Boost Clickbait Detection based on User Behavior Analysis

Hai-Tao Zheng *, Xin Yao, Yong Jiang, Shu-Tao Xia, and Xi Xiao

Tsinghua-Southampton Web Science Laboratory,
Graduate School at Shenzhen, Tsinghua University, China
{zheng.haitao,yaox14,xiast,jiangy,xiaox}@sz.tsinghua.edu.cn

Abstract. Article in the web is usually titled with a misleading title to attract the users click for gaining click-through rate (CTR). A clickbait title may increase click-through rate, but decrease user experience. Thus, it is important to identify the articles with a misleading title and block them for specific users. Existing methods just consider text features, which hardly produce a satisfactory result. User behavior is useful in clickbait detection. Users have different tendencies for the articles with a clickbait title. User actions in an article usually indicate whether an article is with a clickbait title. In this paper, we design an algorithm to model user behavior in order to improve the impact of clickbait detection. Specifically, we use a classifier to produce an initial clickbait-score for articles. Then, we define a loss function on the user behavior and tune the clickbait score toward decreasing the loss function. Experiment shows that we improve precision and recall after using user behavior.

1 Introduction

Nowadays, headline of news or online article is inclined to be titled more attractively in order to attract more click. Various strategies such as building suspense, sensation, luring and teasing are used to make the title attract users' click. Users have different tendencies for the articles with such a clickbait title. Some of users hate these kinds of articles and think they are fooled. Others are willing to browse these kinds of articles. Hence, it is important to identify whether an article has a clickbait title and block them in the information stream for the specific users.

Existing methods only take text feature and meta-data (source, url feature etc.) into account. There is not a great difference between clickbait and non-clickbait in the body. Hence, traditional text classification methods have little effect in clickbait detection. Although the title text looks helpful for our task, length of the title is too short to make results reliable. As for meta-data, it just offers little help in our experiment. We do not have enough cues to finish our task well from the information of article itself. However, user behavior provides us with extra evidence. Users have different tendencies for articles with clickbait

* corresponding author

title. The effect of the classifier is enhanced by analyzing user behavior based on the above assumption.

In this paper, we firstly use a classifier using a serial of features to produce an initial clickbait score for every article. We considers the effect of a clickbait title as the residual between click-through rate (CTR) predictor and real value. Then we learn the residual from real data. We define a loss function based on user behavior. At last, we minimize the loss function by tuning user interest and article clickbait-score. Experiments show that our strategy is effective. Our main contributions are:

1. we present a model combined with user behavior for clickbait detection.
2. we define a loss function on user behavior and tune clickbait score of article toward decreasing the loss function.
3. we conduct a series of experiments to verify effectiveness of our method.

The rest of the paper is organized as follows. Section 2 gives an overview of the related areas. Section 3 introduces our method Section 4 introduces our experiments. At last, we summarize our work in Section 5.

2 Related Work

Recently, as the development of information stream application, clickbait detection has attracted some interest. Potthast etc. [13] studies clickbait detection in Twitter. They make use of three kinds of features: teaser message, linked web page and Twitter meta information. Teaser message and content of linked web page in their work are similar to title and body of article in our work. They use three kinds of classifier respectively (logistic regression, naive Bayes and random forest) with these features.

Chakraborty etc.[4] compare clickbaits and non-clickbaits from different angles. They detect clickbait mainly based on topical similarity and linguistic patterns.

Biyani etc.[2] also purpose a serial of features similar to the above paper. None of them considers user behavior.The above researches mainly focus on textual features.

Yimin Chen etc.[5] discus what kinds of cue would help clickbait detection. They also believe lexical and syntactic features are helpful. Besides, they propose that image analysis and user behavior analysis, which we mainly discuss in this paper, may be useful. However, there are not any experiment to prove their idea in the paper.

3 Clickbait Detection Based on User Behavior

In this section, we will introduce our method. Firstly, we establish model to estimate whether or not an article is clickbait. The model gives an article an initial click-bait score. We introduce this part in Section 3.1. Secondly, we model the effect of clickbait title on the users click as the residual between the real action and Click-Through Rate of our CTR predictor. We introduce this part in Section 3.2. At last, we define a loss function and tune the score by minimizing our defined loss function to product a final clickbait score. We introduce this part in Section 3.3.

