Outbrain click prediction project report

Harshit Krishnakumar (2000114341), Manasa Makam (2000112843), Sushmita Sivaprasad (2000101429)

Abstract
On-line marketing is a sophisticated and inexpensive form of advertising that has extensive reach. It has evolved through time to be one of the most influential methods compared to the traditional means. The objective of this project is to create an active engagement between the audience and the ads displayed on a web page. We aim to create a ranking system for a set of ads the relevancy between the web content and the nature of advertisements.

Keywords
Outbrain Click Prediction, ad recommendation engine, click through rate

[1] *Computer Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*

Contents

1 Introduction 1
2 Background 1
3 Algorithm and Methodology 2
3.1 Exploratory Analysis of Data 2
3.2 Click Through Rate Calculation 3
3.3 Relevance Of Documents 3
Overlap Measure • Goodall Measure • Relevance Score
4 Experiments and Results 3
5 Conclusions and Future Work 4
Acknowledgments 5
References 5

1. Introduction

Internet marketing is a fast growing platform and businesses have been evolving at a fast pace to embrace and adapt to this domain of marketing. Internet is a major platform for collecting data and generating consumer insights in real time, which leads to the evolution of new methods to improve the customer engagement. It has been analysed that an average American spends about 23 hours per week surfing on-line, which shows us that there is an enormous potential which can be tapped by pointing the right advertisements to the right audience.

Display ads fall under the umbrella of inbound marketing, and it has been estimated that the average cost to implement inbound marketing through Internet is about 143$ which is approximately half of that required by outbound marketing ($373)[1]. Historically, banner ads have known to generate 0.1 percent average click through rate in the US, it thus helps to improve the profit margins by displaying the right set of ads to the target audience.

In the scope of this project, we are trying to find the strength of relevancy between a web page content and a cohort of ads that can be promoted on the page. This relevancy factor indicates the probability of an ad being clicked on a particular web page. To illustrate with an example, Figure 1 shows a web blog with scoops and news from the auto industry. The blog talks about a new launch by Mercedes and it hosts different ads on this page out of which there are relevant ads about other models of Mercedes and Land Rover but we also see an ad about gold investment which is not so relevant to the blog content. A user who is interested in reading this blog is less likely to click on an ad related to gold. We developed a ranking system for the ad recommendations based on this relevancy using document category, document topic and document entity. In the given example, category could be auto industry, topic could be cars and entity could be Mercedes. These attributes are compared between web page content and promoted content.

2. Background

Data given by the Kaggle competition ”Outbrain Click Prediction” has seven data sets. Every user is given a unique id called ’uuid’, a group of recommendations is given a display_id which is specific to a page and user. Every recommendation is given an id called ’ad_id’. Publisher_id is the id given to the website hosting the ad. Every context is given a document_id, which will map to 3 dimensional tables that give information about category, entity and topic. To make the terms clear, there is an example shown in Figure 2. This is paid content recommended by Outbrain to CNN.com, here CNN is the publisher, Technology is the document, display_id id given for the set of recommendations, each of the ads paid content is ad_id.

In the context of ad recommendations, we can implement Self Learning Algorithms which can be used to identify users and personalize the ads based on their browsing and purchase

