Simple and Scalable Response Prediction for Display Advertising

Criteo

Criteo is a secure advertising solution

Our Business Model

Our market:

- Banners
- Display
- Performance

Business Model:

- Buying CPM
- Selling CPC

Criteo: Performance and Display = how to « predict the click »

Criteo buys large inventories from Publishers on a CPM model

Criteo sells its campaign on unitary clicks to Advertisers on a CPC model

CPM: Paying per impression
CPC: Cost-per-Click

Goal

- CTR and CVR Prediction
  - CTR: Click-Through Rate
  - CVR: Conversion (i.e. purchasing a product, making a reservation, or subscribing to an email list) Rate
Model: Bayesian Logistic Regression

\[
M_0 \sim \text{normal} \quad S_0 \sim \text{normal}
\]

\[
M_j \sim j
\]

\[
\text{logistic}(\beta_0 + \sum_j \beta_j x_{j,i})
\]

\[
\mu_i \sim \text{Bernoulli}
\]

\[
y_i \sim i
\]

Benefits of Bayesian Logistic Regression

- It serves as a prior on the weights to update the model when a new batch of training data becomes available.
- It is also the distribution used in the exploration/exploitation heuristic (= Thompson sampling).
Features

- All features are categorical
  - Continuous features have been quantized appropriately

- Conjunction features (interaction terms) have been created for non-linearity learning
  - Conjunction features have high cardinality
  - Hashing trick for dimensionality reduction

<table>
<thead>
<tr>
<th>Feature Family</th>
<th>Feature Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertiser</td>
<td>advertiser (ID), advertiser network, campaign, creative, conversion id, ad group, ad size, creative type, offer type ID (ad category)</td>
</tr>
<tr>
<td>Publisher</td>
<td>publisher (ID), publisher network, site, section, URL, page referrer gender, age, region, network speed, accept cookies, geo</td>
</tr>
<tr>
<td>User (when avail.)</td>
<td>serve time, click time</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>
ALGORITHM 1: Hashing Trick

Require: Values for the $F$ features, $v_1, \ldots, v_F$.
Require: Family of hash function $h_f$, number of bins $d$.

$x_i \leftarrow 0, \ 1 \leq i \leq d$.

for $f = 1 \ldots F$ do
  $i \leftarrow [h_f(v_f) \mod d] + 1$.
  $x_i \leftarrow x_i + 1$
end for

Return $(x_1, \ldots, x_d)$.

Example

Publisher: "Instagram", "Snapchat", ...
Age: "0-20", "20-30", "30-40", "40-50", ...
Gender: "Female", "Male"

["instagram", "20-30", "Female"]
d = 5

(h$_{\text{pub}}$("instagram") mod 5) + 1 = 3
=> [0, 0, 1, 0, 0]

(h$_{\text{age}}$("20-30") mod 5) + 1 = 2
=> [0, 1, 1, 0, 0]

(h$_{\text{gen}}$("Female") mod 5) + 1 = 2
=> [0, 2, 1, 0, 0]
Dimensionality Reduction Strategies
Feature Selection: Mutual Information

Standard Mutual Information (SMI)

\[ I[x, y] \equiv \text{KL}(p(x, y) \| p(x)p(y)) \]

\[ I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \]

MI score utilizing a reference distribution (RMI)

\[ I_{\tilde{p}}(X_i, Y) = \sum_{x_i, y} \tilde{p}(x_i, y) \log \frac{p(x_i, y)}{\tilde{p}(x_i)p(y)} \]
Forward Feature Selection

(1) Start with a set of base features and no conjunction features
(2) Train a model with all the selected features
(3) Compute the conditional mutual information for all conjunctions not yet selected
(4) Select the best conjunction
(5) Go back to (2)

Table VI. Top Features for Click Prediction Along with Their Mutual Information

<table>
<thead>
<tr>
<th>Single Feature</th>
<th>SMI (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_guid</td>
<td>0.59742</td>
</tr>
<tr>
<td>query_string</td>
<td>0.59479</td>
</tr>
<tr>
<td>user_identifier</td>
<td>0.49983</td>
</tr>
<tr>
<td>user_segments</td>
<td>0.49842</td>
</tr>
<tr>
<td>section_id</td>
<td>0.20747</td>
</tr>
<tr>
<td>creative_id</td>
<td>0.20645</td>
</tr>
<tr>
<td>site</td>
<td>0.19835</td>
</tr>
<tr>
<td>campaign_id</td>
<td>0.19142</td>
</tr>
<tr>
<td>rm_ad_grp_id</td>
<td>0.19094</td>
</tr>
<tr>
<td>Conjunction feature</td>
<td>RMI (bits)</td>
</tr>
<tr>
<td>section_id x advertiser_id</td>
<td>0.24691</td>
</tr>
<tr>
<td>section_id x creative_id</td>
<td>0.24317</td>
</tr>
<tr>
<td>section_id x IO_id</td>
<td>0.24307</td>
</tr>
<tr>
<td>creative_id x publisher_id</td>
<td>0.24250</td>
</tr>
<tr>
<td>creative_id x site</td>
<td>0.24246</td>
</tr>
<tr>
<td>site x advertiser_id</td>
<td>0.24234</td>
</tr>
<tr>
<td>section_id x pixeloffers</td>
<td>0.24172</td>
</tr>
<tr>
<td>site x IO_id</td>
<td>0.23953</td>
</tr>
<tr>
<td>publisher_id x advertiser_id</td>
<td>0.23903</td>
</tr>
</tbody>
</table>

