Fake News Detection

Machine Learning - CSL603

Prasad Kshirsagar
Computer Science Department
IIT ROPAR
2016csb1041@iitrpr.ac.in

Harshvardhan Dogra
Computer Science Department
IIT ROPAR
2016csb1120@iitrpr.ac.in

Sahil Kumar
Computer Science Department
IIT ROPAR
2016csb1057@iitrpr.ac.in

ABSTRACT

As we know, In today's world social media has become largest & fastest medium of news spread. Some of the news are rumours while some contain truth. So, it is important for us to identify which ones are Fake and which ones are not. Thus, our goal is to build a model which classifies the news as either Fake or True. Our model is designed to emulate the functionality of BS Detector, a popular chrome extension that automatically labels articles,websites as BS.

KEYWORDS

Fake, True, Neural Network, SVM, Logistic Classification, Decision Tree, Naive Bayes Classifier

1 INTRODUCTION & MOTIVATION

We get to hear news on television, through newspapers and also from social media platforms. Some of these media platforms provide true news while some provide fabricated news. Thus, we need a freely available tool to decide trustworthiness of a news. With this motivation, We started working on this project. We trained various machine learning models such as Decision Tree, Logistic Classification, SVM, Neural Network etc. We took 6000 True samples & 6000 Fake samples, We converted words into numbers using NLP in python. Having this big dataset, We preprocessed & cleaned the data and tested it using SVM, decision tree, Logistic classification, neural network. After this we analyzed the results & found results parallel to our expectation.

2 RELATED WORK

2.1 Text Processing

In this, We use tokenization with removal of stop words and also taken into account lemmatization. The context of the words, however is not taken into consideration. All the words having length less than 3 were removed & also all the words containing special characters were removed. Also, some manual improvements were done to improve attributes set.

2.2 Machine Learning Tools

Efforts have been made to realize different types of fake news in [6],Three types of fake news are presented. It includes Serious Fabrications, Large-Scale Hoaxes and Humorous Fakes. Serious Fabricated news may take substantial efforts to collect, case by case. Large-Scale Hoaxing attacks are creative,unique and often multi-platform which may require methods beyond text analytics(eg: network analysis).

3 METHODOLOGY

We have implemented 5 Machine Learning models. They are as follows :

- Decision tree
- Logistic classification
- Neural Network
- Naive Bayes Classifier
- Support Vector Machines.

Our training data-set contains 12000 examples (6000 for fake news[4] and 6000 for true news[5]) and and test set consists of 2500 randomly selected examples. The attributes for the above methods are selected based on their frequency from the vocabulary of the dataset.

4 EXPERIMENTAL SETTINGS

4.1 Decision Tree

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller subsets with increase in depth of tree. The core algorithm for building decision trees is called ID3.

ID3 uses Entropy and Information Gain to construct a decision tree. The formula for information gain is given by:

$$IG = E1 - E2$$

Where,
IG = Information gain
E1 = Entropy of distribution before the split
E2 = Entropy of distribution after split
and
$entropy(p_1, p_2, \ldots, p_n) = -p_1 log(p_1) - p_2 log(p_2) - \cdots - p_n log(p_n)$
We have used 200 words in the vocabulary (after preprocessing) as our features. These 200 words were given by the naive bayes classifier with 100 words having maximum posterior probability of being in fake news and 100 with minimum posterior probability. The ID3 algorithm was implemented without and with early stopping criteria of stopping at less than or equal to 50 examples per node. The early stopping ID3 gave better accuracy on the training set and hence was chosen for our results. Decision trees as we know is prone to overfitting and the best way to correct this problem is to use random forests i.e. create multiple trees using feature bagging or instance bagging and then the majority of the results given by the trees is the acepted answer. We implemented random forests using feature bagging. The forest comprised of 15 decision trees with each tree having 120 attrbutes.

