# Identifying Repeat Buyers by Ensemble Learning with Historical Behavioral Features

Shaohua Jiang[1], Yunlei Mu[2], Qingyu Fan[1], Zhipeng Wang[3], Kang Liang[4], Yingjie Cai[1]
Peng Han[1], Yulei Niu[1], Zhiwu Lu[*1], and Ji-Rong Wen[1]

[1]School of Information, Renmin University of China, Beijing, China
[2]College of Computer, Hangzhou Dianzi University, Hangzhou, China
[3]FreatOut Inc., Tokyo, Japan [4]Alibaba Inc., HangZhou, China

## Abstract

IJCAI-15 Repeat Buyers Prediction Competition is about identifying loyal customers for online stores such as Tmall.com. The main task of this competition focuses on predicting customers' future behaviors after analyzing their activity logs in the past. In this paper, the solutions of our Linkin Park Team are described in detail. In the first stage, we built a feature engineering method based on deeply analyzing the historical behaviors of buyers, and then utilized regression and classification methods as indiviual models to solve the repeat buyers prediction problem. Individual models are finally combined by stack learning to achieve better results. In the second stage, we further improved our feature engineering method to extract more effective behavioral features, and simply made use of Logistic Regression to achieve the final results. Our Team ranked #9 at the first stage of the competition and #4 at the second stage.

## 1 Introduction

Merchants tend to run big promotions on particular dates (e.g. Boxing-day Sales, "Black Friday" or "Double 11 (Nov 11th)"), in order to attract a large number of new buyers. Unfortunately, many of the attracted buyers are only one-time deal hunters, and these promotions may have little long-term impact on sales. To solve this problem, it is important for merchants to identify who can be converted into repeated buyers. By targeting on these potential loyal customers, merchants can greatly reduce the promotion cost and enhance the return on investment (ROI). It is well known that in the field of online advertising [Dalessandro *et al.*, 2014; Ghosh *et al.*, 2015; Tang *et al.*, 2015], customer targeting is extremely challenging, especially for fresh buyers. However, with the long-term user behavior log accumulated by Tmall.com, we may alleviate this problem to some extent.

IJCAI-15 Repeat Buyers Prediction Competition aims to promote applications of advanced techniques from AI research to solve the above problem. Contestants are assumed to have access to vast amount of long-term user behavior log data provided by Tmall.com, the largest B2C platform in China. The task is thus to predict the probability that the new buyers would purchase items from the same merchants again within next 6 months. As compared to most of the other AI competitions in the past, this competition appears to be quite different. Firstly, a large-sale promotion data set is provided for public usage. Secondly, a free distributed computation platform is prepared for top 50 teams from the first-stage competition. Thirdly, a great opportunity for winners to finally deploy their algorithms online.

The competition consists of two stages. In the first stage, a data set of 54,901,871 user behavior logs from around 200,000 users were used for training, and another data set of similar size for testing. Similar to other competitions, we are permitted to extract any features and perform training with arbitrary tools. The prediction results of the testing set are finally submitted for evaluation. In the second stage, the top 50 teams from from the first-stage competition have the opportunity to work on a double size of data set on Alibaba's cloud platform. We need to submit our codes in JAVA (different from the first stage), which will be run automatically in a distributed computation manner on the cloud platform.

The standard evaluation criterion used for IJCAI-15 Repeat Buyers Prediction Competition is the Area Under the ROC Curve (AUC) [Bradley, 1997; Fawcett, 2006; Flach *et al.*, 2011; Kaymak *et al.*, 2012]. According to [Bradley, 1997], this ranking measure is defined as the area under the curve obtained by plotting the specificity against the sensitivity. Specifically, sensitivity is computed as the percentage of correctly classified positives, whereas specificity is computed as the percentage of correctly classified negatives.

In the first stage, we used an ensemble solution in which we chose Gradient Boosting Regression Tree (GBRT) [Friedman, 1999] as the main algorithm. A total of 381 features were extracted from the training data and then integrated into the GBRT models. We also used Multi-instance Factorization Machines (MFM) and Random Forests (RF) [Breiman, 2001] to build our ensemble model as illustrated in Figure 1, where the three models GBRT, MFM, and RF are combined together by stack learning [Smyth and Wolpert, 1999]. In the second stage, we further improved our feature engineering method and simply made use of Logistic Regression (LR) [Chapelle *et al.*, 2014] to achieve the final results. A total of 1,139,045 features were extracted from the training data and

[*]The corresponding author. Email: luzhiwu@ruc.edu.cn.