First table: standard mutual information; second and third table: modified mutual information (RMI). Bottom section contains the top conjunction features.
Conditional Mutual Information

- The difference between
  - log likelihood of the previous model and
  - log likelihood after a conjunction added

- Added weight is approximated to the value after one Newton step starting from 0
  \[ w_k \approx -\frac{L'_k(0)}{L''_k(0)}. \]

- The difference between likelihoods is calculated on a validation set
- The authors claim that the proposed method can be seen as an extension of conditional mutual information because the difference of likelihoods is equal to mutual information between x and y in special case
Subsampling

- The negative class is subsampled at a rate $r << 1$.
- The intercept of the model has to be corrected by adding $\log r$ to it.

\[
\frac{\Pr(y = 1 | x)}{\Pr(y = -1 | x)} = \frac{\Pr(x | y = 1) \Pr(y = 1)}{\Pr(x | y = -1) \Pr(y = -1)} = \frac{\Pr'(x | y = 1) \Pr'(y = 1)}{\Pr'(x | y = -1) \Pr'(y = -1) / r} = r \frac{\Pr'(y = 1 | x)}{\Pr'(y = -1 | x)}
\]
Parameter Estimation: Laplace Approximation

- Locate mode of posterior

\[ w^* = \arg \max_w p(w \mid D) = \arg \max_w \log p(w, D). \]

\[ E(w) = -\log p(w, D), \quad w^* = \arg \min_w E(w). \]

- Estimate standard deviation with Hessian

\[ H_{ij} = \left. \frac{\partial^2 E(w)}{\partial w_i \partial w_j} \right|_{w = w^*}. \]

http://www.sumsar.net/blog/2013/11/easy-laplace-approximation/
Comparison with Feedback Model

- Feedback Model: Response rate is not encoded in the weights but in the feature values. (e.g. CTR, CVR, number of impressions or clicks)

<table>
<thead>
<tr>
<th></th>
<th>auROC</th>
<th>auPRC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+0.9%</td>
<td>+1.3%</td>
</tr>
</tbody>
</table>
Exploration/Exploitation Tradeoff

- Thompson sampling: Each weight is drawn independently according to its Gaussian posterior approximation. A variant is considered in which the standard deviations are multiplied by a factor $\alpha \in \{0.25, 0.5\}$.
- LinUCB: LinUCB selects the ad for which $w^T x + \alpha \sqrt{x^T \Lambda^{-1} x}$ is maximum where $\Lambda^{-1}$ is covariance.
- Exploit-only: Select the ad with the highest mean.
- Random: Select the ad uniformly at random.
- $\varepsilon$-greedy: With $\varepsilon$ probability, select a random ad. Otherwise, select the one with the highest mean.
Thompson Sampling

For each round:

1. Sample random variable $X$ from each arm's Beta distribution
2. Select the arm with largest $X$
3. Observe the reward of selected arm
4. Update distribution of selected arm
UCB1

\[ \text{ucb}_i = \mu_i + \sqrt{2 \cdot \log t / n_i} \]

- \( \mu_i \): Empirical mean of rewards of ith arm
- \( t \): Number of total trial
- \( n_i \): Number of trial of ith arm

https://www.youtube.com/watch?v=JULzYBLp678&list=PLBjazsVtMXvI9miqmU2LT7HOnGGB7Yo&index=150
Exploration/Exploitation Tradeoff

Table VIII. CTR Regrets for Different Explore/Exploit Strategies

<table>
<thead>
<tr>
<th>Method Parameter</th>
<th>TS 0.25</th>
<th>TS 0.5</th>
<th>TS 1</th>
<th>LinUCB 0.5</th>
<th>LinUCB 1</th>
<th>LinUCB 2</th>
<th>ε-greedy 0.005</th>
<th>ε-greedy 0.01</th>
<th>ε-greedy 0.02</th>
<th>Exploit</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regret (%)</td>
<td>4.45</td>
<td>3.72</td>
<td>3.81</td>
<td>4.99</td>
<td>4.22</td>
<td>4.14</td>
<td>5.05</td>
<td>4.98</td>
<td>5.22</td>
<td>5.00</td>
<td>31.95</td>
</tr>
</tbody>
</table>