3.1 Initializing Clickbait Score for Articles

Firstly, we train a classifier to produce an initial clickbait score for each article. Intuitively, the more reliable initial clickbait score is, the better result will be. Thus, we train a classifier as well as possible. In order to select suitable features, we refer to the related work [11][10][12][1]. A complete description of all features is shown in Table 1.

Table 1. Description of features. Source T,B,M imply that the feature is extracted from the title, the body and the mixture of them respectively. Feature categories N,B,VEC denote numeric, binary and vector respectively.

Feature Name	Description	Source	Type
{Has,Num}exclm	Presence/Number of exclamation mark	T	B/N
{Has,Num}ques	Presence/Number of question mark	T	B/N
Haspron	Presence of pronoun	T	B
Hasint	Presence of interrogative	T	B
Numwords	Number of words	T	N
Numdots	Number of dots	T	N
Stoprate	Stop words ratio	T	N
Unigram	tf-idf weight of words	T,B	VEC{N}
Bigram	tf-idf weight of bigrams	T,B	VEC{N}

We compare with the five well-known machine learning algorithms including Logistic Regression (LR) [3], Naive Bayes (NB) [9], Random Forest (RF) [7], Support Vector Machine (SVM) [6], and Gradient Boost Decision Tree (GBDT) [8]. We choose GBDT to produce initial clickbait score based on the experiment due to the best performance. After training, we initial the clickbait score of an article as output of the classifier.

3.2 Fitting Residual Error

The click-through rate prediction should take thousands of, even millions of features into account. Many factors influence whether a user clicks an article. We have had a ready-made predictor, which use 20 thousands-dimension features based on the personas and attributes of articles. We model the effect of a clickbait title as the residual between the real click and the predicted value given by original CTR predictor. We call CTR predicted by original CTR predictor original CTR (a given value in our dataset). We define real CTR as the modified CTR after considering clickbait factor. The residual is equal to real CTR minus original CTR as equation 1.

$$residual = realCTR - originalCTR \tag{1}$$

We build a logistic regression model with a linear scaling to fit the residual between the real click and the predicted value. Two factors could help us fit the residual: the user interest in clickbait title and the clickbait-score of article. We choose three combinations of them as features:

1. The Product of User Interest and Clickbait-score. It is easy to assume that users tend to click the article with a clickbait title if he/she is interested in clickbait. This value will be large on the above condition.
2. Difference between Clickbait-score and User Interest through Rectified Linear Function. A user with a low interest is almost impossible to click an article with a clickbait title. This feature will be large, when this happens. But a user with a high interest can still click an article with a normal title. Therefore, we add a rectified linear function to handle this situation.
3. The Clickbait-score itself. Our statistics shows that the articles with a clickbait title gain more traffic. Therefore, we choose it as one of our features.

The input of the model is the three features above. The output indicates the impact of features on click (1 for positive impact and 0 for negative impact). Logistic regression model is as equation 2.

$$p_r(click|pair_i) = (1 + e^{-w^T x})^{-1} \tag{2}$$

where $pair_i$ is a pair of an article and a user, w is a four-dimensional vector including weights of the above three features and a bias term.

The output value of logistic regression model is between 0 and 1. It does not meet our demand. Thus, we take the probability through the linear scaling to be the residual as equation 3.

$$residual = k(p_r(click|pair_i) - 0.5) \tag{3}$$

where k is scaling parameter, which indicate how important is the effect of a clickbait title.

Now the real click-through rate is represented as the standard click-through rate plus residual as equation 4.

$$p(click|pair_i) = p_b(click|pair_i) + residual \tag{4}$$

where $p_b(click|pair_i)$ is the standard CTR of $pair_i$.

3.3 Tuning Clickbait Score based on User Behavior

We define the loss function as equation 5.

$$L(I, C; Pair) = \sum_{i}^{m}(y_i - p(click|pair_i))^2 + \alpha(\sum_{x \in C}(x - x^2) + \sum_{y \in I}(1 - y)^2) \tag{5}$$

where I is the user interest vector, C is the clickbait-score of articles vector, is a weighting parameter, m is the size of training set and Pair is a serial of records of user action. Each record is represented as a quadruple {user_id, article_id, stardard CTR, action (1 for click and 0 for only-view)}.

