2015 International Conference on Soft Computing Techniques and Implementations- (ICSCTI)
Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015

A comparative study of Linear learning methods in Click-Through Rate Prediction

Antriksh Agarwal
Department of Computer Engg,
Jamia Millia Islamia
Okhla, Delhi
antriksh5235@gmail.com

Avishkar Gupta
Department of Computer Engg,
Jamia Millia Islamia
Okhla, Delhi
avishkar.gupta.delhi@gmail.com

Dr. Tanvir Ahmad
Department of Computer Engg,
Jamia Millia Islamia
Okhla, Delhi
tahmad2@jmi.ac.in

***Abstract*—A major challenge in the current era of search engine advertising is choosing which advertisements to show in response to a user query. This significantly impacts the overall user experience, and more importantly the advertising revenue stream for the search engine provider. Predicting click-through rates (CTR) for an advertisement is a massive-scale learning problem that is central to the multi-billion dollar online advertising industry. This study examines the performance of some well-known statistical learning methods (linear and logistic) with respect to their efficiency in predicting the click through rate of an impression, where an impression can simply be defined as an instance of a particular advertisement, with each instance defined in terms of the learning parameters in our data set. Our data set consisted of three types of independent attributes to act as a regressor in predicting our dependent variable - the app through which it was clicked, the site type and the domain to which it led – with the help of other anonymised variables. Fine tuning of the algorithm parameters was done to get promising results. Besides that a dimensionality check on the data set was conducted to observe the possibilities of dimensionality reduction. Logistic loss (log-loss) was used as the validation index in all cases. Our observations led us to the conclusion that with minimal data pre-processing, linear models give competitive on-par results suited for most practical applications, where the learning method chosen should not be computationally expensive. We go on to further verify this claim by comparing the performance of linear models on various subsets of the data set attributes, showing that the performance of the linear techniques was consistent all across.**

Keywords—Logistic Regression; Click-Through Rate; Linear Models; Logistic Classifier-Regressor; Advertising

I. INTRODUCTION

Advertising via sponsored search results has become the platform for companies to gain a reputation for themselves beyond local markets, acting as a major income/revenue source for search engine providers such as Google/Yahoo/Bing etc., generating revenues of the order of 25 billion dollars and upwards[2].

A way to predict the effectiveness of a marketing campaign would be to record the user's reaction to the ad when it shows up. However, seeing as how that is not feasible with the current technology, click through rate, which tells us about how many visitors merely "initiated action" in response to the showing of the ad servers as a metric in understanding user behavior. Different advertisers target different kinds of users: a mountaineering equipment company will be interested in users who may be bought some sporting gear recently, and an airline would prefer to display its ads to people who are frequent fliers.

Click through rate prediction plays an important role in this area of sponsored advertising. A higher click through rate is a clear indicator for predicting the success of an online marketing campaign, as well as the success of an e-mail marketing campaign. A higher click through rate means more number of users are clicking the ad, which means our campaign is reaching the target audience. Click through rate prediction is therefore necessary to be able to further optimize ad placement in the sponsored search market. The sponsored search advertising model exploits two key aspects of on-line advertising [3]. First, the user enters a query to the search engine, which is a give away of their intent and determines the type of advertisement that would be shown to them. Also, if a user is to follow the said link, then the success can be attributed directly to the search engine provider in the case of sponsored search.

However, in the cases where these advertisements are placed on websites as banner ads, etc. a large number of factors come into play and things are not so straight forward. The positioning of the ad on the site, the device being used to surf the site, are some common ones. Also, the advertising on these sites is directly linked to the traffic volume on the original website where the ad is displayed.

Because of this, it is necessary to factor in these attributes when trying to calculate the click through rate for an advertisement on a site other than that of a search engine provider. Our work is aimed at predicting CTR in these cases where the advertisement display is not necessarily on a search portal. In these scenarios user queries are no longer available to exploit, and factors such as the theme of the website, etc. then have to be taken into account.

Most work in this area has been carried out by Search Engine Providers, but the techniques given by them are not applicable 'as-is' here because they in most cases, do not accommodate the said metrics.

Click through rate can be defined as:

$$CTR = \frac{(\text{No. of Clicks})}{\text{No. of Impressions}} * 100 \qquad (1)$$

where each impression refers to one showing of the ad.

This paper attempts to make a comparison of the performance of some well-known linear and logistic learning methods in click-through-rate prediction and touches on the key role that CTR prediction plays in sponsored search. We chose linear models, since training a single layer model would allow us to handle significantly larger data sets and larger models than have been reported elsewhere. Also, the data pre-processing was kept to minimum as our objective was to draw a comparison based on the performance of classification on the data set.

