Sentiment Analysis for the Restaurant Reviews

Maulik Lalani mlalani@ufl.edu

***Abstract*— In the recent years, there has been tremendous increase in the data because of the boom in the IT industry. According to one study there was more than 1.2 million terabytes of data on the internet in 2013. Many multinational companies and universities are developing and doing research on topics like data extraction, data mining, machine learning, data science, and pattern recognition. Many tools are being developed to find the valuable insights from huge amount of data. This technique is called sentiment analysis. It has become very crucial and important to find relevant, accurate and vital information from huge amount of data which would help many multinational companies to make important business decisions. Sentiment analysis basically studies people's reaction and sentiments towards some product. The main source of data for the sentiment analysis is the internet. Also, now a day's researchers are getting data easily because social media company releases API to extract the data from their web site.**

In this project, I have chosen hotel domain to do the sentiment analysis of the hotel and restaurant reviews and divide it into two categories like positive or negative and deceptive or truthful.

***Index Terms*—sentiment analysis, Naïve Bayes, Term Frequency Inverse Document Frequency(TF-IDF), NLTK library, requests API, Yelp API.**

I. Introduction

21st century is all about the internet. Every domain like aerospace, computer science, electronics, health care, event management, hotel management, tourism, transportation, and many more uses internet in one or the other form. Because of this, there has been tremendous increase in the data and it is increasing rapidly. Till 2016, there were around 4.62 billion web pages on the web. Also, as of 2014, Google had calculated 200 terabytes of data. Companies like amazon, Microsoft, Google are creating more and more data centers and finding ways to store and access this data efficiently. Different technologies like python, Hadoop, R, MongoDB, AWS services etc. are used to efficiently calculate and store this data. However, one more challenge that researchers are facing today is how to interpret the data and find the valuable insights out of it. It is not possible to visually analyses tera bytes of data and find the results. Hence different techniques of machine learning, statistical modeling and pattern recognition is used. Many of the techniques use past data to predict the future results. There are statistical models like regression which keeps few matrices constant and tries to find the effect of other matrices on the output.

In this project, I have tried to solve this problem for the restaurant reviews using Naïve Bayes algorithm and using python libraries. It uses the previous data to create the model. Using this project, user can categorize the reviews into two categories as either truthful or deceptive and either positive or negative based on the past data. Using the first category, user can determine whether the review is genuine or whether it is fake. Second category is used to tell the user whether it is a positive review or negative review. Also, one more advantage of the project is that user can feed thousands of reviews in one go and within milliseconds he can see the results and decide whether to choose that restaurant or go for another one. Along with that other kind of classifiers are also studied to find which one is suitable for any scenario.

The main source of data for these kind of project is the internet. Also, now a day's researchers are getting data easily because social media company releases API to extract the data from their web site. In this project, I had extracted the data from the Yelp website. However, because of the limited access and security reasons, I was not able to extract the whole text of the hotel review. Hence, I have used dataset from the USC's NLP dataset.

II. Description

The main problem today that I described in the previous section was about the data interpretation and prediction. I tried to solve this problem using the past data and using Naïve Bayes classifier.

Naïve Bayes classifier is very simple but powerful algorithm for predictive analysis. It provides the way to calculate the probability of hypothesis based on the prior knowledge. Advantages of Naïve Bayes classifier is that it is very simple, it works when there is logical independence between features. The main disadvantage of Naïve Bayes algorithm is that it does not account for the interaction between different features.

P(h|d) = (P(d|h) * P(h)) / P(d)

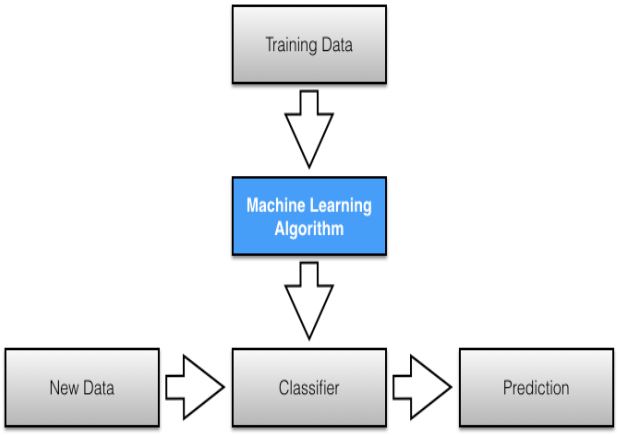
Where

- **P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.
- **P(d|h)** is the probability of data d given that the hypothesis h was true.
- **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- **P(d)** is the probability of the data (regardless of the hypothesis).

