CS229 Project Final Write-up
Predictive Analytics for E-Commerce Customer Behavior and Demand Forecasting
Team members: Shravan Surineni, Shuyu Mao{shravans,surimao}@stanford.edu

Abstract

We developed two robust classifiers to predict buying intentions of users based on past behavior for a large e-commerce website that sells nutrition products. In this project, we reviewed the traditional and advanced deep learning approaches, and recent advances in this field. We show that with sufficient preprocessing and selecting the right feature set, and resampling to address imbalanced datasets, even simple logistic regression and neural network models give superior performance.

Introduction

Predictive analytics to predict user intentions towards a specific product or category on an E-commerce website, based on historical website interaction data, is very useful especially for advertising, product recommendation engines and for demand forecasting. In online retail, it is possible to capture clickstream data on the website and gather helpful analytics through machine learning models.

Clickstream data is the electronic record of a user's activity on the Internet, and specifically on an e-commerce site for our purpose. Thus, the data trace the path a visitor takes while navigating the Web. This path reflects choices, often very large in number, made by the user both within and across websites (Bucklin & Sismeiro, 2008). This rich source of information makes it possible to gain more knowledge about the behavior of customers online and has the potential to predict future online behavior.

Machine learning models can be used to analyze clickstream data and predict future behavior. The final prediction of customer behavior is a binary classification problem (buy/no buy), so Logistic regression (LR) is the most popular model. Logistic regression requires a linear relationship between input features and the output, and may not model customer behavior accurately. It is possible to capture these non-linear relationships with deep learning using a Neural Network(NN).

In this project our goal is to analyze user activity on an e-commerce site selling nutrition products, identify the patterns to predict future behavior, through LR and NN models. **We achieved 95% accuracy on test set.**

Related Work

Clickstream data analysis is an established field with lot of research done on this topic. Most of the traffic analysis on websites and product recommendations use this analysis for predicting user behavior and demand forecasting.

The references in the end of the section refer to multiple academic projects done on this topic. Most users used logistic regression and deep learning models that used neural network models with random forests. Deep learning models are further enhanced in some cases using deep belief networks (DBN), a class of unsupervised learning models composed of a stack of restricted Boltzmann machines. A the core component of the complicated DBN is a greedy, layer by layer learning algorithm which optimizes DBN weights. Auto encoders are often used to learn good representation of the data transform and reduce the dimensionality of the problem in order to facilitate the supervised learning stage.

However, in our project, we intend to use simple logistic regression and neural network model as we feel that would suffice for the dataset we have at hand, if we can do sufficient data preprocessing to prepare the dataset to use in LR and NN models.

Dataset and Features

Data consists of one week of clickstream data from an e-commerce website that sells a vast collection of nutrition products. Each clickstream record includes 11 different fields, including unique user id, session id, timestamp and product id. In case of a buy or shopping cart view, we have information about the price and extra details. The table below shows the full list of fields provided in each clickstream record. This data is disaggregated and anonymous, without demographic information and other user attributes.

The first step in developing a machine learning mode, is to understand the dataset and to get an intuition of how it behaves, and the various fields and how they affect the model. Datatset analysis was one of the most critical steps in this project, as we spent a significant amount of time and effort on data analysis and preprocessing. The dataset contained all requests sent to the server, lacking any preprocessing. The original log included *1 245 378* rows of data, with *20 330* buy session. A vast part of the sessions from anonymous users (~70%). Since the website/store requires visitors to authenticate an e-mail id or phone number before they can make a purchase, all the anonymous sessions are not relevant and discarded.

The data consisted of a set of sessions and each session contains the user information, the product id from the product catalog. The sessions are two types, the sessions that ended up in a purchase, and the sessions that do not end up in a transaction. The clickstream records did not classify the sessions into buy/non-buy, so the first task was to find all the sessions that contained a buying event (classification), and the product bought (prediction).

The data is also highly unbalanced for the two classes considered (buy and non-buy), so we faced a severe imbalance problem. From the data of about 100,000 samples,

- 98% page views were without a transaction
- 2% sessions resulted in a transaction

Data preprocessing

Each session consisted of a session id, product id and timestamp, but no order information. There is a separate record containing only order information. The classification of events in clickstream was achieved by getting the session ID and product ID from the order information, and finding the sessions that match with this information to mark the full set of data into purchases.

Analyzing the buying records indicated that about 250 products were among the highly ordered set. We examined the buying records to categorize product ids into classes, with scores ranging from 1-10, with 1 being the lowest ordered items, and ten being the highest.

The records also provided a text description of the device used to access the site, client type and other textual information. As the text data cannot be processed in regression, we categorized we the text data into numerical values, to use as features with our ML models. The full list of fields, values and interpretation is shown below.

Field	Values	Meaning
Client Type	'Web_site', 'post_man', 'androiod_app','ios_app',' null'	The application used by user to access the site. Provides insights into user access to site
Count	'0', '12'	Corresponds to whether customer needs AI engine to provide recommendation or no
Device	MOBILE, null, COMPUTER, TABLET	Device customer used to access site. Expect higher conversion Computer/Mobile usage
Search String	Information on landing page	Whether customer got to the webpage through search engine or directly to site. Provides information on ad effectiveness vs. customer loyalty
Session ID	Cookie – Same Device, User, IP address	Unique session ID. Identifier to know single use vs. repeated visits
Site Page Type	Type of the webpage: Product_detail, pack,	Once on the site, how customer go to current webpage. Customer looking for specific product (through search or clicking popular products etc.)
Site Product ID	Product ID/Catalog	Unique product ID
URL	Infinite Analytics API	The recommendation tag that customer followed
User ID	User Identification	Anonymous user vs. known user
User Type	Session = Anonymous Site User = Authenticated	
Raw	Timestamp (Day, Month, Year, Time)	

The problem of predicting purchase engagement only refers to a subset of data from the original click stream data, as only the sessions that correspond to authenticated users are considered. From the original dataset, only *100 000* user sessions are considered. From this pool of sessions 80% of the data is used as the training set, and 20% of the data as the test set.

