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)

Features

<table>
<thead>
<tr>
<th>display_id</th>
<th>ad_id</th>
<th>clicked</th>
<th>document_id</th>
<th>platform</th>
<th>timestamp</th>
<th>country</th>
<th>campaign_id</th>
<th>topic_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>5383232</td>
<td>2623256</td>
<td>0</td>
<td>180751341</td>
<td>1</td>
<td>40536000</td>
<td>PH</td>
<td>20077</td>
<td>88</td>
</tr>
<tr>
<td>16465403</td>
<td>221239</td>
<td>0</td>
<td>7265974</td>
<td>3</td>
<td>10986908</td>
<td>US</td>
<td>24776</td>
<td>234</td>
</tr>
<tr>
<td>5782925</td>
<td>64255</td>
<td>0</td>
<td>8049322</td>
<td>2</td>
<td>58486713</td>
<td>US</td>
<td>8568</td>
<td>1</td>
</tr>
<tr>
<td>12999603</td>
<td>479112</td>
<td>0</td>
<td>2111211</td>
<td>3</td>
<td>854135235</td>
<td>CA</td>
<td>32329</td>
<td>137</td>
</tr>
<tr>
<td>8929202</td>
<td>172955</td>
<td>0</td>
<td>2342266</td>
<td>2</td>
<td>595239728</td>
<td>GB</td>
<td>20933</td>
<td>52</td>
</tr>
</tbody>
</table>

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 categorical values rather than numerical values.

Machine Learning Pipeline

Data Sets

Data Wrangling
Events
Promoted Content
Clicks_train/test

Feature Engineering

Train/Test Set
TRAIN Training Set 1-16
TEST Training Set 17-20

Model
Logistic Regression (LR)
Support Vector Machines (SVM)
Random Forests (RF)
Naive Bayes (NB)

MAP@12

Performance Metric: MAP@12

\[
\text{MAP} = \frac{1}{|U|} \sum_{d=1}^{|U|} \sum_{k=1}^{K} P(k) = \text{precision at cutoff } k
\]
\[
|U| = \# \text{ of display_id}
\]
\[
n = \# \text{ of predicted ad_ids}
\]

Learning Curves

Logistic Regression
\[
L = \sum_{i=1}^{n} -T_i \log S_i + (1 - T_i) \log (1 - S_i)
\]
\[
S_i = \frac{1}{1 + e^{-z_i}}
\]
\[
z_i = \sum_{j=1}^{p} \beta_j x_{i,j}
\]
B = # of samples/trains
b = 1, \ldots, B

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

Random Forest

Slight overfitting (Test Score < Train Score)
100% training set for best performance
Best score on kaggle to date: 69.553

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