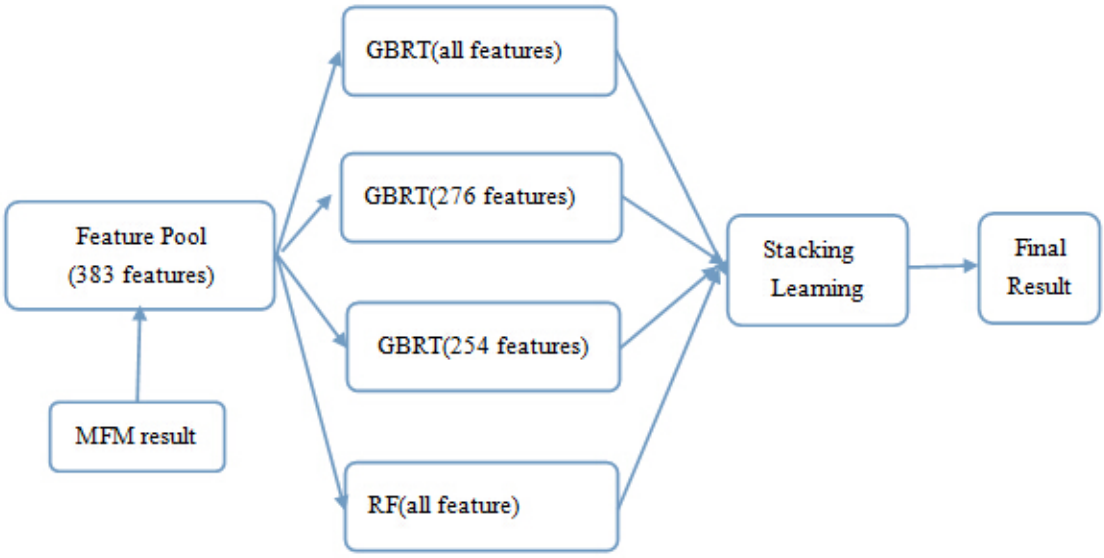


Figure 1: The framework of our ensemble model for repeat buyers prediction.

| Data format | File Name | Schema |
|---|---|---|
| data format 2 | train_format2.csv | user id,age_range gender,merchant id label,activity_log |
| | test_format2.csv | user id,age_range gender,merchant id label,activity_log |
| data format 1 | user_log_format1.csv | user id,item id cat id,seller id brand id,time_stamp action_type |
| | user_info_format1.csv | user id,age_range gender |
| | train_format1.csv | user id,merchant id label |
| | test_format1.csv | user id,merchant id label |

Table 1: Schema of the dataset's files

then integrated into the LR models. Our Linkin Park Team finally ranked #9 at the first stage and #4 at the second stage, which shows the effectiveness of our feature engineering and ensemble learning methods for repeat buyers prediction.

## 2 Data Analysis and Preprocessing

In this section, we firstly analyze the data in the first stage, and then introduce dataset splitting for offline validation, as well as data transforming. Since the data in the second stage is similar to that in the first stage, we take almost the same data analysis method.

### 2.1 Data Analysis

The dataset is provided in two formats, named as data format 1 and data format 2, respectively. Both data formats contain anonymous users' shopping logs in the past 6 months (from May 12 to Nov 10) which include data before and on the "Double 11" day, as well as the label information indicating whether they are repeated buyers for the next 6 months (from Dec.12 to May 11 of the next year).

The schema of data format 2 is shown in Table 1. Values of the label field are {0, 1, -1, NULL}, where '1' denotes that 'user id' is a repeat buyer for the specific 'merchant id', while '0' denotes the opposite. Moreover, '-1' represents that 'user id' is not a new customer of the given merchant and thus out of our prediction, while 'NULL' occurs only in the testing data, indicating that it is used for prediction. Field 'activity_log' is a set of interaction records between {user id, merchant id}. Data format 1 contains four files, which are listed in Table 1. This data format is more user-friendly for feature engineering. The four files are extracted from data format 2, but the records with label = -1 are ignored and thus may cause information missing. Values of 'action_type' are {0,1,2,3}, where 0 is for click, 1 for add-to-cart (cart), 2 for purchase (buy) and 3 for add-to-favorite (favor).