In the above equation p(h|d) is the posterior probability and P(d|h), P(h) and P(d) are the prior probability. Naïve Bayes classifier is easiest to understand because each of the classes d1, d2, d3 are considered to be independent and rather than calculating P(d1,d2,d3), values P(d1|h)*P(d2|h)*P(d3|h) are calculated for different hypothesis. Result is the maximum of all the classes and is written as follows:

MAP(h) = max(P(d|h))

P(h) has been dropped because it is the constant term.



I have used the above approach to find the outcome. I have divided the data into training dataset and test dataset. The input data have unique ID for each review followed by a review. The program has 2 python files one for creating the model and other for the development. The input files for creating the models are the reviews file and desired output file respectively.

Before creating the model, the reviews are converted to lower case, punctuations are removed, duplicates are removed, stemming is done using python libraries and finally the data is ready for the modeling. I have used four classes called Truthful, Deceptive, Positive and Negative. There are two kinds of probabilities stored in the model files for the classes.

Class Probabilities: Overall probability of each class in the training dataset.
Conditional Probabilities: The conditional probability of each input word from the review given each class value.

Following is the representation of the Bayes Theorem for my project.

P(class|word) = (P(word|class) * P(class)) / P(word)

Where
P(class|word) = probability of a review being in a class given the word in a sentence is deceptive.

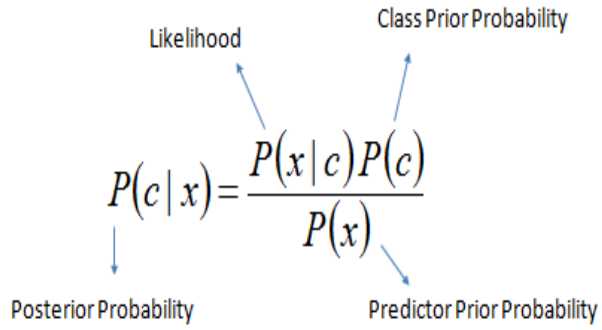
P(word|class) = probability of a word in a sentence is a class given the sentence is present in that class.

P(class) = Probability of a review to be in class.

P(word) = Probability of a word in a sentence to be in class.

P(word) is a normalizing term and helps us in finding the probability. Hence, I can drop it because we are interested in the most probable hypothesis as it is used to normalize and constant.

Finally, my model files have prior probability of all the classes and probability of all the unique words for all the four classes i.e. positive, negative, deceptive and truthful. This model is saved in a text file. Now when a user enters the review, sentence is broken down into words, it is converted to lower case, punctuations are removed, duplicates are removed, stemming is done using python libraries. Now, probability of all the four classes are calculated for each word using following formula:



$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

So, for each class, probability of all the hypothesis are multiplied along with the probability of total probability as follows:

P(x1,…, xn | c) = P(x1 | c)•P(x2 | c)•P(x3 | c)•...•P(xn | c)

FinalOutput = max(P(cj) P(x | c)) where x =x1, x2, …xn
Finally, one output is the maximum of deceptive and truthful and the other output is the maximum of positive and negative.

III. EVALUATION

I have divided my dataset into training dataset for creating the model and remaining data for testing purposes. Hence, I will run my test dataset in the classifier and check whether my output is correct or not.

Time taken to run my dataset is around **0.2 seconds** and efficiency is **79%.**

To improve the evaluation process, F score is also used. F score is basically used when you must measure the precision and recall. It is the harmonic mean of precision and recall.

Precision is defined as follows:

p=tp/(tp+fp)

tp = true positives and fp = false positive

Recall is defined as follows:

r=tp/(tp+fn)

tp = true positive and fn = false negative
Hence, recall is the ratio of true positives to all that are classified correctly while precision is the ratio of true positives to all the positives.

Finally, F score is the harmonic mean of recall and precision.

F = 2tp/(2tp+fp+fn)

We have used harmonic mean because it is best when dealing with ratios.