II. RELATED WORKS

Craswell, Ramsey, et. al. [5] analyzed the effect of a links' position in determining the probability that the link will be clicked. They compared four real world situation models to that of logistic regression. They proposed a "cascade" model that can be applied without the need for training data, and parameter-free to click observations. Thir model however performed badly in lower ranks. Their results went into depth about how the position of an ad will affect its probability of being clicked, just like a search result.

Azin Ashkan, Charles L.A. Clarke et. al. estimated ad click-through rate by exploring user queries and click-through logs. Their findings go on to prove that rank of an ad, query intent, no. of ads displayed on result page etc. are effective in estimating click-through rate[6].

A related paper [7] by Zhong , Wang et. al. explored the user's post-click behavior (such as the dwell time on the clicked document, and whether there are further clicks on the clicked document). They worked on monitoring the user's activity post -click after leaving the search page and proposed a click model.

The works of Ye Chen, Tak W. Yan[8] and several others hint at the positional-bias problem in Unison. The works of Jingfang Xu et. al[9] and Ben Carterette & Rosie Jones[10] touches on the problem of minimizing relevance judgment errors. Their findings provided a way for comparing raking functions by predicting relevance from click-through rates.

In addition to this, previous eye-tracking experiments and studies on explaining position-bias of user clicks provide a spectrum of hypotheses and models on how an average user examines and possibly clicks web documents returned by a search engine with respect to the submitted query.

III. PROPOSED ARCHITECTURE

Fig. 1 describes the proposed methodology we employed for the prediction of click-through rate based on the independent variables, having following modules - Data Pre-processing, Logistic Loss based Classifier Selection, Linear Models and Dimensionality Reduction.

A. Procuring the Data-set

This is the primary step during which data obtained from logs of websites are used to derive the independent variables. A raw (not scaled) data set is obtained and saved in a standard format (eg. CSV).

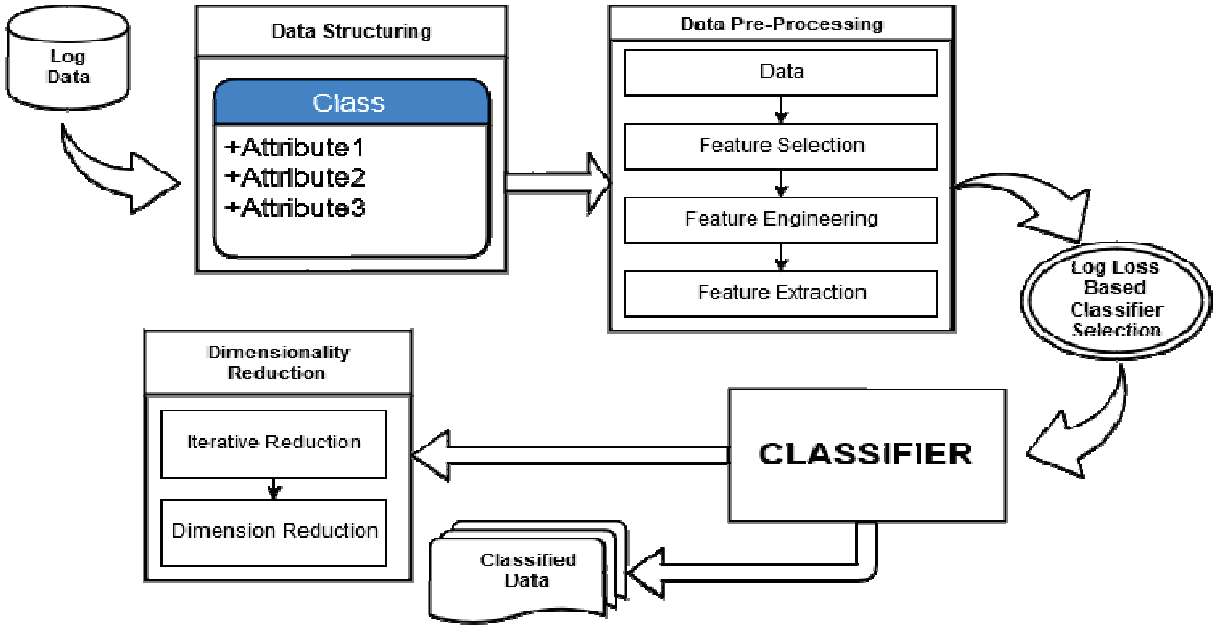

Fig. 1.Proposed Architecture of Prediction

B. Data Pre-processing

Various data pre-processing steps like data scaling, field removal and format conversion were applied that can be summarized as follows.