We chose data format 1, but also considered the records with label = -1, which is thus named as data format 3. With this data format, we can exploit more information about users and merchants. For convenience, we changed the field name of seller id to merchant id. Statistics of data format 1 and data format 3 are shown in Table 2. Moreover, we counted records of different action_type in different time periods: the whole time period (from May 12 to Nov 11), the day of 'double 11'. We also counted average records per day before 'double 11' for comparison (see Table 3). We found that the records of different action_type of 'double 11' occupy prodigious proportion, far larger than daily average before 'double 11'. We also found that, 'cart' is much sparser than 'click' and 'buy', and thus is less important for the competition. The overall ratio between 'click' and 'buy' is about 16:1.

| Data format | # Pairs of (user,merchant) | #User | #Merchant | #Item | #Cat | #Brand |
|---|---|---|---|---|---|---|
| data format 1 | 4215979 | 126474 | 4995 | 800322 | 1542 | 7936 |
| data format 3 | 14052684 | 424170 | 4995 | 1090071 | 1658 | 8444 |

Table 2: Statistics of data format 1 and data format 3

| | #Click | #Cart | #Favour | #Buy | Total |
|---|---|---|---|---|---|
| All-Period | 48550712 | 67728 | 3277708 | 3005723 | 54901871 |
| 11-11 | 9188150 | 12621 | 1220285 | 156450 | 10577506 |
| Daily Average Before 11-11 | 224928 | 299 | 11181 | 15485 | 240594 |

Table 3: Records of different action_type

### 2.2 Offline Evaluation

Due to the fact that the labels of test data (T) (from Nov. 12 to May 11 of the next year) are unknown, we cannot build an offline test set based on user historical behaviors. However, since the labels of the training set (V) are given explicitly and its size is 260,864, we use 1/5 of the training set (nearly 52,000 samples) as the offline test set (OT), and the rest 4/5 (nearly 208,864 samples) as the offline training set (OV). We use OV to train models (also tune the parameters), and use OT to validation. As for the online prediction process, we use V to train models and use T for online evaluation.

### 2.3 Data Transforming

Data transforming is necessary. We have observed some wrong records of '11-12' time_stamp. It is officially said that if a user clicks the item at 23:59 on Nov 11, it may be recorded as Nov 12. Therefore, we regard these records as noisy data and modify their time_stamp from '11-12' to '11-11'. Besides, we have also observed some item id belonging to numerous brand id, which can be considered as noisy data too. We set the smallest brand id for these item id to offset the effect of the noisy data. We also add columns as days and weeks by counting days and weeks from current time to '05-12' for easier feature selection.

## 3 Models and Platform

Two types of individual models were implemented: regression and classification. Random forests (RF) is used as a classification model, while Logistic Regression (LR), Gradient Boosting Decision Tree (GBRT) and Multi-instance Factorization Machines (MFM) are used as individual models. According to the characteristics of different models, we develop different Feature Engineering methods.

### 3.1 LR

Logistic regression is widely deployed in large-scale state-of-the-art prediction systems [Chapelle *et al.*, 2014]. The model scales well with feature dimensions and can encode interrelations between different information channels by feature conjunctions. The convexness of its lost function makes the learning procedure robust to noise, makes incremental model updates feasible, and also makes parallelization easy to implement. Logistic regression is also preferred for its robustness against sub-sampling [Maalouf and Trafalis, 2011] and sparseness gained by well designed online learning algorithms [McMahan *et al.*, 2013].

### 3.2 RF

RF is a perturb-and-combine technique [Breiman, 2001], specifically designed for trees. This means that a diverse set of classifiers is first created by introducing randomness in the classifier construction. The prediction of the ensemble is then given as the averaged prediction of individual classifiers. RF performs well in many Kaggle Competitions because of its simplicity, predictive power and ability of avoiding overfitting. We use it in the first stages of the competition.

### 3.3 GBRT

GBRT [Friedman, 1999] can produce competitive, highly robust, and interpretable procedures for both regression and classification. GBRT is used in a variety of areas including Web search ranking and ecology. Besides, due to its insensitivity to long-tailed distributions and outliers, GBRT is also suitable for our e-commerce business scenarios. Since all trees tend to be robust against the addition of irrelevant input features, GBRT is an ideal method for building feature engineering. So we use it as our main model in the first stage. However, GBRT seems unable to handle large-scale sparse features well because its computation and storage requirements increase rapidly with respect to the number of features. Hence, we do not use it in the second stage.

