Predict Commercial Promoted Contents Will Be Clicked By User

Gary(Xinran) Guo | garyguo@stanford.edu | SUNetID: garyguo | Stanford University

1. Introduction

As e-commerce, social media grows rapidly, advertisements appear everywhere in human daily activities. Ad clickstream data becomes incredibly large and it contains tremendous treasures that we can learn for people's interests, hobbies, and personalities. Giving properly customized recommendations, advertisement, coupons help a business grow faster, and attract more target customers. With huge passion about ad clickstream analysis, I will start with public data set from Ourbain.com, the web's leading content discovery platform delivers these moments or advertisements while we surf our favorite sites. To predict which pieces of recommended content each user will be likely to click on.

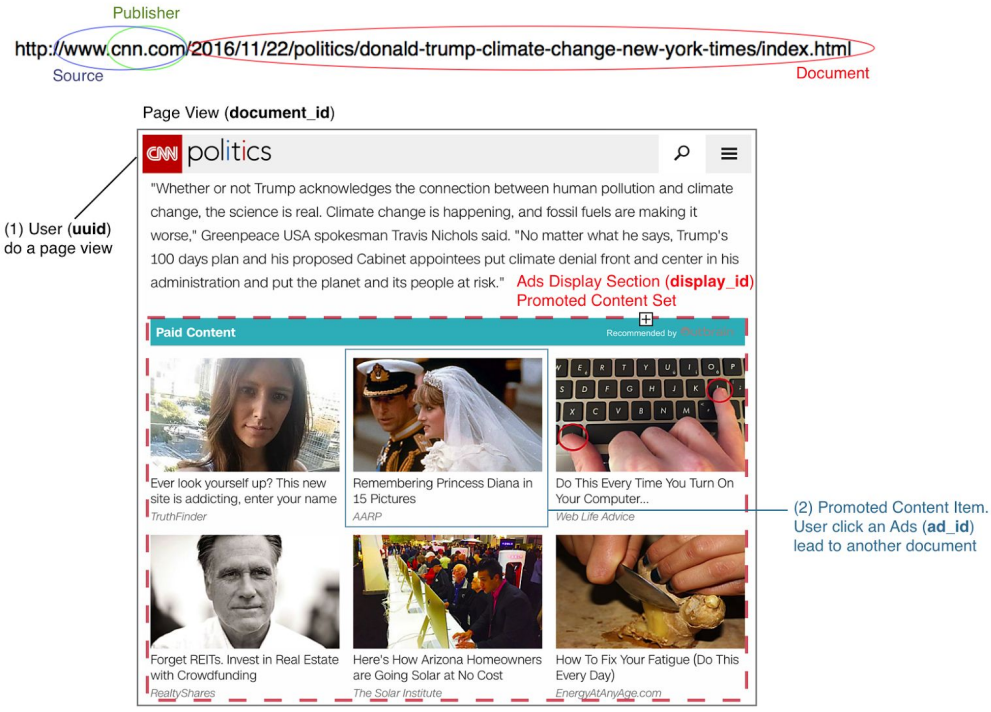

Figure 1. General process of user click ad event

Briefly, as figure 1 shown, whenever users visit a set of web pages, they will be also served by advertisements in many places (refer to Ads display section). The dataset 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 idea is to rank the recommendations in each group by decreasing predicted likelihood of being clicked.

2. Approach

2.1 Dataset Analytics

Data sets will roughly have 10 database tables (~40 GB), which can primarily divide into 3 parts, user clickstream, page document, and promoted content. As shown below in Figure 2,

- **User Clickstream**: the log of users visiting documents, where contains document_id, platform (desktop, mobile, tablet), geo_location, traffic source (internal, search, social)
- **Page Document**: The detail metadata of document include, sources, publisher, topics, entities, and a taxonomy of categories
- **Promoted Content**: The detail metadata of advertisement, that each ad belongs to a campaign (campaign_id) run by an advertiser (advertiser_id)

Start to do exploratory data analysis (EDA), and understand data distribution.

