INTER IIT TECH MEET 2015, IIT KHARAGPUR

STAY RANKING INDEX

January 28, 2015

Abhishek Singh, Puneet Singh, Ved Prakash Gupta
Indian Institute of Technology, Kanpur
http://home.iitk.ac.in/ puneets/Inter-IIT/

Contents

INTRODUCTION

In the current world which is very much connected through social media and relies heavily on user-generated data, it has become essential for businesses to make use of such data in the most efficient way possible. Particularly, for the e-commerce industry, social media analytics and user to user advertising has become an important resource to tap into. Same is true for the tourism industry, especially hotel booking services like, *Stayzilla* or *Tripadvisor*. New user and tourist would often pay a lot of attention to reviews and ratings provided by users who have earlier used their services. 77% of the consumers read reviews before purchasing online and Customer reviews can increase e-commerce conversion rate by 14-76 %. Hotel booking industry is looking at new methods and technical solutions to provide the best information to their users. The aim of this project is to provide such solution which makes use of various techniques from machine learning and text data analysis.

A study illustrated that determinants of customer satisfaction in hospitality venues can be identified through an analysis of online reviews. Using text mining and content analysis of 42,668 online traveler reviews covering 774 star-rated hotels, the study found that transportation convenience, food and beverage management, convenience to tourist destinations and value for money are identified as excellent factors that customers booking both luxury and budget hotels consider important and for which the performance is much satisfactory to them. Customers paid more attention to, but were less satisfied with, bed, reception services and room size and decoration. Most determinants of customer satisfaction also showed a consensus over luxury versus budget hotels, except for factors referring to lobby and sound insulation. So clearly, analyzing user reviews can really help the tourism industry.

PROBLEM FORMATION

Our solution involves a Stay Ranking Index on a scale of 1 to 10 (with 1 being the lowest and 10 being the highest). The Stay Ranking index is based on a model which feeds in the different customer reviews (+ve or-ve) of the stay across the web (includes other travel websites). We have used a sentiment analysis algorithm to give a numerical value to the review and then we aggregate this scores given to various reviews. The Stay ranking is an aggregate of each individual customer review. Once a score is given to the different stays in particular city that is used a valuable sub filter to increase conversions for Stayzilla apart from other filters like distance, price range, star rating and facilities.
The problem can be broken down into 2 parts.

1. Script for a "Crawl" job to get the reviews from different sites.
2. Effective algorithm to rank hotels based on the score for review, price, distance from city, amenities and so on.

TECHNIQUES USED

Latent Dirichlet Allocation (LDA)

Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document.

Following are the notation of LDA :-

- A word is the basic unit of discrete data, defined to be an item from a vocabulary indexed by (1,. . . ,V). We represent words using unit-basis vectors that have a single component equal to one and all other components equal zero.
- A document is a sequence of N words denoted by $w = (w_1, w_2, ..., w_N)$ where w_N is the n^{th} word in the sequence.
- A corpus is a collection of M documents denoted by D=$w_1, w_2, ..., w_M$.

Idea behind LDA : The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. We wish to find a probabilistic model of a corpus that not only assigns high probability to members of the corpus, but also assigns high probability to other "similar" documents.

In Mathematical terms :
LDA assumes the following generative process for each document w in a corpus D:

1. Choose $N \sim Poisson(\zeta)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_N :
    - Choose a topic $z_N \sim Multinomial(\theta)$.
    - Choose a word w_N from $p(w_N|z_N, \beta)$, a multinomial probability conditioned on the topic z_n .

How the words in documents generated :-
Each word comes from different topics (bag of words)

$$P(word) = \sum_{k=1}^{K} P(Topic_k)P(word|Topic_k)$$

Extreme Learning Machine

Gradient descent based algorithms require all the weights be updated after every iteration. So, gradient based algorithms are generally slow and may easily converge to local minima. On the other hand, ELM, proposed by Huang et al. in 2004, randomly assigns the weights connecting input and hidden layers; and hidden biases. Then it analytically determines the output weight using the Moore-Penrose generalized inverse. It has been proved in that given randomly assigned input weights and hidden biases with almost any non-zero activation function, we can approximate any continuous function on compact sets. Unlike the traditional algorithms, ELM not only achieves the minimum error but also assigns the smallest norm for the output weights. The reason for using Moore-Penrose inverse is that according to BartlettâĂŹs theory, smaller norm of weights results in better generalization of the feedforward neural network. The advantages of ELM over traditional algorithms are as follows

- ELM can be up to 1000 times faster than the traditional algorithm.
- ELM has better generalization performance as it not only reaches the smallest error but also the assigns smallest norm of weights.
- Non-differentiable functions can also be used to train SLFNs with ELM learning algorithms.