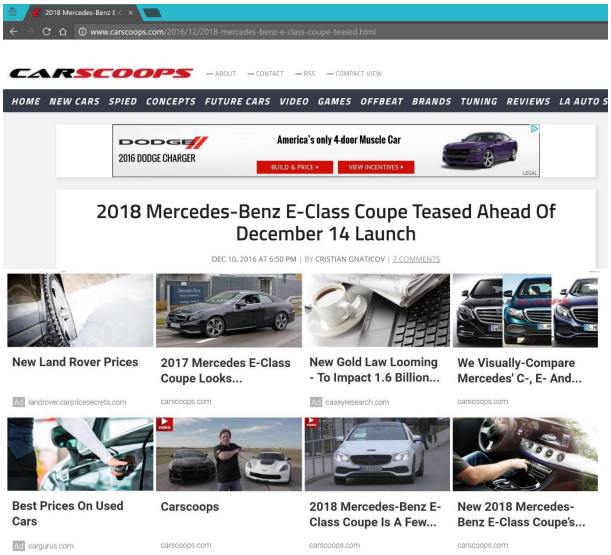

Figure 1. An example of promoted ads on a web page

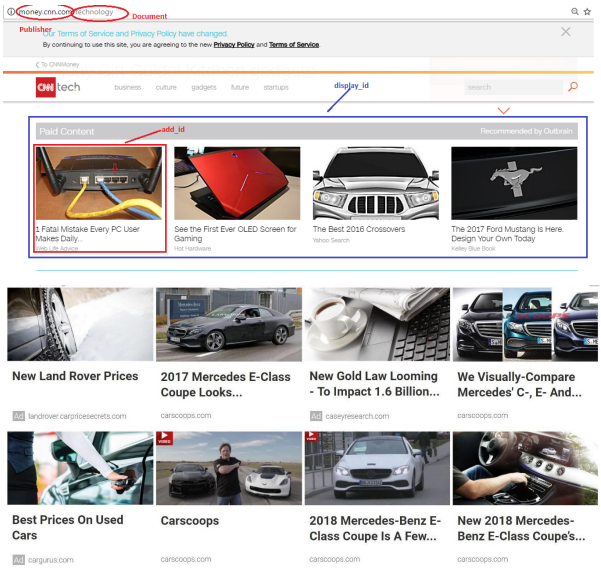

Figure 2. Outbrain Recommendations to CNN

history. Google Adwords uses a similar algorithm to find the most relevant occurring keyword by Frequent Pattern Mining over a large set of documents and their relevant industries. The ads shown are charged when the targeted group of audience click on the ad. This also reduces the total ad cost spent by the company in advertising their product. The advertiser who wishes to advertise their product first chooses their keywords that are related to their website and the product they are advertising. When a person searches for any particular keyword, the Google Adword search engine will determine the position the ad is displayed on based on a combination of the ad's relevance and in case there are multiple advertiser competing on the same keyword, and the order of the ad placement is based on the second highest bidder. The fee that is charged per click is taken as 1% more than the cost to have the ad displayed on that page. In most of the ad recommendation engines, Click Through Rate is used as a major contributor to the Quality score[2].

$$\text{Click Through Rate} = \frac{\text{Number of clicks}}{\text{Number of Impressions}} \quad (1)$$

Another common method used in ad recommendation is the "fp-growth data mining algorithm". It is used for frequent pattern mining to identify keywords related to every industry. Taking the descriptions of the companies within a certain industry, local competitors and companies within the similar domain we apply fp-growth algorithm on it to find out the most frequently occurring keywords. This helps in tagging documents and searches to a particular industry

3. Algorithm and Methodology

3.1 Exploratory Analysis of Data

Data provided by Kaggle consists of 7 tables and one of them (page_views) consists of more than two billion records. We loaded this data into Python to get an idea of the granularity and relationships between the tables. Upon exploratory analysis, we arrived at the data model given in Figure 3.

Detailed analysis of each table in the data is provided below:

1. **Page Views** contains data about all users and every page that they visited, irrespective of whether an ad was displayed or not. This table has UUID, Document ID, timestamp of visit, platform on which the document was visited and geographical location of the user

2. **Events** table maps every document where an ad was diaplayed to a diaplay_id. Display ID can be joined with clicks_train table to get the ads which were displayed. Display ID is termed as the "context" of the document and user

3. **Clicks Train / Test** table is the most important table in this project, since it gives the mapping from display_id to ad_id and has a flag column which gives if the ad was clicked or not. Every display ID has a set of ads which were displayed on a document, and one of the ads were clicked. Ad Ids in this table are repetitive but display IDs are not, which means that the same Ads are shown on multiple pages, whereas display ID is unique for a document and user instance. Clicks_test table is simiar to cicks_train, except it will not have the clicked flag.

4. **Promoted Content** has information that describes an Ad. It maps to a document_id which is a unique ID for the document which the Ad is promoting. It also has other information like campaign_id and advertiser_id

5. **Documents Meta** contains information about all the documents (web pages) that users viewed. It gives the source, publisher and publish time. More importantly, it acts as a bridge between the other document reference tables described next

6. **Documents Entities, Documents Topics, Documents Categories** are document reference tables that describe the entities, topics and categories that a document talks about. Each document to category/topic/entity mapping has a confidence score, which shows the strength of a mapping.

Joins on these tables are shown in Figure 3. Page Views is the master table which has data about all the pages, and events is a subset of pages where an ad was displayed. These tables can be joined on uuid and document ID. Events table can be joined to clicks_train table to map every Ad to its parent document on which it is displayed. Further, clicks_train table can be joined with promoted_content table to get the document information which the ad is about. This is the page where user will land if he/she clicks on the Ad. Every table where document_id is present can be joined to either of the document reference tables to get more information describing the document.

3.2 Click Through Rate Calculation

Click through rate is a metric which describes how well an ad is performing based on previous data. It is the ratio of the number of clicks by the number of impressions. It indicates how relevant and useful users find an ad in a particular context.

We can get Click Through Rate in this data from the clicks_train table by counting the number of times an Ad was displayed and the number of times an Ad was clicked using the clicked flag.

3.3 Relevance Of Documents

The objective of our project is to give a ranking to the recommendations based on the probability of it being clicked. Our approach to the problem is to find out a relevancy score for each of the recommendations in a display id based on category, topic and entity of promoted content and the page which is promoting it. This is equivalent to finding the similarity of the documents and there are some popular methods to find similarity measures like Overlap measure and Goodall measure[4].

