SIMPLE AND SCALABLE RESPONSE PREDICTION FOR DISPLAY ADVERTISING

论文阅读笔记

By OLIVIER CHAPELLE, Criteo   EREN MANAVOGLU, Microsoft   ROMER ROSALES, LinkedIn

Keywords: display advertising, click-through rate, conversion rate, logistic regression, feature hashing, distributed learning

MODEL SETUP

1. Splitting the training (~1 billion) and test sets chronologically
2. Subsampling the negative samples
3. Feature engineering
    a. 30 base features
    b. Automatically find new conjunction features by using the conditional mutual information in a forward feature selection algorithm:
        (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 informations for all conjunctions not yet selected;
        (4) Select the best conjunction;
        (5) Go back to (2).
    c. Feature hashing with 24 bits
4. Model: logistic regression
5. Metrics: negative log likelihood (or root mean squared error or area under the precision / recall curve or area under the ROC curve)

TAKEAWAYS

SOME VOCABULARIES:

- **Display advertising**: advertising on websites or apps or social media through banners or other ad formats made of text, images, flash, video, and audio
- **CPM**: cost-per-mille (cost per thousand impressions)
- **CPC**: cost-per-click (cost per click)
- **CPA**: cost-per-acquisition (cost per click AND specific action)
- **CTR**: click-through rate
- **CVR**: conversion rate

CLICK & CONVERSION

DEFINITION

- Click : action of clicking the ad impression
- **Conversion**: actions after click, such as **subscribing to an email list, making a reservation or purchasing a product**

ATTRIBUTE CONVERSION EVENTS TO CLICK EVENT

- In order to build a conversion model, we need to attribute the conversion event to the correct click event
- **A conversion event can happen minutes, hours or even days after a click event**
- In general **several conversion events could be associated with the same click**
- The longer the time elapsed between click and conversion the more logged events that need to be maintained and matched

DETERMINE THE AMOUNT OF DATA TO BE UTILIZED

- How much data need to be utilized for matching clicks & conversion?
- **Calculate the percentage of conversion events with different attribution time intervals**
- In their dataset, 86.7% of conversion events are triggered within 10 minutes of the click events. 39.2% occur within 1 minute of the corresponding click event. 95.5% occur within one hour of the clicks. Within two days of the click, 98.5% of the conversions can be recovered.
- They limited the time interval to 2 days which ignores approximately 1.5% of the conversion events

FEATURES & ALGORITHMS

FEATURE FAMILY

- **Advertiser**: advertiser (id), advertiser network, campaign, creative, conversion id, ad group, ad size, creative type, offer type id (ad category)
- **Publisher**: publisher (id), publisher network, site, section, url, page referrer
- **User** (when avail.) : gender, age, region, network speed, accept cookies, geo
- **Time**: serve time, click time

CATEGORICAL FEATURES:

- All the features considered in this paper are categorical (real values can be made categorical through discretization)
- Standard way: dummy coding, dimensionality can get very large
- **Hashing trick**
  - Dimensionality reduction : use a hash function to reduce the number of values a feature can take
  - **Vowpal Wabbit**: hash all features into the same space, a different hash function is used for each feature

    ALGORITHM 1: Hasing trick

    Require: Values for the F features, $v_1, \ldots, v_F$.
    Require: Family of hash function h_f, number of bins d.
    $x_i \leftarrow 0, \ 1 \le i \le d.$
    for $f = 1 \ldots F$ **do**
    $\quad i \leftarrow [h_f(v_f) \mod d] + 1.$
    $\quad x_i \leftarrow x_i + 1$
    end for
    return $(x_1, \ldots, x_d)$.

  - Collision analysis
    - *What is the impact of collisions? How to deal with it? TODO*
- Alternatives (keep only the most important values):
  - **Count**: Select the most frequent values.
  - **Mutual information**: Select the values that are most helpful in determining the target
  - **L1 regularization**

- Conjunctions

- A linear model can only learn effects independently for each feature
- A conjunction between two categorical variables is their Cartesian product
- *How to deal with the large cardinality of the conjunctions? TODO (hashing? matrix factorization?)*

SUBSAMPLING

- Subsample the negative class at a rate r<<1
- After training (with logistic regression), the intercept of the model has to be corrected by adding log(r) *[combining (1) and (6)]*

The logistic regression model is a linear model of the *log odds ratio*:

$$\log \frac{\Pr(y=1 \mid \mathrm{x}, \mathrm{w})}{\Pr(y=-1 \mid \mathrm{x}, \mathrm{w})} = \mathrm{w}^\top \mathrm{x}. \tag{1}$$

The model has of course to be corrected for this subsampling. Let us call $\Pr'$ the probability distribution after subsampling. Then:

$$\begin{aligned}
\frac{\Pr(y=1 \mid \mathrm{x})}{\Pr(y=-1 \mid \mathrm{x})} &= \frac{\Pr(\mathrm{x} \mid y=1)\Pr(y=1)}{\Pr(\mathrm{x} \mid y=-1)\Pr(y=-1)} && (4)\\
&= \frac{\Pr'(\mathrm{x} \mid y=1)\Pr'(y=1)}{\Pr'(\mathrm{x} \mid y=-1)\Pr'(y=-1)/r} && (5)\\
&= r\frac{\Pr'(y=1 \mid \mathrm{x})}{\Pr'(y=-1 \mid \mathrm{x})} && (6)
\end{aligned}$$

- or by giving an importance weight of 1/r to the negatives samples
- *What are the impacts of subsampling (in general)? TODO*

MODELING

- Logistic regression (easy to parallelize) with L-BFGS optimizer
- Map-Reduce implementation: a hybrid online+batch approach

ALGORITHM 4: Sketch of the proposed learning architecture

Require: Data split across nodes
  for all nodes k **do**
    w^k = result on the data of node k using stochastic gradient descent.
  end for
  Compute the average $\bar{\mathrm{w}}$ using AllReduce.
  Start a preconditioned L-BFGS optimization from $\bar{\mathrm{w}}$.
  for $t = 1, \ldots, T$ **do**
    for all nodes k **do**
      Compute g^k the (local batch) gradient of examples on node k
      Compute $\mathrm{g} = \sum_{k=1}^{m} \mathrm{g}^k$ using AllReduce.
      Add the regularization part in the gradient.
      Take an L-BFGS step.
    end for
  end for

NON STATIONARITY & MODEL UPDATE MECHANISM

- Display advertising is a non-stationary process as the set of active advertisers, campaigns, publishers and users is constantly changing.
- **Ad creation rates** : when new ads are added to the system (three identifiers: conversion identifiers, creatives and campaigns)
- **Ad life-time** : churn rate of the ads (also three levels: conversion, creative and campaign)
- Exploration / exploitation trade-off
  - In order to learn the CTR of a new ad, it needs to be displayed first, leading to a potential loss of short-term revenue.
  - Algorithms: Upper Confidence Bound (UCB), Bayes-optimal approach of Gittins, **Thompson sampling**