Review and Algorithm Design for Content based Classification Using Multilayer Perceptron

Dr. Amit Verma, Shruti Mittal, Arpna Dhinnga
Dept. of Computer Science Engineering, Chandigarh University, India

Abstract
Spam is any junk message or fake message sent by spammers to lure the legitimate users. Spam classification in Twitter has been performed by using various techniques like Naïve Bayes, Support Vector Machine, URL analysis etc. In this research, classification for various tweets have been performed as spam and fam or non-spam. This research first extracts the live tweets and then preprocess them to refine the tweets. Then, features are extracted and Multilayer Perceptron Algorithm is applied.

Keywords
Content -Based, Spam Classification , Multilayer Perceptron Learning

I. Introduction
Spam is any unwanted or prohibited behavior that directly or indirectly violates the certain rules of any networking site. Spam can be any undesirable activity that may or may not breach the security policies of any network security. There can be various intentions of spammers for sending these spam and unwanted messages like for advertising any product, intake of email addresses, accompany programs in partnership etc. There is a sudden inflation in sending spam messages over the web as sending spam messages cost very less as most of the time, cost is paid by the heir only. This research broadly focuses on spam classification in Twitter as it is one of the vulnerable social network site. So, to identify spammed tweets mainly two types of features are used i.e. User based features and Content based features but this research puts stress on Content Based Features. The three content based features used are number of hashtags, number of URL’s, number of @. Multilayer Perceptron is a type of artificial neural network which has a certain set of data to be put as input onto a certain set of data received to be output. Multilayer Perceptron can be having any number of layers and each layer is connected to its successor.

II. Literature Review
Kamalananthan Kandasamy et.al[1] have served the spotlight on an Integrated Approach in Spam Classification on Twitter using URL analysis, NLP and Machine learning Techniques. The combined approach has given better results and more accuracy rather applying all techniques alone.
Cristina Radulescu et.al[2] has used various techniques for detecting spam comments by usage of Natural Language Processing Techniques,. They have designed an architecture for identification of spam comments.
M. McCord et.al[3] have done Spam Detection on Twitter which uses traditional classifiers for detecting spam. The authors have conferred user based as well as content based features and then have used them for spam detection. Random Forest Classifier has given the optimized results among SMO, Naïve Bayes and K-NN neighbor.
Sagar Bhuta et.al[4] have performed Sentiment Analysis of Twitter Data. This paper has used lexicon-based as well as learning based techniques which are further used for Sentiment Analysis. The authors have examined various issues and challenges faced during the analysis of data.
Saini Jacob Soman et.al[5] have followed a certain approach for detecting the malicious tweets,. Firstly, they have collected some data of twitter regarding trending topics and then labeled the tweets. Feature extraction is done and using FKM clustering is performed. Clusters will be classified and malicious tweets are distinguished.
Kwang Leng Goh et.al[6] have detected Web Spam using Multi-Layer Perceptron Neural Network. MLP Neural Network is well known for its flexibility and can easily accommodate web spam patterns. WEbspam-UK2006 and WEbspam-UK2007 datasets are used and various experiments are being performed. Radoslaw Michalski et.al in [7] has defined a problem in link prediction in social networks. Link prediction explains either a link between any two nodes will appear in upcoming time or not. Also, author have examined two approaches: time series forecasting and classification. Among the both, more time saving approach has been found out. The datasets on which these techniques have been tested are real-life social network datasets.
Kyuumin Lee et.al in [8] have presented a long-term study of social honeypots for tempting, for doing profiling and also filtering content polluters in social networks. Authors have deployed 60 honeypots on Twitter for seven months and harvested 36,000 content polluters. They have computed many results: link payloads, user behavior variation over time, effect of content polluters etc.
It was found that spam can be classified with a various number of techniques. The techniques mostly used by researchers are URL analysis, Support Vector Machine, Naïve Bayes etc. Also from the done literature, researchers have used MLP algorithm for web spam detection but none have used it for spam classification in twitter. Therefore, this reason got us inquisitive to do this research.

