iPinYou dataset as a function of the number of features.

B. Impact of Feature Selection on CTR Performance

Tables IV and V summarize the AuPRC and AuROC relative improvements, correspondingly, through the use of additional conjunction features over the base LR model with no conjunctions used. The conjunction features added to the model were determined using indicated attribute selection methods. Specifically, 33 out of 680 conjunctions were added to the model with 17 base features, resulting in a model with 50 – similarly as the one evaluated in [2]. The total AuPRC and AuROC relative improvement (over the whole iPinYou dataset) as a function of the number of features is shown in Fig. 1 and Fig. 2, correspondingly.

The presented results indicate that in most cases it is beneficial – in terms of AuPRC – to add 2nd and 3rd order conjunction features to the base model. We confirm the findings of [2], stating that using a reference distribution (as in RMI and aRMI) enables to determine more meaningful top conjunction features for the CTR prediction task. Nevertheless,

TABLE III
Top 2nd and 3rd order conjunction features using different selection techniques for iPinYou advertiser 3358.

No	SMI	RMI	aRMI
1	IP × Slot ID × User tag	Slot format × OS × User tag	Slot ID × User tag
2	IP × User tag × Hour	Browser × Slot size × User tag	Domain × User tag
3	IP × Slot size × User tag	Browser × Creative ID × User tag	Slot format × OS × User tag
4	Creative ID × IP × User tag	Browser × Slot format × User tag	OS × User tag
5	IP × User tag × Slot visibility	Slot ID × User tag × Slot visibility	Browser × Slot size × User tag
6	Slot format × IP × User tag	Creative ID × OS × User tag	Browser × Creative ID × User tag
7	Floor price × IP × User tag	OS × Slot size × User tag	Slot format × User tag
8	IP × User tag × Weekday	Slot ID × User tag	Browser × Slot format × User tag

TABLE IV
AuPRC improvement through the use of additional features over the base LR with no conjunction features used; the best result is highlighted by bold font setting.

advertiser	random	SMI	RMI	aRMI
1458	-0.92%	-0.22%	1.30%	**3.22%**
2259	8.94%	-3.53%	-7.86%	**17.46%**
2261	13.30%	1.89%	10.06%	**13.57%**
2821	8.96%	3.53%	3.71%	**14.94%**
2997	-8.57%	**-0.68%**	-4.76%	-9.02%
3358	16.85%	16.84%	33.75%	**46.22%**
3386	8.69%	2.43%	-4.08%	**9.00%**
3427	5.66%	5.33%	10.85%	**11.55%**
3476	0.52%	5.67%	18.54%	**25.95%**
Season 2	4.98%	4.80%	9.62%	**16.10%**
Season 3	8.17%	0.89%	1.20%	**12.94%**
Total	6.21%	3.29%	6.38%	**14.88%**

TABLE V
AuROC improvement through the use of additional features over the base LR with no conjunction features used; the best result is highlighted by bold font setting.

advertiser	random	SMI	RMI	aRMI
1458	-0.11%	-0.03%	0.07%	**0.12%**
2259	1.24%	-0.60%	-0.79%	**2.78%**
2261	**0.09%**	-1.38%	-1.84%	-0.08%
2821	2.09%	-1.33%	-0.10%	**5.82%**
2997	-1.21%	**0.36%**	-3.63%	-4.30%
3358	-0.42%	**0.09%**	-0.54%	0.08%
3386	0.90%	0.18%	-0.45%	**2.48%**
3427	0.08%	-0.20%	-0.31%	**0.16%**
3476	-0.38%	0.29%	0.22%	**0.61%**
Season 2	0.06%	0.06%	-0.16%	**0.74%**
Season 3	1.10%	-0.98%	-1.01%	**2.73%**
Total	0.46%	-0.34%	-0.49%	**1.51%**

the achieved AuROC results (Table V and Fig. 2) are not so unambiguous.

As it can be seen, even random selection of conjunction features enables to improve the recommendation quality over the base model. Although in some cases it surpasses both SMI and RMI, note that the presented result is an average of series of experiments, close to the expected value. In particular, the outcomes of single experiments may give scattered results. The aRMI method enabled to achieve significantly higher AuPRC and AuROC values for most of the iPinYou advertisers, resulting in the highest values over the whole dataset. The deterioration of the result for the advertiser 2997 is probably a consequence of more noisy data collected in a mobile environment, as indicated in [22].