The loss function consists of two terms. The first term is the first sigma symbol. It is an empirical risk term and is represented as a square loss. It define the loss of user behaviors. The second term is used to control the distributions of clickbait score and user interest. The role of first term is easy to understand. We will explicate the role of second term in experiment section.

In order to get more reliable clickbait-score, we minimize the loss function by gradient descent. Firstly, we fix the clickbait-score vector and calculate gradient of interest vector and clickbait score vector respectively as equation 6 and equation 7.

$$\frac{\partial L}{\partial I} = 2k\sum_{i}^{m}(y_i - p(click|pair_i))p_r(click|pair_i)'_I + 2\alpha\sum_{y\in I}(1-y) \tag{6}$$

$$\frac{\partial L}{\partial C} = 2k\sum_{i}^{m}(y_i - p(click|pair_i))p_r(click|pair_i)'_C + \alpha\sum_{x\in C}(1-2x) \tag{7}$$

Then we update interest vectors and clickbait score by gradient descent algorithm.

4 Experiments

4.1 Dataset

We have two kinds of datasets: the article sets and the user behavior record sets. The article set are crawled from four major website portals(Sohu, Tencent, Netease and Sina), WeChat public accounts and some professional BBS.

There are two article sets. The first dataset contains 32,037 articles (7,461 clickbait and 24,576 non-clickbait) which are manually annotated if the article with a clickbait title. It is used for training and testing the basic model. The second contains 11,193 annotated articles (2,638 clickbait and 8,555 non-clickbait). It is user for testing the entire model. The two article sets are crawled in different periods.

The user behavior record set is the behavior records of 12,451 relatively active users. We also have two user behavior sets, which is the corresponding user record in two article sets. The first one has 5,688,369 records. It is used for training and testing the residual predictor. The second one has 6,254,448 records. It is used for testing the entire model.

4.2 Classifier Model Selection

To address the issues of imbalances between the positive examples and the negative examples in first dataset, we randomly select 7,461 examples from all non-clickbait examples as negative examples. We randomly select 80% examples as training set. The remaining 20% is used for testing. We compare the five well-known learning algorithms as implemented in sklearn using default parameters.

Table 2. AUC of CTR preditor of different k.

	LR	NB	RF	GBDT	SVM
Precision	**0.75**	0.71	0.72	**0.75**	0.71
Recall	0.76	0.77	0.74	**0.81**	0.80

The experiment results can be found in Table 2. From the perspective of comparing classifier, GBDT gets the best performance in general. GBDT also performs well in recent kaggle competition. One of the reason is that GBDT has a better anti-noise capability than the other methods above. Our datasets come from real world and inevitably include noise.

4.3 Training the Residual Predictor

We divide the first record set into three parts. The first part is used for initializing user interest. We set user interest to the proportion of clickbait in articles that user click recently. The second part is used for training residual predictor. We train the residual predictor based on Section 4.1. The third part is used for evaluating and choosing an appropriate parameter k. We compare the effects of different k for $p(clickpair_i)$. Table 3 shows the ROC curve and area under roc curve (AUC) value with different k.

Table 3. AUC of CTR preditor of different k.

k	0.00	0.02	0.04	0.06	0.08
AUC	0.7258	0.7484	0.7581	**0.7603**	0.7581

Experiments demonstrate that clickbait title has a nonnegligible impact on users click. Although the increase of AUC is not very significant, it is enough for us. Our goal is not to improve AUC, but to model the effect of clickbait title on users click. Thus, such a performance is enough for us. There is only a little difference between different parameter k. AUC reaches a maximum, when k is equal to 0.06. Therefore, we set k equal to 0.06 in the following experiments.

4.4 The Effect of User Behavior

We choose the result of the GBDT classifier with all features as the initial clickbait-score due to the best performance. The basic classifier gets 75.85% precision and 80.76% recall in article set 2. We also initialize user interest with the proportion of clickbait in articles that user click recently.

Table 4. Performance with different alpha.

