Outbrain Click Prediction

Ankit Saxena, Samvat Rastogi, Shailendra Patil

Abstract
In today's world, Internet is omni-present. From finding a meaning of a simple word to operating a remote machine, Internet helps us in all. As we are advancing in the technology, our need from the technology is changing day by day. We are using technology to make our daily task simpler to more intelligent. One of those task is content recommendation. A content recommendation system is an intelligent system which helps to recommend appropriate content to the users based on their activities like browsing patterns on Internet. Outbrain is one such organization which pairs the relevant content with readers in about 250 billion personalized recommendation every month.
The purpose of our project is to help Outbrain to improve its recommendation system by predicting which contents are likely to be clicked by its global user base. Hence, as our goal, we created a prediction model using the train data to predict whether the user will click a particular content. We trained the model with train data with important attributes like platform, advertisement id and display id from meta data to complete the prediction. Since data is relational in nature, these huge data sets could be joined on a basis of single key resulting into data set with required data or with required features. For the model, we calculated the probability of an advertisement to get clicked from a particular platform on the basis of which we could determine the likeliness of the click on the advertisement.

Keywords
Data Mining — Kaggle — Prediction — Probability

Contents

Introduction

Today, marketing and selling a product is far more easier than old days. Thanks to Internet. Due to Internet, a product can be advertised and purchased from any corner of the world in the matter of few seconds. Moreover, the product in the advertisement can be anything from a simple pen to an insurance policy. This means anything can be advertised on the Internet, be it a product or a service. But the major challenge is to advertise the correct product or service to the target user. Because advertising a shampoo to a bald guy will never help the shampoo company to make a profit. So, how can this accuracy be achieved?

Outbrain is an advertising company which uses behavioral targeting to recommend articles, products, services, photos, videos to the user. Behavioral Targeting is a technique which allows Outbrain to increase the efficiency of its advertisement network by user web browsing behavior information. According to Wikipedia [1], "behavioral targeting uses information collected from an individual's web-browsing behavior (e.g., the pages that they have visited or searched) to select advertisements to display". Because of this technique, Outbrain's recommendations can be found on more than 35,000 websites and it serves over 250 million recommendations and 15 billion page views per month. It reaches over 87% of US Internet users.

1. Background

The problem has been picked up from Kaggle. The challenge stated in the problem is to predict which advertisement are likely to be clicked.

1.1 Data

As this is a Kaggle Problem [2], the dataset was downloaded from Kaggle [3]. The dataset contains a sample of users' page views and clicks on multiple publisher sites in the United

States between June 14^{th} June, 2016 to 28^{th} June, 2016. It contains numerous sets of content recommendations served to a specific user in a specific context. Each context (i.e. a set of recommendations) is given a display_id. In each such set, the user has clicked on at least one recommendation. The identities of the clicked recommendations in the test set are not revealed. This is a very large relational dataset.

1.1.1 Data Fields

Each user is represented by a unique id, namely uuid. A user can view any web page with content, which is denoted by an id (document_id). Each document has a set of ads (ad_id). Each of these ads belong to a campaign (campaign_id) run by an advertiser (advertiser_id). Dataset also provides meta data about the document like mentioned entities, categories, topics etc. Files used for doing further computation are:

1. events.csv
2. clicks_train.csv
3. clicks_test.csv

clicks_train.csv is the training set, showing which set of ads was clicked. This file has total of 87141731 rows. It has no missing values.

Table 1. clicks_train.csv

Column	Distinct Values	Missing	Data Type
display_id	16874593	0	text
ad_id	478950	0	text
clicked	2	0	text

clicks_test.csv is the same as clicks_train.csv, except it does not have the clicked ad. Each display_id has only one clicked ad. This dataset contains display_ids from the entire dataset time frame. It has a total of 32225162 rows with 6245533 distinct display_ids and 381385 distinct ad_id.

Table 2. clicks_test.csv

Column	Distinct Values	Missing	Data Type
display_id	6245533	0	text
ad_id	381385	0	text

events.csv provides information on the display_id context. It covers both the train and test set. This dataset has 23120126 rows.

Table 3. Events.csv

Column	Distinct Values	Missing	Data Type
display_id	23120126	0	text
uuid	19794967	0	text
document_id	894060	0	text
timestamp	22896622	0	text
platform	4	0	text
geo_location	2988	801	text

1.1.2 Data Preprocessing

Data Preprocessing plays a very important role in any data mining problem. This problem have real world data which is collected from users from various geographies. Since data is huge in size, it is highly possible that some of the data is missing or distorted. Table 1-3 shows an overview of Data. There are no missing values for click_train.csv and clicks_test.csv. However, there are 340 missing values and 461 blank values in events.csv for geo_location. But since the data is very huge and total defective rows are much less than 0.05% of total data, we can remove those rows to sanitize the data.

It is quite noticeable that each data set has very limited number of features and are dependent on the features available in different data set. Moreover, data is relational in nature. In the sample submission file, only display_id and ad_id are required. So, in order to obtain the appropriate output, clicks_train and events data were merged, on the basis of display_id and platform. This means that data is joined using **Left Outer join**. The feature, platform, was used because browsing device can actually change the browsing behavior of a user as some websites are not mobile compatible or optimized.

2. Algorithm and Methodology

This problem mainly deals with prediction of user clicks. Hence, to predict the same, we implemented the solution using **regularized probabilities**.

