TalkingData AdTracking Fraud Detection Challenge

Hongzhi Shi[1], Lu Chen [2], Shuhan Fan[3], Haoyue Tang[1]

[1]Department of Electronic Engineering

[2]Institute for Interdisciplinary Information Sciences

[3]Department of Computer Science

Introduction

In pay-per-click online advertising systems like Google, Overture, or MSN, advertisers are charged for their ads only when a user clicks on the ad. These systems are highly susceptible to a particular style of fraudulent attack called click fraud, which happens when an advertiser or service provider generates clicks on an ad with the sole intent of increasing the payment of the advertiser. For online advertisement, fraud clicks cause significant burden for companies to predict their advertising effects precisely. Methods are needed to prevent distinguish these fraudulent clicks.

In this project, we aim at predicting a user behaviour,simply as Conversion Rate (CVR) using statistical learning methods from the details about his click of the advertisement. Take [1, 2, 3] as references, we adopt xgBoost, DNN and ensemble methods to improve the prediction accuracy.

Overview

The main idea of our work is as follows:

- Feature Engineering. We generate new features on full data after evaluating them on sampled data.
- Models. We use xgBoost and DNN models respectively on the full positive data and down-sampling negative data to make sure the amount of this two types of data are equal.
- Ensemble. We ensemble the submission of the two models to meet the higher accuracy.

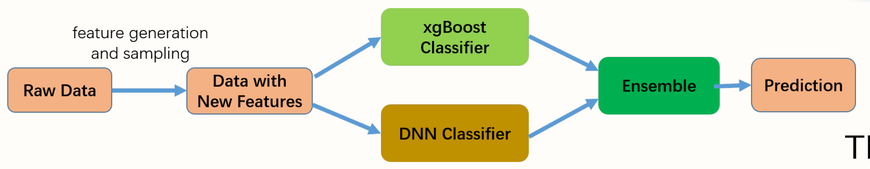

Figure 1: Prediction Flowchart

Feature Engineering

In the problem, last 3-day's click statistics of ip, app, channel, device and click time are provided by TalkingData, which are too limited for detection problem. Hence we generate new features in diverse ways and use xgBoost to derive how important new features are, which help us select features for further prediction.

To deal with extremely imbalanced data, we first use all positive examples and down-sampled negative examples with 5% for feature engineering.

- Clicks by ip : The click number of each ip.
- Confidence rate : We use the attribute rate to deal with the limited number of joint-category samples:$P(\text{is\_attribute}|\text{category})$
  Since some samples about joint category is limited, we use confidence rate as a companion to reweighing the attribute rate. $\text{conf} = \frac{\log(\text{views})}{\log 1000000}$
- Group by aggregation : We apply aggregation function like count/mean/var to different groups of attributes.
- Time until next click : How long it takes for a given combination (like ip-app-channel) before they perform the next click.
- Clicks on app ad before & after : Whether the user previously or subsequently clicked the exact same click combination (like app-device-os-channel).
- LDA / NMF / LSA on 5 basic attributes : we tried categorical feature embedding by using LDA/NMF/LSA, like computing LDA topics of IPs related to app.

To have an intuition of how new features improve performance, we add different new features into basic features, i.e, initial attributes, and train xgBoost models to evaluate the improvement compared with baseline, which is shown in Fig.2. In the models, we take Day-7& Day-8 data as training set, Day-9 data as test set, and adopt fixed hyper-parameters.

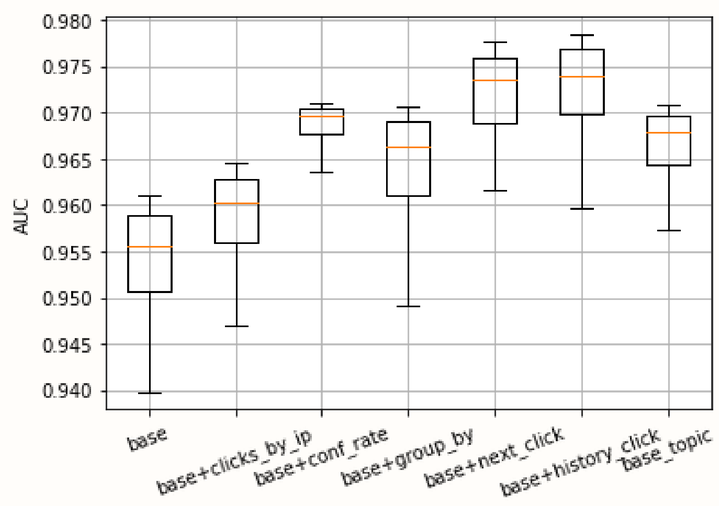

Figure 2: Evaluation of the features

Methods

xgBoost

The flow of xgBoost model process is as Fig.3 shows.

