Comparative Analysis of Machine Learning models for CTR prediction
INFO 7390

Saurabh Salunkhe

NORTHEASTERN UNIVESITY

Contents

Abstract

In online advertisements, click through rate has a significant role in revenue generation since it serves as a very important metric in listing ads to the users. Predicting the relevant ads and click through rate has been a huge problem in this multi-billion dollar ads industry. Earlier, even huge websites like Avito used to rely on general statistics for the placement of the ads in the search results. Off late, various machine learning methodologies have been leveraged for solving the ads relevance problem. For instance, Facebook uses decision trees with logistic regression whereas Twitter uses Logistic regression as the key model for ads prediction. In this paper, I have done a comparative analysis of all the state of the art machine learning and deep learning models that has been used to solve the ads relevance problem. Further, I have included the pros and cons of all the models as per my observation by applying those models to the dataset provided by Avito in one of the Kaggle competition.

Keywords

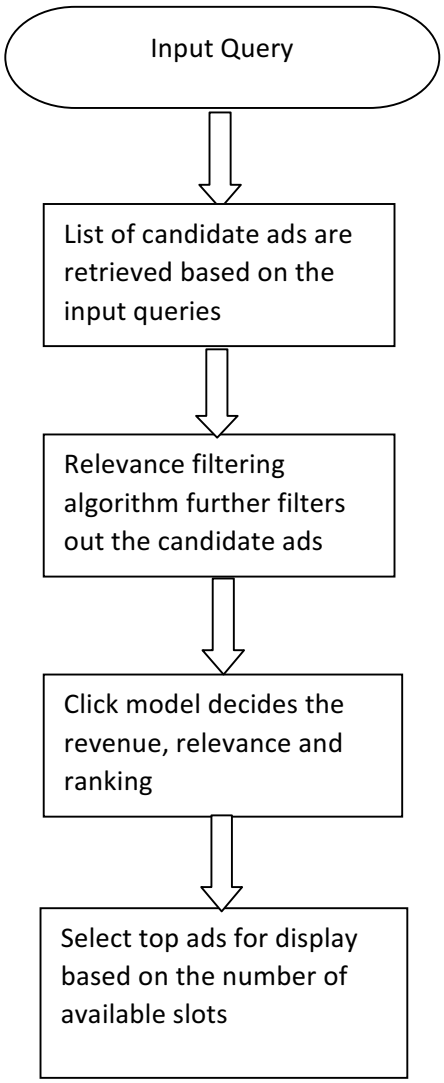
Linear Regression, Logistic Regression, XGBoost, Deep Neural Networks, Random Forests, CTR prediction, Online advertisements

Introduction and Related Work

Ads relevance continues to be a problem in this huge ad industry and in the recent times we have witnessed several machine learning models that tries to solve this problem. In this paper, I have tried to get a detailed analysis, advantages and disadvantages of all the cutting-edge models built and used by huge companies or models that has been used in Machine learning competitions. The paper on [1] very well explains how we can use linear regression to calculate the CTR by retrieving context information in real time. The paper on [2] has explained how model ensembling techniques can be used to construct a robust model and prevent over fitting by leveraging the concept of ensemble trees. However, all the models discussed in this paper are state of the art models which I used on my datasets and noted down my observations to substantiate the differences between those models like accuracy and response time.

Advertisement System Framework

Before straight away delving into Machine learning models and their analysis, let us try to understand how advertisement system framework works. Huge online classified ads and Search engines generally follows this framework for their ads retrieval when user search for a query. Advertisers or Sellers place their ads to target the millions of customers of the websites or search engine. They bid for keywords and when the user submits a query the results are matched with the keyword entered and then presented to the user. After the user submits a query, candidate ads are retrieved from the ads database as per the query. Every candidate ad will have its own relevance score according to the query and the ad. Usually, the relevance score is decided by the relevance model. The relevance model filters out many candidate ads. Further, the ads are narrowed down by the click model given the query and the model. The final list of ads is then sorted and placed according to the available slots.