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

Figure 1: Logistic Regression Cost function

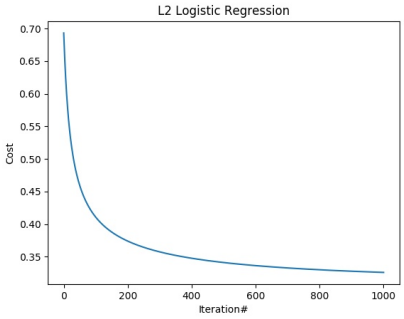


Figure 2: L2 Regression Cost vs Iterations

4.2 Logistic Classification

It uses a logistic function to model a binary dependent variable. Logistic function, S(z) is represented as

$$S(z) = \frac{1}{1+e^{-z}}$$

We use cross-entropy loss function which is represented in the cost function as 1. We have applied gradient descent for convergence of the weights of the model. The weight update has been done both using both *L1 norm* and *L2 norm*. *L1 norm* is represented as

$$\|x\|_1 = \sum_i |x_i|$$

L2 norm is represented as

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

The parameters to consider here are the learning rate for the gradient descent and the number of iterations for the gradient descent to converge and they are determined heuristically.

The features selected here are of two types:

- Take top 5000 frequency words as the features and initialize a random weight vector of size 5000.
- Use Naive-Bayes classifier to learn the top 200 words out of the vocabulary that give the highest posterior probability. We call these words as most *indicative words[3].*

As it turns out later, the second option gives better accuracy as compared to first one. We have calculated the accuracy of correct classification of fake news plus the confusion matrix.

4.3 Neural Network

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. We have used neural network on the dataset of 5000 attributes and 12000 training examples and it is tested on 2000 examples.

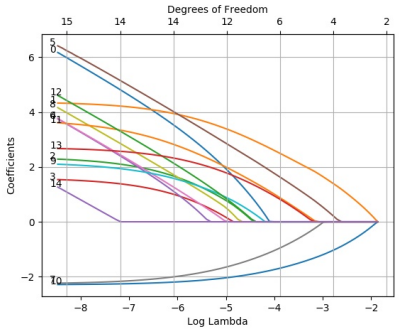


Figure 3: L1 Regression using python glmnet[1]

Binary cross-entropy loss function[2] is used with the metric being the accuracy of the binary classification of true and fake news. The parameters used are learning rate, the batch-size and the epoch-size. They are heuristically determined in our case. Adaptive Moment Estimation(Adam) optimizer is also used in the model.

- When we use 5000 attributes for training, then the architecture of the neural network has 5000 as the input neurons and 1 output neuron with 2 hidden layers. The hidden layer 1 has 500 neurons and hidden layer 2 has 50 neurons. We have tried various activation functions such as *sigmoid, ReLu, tanh, softmax*, but in this case, the maximum accuracy was achieved with all the activation functions being sigmoid.
- When we used top 200 most indicative words as our attributes, we only kept 1 hidden layer with the input layer to hidden layer having *sigmoid* and hidden layer to output having *softmax* activation function. The input layer had 200 neurons and the hidden layer had 20 neurons. This architecture produced experimentally highest accuracy.

4.4 Naive Bayes Classifier

Given the size of our feature space, Naive Bayes classifier estimation is best suited to begin our analysis. So, maximum likelihood parameters are given by

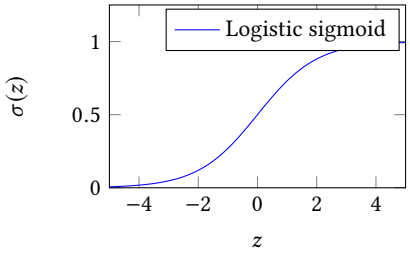
$$X_{j|y=1} = \frac{\sum_{i=1}^{m}\{x_j^i = 1 \cap y_j^i = 1\} + 1}{\sum_{i=1}^{m}\{y_j^i = 1\} + m}$$

$$X_{j|y=0} = \frac{\sum_{i=1}^{m}\{x_j^i = 1 \cap y_j^i = 0\} + 1}{\sum_{i=1}^{m}\{y_j^i = 0\} + m}$$

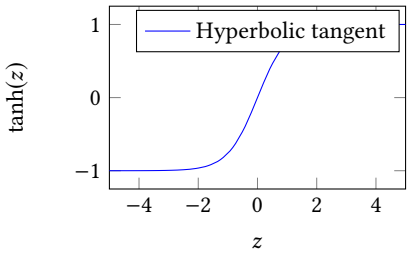
Using our naive bayes algorithm, We identified top 200 tokens that were found to be most indicative of classification of the example. This was computed by finding 100 tokens which have highest posterior probability of being in fake news and 100 tokens with lowest posterior probability of being in fake news.

4.5 Support Vector Machines

As we know, Support Vector Machine can classify the data points even in higher dimensional feature space, if the data points are not separable in the given feature space. The SVM algorithm uses



(a) Logistic sigmoid activation function.