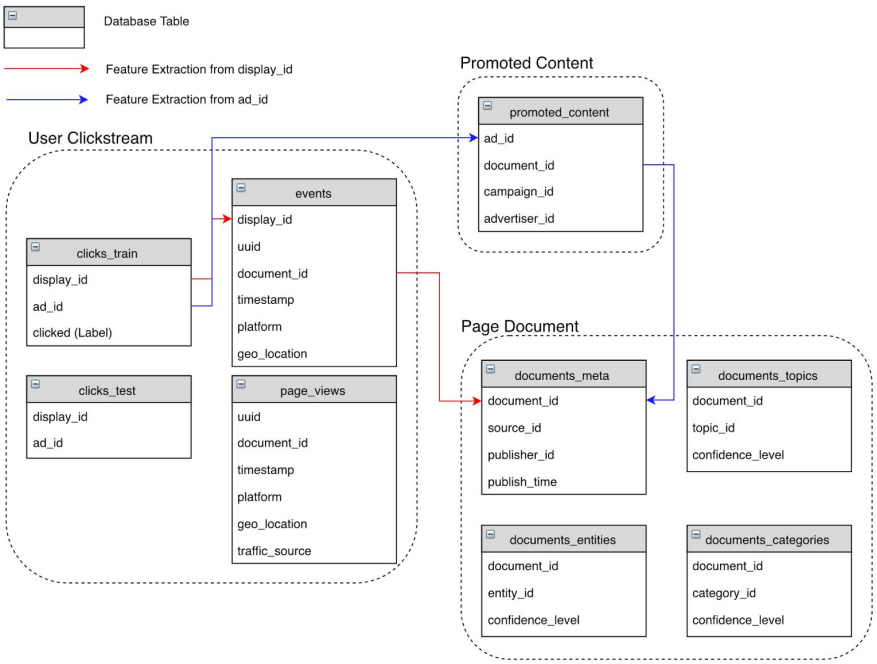

Figure 2: Dataset Structure

click_test and click_train: Each display has a certain number of ads, figure out the distribution of these ads counts. The amount of ads that are very often used, and rare used. It indicates that Ads that appear less than 10 times is 61.74%. The huge number of ads appear just a few times in the training set, with two-thirds having less than 10 appearances. This shows us that I need to predict whether someone will click on an ad not just based on the past data of that specific ad, but also by linking it to other ads.
events: based on analysis of user (uuid), figure out the distribution of uniqueness of user id, I have the result that users that appear less than 2 times is 88.42%. So there will be a little scope for building user-based recommendation profiles here.
Documents_Categories and documents_topics: Outbrain has some content classification algorithms, by looking into the most popular classifications, it shows that a number of topics that appear more than 10000 times are 201, Number of unique categories is 97 and Number of categories that appear more than 1000 times is 88.
Page_View and Event: By analyzing the huge amount of raw user clickstream log, The average page views by users indicates that higher average (2.835) in page views dataset which user view page and not click ads, compared to the events dataset (1.167) with user click ads. After further research, we can see that 65% of the users have only 1 page view, 77%, have at most 2 page views and 89% of the users have at most 5 page_views. We discovered that fields (uuid, document_id, platform, geo_location and day(timestamp)) can be used to effectively join both datasets. And events dataset can be considered to be a subset of page_views dataset, as almost all (99.76%) its rows can be matched in page_views using these fields. 75% of users in events have additional logged visits on page_views.

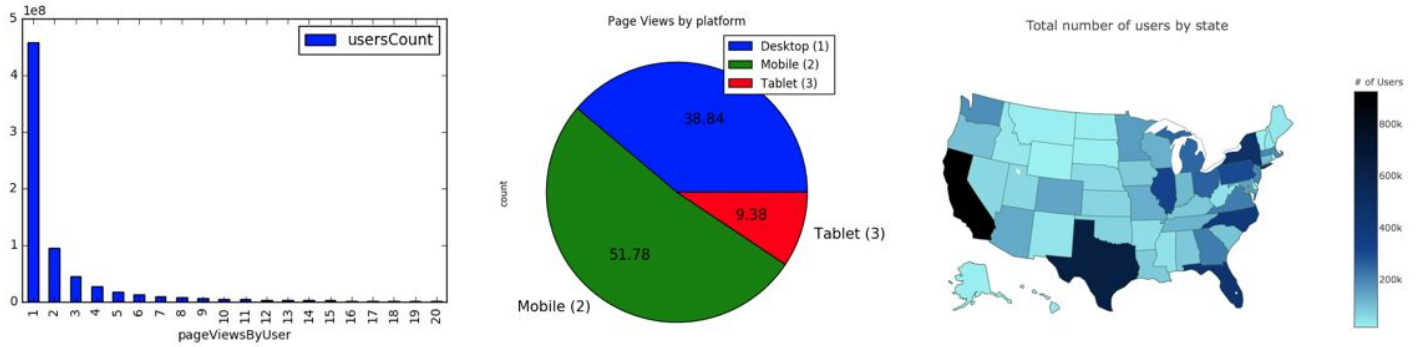

Figure 3. Exploratory Data Analysis For Page View (29.71GB)

2.2 Features Extraction

Basic Features

Based on EDA, user-based recommendation profile is not suitable for click-prediction, instead, document categories and topics have more uniform distribution of data. To keep it simple, I start with selecting features that related to document categories and topics. As shown on figure 2 RED line. Given a training sample on click_train (display_id, ad_id) and label as clicked or not. The events data will tell you what page those ads were embedded in, and the other attributes of how the page was accessed and by whom, so use display_id to find one row with the uuid and the document_id on events dataset. Then follow document_id to retrieve corresponding topics, categories etc. As shown on figure 2 BLUE line, start another direction with ad_id, and get detail document_id by looking promoted_content, then use document_id to retrieve topics, categories from page document. By using these features, I can discover the relationship between the user view page categories and ad categories, how these relate and impact with each other. Similarly, I can select many other document-related features, such as topics, entity, confidence level, source, publisher, campaign, etc.