Thus, CTR or the click model used here is extremely important since it has a huge role to play in the revenue as well as the relevance of the ads. We can assume that CTR can be output predictors in case of regression. There is another approach that simply gets the web logs and study the click data whether it is clicked or not. In this paper, we will discuss both the methods.

Experimental Setup

Datasets

As mentioned earlier, the data set has been picked from Kaggle by Avito website which has over 60 million instances. The dataset is in relational format with 8 tables. For most of the models I have trained, I have used 80/20 training and test sets. Features set for data are as follows

Input Features

SearchID	identifier for the visitor's search event
SearchDate	Date from the search event
IPID	Ip address of user
SearchLocaltion	Location where search was done
UserID	Unique identifier of the user
UserAgentID	Browser family of the device
AdID	Unique identifier of the ad
Position	Position of the ad
ObjectType	Wheter context, regular or highlighted ad
isClick	Whether the ad has been clicked
Price	price of the ad
Title	Title of the ad

Feature extraction

Feature extraction and ranking is probably the most important part of machine learning. There are tons of feature ranking methods out there. In my models used below, I have used the 'Recursive Feature Elimination' and 'Random Forest Selection' methods for coming up with high weight features. Based on the calculations, I found out that the ads position, category and the historical CTR has a substantial weightage. Hence, in most of the models, I have used these variables as my predictors.

Recursive Feature Elimination

The recursive feature elimination model performs well since it iterates the process until all the features of the dataset has been visited. However, it initially requires an estimator like SVM or linear model. We can then use the RFE function of sklearn for feature engineering.

Random Forest Feature ranking

Here, I have used the sklearn random forest regressor and ranking to come up with relevant features.

Models used

Logistic Regression

Logistic Regression is by far the most commonly used method for ads clicks classification problem. Tech giant like Twitter uses Logistic regression as their choice of model for predicting CTR.

Input Features

Features	Explanation
Position	Position of the AD in webpage
HistCTR	Past click information

Observation

By using 6 million instances, I got an accuracy of 98% when estimating the concrete class, 1 or 0. However, since we are more interested in the stochastic metric owing to the highly imbalanced classes, I further carried out the probabilities by using sklearn predict_proba and below mentioned are my results.

Sample Response

Test Id	0 (No Click)	1 (Click)
0	0.981189	0.018811
1	0.998506	0.001494
2	0.981192	0.018808
3	0.998503	0.001497
4	0.981255	0.018745

Advantage

The advantage of logistic regression is that it goes well with frequently mentioned ads or ads with high views.

Disadvantage

In real life scenarios, the classes are highly imbalanced since the ratio of click to non-click is less than ten percent. Also, another problem with logistic regression is that it is not good for ads with low views thereby creating a huge bias.

XGBoost

XGBoost has gained a lot of popularity recently due to its prominence in recent Kaggle Competitions. Almost 50% of the winning models have been implemented in XGBoost training. Major companies like Yandex uses the proprietary software MatrixNet which is an implementation of boosted trees. Feature set table is as follows:

Input Features

Features	Explanation
Position	Position of the AD in webpage
HistCTR	Past click information
Level	Level of category for search

Sample Response

Test Id	0 (No Click)	1 (Click)
0	0.993839	0.006161
1	0.996612	0.003388
2	0.997686	0.002314
3	0.994729	0.005271
4	0.990436	0.009564

Advantage

XGBoost greatly prevents overfitting. Further, another advantage of XGBoost is that we can use feature sets loosely and it deals greatly with all types of loss prediction

Disadvantage

Since gradient boosting builds one tree at a time, the model construction time can be much higher.