Given a training set $N = \{(x_i, t_i) | x_i \in R^n, i = 1, ..., n \text{ and } t_i \in R^m\}$, activation function $g(x)$ and hidden node number Ñ the mathematical equation for the neural network can be written as :

$$\sum_{i=1}^{N} \beta g(w_i x_j + b_i) = o_j, j = 1, ..., N$$

where $w_i = [w_{i1}, w_{i2}, w_{i3}, .., w_{in}]^T$ is the weight vector connecting the i_{th} hidden neuron and the input neurons and $\beta_j = [\beta_{j1}, \beta_{j2}, .., \beta_{jm}]^T$ is the weight vector connecting the i_{th} hidden neuron and the output neurons. t_j denotes the target vector of the input x_j whereas o_j denotes the output vector obtained from the neural network. $w_i.x_j$ denotes the inner product of w_i and x_j. The output neurons are chosen to be linear.
Standard SLFNs with Ñ hidden neurons with activation function $g(x)$ can approximate these N samples with zero mean error means that $\sum_{j=1}^{N} ||o_j - t_j|| = 0$, i.e, there exists β_i, w_i and b_i such that $\sum_{i=1}^{N} \beta g(w_i x_j + b_i) = o_j, j = 1, ..., N$

The above n equations can be written compactly as: $H\beta = T$ where

$$H = \begin{vmatrix} g(w_1 x_1 + b_1) & ... & g(w_N x_1 + b_N) \\ : & : & : \\ g(w_1 x_n + b_1) & ... & g(w_N x_N + b_N \end{vmatrix}$$

$$\beta = \begin{vmatrix} \beta_1^T \\ : \\ \beta_N^T \end{vmatrix}$$

and

$$T = \begin{vmatrix} t_1 \\ : \\ t_N \end{vmatrix}$$

The smallest norm least squared solution of the above linear system is:

$$\beta = H^{\dagger} T$$

The algorithm for the ELM with architecture as shown can be summarized as follows:
Given training sample N, activation function $g(x)$ and number of hidden neurons Ñ

1. Assign random input weights w_i and bias $b_i, i = 1...\tilde{N}$
2. Calculate the hidden layer output matrix H.
3. Calculate the output weight $\beta = H^{\dagger} T$

METHODOLOGY

Web Crawler

The first and the foremost requirement of the project was a web crawler which would be used to get the data from various travel advisory websites such as *Tripadvisor.com*. The type of data to be crawled included the list of hotels and different kinds of stays in a particular city. It also got all the reviews written by users for all the stays. Apart from this *Tripadvisor* also provides ratings of different stays on the basis of rooms quality, sleep quality, location and service. These ratings were also used in calculating the stay ranking index. The web crawler is programmed in the python language. The *.py* file is made available in the appendix.

Hotel Review Analysis

Implementation of LDA for Hotel Review Analysis

We first collected some sample reviews of hotels. Then we divided our corpus into two parts of which one part was used as the training data and accuracy of the model is checked by the remaining part. To **'increase efficiency'** of the model, we have changed our training and test data sets repeatedly.

Accuracy Check Concept

For this analysis,we have collected total 250 txt files of reviews consisting of different categories i.e positive and negative reviews. Now we have made training dataset by picking up 100 files from each category and remaining 50 files from each category are used for the testing the model.
So our training dataset consists of total 200 (100*2) review files. Similarly test data set consists of total 50(25*2) files.
As LDA gives us probability matrix of document and topics where each element represents the probablity of given word of that file occuring in different topics.
Hence by LDA analysis on this training sample,we have got the probability matrix.

How to Check Accuracy

We have performed following analysis to calculate accuracy :
We took the first vector of prob. Matrix of training data and took the difference with test data's prob. Matrix first 25 entries and added all the elements of vector to get a single number,after performing same analysis over all the vectors, we took the average of these 100 elements and that represents the a measure of positive review. Similarly we performed same analysis with remaining 25 entries of test data and got the measure for negative review.

Now
if *min of these two measured indexes == positive review index*
Give that review a "Positive Rating"
otherwise
Give that review a "Negative Rating"

Accuracy = (No. of Reviews we got predicted correct from the accuracy check analysis) /(Total No.of Reviews in the test dataset)

After performing this analysis on different combinations of training and test data sets, we got a consistent measure of accuracy.