Advanced Features

After deep exploratory data analysis, get more information about how data set distributed, select most common data features, and compare results. Such as geographic location, platform, traffic resource.
Start with display and get the corresponding document in the event table, then retrieve all history that related to this particular document in page_view table. Calculate sum of view, and distribution of geographic location, platform, traffic sources as new features. For example, from Figure 3, by knowing user page view distribution corresponding geographic location, we can discover a more explicit relationship between data.

2.3 Model & Algorithm

Regularized Click Probability

Start with the most straight model, we only focus on the ad itself, suppose not to consider any other related features. Calculate the click probability of each single ad via full training set, and store into a table. Then move to test data, given a set of ads within a single display section, look up the probability and rank the probability from high to low to produce the result. Algorithm as follow,

```
REG = 10
Defined Probability table, Ad_id as key, Probability as value.
During Training given (display_id, ad_id, clicked):
For each ad (ad_id):
       Probability (ad_id) = train[train.clicked==1].ad_id.value_counts /
(train.ad_id.value_counts + REG)

During Testing given (display_id, ad_id):
Set_of_ad = Testing set group by “display_id”
For each ad in Set_of_ad:
       If ad_id not in Probability table: Probability (ad_id) = 0
       Sort by Probability(ad_id) from high to low
```

The flaw with using per-ad target rate encoding is that ads with a very small number of views can get inflated click probabilities. For example, let's say there was an ad with two views, and two clicks, and there was another ad with 1000 views and 800 clicks. In normal calculation would have a 100% probability, while the second one would only have 80%, and the first ad would be ranked as a high probability prediction. However, the second ad is actually preferred, because it has a track record of being clicked very often, while the ad with two views does not have enough data to say that the true probability of it being clicked is 1. Because of this, adding a term **REG** that penalizes ads with small amounts of data, therefore making it prefer an ad with large amounts of training data and a reliable probability. In the algorithm, you can see adding a fixed amount to the total of each ad (e.g. 10), which will reduce the probabilities of all ads, but will reduce the probabilities of ads with small ads more. If we apply this to the previous scenario, for the first ad we get an adjusted

probability of 2 / (2+10) = 0.166, while the second ad gets an adjusted probability of 800 / (1000+10) = 0.792, therefore ranking the second ad before the first one in the final output. This also means that in a scenario where two ads have the same probability, it will prefer the one with more data to back up the probability.

Logistic Regression with efficient L1-L2-regularization

As an ad will be predicted as a click or not, the key is to calculate the click probability of ads in each display section. I construct standard machine learning binary classification model by using logistic regression. I split click_train dataset as two parts, 90% of data will be used for training and 10% will be used for validation. Base on a related search "Ad Click Prediction: a View from the Trenches" from Google Inc.[1]. I applied the main algorithm as Follow-The-Regularized-Leader - proximal. In short, this is an adaptive-learning-rate sparse logistic regression with efficient L1-L2-regularization.

Let's set up some notation, We denote vectors like $g_t \in R^d$, where t indexes the current training instance; the i^{th} entry in a feature vector g_t is denoted $g_{t,i}$. We can use the following framework to model our problem using logistic regression. On round t, we are asked to predict on an instance described by feature factor $x_t \in R^d$; given model parameters w_t, we predict $p_t = \sigma(w_t \bullet x_t)$, where $\sigma(a) = 1/(1+exp(-a))$ is the sigmoid function. Then we observe the label $y_t \in \{0,1\}$, and suffer the resulting logistic loss, given as $l_t(w_t) = -y_t log(p_t) - (1-y_t) log(1-p_t)$ the negative log-likelihood of y_t given p. The detailed algorithm as follow,

Input: Parameter $\alpha, \beta, \lambda_1, \lambda_2$

($\forall i \in \{1, ..., d\}$), initialize and

For t = 1 to T do

Receive feature vector and let

for $i \in I$ compute

If $|z_i| \le \lambda_1$, then $\omega_{t,i} = 0$

else, $\omega_{t,i} = -(\frac{\beta+\sqrt{n_i}}{\alpha} + \lambda_2)^{-1}(z_i - sgn(z_i)\lambda_1)$

Predict $p_t = \sigma(x_t * w)$ using the $\omega_{t,i}$ computed above observe label $y_t \in \{0,1\}$