	Precision	Recall	F
Deceptive	0.81	0.9	0.85
Truthful	0.89	0.79	0.84
Negative	0.94	0.93	0.93
Positive	0.93	0.94	0.93

Weighted Avg: 0.89

Although this project gives correct output, we can still improve it if we do further process it using NLTK library. For example, words like go, gone, going etc. can be clubbed together and can be made single word ‘go’ to avoid the duplicates. Also, there are many different algorithms like K nearest neighbors, support vector machines etc. discussed in the next section.

IV. RELATED WORK

There is always a question which classifier algorithm to use whenever we want to do sentiment analysis. There are many factors that we should keep in mind before selecting the right one like size of the training data set, are classes independent, system requirement in terms of performance etc. This section explains different classifiers along with the comparison with the Naïve Bayes classifier.

A. Logical regression:

Logical regression determines the relationship between binary dependent variables (only two states i.e. either 1 or 0) with one or more independent variable. It determines the best fitting model by giving the appropriate coefficients for the formula to predict the logical transformation.

Logistic Regression

❖ A General Model:

$$\text{logit}(p_{disease}) = \log\left(\frac{p_{disease}}{1-p_{disease}}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_J X_J$$

Where:

$p_{disease}$ is the probability that an individual has a particular disease.

β_0 is the intercept

$\beta_1, \beta_2 \ldots \beta_J$ are the coefficients (effects) of genetic factors

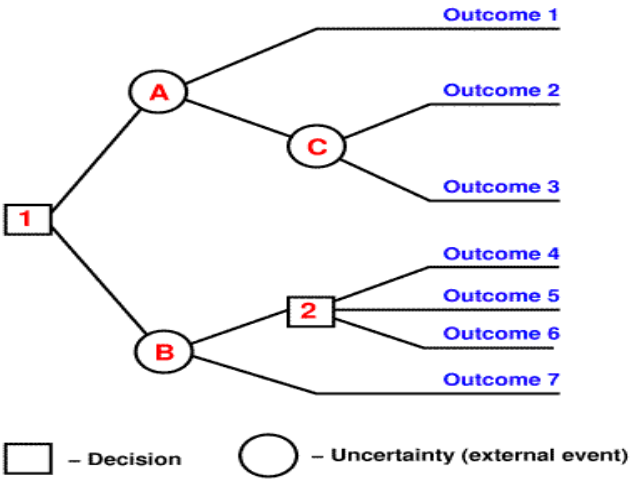
$X_1, X_2 \ldots X_J$ are the variables of genetic factors

The main advantage of logical regression is that there are many ways by which we can make the model and we do not have to worry about the relation between the models. However, this model fails if we do not have good relation between the features.

B. Decision tree:

Decision tree forms the model in the form of tree where each node is a ‘test’ and each branch is the possible outcome of the result as shown in the figure below. It is basically used when you want to make some decision and the path from root to the leaf forms the classifier rules for the model. The main advantage of the decision tree is that it is very easy to understand and explain. Very little effort is required for the

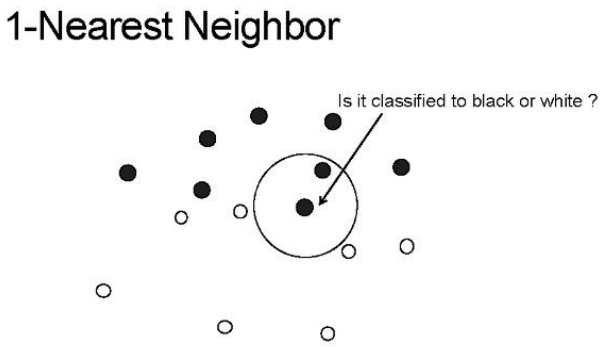
data preparation for decision tree algorithm because there no need to do scaling for the numeric values because it does not affect the tree. Decision trees are also not sensitive to outliers since the splitting happens based on proportion of samples within the split ranges and not on absolute values. Decision trees are also not sensitive to the missing values. Nonlinear relations do not affect the performance. It is very easy to interpret and explain the logic to the collogues. **Hence, decision tree is used when relationships are highly nonlinear and data is not proper.** One disadvantage of decision tree is that tree has to be reconstructed whenever the new data comes.