Fig. 3 shows the precision-recall curves for the iPinYou advertiser 3427. Both aRMI and random selection were used to determine the top 50 conjunction features to be included in the LR model. The additional conjunctions enable to improve the precision especially for the lower recall values (i.e. recall < 0.5). In the cases of higher recall values (i.e. recall > 0.6) the effect can be negligible or even negative in terms of precision improvement. Such a result may be caused by better fitting to the training data through more specific features. Nonetheless, in overall, as presented in Fig. 1 and Table IV, it is beneficial to include these non-linear parameters into the model.

VII. Conclusions and Future Work

In this paper we have focused on the problem of selecting the most useful conjunction features for an LR model in the programmatic advertising domain. In contrast to [15] and [2], in addition to the 2nd order conjunction features, 3rd order conjunctions have been also incorporated into the LR model. Moreover, the mutual information was estimated on pre-processed data, instead of on raw features as in [15]. For instance, user browser and OS information were used in preference to a hashed HTTP cookie. The results of the presented experiments allow us to conclude that, in terms of CTR prediction effectiveness it may be beneficial to use both 2nd and 3rd order conjunction features.

As in [2], it has been confirmed that using a reference distribution enables to address some of the limitations of SMI. In case of selecting top conjunction features that have different order, reformulating the RMI measure as in (5) enables to select features that are more appropriate for the CTR prediction task. Specifically, aRMI is not biased towards higher order conjunction features.

The proposed solution, as it enables to increase the quality of CTR prediction, could be a valuable tool in any LR model used in the programmatic advertising domain. In addition, by

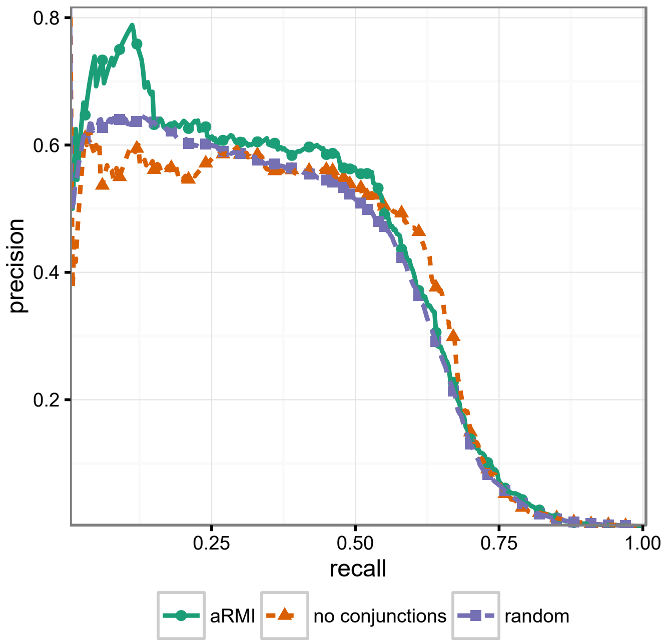

Fig. 3. Precision-recall curve for the advertiser 3427 using different feature selection methods.

increasing the level of automation, one could significantly cut down the amount of manual work required and reduce the need for domain expertise.

A promising future work direction would be to evaluate the ability to incrementally incorporate the most useful features into the model – fully complying with the requirements of streaming data processing. Currently, to determine the top conjunction features, an additional pre-processing step using preliminary collected data is needed.

Acknowledgment

This work is supported by the Polish National Science Centre, grant DEC-2011/01/D/ST6/06788.

References

[1] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling Up Machine Learning: Parallel and Distributed Approaches*. New York, NY, USA: Cambridge University Press, 2011.