For all $i \in I$ do

$g_i = (p_t - y_t)x_i$ # gradient of loss

$\sigma_i = \frac{1}{\alpha}(\sqrt{n_i + g_i^2} - \sqrt{n_i})$

$z_i = z_i + g_i - \sigma_i \omega_{t,i}$

$n_i = n_i + g_i^2$

End for

End for

3. Result & Evaluation

3.1 Prediction Output

Applied different features selection (basic features v.s. advanced features) into above two model (regularized click probability v.s. Logistic-Regression with efficient L1-L2-regularization), I defined the same output format. in output 1, the key will be display_id and ad_id, the value is the probability of that particular ad. I can not set a boundary that an ad with click probability greater than 0.50 treated as clicked because the click probability is much less than 0.50 for most of the ad. I instead constructed output 2, which is group output 1 by display id to get a set of ad that belongs to it, then rank its click probability from high to low. That gives much more clear data presentation to do result validation. As shown below, you can see a sample of output 1 and output 2.

Output 1

display_id	ad_id	clicked (probability)
16874594	66758	0.167213874715
16874594	150083	0.110528833553
16874594	162754	0.247692716539
16874594	170392	0.301802838584
...		

Output 2

display_id	ad_id (Rank probability)
16874594	170392 172888 162754 66758 150083
16874595	8846 143982 30609
16874596	289915 11430 289122 132820 57197
...	

3.2 Prediction Evaluation

By validating with testing data set, gives following prediction accuracy,

Prediction Accuracy	Regularized Click Probability	Logistic-Regression with efficient L1-L2-regularization
Basic Features	0.63854	0.66897
Advanced Features	N/A	0.63269

After applied basic feature to regularized click probability model, where only consider features that related to the ad itself, it gives a great prediction accuracy 0.638, which tells the ad's category and the topic will also be the main part that attracts users' attention. Furthermore, selected more comprehensive related features from both user page document and ad document, (include category, topics, entity, confidence level, source, publisher, advertiser, campaign), and applied into the logistic regression with efficient L1-L2 regularization. It significantly improves prediction accuracy by 3% and reaches highest prediction accuracy 0.66897 in this report. Eventually, I look into more on user click stream on page view log, and pre-calculate more features, such that the distribution of all page views that lead to an ad click. However, the prediction result does not turn out really well. It reduces prediction accuracy even worse than the regularized click probability model, which indicates too many unrelated features will have a negative impact on learning.

4. Discussions

- This project is application based. So the major task is to apply different machine learning models to a practical problem and make a prediction. Features selection and model comparison will be the primary focus of the project.
- Doing exploratory data analysis (EDA) is one of the most important tasks for this project. The cumulative volume of data sets exceeds 40GB. Knowing feature distributions helps significantly on features selection. One mechanism is to use most-frequent feature values. Multiset should be used for feature value counting. Another mechanism is to avoid to generate collinear features as much as possible since collinearity is a major hurt to linear models. In ad data, there are many hierarchical features. One idea is that hierarchical features are grouped together, and there are no quadratic features generated within a group. Grouping, however, is done with eyeballing and basic statistics, such as cardinality. Therefore, model tuning is critical.
- More advanced features are not guaranteed to make a prediction better. Comparing the result of the basic feature, and advanced features, The prediction accuracy is actually decreasing when applying more unrelated features.
- After EDA, finding that the average page views of a single user is just 2.835, it indicates that user-based recommendation profile is not suitable for click-prediction, instead, document categories and topics have uniform distribution among data.
- In Regularized Click Probability model, adding a regularized term REG that penalizes ads with small amounts of data, therefore making it prefer an ad with large amounts of training data and a reliable probability. It improves prediction accuracy from 0.5693 to 0.6072, roughly by 3%.

5. Future Work

- Do more exploratory data analysis to discover more features distributions in the huge amount of data set, such as calculate the mean, sum, standard deviation.
- Join various tables, and select different features to do prediction and compare result, test error.
- Apply different machine learn models, for example, Keras, a deep Learning library for Theano and TensorFlow, high-level neural networks library, will be good start To deeply dig more into user ad click stream (page view & event).
- Changing the parameters of the model in logistic regression algorithm and compare results.

References

- McMahan, H. B., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A. M., ... Davydov, E. (2013). Ad click prediction: a view from the trenches, In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (p. 1222). New York, New York, USA: ACM Press.
- H. Brendan McMahan, Gary Holt, D. 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, Jeremy Kubica. Ad Click Prediction: a View from the Trenches. Pages 2.
- M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th international conference on World Wide Web, pages 521–530. ACM, 2007.
- O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. Transactions on Intelligent Systems and Technology, 2014
- O. Chapelle. Click modeling for display advertising. In AdML: 2012 ICML Workshop on Online Advertising, 2012.