C. K nearest neighbors:

The K nearest neighbor's classifier is based on the premise that the classification of unknown instances of data can be done by relating the unknown to the known using a similarity or a distance function. We can infer that two instances which are far apart from each other in space are less closely related to each other than the case where two instances are closer to each other in space.

Unlike other machine learning algorithms K nearest neighbor's algorithm doesn't gather any information from the training data during the learning phase. In general, the k nearest neighbor's algorithm works in this way. Classification in its most basic form is finding the nearest neighbor in its instance space and labeling the own self with the same class label as that of the located neighbor.



In the above diagram, to find whether the given point is black or white, 1 nearest neighbor is used. i.e. it looks for the nearest one point to find the result. If it was 2 nearest neighbor, then it would look for 2 nearest points and determine the result.
The advantage of this algorithm is that no assumption must be made. Also the cost of learning is zero for this classifier. This algorithm finds the probability of a class from the k closest point to the given input. Unlike other algorithms, it does not need to preprocess the model. However, because of this, speed of the algorithm is reduced when there is huge amount of data because we will have to calculate the minimum distance for each point. **Hence This algorithm is used when the data is less and you don't want to spend resources in training the data.**

D. Maximum Entropy:

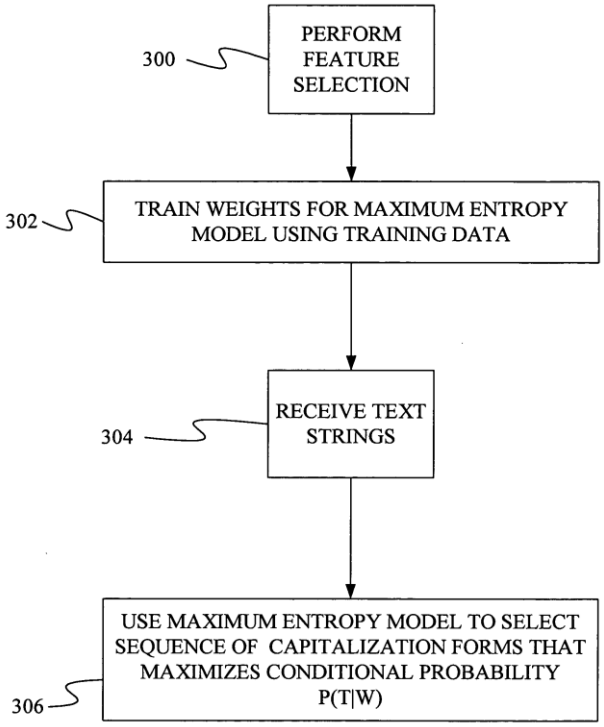
This algorithm tries to maximize the entropy along with satisfying all the constraints. Entropy is nothing but a unit of measure of uncertainty. For example, entropy of the things that you know would be 0 and entropy of the complex data would be very high. It is similar to Naïve Bayes algorithm but the difference is it does not have independent features but rather finds weights of the feature using search based optimization that maximizes the likelihood. It can also handle Booleans and integers.
Steps for implementing maximum Entropy are given below:

1) *Define joint frature f(w,c) =N where N = number of occurrences of c.*
2) *Assign weight to the joint feature using optimization and because of this step, this algorithm is time consuming.*
3) *Formula for calculating the probability of classes is given by the following equation.*

$$P(c|d,\lambda) \stackrel{def}{=} \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c' \in C} \exp \sum_i \lambda_i f_i(c',d)}$$

The step two in the above process is very time consuming and hence Naïve Bayes classifier is faster but it does not handle

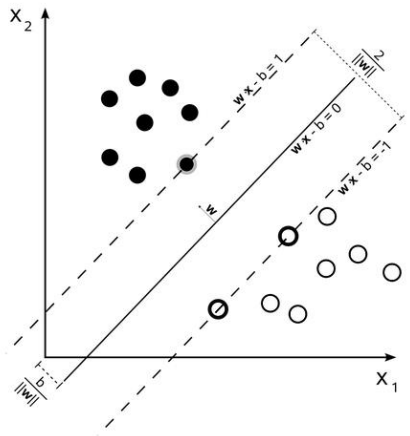
joint features. Below diagram illustrates the steps as mentioned above.