[2] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and Scalable Response Prediction for Display Advertising," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 4, pp. 61:1—-61:34, 2014. [Online]. Available: http://doi.acm.org/10.1145/2532128

[3] J. Chen, B. Sun, H. Li, H. Lu, and X.-S. Hua, "Deep CTR Prediction in Display Advertising," in *Proceedings of the 2016 ACM on Multimedia Conference*, ser. MM '16. New York, NY, USA: ACM, 2016, pp. 811–820. [Online]. Available: http://doi.acm.org/10.1145/2964284.2964325

[4] M. Ciesielczyk, A. Szwabe, M. Morzy, and P. Misiorek, "Progressive Random Indexing: Dimensionality Reduction Preserving Local Network Dependencies," *ACM Transactions on Internet Technology*, vol. 17, no. 2, pp. 20:1–20:21, 2017. [Online]. Available: http://doi.acm.org/10.1145/2996185

[5] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 233–240. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143874

[6] T. Dunning and E. Friedman, *Practical Machine Learning: Innovations in Recommendation*, 1st ed. O'Reilly Media, Inc., 2014.

[7] eMarketer, "Us programmatic ad spending forecast: Most mobile display and video ad dollars to be automated by 2018," Sep. 2016.

[8] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Sine cosine optimization algorithm for feature selection," in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug 2016, pp. 1–5.

[9] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. Candela, "Practical Lessons from Predicting Clicks on Ads at Facebook," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ser. ADKDD'14. New York, NY, USA: ACM, 2014, pp. 5:1—-5:9. [Online]. Available: http://doi.acm.org/10.1145/2648584.2648589

[10] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware Factorization Machines for CTR Prediction," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, 2016, pp. 43–50. [Online]. Available: http://doi.acm.org/10.1145/2959100.2959134

[11] J. Langford, "Vowpal wabbit," 2016. [Online]. Available: https://github.com/JohnLangford/vowpal_wabbit/wiki

[12] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad Click Prediction: A View from the Trenches," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 1222–1230. [Online]. Available: http://doi.acm.org/10.1145/2487575.2488200

[13] T. Parlar and S. A. zel, "A new feature selection method for sentiment analysis of turkish reviews," in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug 2016, pp. 1–6.

[14] S. Rendle, D. Fetterly, E. J. Shekita, and B.-y. Su, "Robust Large-Scale Machine Learning in the Cloud," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1125–1134. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939790

[15] R. Rosales and O. Chapelle, "Attribute Selection by Measuring Information on Reference Distributions," *Tech Pulse Conference, Yahoo!*, 2011.

[16] L. Shan, L. Lin, C. Sun, and X. Wang, "Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization," *Electronic Commerce Research and Applications*, vol. 16, pp. 30–42, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1567422316000144

[17] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 257–297. [Online]. Available: http://link.springer.com/chapter/10.1007/978-0-387-85820-3{_}8

[18] A. Szwabe, P. Misiorek, and M. Ciesielczyk, "Evaluation of Tensor-Based Algorithms for Real-Time Bidding Optimization," in *Intelligent Information and Database Systems: 9th Asian Conference, ACIIDS 2017, Kanazawa, Japan, April 3-5, 2017, Proceedings, Part I*, N. T. Nguyen, S. Tojo, L. M. Nguyen, and B. Trawiński, Eds. Cham: Springer International Publishing, 2017, pp. 160–169. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-54472-4_16

[19] A. Szwabe, P. Misiorek, and M. Ciesielczyk, "Tensor-Based Modeling of Temporal Features for Big Data CTR Estimation," in *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery*. Springer International Publishing, 2017. [Online]. Available: http://ncn6788.cie.put.poznan.pl/images/ncn6788_bdas2017.pdf

[20] W. C.-H. Wu, M.-Y. Yeh, and M.-S. Chen, "Predicting Winning Price in Real Time Bidding with Censored Data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1305–1314. [Online]. Available: http://doi.acm.org/10.1145/2783258.2783276

[21] W. Zhang, T. Du, and J. Wang, *Deep Learning over Multi-field Categorical Data*. Cham: Springer International Publishing, 2016, pp. 45–57. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-30671-1_4

[22] W. Zhang, S. Yuan, J. Wang, and X. Shen, "Real-Time Bidding Benchmarking with iPinYou Dataset," *CoRR*, vol. abs/1407.7, 2014. [Online]. Available: http://arxiv.org/abs/1407.7073