(b) Hyperbolic tangent activation function.

Figure 4: Common used activation functions include the logistic sigmoid $\sigma(z)$ and the hyperbolic tangent $tanh(z)$.

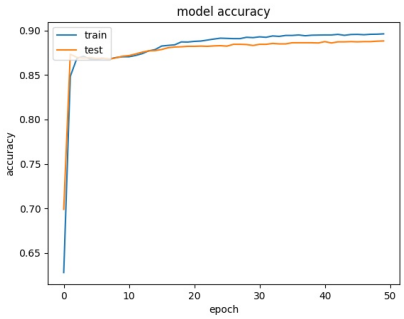


Figure 5: Neural Network accuracy

Hinge loss that seeks to maximize the margin between the two classes of the data. The SVM algorithm uses a second-order Gauss Kernel that operates on 5000 chosen attributes feature space. The expression for this kernel is given by following expression :

$$G(x;\sigma) = \frac{1}{\sqrt{2\pi}} \exp(x^2/2\sigma^2)$$

Note that, this expression is provided for 1-D case. In retrospect, the selection of this high-order kernel seems rather naive, since it may have caused the SVM model to over fit the training set.

5 RESULTS & DISCUSSION

- As Table 1 shows confusion matrix for SVM, We can clearly conclude that, Though SVM classifies the data most accurately, still it predicts significant number of true news as Fake.

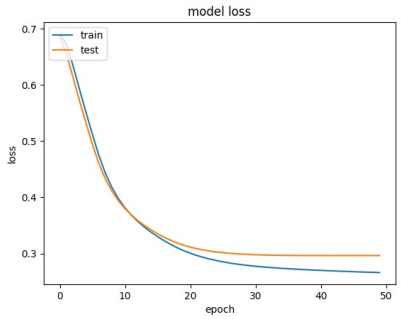


Figure 6: Neural Network loss

Table 1: Confusion Matrix for SVM

	Predicted True	Predicted Fake
Actual True	775	225
Actual Fake	74	926

Table 2: Results

Model	Accuracy (%)
Decision Tree	76.62
Logistic Classification	70.29
Neural Network	78.36
Naive Bayes Classifier	75.11
Support Vector Machines	83.67

- Now, Coming to the Neural Networks, here NN is giving reasonable results but due to sparsity of dataset, results are not good as expected.
- For decision tree, When we apply feature bagging accuracy increases from 76% to 81 %.
- As data points are sparse, it it justified to apply L1 norm using logistic regression in comparison with L2 norm since L1 norm remains unaffected by outliers.
- The most indicative points(200 attributes) dataset gives better results than the dataset of 5000 most frequently occurring words in the vocabulary.

6 SUMMARY & FUTURE WORK

To summarize our project, As expected SVM gives the best results but surprisingly neural network and logistic classification are not giving results as hoped. A lot of our results circle back to the need for acquiring more data. Generally speaking, simple algorithms perform better on less (less variant) data. Since we had less data, SVM and Naive Bayes outperformed Neural Networks, and Logistic Regression did not perform well. Given enough time to acquire more fake news data, and gain experience in python, we will try to better process the data using n-grams, and revisit our deep-learning algorithm. We tried using our own codes for the project, and the algorithms were relatively slow. To tweak all knobs of various algorithms, we shall use available robust packages in the future.

REFERENCES

[1] [n. d.]. Glmnet Vignette (for python). https://glmnet-python.readthedocs.io/en/latest/glmnet_vignette.html

[2] Jason Brownlee. Jun 7, 2016. Binary Classification Tutorial with the Keras Deep Learning Library. https://machinelearningmastery.com/binary-classification-tutorial-with-the-keras-deep-learning-library/

[3] Christopher D. Manning Jeffrey Pennington, Richard Socher. [n. d.]. *GloVe: Global Vectors for Word Representation.* Ph.D. Dissertation. Computer Science Department, Stanford University, Stanford.

[4] Sriraman R. Krishnamurthy. [n. d.]. Fake News NLP Stuff. https://www.kaggle.com/rksriram312/fake-news-nlp-stuff/notebook

[5] Andrew Thompson. [n. d.]. All the news. https://www.kaggle.com/snapcrack/all-the-news

[6] Yimin Chen Victoria L. Rubin and Niall J. Conroy. 1985. *Deception Detection for News: Three Types of Fakes.* Ph.D. Dissertation. University of Western Ontario, London, Ontario, CANADA.