Regularization refers to a process of introducing additional information in order to solve an ill-posed problem or to prevent over fitting. Mathematically, a regularization term $R(f)$ is introduced to a general loss function:

$$\min_f \sum_{i=1}^{n} V(f(\hat{x}_i), \hat{y}_i) + \lambda R(f)$$

for a loss function V that describes the cost of predicting $f(x)$ when the label is y , such as the square loss or hinge loss, and for the term λ which controls the importance of the regularization term. $R(f)$ is typically a penalty on the complexity of f, such as restrictions for smoothness or bounds on the vector space norm [4].

For this, we combined the ad_id from clicks_train and platform from events into a single key, and the probability for each ad_id corresponding to a display id was calculated. And we also calculate the probability of total advertisement clicked, so that if at all there is any ad_id which was not trained in the training data, then we make the probabilities of such entries equal to the probability of advertisements clicked. This is known as Smoothing. Using these probabilities we predict the probabilities for the given display and ad_id in test data.

2.1 Training of the model

In order to train the model, we selected the events and clicks_train data sets and merged them using a left outer join. Then, the columns of ad_id and platforms are joined together

in order to form a unique key, say ad_id_pl. Now, the probability of all ad_id corresponding to its display_id is calculated and also the total probability of ads being clicked i.e. 0 or 1 is calculated. For this training data, the total probability of ad being clicked is 0.1936454 as there are 16874593 advertisements that were clicked. These probabilities were calculated using regularization, where we tuned the model by applying penalty for extreme cases.

Further, these calculated probabilities were used to predict the probabilities of the test data corresponding to their ad_id and if there is any untrained data point present then 0.1936454 was used as its probability. Hence, as per the requirement of Kaggle, the ad_id obtained were sorted in ascending order according to its probability and then grouped by display_id.

3. Experiment and Result

3.1 Hardware used

All the experiments were performed on a machine with two Intel Xeon E5-2650 v2 8-core processors with 32GB of memory, available at Indiana University, Bloomington. We have used R version 3.3.0, R Studio version 0.99.902 for the coding involved in this project.

3.2 Experiment

As data was provided by Kaggle, the complete data set is divided into two major parts of train and test. The train data consists of all the ad_id with respective display_id and a click flag. The test data consists of same type of data apart from the clicked flag.

The standard technqiue used by Kaggle, Mean Average Precision (MAP) for precision evaluation is being used to compare the models.

Mean Average Precision (MAP) is a number metric which gives the measure of performance of an algorithm. MAP is an average of precision values of all points where relevant information can be found. The average precision values at the ranks of relevant information are found. This measure is very different from precision as precision only considers a single threshold in the rank and average precision considers the entire ranking, assigning it a quality number to it. But while executing, it gives more importance to the things which happens in the early stages of ranking. This means that the information retrieved at rank 1 is much more important than rank 2 [5].

Mathematically, average precision is defined as below:

$$\text{ap@n} = \sum_{k=1}^{n} \frac{P(k)}{\min(m,n)}$$

where m are the total edges in any graph and n is the maximum number of nodes, a user is likely to guess. And, The mean average precision for N users at position n is the average of the average precision of each user, i.e.,

$$\text{MAP@n} = \sum_{i=1}^{N} ap@n_i/N$$

Here, we have leveraged $MAP@12$ algorithm in Kaggle, which is defined as

$$\text{MAP@12} = \frac{1}{|U|} \Sigma_{u=1}^{|U|} \Sigma_{k=1}^{min(12,n)} \text{P(k)}$$

where $|U|$ is the number of display_ids, P(k) is the precision at cutoff k, n is the number of predicted ad_ids [6] [7].

3.3 Results

On evaluating, the performance of our algorithm with regularization parameter of 18, we received a MAP score of **0.63853** and initial ranking of **121**.

4. Summary and Conclusions

The given data was tremendously huge and required proper prior knowledge before taking any step. Hence it is very important to study the data very well. We were able to predict the likelihood of user clicks using very less features to a certain extent of accuracy. This project helped us a lot in gaining insight about handling large set of data, importance of infrastructure dependability and preprocessing of data. In the end, we were able to secure a score of **0.63853** and initial ranking of **121** (As on Dec 14^{th}, the rank was 189).

4.1 Future Work

We would like to extend this project further till the actual deadline on Kaggle. As we mentioned, only few features were used to create model. Hence, there is possibility to create more accurate model using other features. We would further like to consider creating model based on logistic regression, SVM and neural network as we believe we can achieve better results with those techniques.

Acknowledgments

We have put in considerable efforts to reach this stage of the project but it would not have been possible without the continuous guidance and support of many other individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to our Professor Mehmet Dalkilic, for his supervision as well as for providing the necessary resources required for the completion of this project report.

We would also like to thank our Associate Instructor Hasan Kurban for his help and constant support throughout our coursework.

References

[1] *Behavioral targeting*. Wikipedia.

[2] *Outbrain Click Prediction*. Kaggle, 2016.

[3] *Data - Outbrain Click Prediction*. Kaggle, 2016.

[4] *Regularization (mathematics)*. Wikipedia.

[5] Victor Lavrenko. *Mean Average Precision*. Youtube, 2014.

[6] Mikel Bober. *Mean Average Precision*. Kaggle, 2016.

[7] *Outbrain Click Prediction: Evaluation*. Kaggle, 2016.