Linear Regression

It might come as a surprise as how can be use linear regression model in this classification problem since a lot of variables are ordinal but borrowing some of the ideas from [2], I have used a lot of techniques to make sure we can leverage the linear regression capabilities by getting the continuous values from the variables. This may result in dense feature but works well. Feature Set:

Input Features

Features	Explanation
Position	Position of the AD in webpage
HistCTR	Past click information
AdSize	Size of the ad
Location	Location of the user
Category	Category according to classification model

Advantage

It is a very helpful and efficient method in calculating the click through rate and the output is very relevant since we are deriving context information and applying it then and there.

Disadvantages

This method may have a large response time.

Deep Neural Networks

This is much of an experimentation since deep learning alone are not usually the choice of model for predicting the click through rate although in some paper [2] , deep learning has been combined with boosted trees in finding the CTR. However, I thought it would be worth a try to use neural networks for this task. I used the Keras library for this purpose. Keras is a rather straightforward approach for building neural networks but care must be taken not to add more layers as it might have led to the problem of overfitting. I have utilized the sequential model API for building the model wherein we will add layers to an empty model we create. Our final layer will have one node as that will be our output variable. Here, I have used Rectified Linear Unit (ReLu) as my activation function. Further, I have used 'adam' as my optimizer function and 'rmse' as my loss function.

Feature sets

Features	Exaplanation
SearchID	Identifier of the AD
AdPosition	Position where the ad is in the website
HistCTR	Historical click through rate

Epoch Values

```
Epoch 1/50 - 1s - loss: 0.0082 Epoch 2/50 - 1s - loss: 0.0069 Epoch 3/50 - 1s - loss: 0.0068 Epoch 4/50 - 1s - loss: 0.0068 Epoch 5/50 - 1s - loss: 0.0067 Epoch 6/50 - 1s - loss: 0.0067 Epoch 7/50 - 1s - loss: 0.0067 Epoch 8/50 - 1s - loss: 0.0066
```

Observations

The rmse value turns out to be 0.004537 which is very small. But again, as discussed in logistic regression, the highly imbalanced classes are playing its roles. Here it makes a lot of sense to go for stochastic model instead of predicting the concrete class.

Sample Response

Test Id	0 (No Click)	1 (Click)
0	0.993839	0.006161
1	0.996612	0.003388
2	0.997686	0.002314
3	0.994729	0.005271
4	0.990436	0.009564

Advantage

Relatively new method and can leverage nonlinear features and which has a substantial edge over logistic regression

Disadvantages

The model may introduce some bias. Also, huge problems of overfitting may occur if lots of parameters are used.

Conclusion

We first started by saying the importance of the parameter click through rate in the ads framework model and how significant it is in the revenue generation and relevancy for both the publisher and advertiser. We further stated the current models being used by tech giants like facebook and Yahoo. Then, we applied different models on the Avito dataset for comparative analysis of all the state of the art machine learning and deep learning model and state its advantage and disadvantages. Here, I have made a tabular data for quick look of all the pros and cons of the model we discussed so far.

Model	Pros	Cons

Linear Regression	Efficient and relatively new so lot of scope for improvement	Since we are fetching the context during the search event and performing computations, response time can be higher
Logistic Regression	Performs excellent with high view ads and frequently mentioned ads	Performs poorly for new ads which are usually low view ads
XGBoost	XGBoost greatly prevents overfitting. We can use feature sets loosely and it deals greatly with all types of loss prediction	Since it models one tree at a time, the model training time can be large
Deep Learning	Relatively new method and can leverage nonlinear features and which has a substantial edge over logistic regression	Model may be biased. Also, huge problems of overfitting due to many parameter

References

[1] https://arxiv.org/ftp/arxiv/papers/1701/1701.08744.pdf

[2] https://www.microsoft.com/en-us/research/wp-content/uploads/2017/04/main-1.pdf

[3] https://www.kaggle.com/harlfoxem/housesalesprediction

[4] https://arxiv.org/pdf/1603.02754.pdf

[5] http://wan.poly.edu/KDD2012/forms/workshop/ADKDD12/doc/a3.pdf

[6] https://www.slideshare.net/ManishaSule/ctr-prediction-using-spark-machine-learning-pipelines