### 3.4 Multi-Instance Factorization Machines

In the first stage of the competition, we employ GBRT to predict the probability of repeat buyers based on all the designed features. The designed features include user features, merchant features, conjuncted features, and high-level features. In this section, we introduce a sub-module in our system, multi-instance factorization machines, which is used to generate one of the high-level features. This method places factorization machines [Rendle, 2010] in a multi-instance framework [Maron and Lozano-Pérez, 1997]. The output of this sub-module is the probability that one user will be a repeater buyer to one merchant, and the predicted probability occupies one dimension of all the features.

Current state-of-the-art methods for user behavioral prediction often rely on careful feature engineering on logs of user activities. Instead of tedious feature engineering efforts, we employ multi-instance factorization machines, which aims

to identify critical logs of all raw logs and model relationships between different feature fields. This sub-module alone achieves an AUC of 0.689, and enhances the overall performance when it is fused in the whole system.

During predicting the probability of one user's being a repeat buyer to one merchant, all the logs of this user's activities are used as a bag of instances. Here, each instance is just a raw log item with an additional flag (whether this log item is recorded at the merchant or not). So the probability of a repeat buyer to a merchant depends on all the user's logs at (or not at) the merchant.

Factorization machines are employed to model one instance's positive contribution to a repeat buyer as defined in Eq. (1), with $M(\mathbf{x})$ being defined in Eq. (2):

$$\Pr(y(\mathbf{x}) = 1; \mathbf{w}, \{\mathbf{v}\}) = \frac{1}{1 + e^{-M(\mathbf{x})}} \quad (1)$$

$$M(\mathbf{x}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=i+1}^{D} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (2)$$

The negative contribution is defined as $\Pr(y(\mathbf{x}) = -1; \mathbf{w}, \{\mathbf{v}\}) = 1 - \Pr(y(\mathbf{x}) = 1; \mathbf{w}, \{\mathbf{v}\})$.

The probability of a non repeat buyer is defined by all instances contribute negatively, as in Eq. (3):

$$\Pr(y(B) = -1) = \prod_{\mathbf{x} \in B} \Pr(y(\mathbf{x}) = -1) \quad (3)$$

Then the probability of a repeat buyer is defined as $\Pr(y(B) = 1) = 1 - \Pr(y(B) = -1)$.

The probability of all users being repeat buyer or not at their related merchants is defined in Eq. (4):

$$\arg \max_{\mathbf{w}, \{\mathbf{v}\}} \prod \Pr(y(B^+) = 1) \prod \Pr(y(B^-) = -1) \quad (4)$$

Stochastic gradient descent is used to learn the weights which maxmize Eq. (4). The wel-known L-BFGS [Liu and Nocedal, 1989] is then employed to further tune all the weights with $L_2$-normalization.

### 3.5 Distributed Computation Platform of the Second Stage

In the second stage, the competition sponsor provides a distributed computation platform to perform feature extraction and machine learning algorithms. The platform uses Alibaba's open data processing service (ODPS) as basic computation platform, and uses graph as feature extraction framework. Xlib algorithm package provides the implementations of logistic regression and gradient boost decision tree.

The graph framework splits the data into 10 workers randomly. There are 10 vertexes in each worker. Then the framework performs feature extraction programs on every vertex. It makes feature extraction parallel well.

## 4 Feature Engineering

In this section, our work on feature engineering is described in detail. Firstly, we introduce the baseline features provided by the official. Secondly, we split the features into two categories: information description features and dummy features. Moreover, we also apply feature processing on some information description features.

### 4.1 Baseline Features

The official provides a set of features for LR as baseline. These features are built as following. Firstly, the top 5 similar partners are found for each merchant. The similarity is simply defined as the normalized number of overlapping costumers. Furthermore, for a (user,merchant) pair, the feature is defined by the activities of the user at merchant $m_1, m_2, m_3, m_4, m_5$, where $m_i$ is one of the top 5 similar partners of the specified merchant. Finally, merchant id is also used as features. For example, there are 5,000 dimensions corresponding to 5000 merchants, and only the value of specified merchant is set to one, while the others to zeros.