Feature Selection: In the field removal steps columns like ID, Serial No. were removed from the data since these columns were used for identifying the rows and have no role in classification. Candidate features are chosen out of the features obtained in the previous step, such that, their removal does not affect the accuracy of classification model. Among those candidates for the pair about which we have a rationale for their removal are removed such as the identities of each of the table as well as the features provided in the table.

Feature Engineering: Features such as time and hour which have been given in a date time format in the table had to be separated and special functions were created for the same.

1) Feature Extraction: Often features are not given as continuous values but categorical. When discrete values constitute the data of a particular feature, instead of the continuous values that are usually used to classify, we cannot use these features directly with the estimators. The estimators expect the input to be continuous and would interpret the categories as being ordered, which is not often desired. One possibility to convert categorical features to features that can be used is feature hashing. Feature hashing, also known as the hashing trick, is a fast and space-efficient way of vectorising features, i.e. turning categorical features into indices in a vector or matrix. It works by applying a hash function to the features and using their hash values as indices directly, rather than looking the indices up in an associative array and creates features to determine column index in sample matrices directly.

C. Log-Loss Based Classifier Selection

In this step the emphasis is on the selection of the classification algorithm. The data set should be tried on various Machine learning (ML) algorithms. This aids in selection of the base learner. Logarithmic Loss is the loss function used in multinomial logistic regression and extensions of it such as neural networks, defined as the negative log-likelihood of the true labels given a probabilistic classifier's predictions. Logistic loss for $y_i \in \{0,1\}$:

$$L = -\left(\frac{1}{m}\right)\sum_{i=1}^{m}(y_i \log(\overline{y_i}) + (1 - y_i)\log(1 - \overline{y_i})) \qquad (2)$$

where $\overline{y_i}$ is a prediction for the i^{th}. Logistic Loss for $y_i \in \{-1,1\}$:

$$L = -\left(\frac{1}{m}\right)\sum_{i=1}^{m}(\log(1 + e^{-y_i p_i})) \qquad (3)$$

where p_i is a raw score from the model and $y_i = \sigma(p_i)$, $\forall i \in \{1,\ldots,m\}$.

D. Classifier

Identifying to which set of categories a new observations belongs, on the basis of a training set and the statistical relationship among variables in the training set whose category membership is known is commonly referred to as classification. It includes many techniques for modelling and analysing several variables and finally derives a relationship between a dependent variable and the independent variables. Classifier performance depends greatly on the characteristics of the data to be classified. Various empirical tests have to be performed to compare classifier performance and to find the characteristics of data that determine classifier performance. A large number of algorithms for classification can be phrased ni terms of a linear function that assigns a score to each possible category k by combining the feature vector of an instance with a vector of weights, using a dot product. The predicted category is the one with the highest score. This type of score function is known as a linear predictor function and has the following general form:

$$\text{score}(\mathbf{X}_i, k) = \boldsymbol{\beta}_k \cdot \mathbf{X}_i \qquad (4)$$

where $\mathbf{X}_i$ is the feature vector for instance i, $\boldsymbol{\beta}_k$ is the vector of weights corresponding to category k, and score($\mathbf{X}_i$, k) is the score associated with assigning instance i to category k. Algorithms with this basic setup are known as linear classifiers.

E. Dimensionality Reduction (DR)

DR is the process of removal of variables from the data set which are correlated with each other and might degrade the classifier accuracy. Following steps were performed in order to improve the accuracy.

1) Iterative Classification: Each variable in the data set is excluded one by one and a model is built using Logistic Regression, features whose exclusion results in logistic loss lower than the default logistic loss (when no variable is removed) are noted down.

Dimension Reduction: Candidate features are chosen out of the features obtained in the previous step, such that, their removal does not affect the accuracy of classification model. Among those candidates for the pair about which we have a rationale for their removal are removed. This accuracy driven DR approach is also known as the wrapper approach.

IV. EXPERIMENTAL SETUP

Experimental Data-Set

We were provided with ten days of sub sampled click-through rate data by Avazu[1], made available on kaggle, an online portal for data-science. This set consisted of about 40 million lines of training data, which we further sub-sampled this data set to create a 60-40 split of training and testing data using the first 500,000 records. The sets had the following attributes.