3.3.1 Overlap Measure

Overlap measure is a basic method, which gives a score of 1 when categories of two documents match and 0 when they do not. A score is given to get relevance strength by combining these scores, by taking weighted average or regular average or direct summation

3.3.2 Goodall Measure

Goodall method uses a measure that normalizes the similarity between two attributes by considering the probability that the similarity value observed could be observed in a random sample, which means that it assigns low similarity for a match that is less frequent than the one with more frequency.

3.3.3 Relevance Score

Both Overlap and Goodall methods assume either a complete match or complete mismatch of two objects, but in this case we are given a confidence level to the category, entity and topic. Confidence level of 0.80 indicates that a document can belong to a particular topic with 80% confidence, which is not a 1 or 0 comparison. So we used a modified approach to calculate a relevancy score.

$$\text{Relevance Score} = (CL\_x + CL\_y) \times (CL\_x \times CL\_y) \quad (2)$$

CL_x = Confidence level with which host website falls under a matching category
CL_y = Confidence level with which promoted

Lets take an example where a display_id has three ad_ids and each of them fall under different categories and the web page which is hosting also falls under different categories with some confidence level. When the categories of objects match-, sum of confidence levels is multiplies by the confidence levels to get a score that represents the strength of relevancy. This is shown in Figure 4. Figure 4 is a simple sample data, display_id 1 has five recommendations, each of these belong to different categories with different confidence levels and each of these categories are given confidence levels based on the web page content it is hosted in. We have tried different functions and finally decided this function

This function gives sensible results in the above example, ad_id 1 belong to category 3 with 80% CL and the hosting website falls in the same category with 90% CL which is a best match in this data set. It got a rank one, similarly for ad_id 5 there is a poor match and poor scores which has got a rank of 5. We improved this rank further by adding the relevance score calculated using document topics and document entities. The aggregate score is then used to rank, the one with highest score gets the top rank. This function resulted in a ranking system for all the recommendations at display_id level

4. Experiments and Results

The biggest challenge in this project was to handle data efficiently, since we had close to 2 billion records in the page_views table. The total raw data file size was 100 GB, and we tried to import this data into SQL Server 2012, but due to our local system memory constraints, the server crashed every time we tried to run commands on the data.

We applied for access to the Indiana University Super Computer "Karst" system. We accessed the Karst system using ThinLink client remote desktop, and we were able to process this data efficiently by loading into Python.

We had two methods to rank Ads for every display ID. The first one used a simple click through rate for every Ad which was already displayed. We calculated CTR metric from the clicks_train table and assigned the same CTR to all the old ads in clicks_test table. For all the new Ads that we saw in clicks_test table, we assigned an average CTR from the