### 4.2 Information Description Features

Table 4 shows the details of the information description features. These features can convey some information about relationship between users and merchants, and then profile users and merchants well. We split these features into three parts for each sample (user,merchant): pair features, user features and merchant features. Horary granularity can be categorized as overall features throughout all period and features in a monthly time base. In column 'Period' of Table 4, we label the features with monthly partition as M and all-period features as A. The three types of statistics are counting, ratio and lifecycle. Counting features are basic statistics on various granularities. Ratio features are generated by division between counting features, and the lifecycle features describe the lifecycle of users' action.

### 4.3 Dummy Features

For categorical features such as user gender, user age, merchant id, brand id and merchant id, we apply one hot coding. This means that we denote these categorical features using a series of sparse vectors (the corresponding dimension value is 1, while the rest are 0).

In the baseline method, we apply one hot coding to merchants with similar customers. We also apply one hot coding to the following features:

- The user's age and gender;
- The brand id, category id and item id which the user purchased at the merchant;
- The brand id which the user repeatedly purchased at the merchant.

### 4.4 Feature Processing

It should be noted that logistic regression is sensitive to features' dimensions. Hence, for non-linear features, we transform them primarily using log transformation, which can avoid huge difference of dimensions between features. We also inverse the lifecycle features to scale the day count.

## 5 Blending and Ensemble

In this section, we first introduce our blending and ensemble approach to aggregate the models mentioned in Section 3 for performance improvements. We then introduce a post-processing method to obtain better results.

| Part | Type | Description | Period |
|---|---|---|---|
| Pair | Counts | action count | A&M |
| | | action day count | A&M |
| | | action item count | A&M |
| | Ratios | action count/action day count | A&M |
| | | action item count/action count | A |
| | | action count of every month/all-period action count | M |
| | Lifecycle | first and last active day | A |
| | | the length of active span | A |
| User | Counts | action count | A&M |
| | | action day count | A&M |
| | | action merchant count | A&M |
| | | action merchant count which action day over 2,5 or action month over 2,3 | A |
| | Ratios | action count/action day count | A&M |
| | | click,cart,favor count/Buy count | M |
| | | action merchant count/action count | A |
| | | percentage of action count of every month | M |
| | | percentage of merchant which action day over 2,5 or action month over 2,3 | A |
| | Lifecycle | first and last active day | A |
| Merchant | Counts | action count | A&M |
| | | action user count | A&M |
| | | action user count which action day over 1,5,10 or action month over 1,2,3 | A |
| | | action day count | A&M |
| | | sold brand count | A |
| | | sold item count | A |
| | Ratios | action count/action user count | A&M |
| | | action user count/action day count | A&M |
| | | buy user count/click,cart,favor count | A |
| | | percentage of user count of every month | A |
| | | percentage of user whose action day over 1,5,10 or action month over 1,2,3 | A |

Table 4: Details of information description features (In column 'Period', features with monthly partition are labeled as M, and all-period features labeled as A).

### 5.1 Model Combination

We firstly use MFM's results as features of our feature pool. Because the scores of MFM are lower than those of GBRT and RF, it may not wise to simply merge MFM to the final prediction model. However, it does hold the advantage different from both GBRT and RF, and thus adding MFMs results as features of our feature pool may be a good choice. Without surprise, it improves our score by nearly 0.0018, making our score raising from 0.700936 to 0.702711.

To train GBRT model, three feature sets of feature pool are used. We also make use of the while feature set to train RF model. The sizes of the feature sets are listed in Table 5. We then utilize the stacking learning method [Smyth and Wolpert, 1999] to combine these models together by regarding their outputs as the features of a simple Linear Regression. Since linear regression has strong generalization ability, it can avoid overfitting. The model combination framework is also illustrated in in Figure 1.

### 5.2 Post Processing

According to the results of offline evaluation, we find that the predicted probabilities of some users are obviously higher than the average level. In fact, these users are only active on the date of "double 11", and then are not the loyal customer to any merchant. Hence, we lower their probabilities by multiplying an adjustment factor $f$. In our experiments, we directly set the value of $f$ as 0.9.

## 6 Experimental Results

In this section, we evaluate our solutions for IJCAI-15 Repeat Buyers Prediction Competition in both of the two stages. In our experiments, we split 1/5 of the training set to generate an offline test set and the others to generate an offline training set for modeling training and parameter tuning. The results of individual models at the first stage are listed in Table 5, while the top 10 teams at the second stage are listed in Table 6. In particular, Table 5 shows the effectiveness of ensemble of individual models, while Table 6 shows the effectiveness of our stronger feature engineering method.