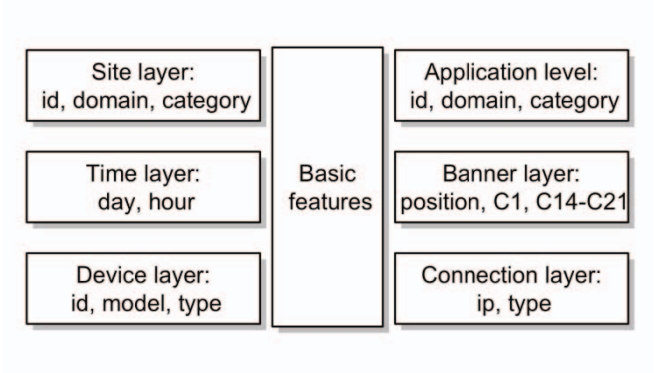

Fig. 2. Types of attributes in the data-sets, categorized according to their physical intuition

The feature we had to predict in the attributes above is 'click' and the others function as the independent features. Out of the independent features, the parameters C1, C14-C21 represent categorical features(where each value represented an ID and has no quantitative significance) whose significance was anonymized by Avazu for business reasons. These anonymized features represent variant attributes such as the dimensions of the advertisement. The features whose attributions were known even though also containing hashed strings were also categorical (discrete) features that covered the following attributes:

site_id, site_domain and site_category – Features that specify the site on which an impression of the advertisement was put.
app_id, app_domain and app_category – Specify the app in which the advertisement/webpage with the advertisement was shown.
device_id, device_ip, device_model, device_type, conn_type – Identify the device of the user on which the impressions were shown.

Prototyping Tools

We used the classifiers provider in the scikit-learn toolkit[11] available for the Python programming language. The SciPy toolkit featuring Numpy, Scipy, and all associated packages was also used.

B. Experimental Procedure

The experiment was then conducted using the architecture proposed in section III. We tested the architecture using three learning methods – vanilla logistic regression, Stochastic Gradient Descent (SGD Classifier) and a Bayesian method(multinomial Bayes). Logistic regression and Stochastic Gradient Descent were used as logistic regression attempts to minimize log-loss and SGD for its ability to support supports different loss functions and penalties for classification. We used a Bayesian method so as to show that the features are not independent of each other, as otherwise the naïve Bayes assumption of feature independence would make it also a viable option. This was done to find which classifier performed best given our set of chosen features. Some variables, such as id, app_id, site_id, site_domain, were removed at the start, so that the models do not use these distinct valued attributes to create additional features that are too-specific to an impression. This was also done to not un-necessarily increase the size of the dataset. Other variables were removed and tested to see which variables best fit our classification. The logistic loss computed provided a fair deal of insight on how good the algorithm was performing on our sub sampled data, with click, the click through rate of the impression being the binary attributed target feature for which probability of classification was calculated.

Table I. Comparison of different learning methods with their output log-losses.

Learning Method Applied	Logistic Loss
SGD Classifier	0.411
Logistic Regression	0.403
Multinomial Bayes	1.476

We applied various learning methods to check out which one gave us the best results. As you can see above, Logistic Regression gave us the best output We conducted experiments on them to find out what set of attributes, taken together, gave us the best estimate of the click-through rate. For this we tested our results with iterative reduction and dimension reduction.

Table II. Using linear models to see how attribute removal changed the output.

Attributes Removed	Log-Loss
None	0.403
app_category	0.404
site_category	0.404
device_conn_type	0.403
C1, C14-C21	0.432
C1, app_category, site_category	0.404
C1, device_conn_type	0.403

C. Logistic Regression

Logistic regression is a regression model in which the dependent variable is categorical. Logistic regression measures the relationship between the dependent variable and the independent variables by estimating probabilities using a logistic function. The mathematics of logistic regression

begins with the explanation of logistic function. The logistic function is useful because it can take an input with any value from negative to positive infinity, whereas the output always takes values between zero and one and hence is interpretable as a probability. The logistic function $\sigma(t)$ is defined as follows:

$$\sigma(t)=\frac{e^t}{1+e^t}=\frac{1}{1\text{-}e^{-t}} \qquad (5)$$

If t is viewed as a linear function of an explanatory variable x (or of a linear combination of explanatory variables), then we express t as follows:

$$t=\beta_0+\beta_1 x \qquad (6)$$

And the logistic function can now be written as:

$$F(x)=\frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \qquad (7)$$

Note $F(x)$ is interpreted as the probability of the dependent variable equaling a "success" or "case" rather than a failure or non-case. It's clear that the response variables Y_i are not identically distributed: $P(Y=1|X)$ differs from one data point X_i to another, though they are independent given design matrix X and shared with parametersβ.