In the above diagram, label 300 defines the joint feature and next step labelled 302 trains the model by assigning the weight and the last step is defining the classifier which gives the result. **Maximum entropy is used when there is a relationship between features.**

E. Support Vector Machines

It is a simple linear regression algorithm. It divides two classes using hyperplane as optimally as possible. There should be as much points as possible on the one side as much as they are on the other maximizing the distance of each point to this hyperplane.



White circles and black circles are the two categories in which the dataset is divided and SVM will always choose those classes that maximizes the margin between them. The hyper plane is given by the following formula:

$$\langle \vec{w} \cdot \vec{x} \rangle + b = \sum_i y_i \alpha_i \langle \vec{x_i} \cdot \vec{x} \rangle + b = 0$$

where xi are the n dimensional input vectors, yi is the output values, wi is the weight vector and ai is the Lagrangian multiplier.

Hence, once we have this vector using the training dataset class of other vector are decided using w* xi+b>=0 then it belongs to the positive class otherwise negative. If we have more than 2 classes than we can use the kernel function to map the input data into higher dimensional vectors. If the kernel is linear than SVM is like the logical regression. But people use SVM because it is very accurate and it is very popular for the text classification. **SVM is used when there is a very complex relationship between the features and when you want to perform both regression and classification.**

V. Conclusion

To summarize, I have implemented Naïve Bayes classifier to do the sentiment analysis on the restaurant reviews and find out whether the review is negative or positive and whether it is truthful or deceptive. Through this project, I have learnt at least 5 different kinds of classifiers, their advantages, disadvantages and most important, when to use which classifier. K nearest algorithm is used when the data is less and you don't want to spend resources in training the data. Decision tree is used when relationships are highly nonlinear and data is not proper.

Maximum entropy is used when there is a relationship between features. SVM is used when there is a very complex relationship between the features and when you want to perform both regression and classification. I have learnt a lot about steps to be followed to do the sentiment analysis, what is a model and a F score. Along with that, this project helped me improve my coding skills in python. This project is the basic version and we can still add many things to it. One of them is stemming as a part of pre processing step. Also, because of this project I learnt about the Yelp data challenge contest where you compete with people all around the world and work on the datasets given by them to do some research in topics like Location mining, Seasonal trends, NLP, and social graph mining. I will be participating in this contest this year.

References

[1] http://machinelearningmastery.com/naive-bayes-for-machine-learning/
[2] http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/
[3] https://www.quora.com/What-are-the-best-supervised-learning-algorithms-for-sentiment-analysis-in-text
[4] https://web.stanford.edu/class/cs124/lec/naivebayes.pdf
[5] https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering
[6] http://machinelearningmastery.com/naive-bayes-for-machine-learning/
[7] http://acl-arc.comp.nus.edu.sg/archives/acl-arc-090501d4/data/pdf/anthology-PDF/W/W04/W04-3253.pdf
[8] http://ieeexplore.ieee.org/document/6914200/?reload=true
[9] https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_nb.htm#BABIIDDE

[10] https://www.quora.com/What-are-the-advantages-of-different-classification-algorithms
[11] http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/
[12] http://www.cs.upc.edu/~bejar/apren/docum/trans/03d-algind-knn-eng.pdf
[13] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[14] http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/
[15] https://pdfs.semanticscholar.org/8a73/74b98a9d94b8c01e996e72340f86a4327869.pdf
[16] https://en.wikipedia.org/wiki/Decision_tree
[17] http://www.saedsayad.com/decision_tree.htm
[18] https://www.medcalc.org/manual/logistic_regression.php
[19] https://en.wikipedia.org/wiki/Logistic_regression
[20] http://www.statisticssolutions.com/what-is-logistic-regression/
[21] https://www.google.com/search?q=maximum+entropy+model+diagram&rlz=1C1CHBF_enUS717US718&source=lnms&sa=X&ved=0ahUKEwiSz4_8zsHTAhVs2IMKHZihBrkQ_AUIBSgA&biw=1280&bih=566&dpr=1.5#q=what+is+support+vector+machine
[22] http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
[23] https://en.wikipedia.org/wiki/Support_vector_machine
[24] http://acl-arc.comp.nus.edu.sg/archives/acl-arc-090501d4/data/pdf/anthology-PDF/W/W04/W04-3253.pdf
[25] https://nlp.stanford.edu/pubs/sidaw12_simple_sentiment.pdf
[26] http://sentiment.christopherpotts.net/classifiers.html