Once the desired level of accuracy is achieved, then only we applied this model to new corpus of review files of different hotels.

Extreme Learning Machine on Stay Data

A comma separated data file with 235 data vectors corresponding to every stay in the city of Goa, India. Each vector contains the name of the stay, its ranking on the parameters of the reviews provided by the users, location, sleep quality, rooms, service, value and cleanliness.The set of 235 data points was divided into a training set of 180 data vectors and 55 cross validation set for checking the validity of the model.
The mathematical technique of extreme learning machine has been used to this analysis and make the model. The code is written in R programming language and the *'elmNN'* package is used to train the model.
A successful model implementation is obtained by first changing all the parameter values to lie in a range of o to 1. This is done by using the *min-max normalization.*

$$Y = \frac{X - min(X)}{max(X) - min(X)}$$

Then the extreme learning machine algorithm is applied on the training data using the *elmtrain* function which uses implementation of ELM (Extreme Learning Machine) algorithm for SLFN (Single Hidden Layer Feedforward Neural Network). ELM algorithm is an alternative training method for SLFN (Single Hidden Layer Feedforward Networks) which does not need any iterative tuning nor setting parameters such as learning rate, momentum, etc., which are current issues of the traditional gradient-based learning algorithms (like backpropagation).
Training of a SLFN with ELM is a three-step learning model:
Given a training set P = $(x_i, t_i)|x_i \in R, t_i \in R, i = 1, ..., N$, hidden node output function G(a, b, x), and the number of hidden nodes L

1. Assign randomly hidden node parameters (a_i, b_i), i = 1,..., L. It means that the arc

weights between the input layer and the hidden layer and the hidden layer bias are randomly generated.

2. Calculate the hidden layer output matrix H using one of the available activation functions.

3. Calculate the output weights B: B = ginv(H) % ∗ % T (matrix multiplication), where T is the target output of the training set.

ginv(H) is the Moore-Penrose generalized inverse of hidden layer output matrix H. This is calculated by the MASS package function ginv. Once the SLFN has been trained, the output of a generic test set is simply Y = H% ∗ % multiplication). Salient features:

- The learning speed of ELM is extremely fast.
- Unlike traditional gradient-based learning algorithms which only work for differentiable activation functions, ELM works for all bounded non constant piece wise continuous activation functions.
- Unlike traditional gradient-based learning algorithms facing several issues like local minima, improper learning rate and over fitting, etc, ELM tends to reach the solutions straightforward without such trivial issues.
- The ELM learning algorithm looks much simpler than other popular learning algorithms: neural networks and support vector machines.

The number of nodes in the hidden layer were varied till the best results were obtained. The number of nodes in the single hidden layer was decided to be five. The validity of the model was checked in terms of mean square error.
Once the model was developed, the same model was used to predict the ratings as a function of explanatory variables. An observation was made that the mean square error in the cross validation set was approximately double of that of training set which is same as many other studies. Once the predicted value of ratings were obtained, they were to be converted back to a ascale of 0 to 10. This was done using the formula:

$$R = \frac{X - min(X)}{max(X) - min(X)}$$

where X is the obtained rating vector.

REFERENCES

David M. Blei, Andrew Y. Ng, Michael I. Jordan, *Latent dirichlet allocation*, The Journal of Machine Learning Research, Volume 3, 3/1/2003, Pages 993-1022

G.B. Huang, Q. Zhu, C.Siew, *Extreme Learning Machine: A New Learning Scheme of Feed-forward Neural Networks*, International Joint Conference on Neural Networks (2004), Vol. 2, pp:985-990

G.B. Huang, Q. Zhu, C. Siew, *Extreme Learning Machine: Theory and Applications*, Neuro-computing (2006), Vol. 70, pp: 489-501

G.-B. Huang, H. Zhou, X. Ding, R. Zhang (2011) *Extreme Learning Machine for Regression and Multiclass Classification* IEEE Transactions on Systems, Man, and Cybernetics - part B: Cybernetics, vol. 42, no. 2, 513-529

http://cran.r-project.org/web/packages/elmNN/elmNN.pdf

Thelwall, M. (2001). *A web crawler design for data mining.* Journal of Information Science, 27(5), 319-325. Chicago

Huiying Li, Qiang Ye, Rob Law, *Determinants of Customer Satisfaction in the Hotel Industry: An Application of Online Review Analysis*,Asia Pacific Journal of Tourism Research,01/2012

ANNEXURE

Results from LDA

Accuracy of LDA : 90%

Results from ELM

Training Error	Cross Validation Error
0.68	1.99