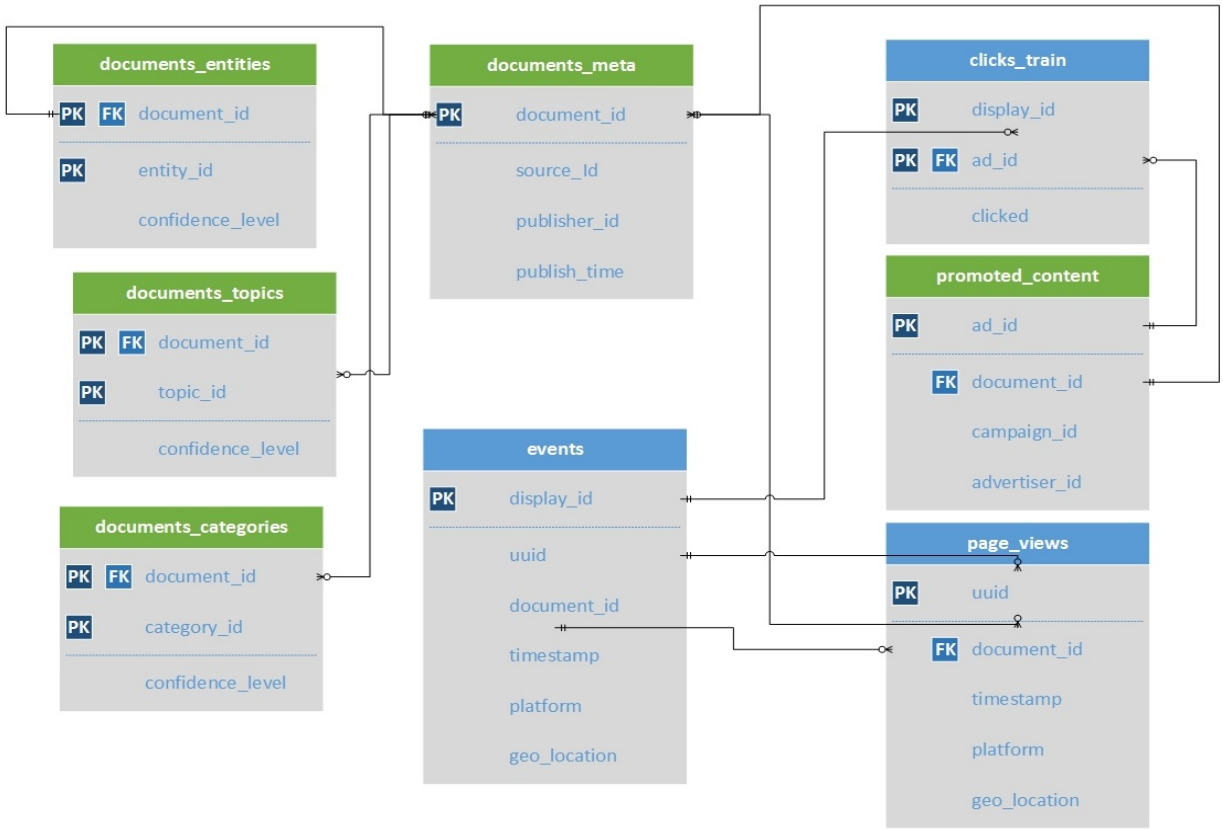

Figure 3. Data Model for Outbrain Click Prediction

clicks_train table. Based on CTR values, we ranked the set of Ads in every display_id.

This method might not be completely reliable, since most of the Ads were displayed less than five times in the entire time frame of data given. This decreases the strength of the CTR we calculate for the Ad. Further, an Ad might have been clicked in a particular context, and a CTR calculated for that context might not be right for another context.

A more compelling method is to find how relevant an Ad is with respect to the document where it is getting displayed. This can be calculated using the Relevance Score described in Section 3.3.3 (Equation 2). This compares the document which the Ad is promoting and the document where the Ad is displayed upon, and calculates the relevance score.

Depending on the Relevance Score for each Ad in a display_id, we rank them based on the increasing order. This method will work for old and new Ads and does not depend on the number of times an Ad was displayed.

5. Conclusions and Future Work

This project was primarily about data exploration and analysis. The challenge was to handle large data-sets efficiently and get an idea of the data and joins. We considered all the free database software like SQL Server Express edition, MYSQL Server and Mongo DB, but because of our local system memory constraints, we were not able to use either one of these. We did the analysis using pre-installed Python software in the Karst super computing system provided at Indiana University.

The current solution we provided for this problem calculates a relevance score between Ad and document. As part of the future work, we can arrive at a more refined solution which weights several aspects and gives an overall score:

1. Click Through Rate
2. A strength score for the CTR, considering the number of times an Ad was displayed in its lifetime
3. Relevance Score
4. Interests of the user, which can be calculated using page_views table, which stores the pages a user has visited in his/her lifetime
5. Geographical Location, to give more location based ads, can be calculated using geo_location column

Acknowledgments

This project was submitted as part of the Data Mining course work (Fall 2016) at Indiana University Bloomington. We would like to extend our thanks and express our sincere gratitude to our advisor/prof Dr. Mehmet Dalkilic for motivating us to pursue this project. We would also like to thank Mr.

		Host Website		Promoted Content			
display_id	ad_id	document_category	category_confidence_level	document_category	category_confidence_level	Function ((CL1+CL2)*CL1*CL2)	Rank
1	1	3	0.9	3	0.8	1.224	1
1	2	5	0.5	5	0.9	0.63	2
1	3	2	0.4	2	0.5	0.18	4
1	4	1	0.8	1	0.4	0.384	3
1	5	7	0.1	7	0.3	0.012	5

Figure 4. Implementation of Relevancy Score in an Example

Hasan Kurban and other AI's for providing us with resources and guiding us through the course.

Appendix

Hereby stating that all work herein is solely ours. We used the following packages in Python 2.7:

- numpy version 1.11.2
- pandas version 0.19.1

We used the IU Super Computer "Karst" with the following specifications to run this program:

- 32 GB ram per node
- 100 GB storage space per node
- 3 PB shared scratch (temporary) storage space
- RedHat Linux Operating System

References

[1] http://maximizebusinessmarketing.com/inbound-outbound-marketing-better-b2b-company.

[2] https://support.google.com/adwords/#topic=3119071

[3] Identifying and Overcoming Common Data Mining Mistakes, Doug Wielenga, SAS Institute Inc., Cary, NC.

[4] Similarity Measures for Categorical Data: A Comparative Evaluation, Shyam Boriah Varun Chandola Vipin Kumar, Department of Computer Science and Engineering, University of Minnesota

Github Link to Source Code

`https://github.com/ManasaMakam/DM/blob/master/Outbrain_Click_Prediction`