Outbrain: Click Prediction

Julien Hoachuck, Sudhanshu Singh, Lisa Yamada **{juhoachu, ssingh9, lyamada}@stanford.edu**
CS 229: Machine Learning, Stanford University

Motivation

State of **constant information consumption**
Outbrain's mission:
- Increase user's engagement
- Provide more personalized experience

We propose an advertisement **recommendation algorithm** to prioritize content presented to users to provide an **improved user experience**.

Datasets

Outbrain click prediction competition provided **large datasets** (> 100 GB, 2 billion training examples)
- 6/14/16–6/28/16: page views with click labelling (1 if clicked, 0 if not clicked)

Page_views
uuid
document_id
timestamp
platform
geo_location
traffic_source

Events
display_id
uuid
document_id
timestamp
platform
geo_location

Promoted_content
ad_id
document_id
campaign_id
advertiser_id

Clicks_train/test
display_id
ad_id
clicked

Data sets can be mapped to each other with a given key.

Documents_meta
document_id
source_id
publisher_id
publish_time

Documents_entities
document_id
entity_id
confidence_level

Documents_topics
document_id
topic_id
confidence_level

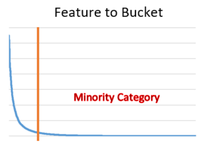
Document_categories
document_id
entity_id
confidence_level

Features

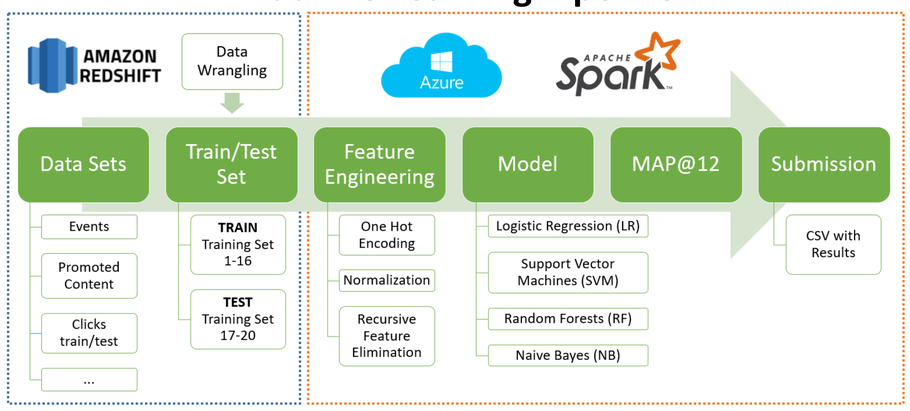
display_id	ad_id	clicked	document_id	platform	timestamp	country	campaign_id	topic_id
538328	263236	0	1807341	2	40530800	PH	20977	89
16465403	221239	0	2765974	3	1.098E+09	US	24776	234
5782925	64255	1	804322	2	384867173	US	8568	1
12959901	479112	0	2111911	3	854135235	CA	32329	137
8924202	172955	0	2342266	2	599239718	GB	20933	52

...

24 features provided + 939 features derived = 963 features
Features were derived using **one hot encoding**:
platform (3), geo_location (10), advertiser_id (926).
Treat as a *categorical* values rather than *numerical* values.



Machine Learning Pipeline



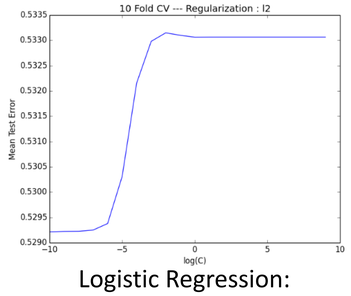
LR
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

SVM
$$\max_\alpha \quad W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$
$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, m$$
$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0.$$

RF
$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x')$$
Regression tree f_b
B = # of samples/trees
b = 1, ..., B

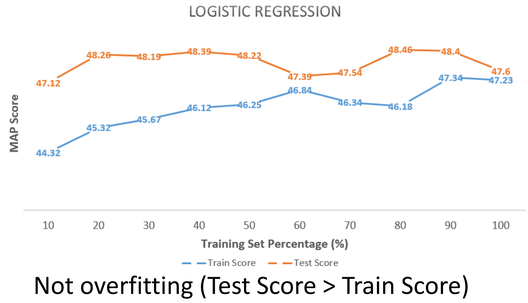
NB
$$\mathcal{L}(\phi_y, \phi_{j|y=0}, \phi_{j|y=1}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$$

Parameter Fitting

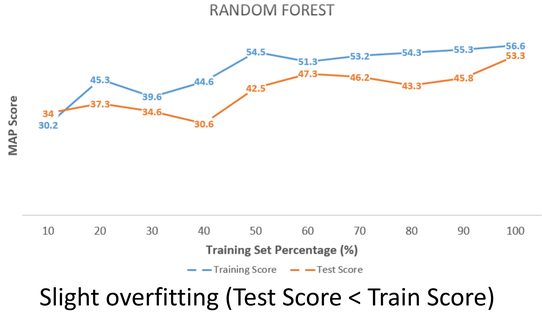


Logistic Regression:
l2 regularization fitting

Learning Curves



Not overfitting (Test Score > Train Score)
80% training set for best performance



Slight overfitting (Test Score < Train Score)
100% training set for best performance

Best score on kaggle to date: 69.553

Mean Average Precision (MAP)

Performance Metric: MAP@12

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

P(k) = precision at cutoff k
$|U|$ = # of display_id
n = # of predicted ad_ids

For every display_ad, only one ad_id is clicked.

display_id	ad_id	clicked
123	989	0
123	990	1
123	999	0
234	783	0
234	777	0
234	767	1
234	798	0

Example:
(ad_id listed in decreasing likelihood)
display_id = 123
P(1) = 0, P(1) = 1/2, P(2) = 0
display_id = 234
P(1) = 0, P(2) = 0, P(3) = 1/3, P(4) = 0
MAP@12 = 5/12

Discussion

The most challenging aspect of this project was understanding **MAP@K** and also dealing with extremely **large datasets**. By using **one hot encoding**, categorical data were mapped to an appropriate format for use with conventional machine learning algorithms. We determined thresholds for features with large cardinality to group rare examples into **minority categories**, reducing run-time. We chose to use **random forest** *(not taught in class)* due to its implicit feature selection, which performs well with missing categorical values. From our analysis so far, random forest has resulted in the best MAP score.

Future Works

To implement:
- **field aware factorization machines (FFM)**, which outperformed existing models in click prediction tasks for classifying large sparse datasets.
- **k-modes** to cluster users and user contexts for feature reduction while examining the user base.

References

[1] L. Breiman, "Random Forest," in *Machine Learning*, vol. 45. Springer US, 2001, pp. 5–32.
[2] C. D. Manning, P. Raghavan, and H. Schütze, "Evaluation in Information Retrieval," in *An introduction to information retrieval*. New York: Cambridge University Press, 2008.
[3] T. Y. Liu, "The Pointwise Approach," in *Learning to rank for information retrieval*. Berlin: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2011.