V. RESULTS AND DISCUSSION

The experiment resulted in finding out that among the listed linear models (in table 1), Logistic Regression was the best algorithm for finding the click-through rate. Such a result could be possible because logistic regression returns well calibrated predictions as it directly optimizes log-loss. This is because in the gradient descent of logistic regression the logistic regression is trying to minimize the cost,

$$\theta_j=\theta_j+\alpha\sum_i\left(y_i\text{-}p\left(x^{(i)}\right)\right)x_j^{(i)} \qquad (8)$$

which is represented by equation (2). Hence, in a way logistic will always be giving better log-loss values.

While Stochastic Gradient Descent (SGD), did not give a better result than logistic regression but it is an online classifier which does not need to be given all the data at the same time. For logistic regression, we have to feed all the data at the same time and with the amount of data we had in the dataset, we did consider using other algorithms before trying to use logistic for better results. It can be argued that the cost function being minimized in SGD is,

$$\theta_j=\theta_j+\alpha\left(y_i\text{-}p\left(x^{(i)}\right)\right)x_j^{(i)} \qquad (9)$$

which is very close to equation (2), but the summation over the terms, gave a better result, than one which was not being summed over. We also know that the performance of SGD improves as the size of data increases exponentially for it. So, the improvement of SGD with the size of data is good. Thus, it is very much possible that if would have supplied the whole of the data set that was available to us, we might just have been able to show that SGD was better than Logistic Regression.

Other methods like Naïve Bayes tend to push probabilities to 0 or 1.This is mainly because it makes the assumption that features are conditionally independent given the class, which was not the case in this dataset.

Another result that caught our eye was that increasing or decreasing any of the variables did not much contribute in improving the log-loss value. This shows that, as stated in the book [10], a regression model does not imply a cause-and-effect relationship between the independent and the dependent variables. Even though a strong empirical relationship may exist between them, it cannot be considered as evidence that the classifier features and the response are related in a cause-and-effect manner. To establish casualty, the relationship between the classifiers and the response must be outside the sample data.

VI. CONCLUSION AND FUTURE SCOPE

The excellent performance of Logistic Regression in comparison to other models, and the consistency in results shown when using this technique across all sets of features, the recommendation based on our results would be to use logistic regression in a practical situation where once can afford to run batch learning jobs frequently. However, given that the SGD classifier came in as a close second, and given the fact that it is an online learning method, classification can be improved by partially fitting any new data that comes in, SGD is ideal for situations where data is constantly flowing in rather than arriving in batches. For such work flows, Logistic Regression would need one to train the classifier with the entire dataset each time some modification needs to be done. Also, from our dimensionality reduction efforts, it is clear that classification will remain consistent even if one of the key features is not present in the data set.

We can look into more robust data preprocessing models and observe how preprocessing in different ways affects our results. We can also look into up and coming data-intensive, parallel programming techniques and GPU based programming to incorporate larger data sets, since at present we were able to use only part of the training data.

Other potentially interesting future work would be to observe how variety websites such as aggregation or social media platforms compare to theme specific sites that focus on only one aspect of content, such as sports portals, etc. This constitutes our future work.

REFERENCES

[1] https://www.kaggle.com/c/avazu-ctr-prediction/data
eMarketer, April 2009

[2] Broder, Josifovski, Introduction to Computational Advertising at Stanford, Lecture Notes, 2009

[3] Nick Craswell, Onno Zoeter, Michael Taylor, Bill Ramsey, An Experimental Comparison of Click Position-Bias Models.
[4] Azin Ashkan, Charles L.A. Clarke, Eugene Agichtein, Qi Guo, Estimating Ad Click-through Rate through Query Intent Analysis.
[5] Zhong et. al, Incorporating Post-Click Behaviors into a Click Model.
[6] Ye Chen, Tak W. Yan, Position-Normalized Click Prediction in Search Advertising
[7] Xu et. al, Improving Quality of Training Data for Learning to Rank Using Click-Through Data.
[8] Ben Carterette, Rosie Jones, Evaluating Search Engines by Modeling the Relationship Between Relevance and Clicks.
[9 M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[10] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *The* Journal of Machine Learning Research 12 (2011): 2825-2830.