## 7 Conclusions

In this paper, we present our solutions for IJCAI-15 Competition: Repeat Buyers Prediction after Sales Promotion. We establish an efficient feature system based on historical behavior logs to describe the characteristics of the data. In the first stage, We mainly use three kind of individual models: GBRT, RF and MFM. We use MFM's results as features and

| Model | GBRT | GBRT | GBRT | RF | Final Model |
|---|---|---|---|---|---|
| Feature Set Size | 383(all) | 276 | 254 | 383(all) | mix |
| AUC | 0.702711 | 0.702645 | 0.702585 | 0.701084 | 0.70319 |

Table 5: The results of individual models for repeat buyers prediction at the first stage.

| Rank | Team Name | AUC |
|---|---|---|
| 1 | hrem | 0.711373 |
| 2 | LeavingSeason | 0.711287 |
| 3 | FAndy&kimiyoung&Neo | 0.710163 |
| 4 | **Linkin Park** (ours) | 0.709877 |
| 5 | OneP | 0.709464 |
| 6 | parameicnm | 0.709070 |
| 7 | 9*STAR | 0.708976 |
| 8 | senochow | 0.706037 |
| 9 | luosu | 0.705180 |
| 10 | Farewell | 0.704467 |

Table 6: Top 10 teams at the second stage.

exploit Stacking Learning to combine GBRT and RF models. In the second stage, We use LR to train with large-scale features. Finally, our Linkin Park Team ranked #9 at the first stage of the competition and #4 at the second stage.

## Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants 61202231 and 61222307, National Key Basic Research Program (973 Program) of China under Grants 2014CB340403 and 2015CB352502, the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China under Grant 15XNLQ01, and Beijing Natural Science Foundation of China under Grant 4132037.

## References

[Bradley, 1997] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, pages 1145–1159, 1997.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Chapelle *et al.*, 2014] Olivier Chapelle, Eren Manavoglu, and Rómer Rosales. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology*, 5(4):61:1–61:34, 2014.

[Dalessandro *et al.*, 2014] Brian Dalessandro, Daizhuo Chen, Troy Raeder, Claudia Perlich, Melinda Han Williams, and Foster J. Provost. Scalable hands-free transfer learning for online advertising. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1573–1582, 2014.

[Fawcett, 2006] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[Flach *et al.*, 2011] Peter A. Flach, Jose Hernandez-Orallo, and Cesar Ferri Ramirez. A coherent interpretation of AUC as a measure of aggregated classification performance. In *International Conference on Machine Learning (ICML)*, pages 657–664, 2011.

[Friedman, 1999] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 1999.

[Ghosh *et al.*, 2015] Arpita Ghosh, Mohammad Mahdian, R. Preston McAfee, and Sergei Vassilvitskii. To match or not to match: Economics of cookie matching in online advertising. *ACM Transactions on Economics and Computation*, 2015.

[Kaymak *et al.*, 2012] Uzay Kaymak, Arie Ben-David, and Rob Potharst. The AUK: A simple alternative to the AUC. *Engineering Applications of Artificial Intelligence*, pages 1082–1089, 2012.

[Liu and Nocedal, 1989] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1-3):503–528, 1989.

[Maalouf and Trafalis, 2011] Maher Maalouf and Theodore B. Trafalis. Robust weighted kernel logistic regression in imbalanced and rare events data. *Computational Statistics & Data Analysis*, 55(1):168–183, 2011.

[Maron and Lozano-Pérez, 1997] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10*, pages 570–576, 1997.

[McMahan *et al.*, 2013] H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. Ad click prediction: a view from the trenches. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1222–1230, 2013.

[Rendle, 2010] Steffen Rendle. Factorization machines. In *IEEE International Conference on Data Mining series (ICDM)*, pages 995–1000, 2010.

[Smyth and Wolpert, 1999] Padhraic Smyth and David Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1-2):59–83, 1999.

[Tang *et al.*, 2015] Jian Tang, Ping Zhang, and Philip Fei Wu. Categorizing consumer behavioral responses and artifact design features: The case of online advertising. *Information Systems Frontiers*, pages 513–532, 2015.