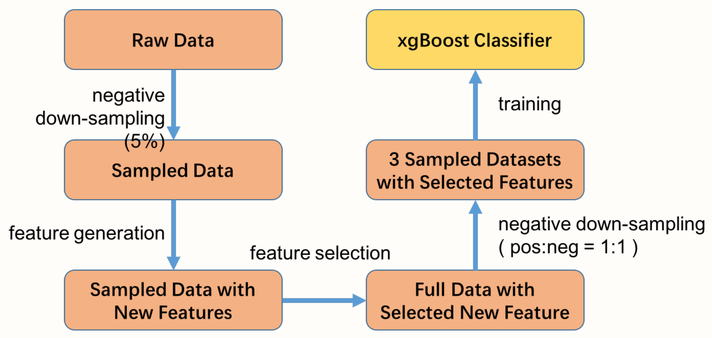

Figure 3: xgBoost process flow chart

For model training, we use 3-fold cross-validation of data from day 7, day 8 and day 9 with all positive examples and down-sampled negative examples on model training such that positive:negative = 1:1. We then choose the best parameters of XgBoost by GridSearchCV. Lastly, We sample 3 different down-sampling datasets and train XgBoost models on them.

DNN

Because of its deep layers and large quantity of neurons, deep neural networks is especially effective for complex prediction and classification tasks.

To promte diversity for future ensenble and prediction, we use two networks for training. One network consists of four layers, an embedding layer, a fully convolution layer (1D), two hidden layers using ReLU as an activation function and an output layer using sigmoid as an activation function. To reduce overfitting, a dropout rate of 0.2 is used at each hidden layer. Another network is different in that it does not has a fully convolution layer.

Ensemble

An ensemble method is used to generate the final detection results based on the above models. To improve accuracy, we adopt ensemble methods based on 3 xgboost models and 3 DNN learning models. The flow figure is provided below.

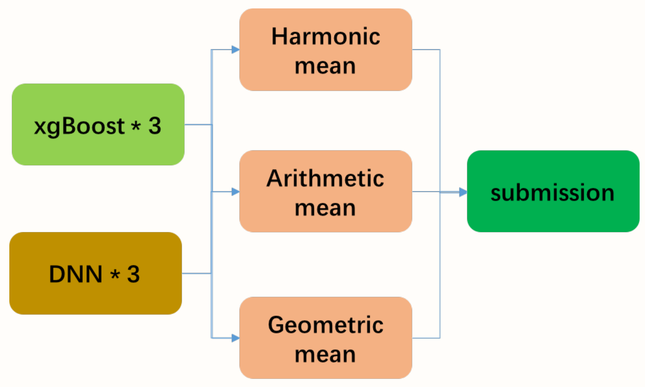

Figure 4: Ensembling methods

Experiments and Results

Method	AUC
xgBoost1	0.9663286
xgBoost2	0.9652461
xgBoost3	0.9658363
DNN1	0.9743403
DNN2	0.9721539
DNN3	0.9656231
Ensemble	0.9748226

Summary and conclusions

In this project, we focus on the feature engineering by feature construction to get more than 300 dimension features and feature evaluation to check the value of the new features. As for the models, we use xgBoost and DNN respectively to train the models and tune the parameters by GridSearchCV for xgBoost and manual for DNN. Finally, we ensemble these two models to improve our results.

It should be noted that, due to time limitations, our course project is still not mature. We will adjust the network structure and parameters and hopefully get a better prediction result.

References

[1] W. Ji, X. Wang and F. Zhu Time-aware conversion prediction, *Frontiers of Computer Science*August 2017, Volume 11, Issue 4, pp 702–716

[2] Kitts B, Click Fraud Detection: Adversarial Pattern Recognition over 5 Years at Microsoft. In: Abou-Nasr M., Lessmann S., Stahlbock R., Weiss G. (eds) Real World Data Mining Applications. *Annals of Information Systems*, vol 17. Springer

[3] W. Zhang, T. Du and J. Wang, Deep learning over multi-field categorical data, *European conference on information retrieval*, pp45-57, 2016