Methods

Logistic Regression

Predictive analysis of the clickstream data is carried out using two different algorithms: logistic regression and neural network. Logistic regression (LR) is commonly used for a binary classification problem. In our prediction, we want to measure the relationship between a dependent variable with binary classification (buy/no-buy decision), and various independent variables, we chose to start with the LR model.

LR model does not assume that the relationship between independent variable and the dependent variable to be linear. It has become a standard classification method as it is easy to use and provides quick and robust results. In our application, the task is to build a classification model that estimates the probability of a user session on a website leading to a purchase based on the parameters we extracted from the clickstream data.

In logistic regression, hypothesis is defined as:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

Where the function g is a sigmoid function. The associated cost function and gradient are [reference]:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right],$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

Neural Network

Deep learning networks use many layers of non-linear processing to model the behavior of the underlying system, i.e. it uses the examples to infer rules for predictive analysis of our clickstream data. For both neural network and logistic regression, we used a regularized cost function to avoid overfitting to data.

Neural network with regularization, the cost function given by [reference],

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\theta(x^{(i)}))_k)\right] + \frac{\lambda}{2m}\left[\sum_{j=1}^{25}\sum_{k=1}^{400}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{10}\sum_{k=1}^{25}(\Theta_{j,k}^{(2)})^2\right].$$

In our model, we used a neural network with 14 input layers and 16 hidden layers. The input layer consists of 14 features form the clickstream data, and we used the sigmoid function for both hidden layer and output layer.

Result & Discussion

First we run the algorithm using only the variables directly available from the clickstream data, and all of the *80 000* samples from our training set, to achieve an accuracy of higher than 90% with both approaches on the first attempt. On further analysis we noticed that both algorithms produced unsatisfactory classifiers and predicated ‘0’, i.e., ‘no buy’. On investigation, it dawned on us that

conventional machine learning approaches do not accurately measure model performance when faced with imbalanced datasets. Since our (or any website) clickstream data is severely imbalanced towards the negative class, the positive class is probably treated by the model as a rare event.

Data imbalance is a well-known problem within e-commerce and banking (e.g., loan default), with several approaches to deal with such datasets. The prominent ones are using resampling techniques, with the main objective of balancing to either increase the frequency of the minority class or decreasing the frequency of the majority class. This is done to obtain approximately same number of instances for both the classes.

In our model, we chose to use random under sampling, i.e., taking only 40% of the samples from the majority class (negative class). This also helped run the model faster, as we are only operating on a fraction of the overall training set vs. the entire set.

The second problem was with getting the model to train better. The set of parameters directly used from the clickstream could not give an accuracy higher than ~65%, and since most features are related, mapping features to a higher dimension did not help either. As a solution, we introduced two new parameters not available in the click stream data. One parameter is introducing a session id index, and the intuition is that a product with higher number of session ids clearly shows consumer preference for that product and purchase as a result. The second parameter is to create a 'product classification list' and to diving the product ids into hot categories, based on the number of sales of these products within the given period.

These two approaches were successful, and helped us achieve accuracy close to 90-95% for both training set and test sets. NN model was better in classification and predication, since it seems to infer the underlying behavior better. It may be possible to increase the accuracy, by increasing input metrics, using other derived metrics based on primary metrics from the data.

Model	**Training Accuracy**	**Test Set Accuracy**	**Sample Size(Train/Set)**
Logistic Regression	**85%**	**91%**	**80k/20k**
Neural Network	**93%**	**95%**	**80k/20k**

Future Work

As mentioned in the prior section, there is potential to improve classification, and NN model for better accuracy, through incorporating more input metrics. Few that come to mind are, time of the day, day of the week, time between sessions and time spent in each session. Getting some more metrics such as user demographics, longtime purchase history from authenticated users could improve prediction quality,

Using longer time-series data (e.g. 3 months or longer) for training to avoid the effect of certain month over customer behavior. Expand our model to predict top 10 selling products next week, i.e., demand forecasting would be few other additions. Also using other techniques such as random forests and deep belief networks and SVMs, etc. In addition to these techniques, further evaluation criteria and model selection methods could be studied to better understand the options suited for different occasions.

References:

http://infiniteanalytics.com

https://www.coursera.org/learn/machine-learning/home/welcome

https://www.coursera.org/learn/machine-learning/home/week/3

https://www.coursera.org/learn/machine-learning/home/week/5

http://164.67.163.139/Documents/areas/fac/marketing/bucklin_clickstream.pdf (last accessed 12/15/2017)

Geoffrey E. Hinton and Ruslan Salkhutdinov. *Reducing the dimensionality of data with neural networks.* *Science* 28 Jul 2006: Vol. 313, Issue 5786, pp. 504-507

https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

https://stats.stackexchange.com/questions/168929/logistic-regression-is-predicting-all-1-and-no-0

MLA: "By Harold D. Hunt." *https://assets.recenter.tamu.edu/documents/articles/1615.pdf*. N.p., n.d. Web. 16 Dec. 2017

https://pdfs.semanticscholar.org/dc25/9823d2e8e66758328dbe533758164384c157.pdf