Alpha	Initial precision	Initial recall	Terminate precision	Terminate recall
0	0.7585	0.8076	/	0.0000
2	0.7585	0.8076	0.9535	0.6846
4	0.7585	0.8076	0.8127	0.8203
6	0.7585	0.8076	0.7680	0.8097

From the Table 4, Initial precision and initial recall are evaluated by the basic classifier in article set 2. Terminate precision and terminate recall are the value that precison and recall reach a stable point in iteration. We can see that precision gradually reaches 1.0 and recall falls dramatically if we do not add the second term into the loss function. Users usually click many non-click articles whether they are interested in clickbait or not. User interest decline because of this behavior. Then all clickbait scores decline. Clickbait scores and user interest will have a further decline in next iteration. Thus, precision increases and recall decreases gradually at the beginning. At last, all articles are classified as non-clickbait. Therefore, we need to add a term into the loss function to control the distributions of user interest and clickbait score. The term $(1-y)^2$ make it easy to increase and hard to decline for user interest. Besides, the term $(x-x^2)$ is close to 0, when clickbait score is close to 0 or 1. It is close to maximum, when clickbait score is close to 0.5. This term make it hard to change their result of classification for articles with explicit result and stop the trend of overall deviation toward high clickbait score and user interest.

We can see that the distribution control term play a part role when alpha is equal to 2. However, it is not enough. In this condition, a large proportion of articles are classified as non-clickbait. It results in a high precision and low recall. When alpha is equal to 4, we reach the best performance.Precision increase 0.05 to 0.8127 and recall increase 0.01 to 0.8203 comparing with initial precision and recall. The performance indicate that our model confirm to the real situation. As alpha increases, precision and recall are almost constant and keep the original level. This is because the weight of distribution control term is too large, which make the cost too much to change categorized result. Thus, articles tend to keep the original result.

5 Conclusion and Future Work

In this paper, we look for a new way to help us achieve the higher precision and recall. We propose a method that models user behavior and conduct relative experiments. The main innovation of our algorithm is the following two ideas:

1. we model the effect of a clickbait title as the residual between the real click and the predicted value from standard click-through rate predictor.
2. we tune the clickbait score based on user behavior.

Experiments show that we increase 0.05 precision and 0.01 recall comparing with the methods not considering user behavior by utilizing user behavior. It demonstrates that user behavior indeed is useful for clickbait detection. Besides, our method can integrate with any other clickbait detection method. The only need is to put the result of other methods as the initial clickbait score in our method.

Ackonwloedgements This research is supported by National Natural Science Foundation of China (Grant No. 61375054), Natural Science Foundation of Guangdong Province Grant No. 2014A030313745, Basic Scientific Research Program of Shenzhen City (Grant No. JCYJ20160331184440545), and Cross fund of Graduate School at Shenzhen, Tsinghua University (Grant No. JC20140001).

References

1. Abbasi, A., Chen, H.: A comparison of fraud cues and classification methods for fake escrow website detection. Information Technology and Management 10(2), 83–101 (2009)
2. Biyani, P., Tsioutsiouliklis, K., Blackmer, J.: "8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality (2016)
3. Cessie, S.L., Houwelingen, J.C.V.: Ridge estimators in logistic regression. Journal of the Royal Statistical Society 41(1), 191–201 (1992)
4. Chakraborty, A., Paranjape, B., Kakarla, S., Ganguly, N.: Stop clickbait: Detecting and preventing clickbaits in online news media. In: Ieee/acm International Conference on Advances in Social Networks Analysis and Mining. pp. 9–16 (2016)
5. Chen, Y., Conroy, N.J., Rubin, V.L.: Misleading online content: Recognizing clickbait as "false news". In: ACM on Workshop on Multimodal Deception Detection. pp. 15–19 (2015)
6. Cortes, C., Vapnik, V.: Support-Vector Networks. Kluwer Academic Publishers (1995)
7. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. Machine Learning 45(1), 157–176 (2004)
8. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
9. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338–345 (1995)
10. Kolari, P., Java, A., Finin, T., Oates, T., Joshi, A.: Detecting spam blogs: A machine learning approach. In: National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, Usa (2006)
11. Lahiri, S., Mitra, P., Lu, X.: Informality Judgment at Sentence Level and Experiments with Formality Score. DBLP (2011)
12. Ntoulas, A., Najork, M., Manasse, M., Fetterly, D.: Detecting spam web pages through content analysis. In: World Wide Web Conference. pp. 83–92 (2006)
13. Potthast, M., Kpsel, S., Stein, B., Hagen, M.: Clickbait detection (2016)