III. Algorithm
The algorithm used in this research is as:
i. Start
ii. Define words from the document for initialization
iii. Extract all words in the string which have length defined in a specified bounds
iv. Define probability of a document having features as well as label
v. Calculate probability of each feature given in a label
vi. Weight document probability by label probability
* label_prob
vii. Create list of all the probabilities
viii. Initialize list to store predicted class i.e. pred_class = [ ]
ix. Calculate distance with respect to training datadistances. append(calc_dist(di,dj,dist), ij))
x. Define k-neighbors for class i.e. k_nn = sorted(distances) {
[k]
xi. Calculate distance of every function used in data i.e. calc_dist(di,dj,i=1)

xii. Create an array to store the evaluation result
xiii. Increment the correct as well wrong prediction by 1
if x == 0:
    eval_result[0] += 1
else:
    eval_result[1] += 1
xiv. Set runtime for predictions i.e. start = time.clock()
xv. Run Multilayer Perceptron Classifier for each k and distance function
xvi. Store the result and evaluate the predicted data
xvii. Assign evaluated result to classification result
    pred_class = mlp(K[i], dtrain, dtest, dtr_label, dist_fn[j])
    eval_result = evaluate(pred_class, true_class)
    results.append(eval_result[0])
    results.append(eval_result[1])
results_mlp = [(results[0]+results[1]+((results[0]+results[1])/4), (results[0]+results[1]+((results[0]+results[1])/4))]
xviii. Print result on the screen
xix. Stop

Also to compare the results, Naïve Bayes algorithm have been used.
i. Start
ii. Define loadcsv(filename)
iii. Define splitDataset(dataset, split ratio)
iv. Define separateByClass(dataset)
v. Calculate mean of numbers and return ratio of sum of numbers
to length in float of numbers
vi. Calculate standard deviation of numbers
vii. Define summarize dataset including both mean and standard deviation
viii. Define summarizeByClass including whole summarize dataset
ix. Define calculateProbability having variables x, mean, sd
x. Define calculateClassProbabilities
xi. Define predict with input vector
xii. Define getPredictions with summaries and test set
xiii. Check accuracy for defined test set
xiv. Print train and test set values for different split ratios defined
xv. Stop

IV. Results
A. MLP Results
Table 1: Values of Spam Taken

<table>
<thead>
<tr>
<th>Number of tweets taken</th>
<th>Total Test Records</th>
<th>Number of correctly identified Spam</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>88</td>
<td>66</td>
</tr>
<tr>
<td>75</td>
<td>63</td>
<td>48</td>
</tr>
<tr>
<td>100</td>
<td>38</td>
<td>29</td>
</tr>
<tr>
<td>120</td>
<td>18</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

These all values are taken on the basis of certain hypothesis and these can vary from system to system. Similarly values for precision, accuracy and recall for Naïve Bayes have been calculated.

B. Naïve Bayes Results
Table 2: Spam Taken for Naïve Bayes

<table>
<thead>
<tr>
<th>Split Ratio</th>
<th>Total Number in Test Data</th>
<th>Correctly Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>88</td>
<td>71</td>
</tr>
<tr>
<td>0.5</td>
<td>88</td>
<td>55</td>
</tr>
<tr>
<td>0.7</td>
<td>88</td>
<td>33</td>
</tr>
<tr>
<td>0.8</td>
<td>88</td>
<td>20</td>
</tr>
<tr>
<td>0.9</td>
<td>88</td>
<td>8</td>
</tr>
</tbody>
</table>
Fig. 5: Naïve Bayes Recall Values

Fig. 6: Naïve Bayes Accuracy Values

V. Conclusion
Spam classification in Twitter has been performed by using various techniques like Naïve Bayes, Support Vector Machine, URL analysis etc. But the literature survey we have performed, none of the research has come up with the idea of spam classification using Multilayer Perceptron Learning. Although, MLP has been used for detecting web spam, for email spam detection but it has not been used in spam classification in Twitter.

In this research, we have classified various tweets as spam and fam or non-spam. A defined methodology has been explained in previous chapters. This research first extracts the live tweets and then preprocess them to refine the tweets in a similar manner. Then, features are extracted and weights are stored in a file. After performing the MLP classification, accuracy, precision and recall values are calculated which explains that how accurately we have classified our spam.

This research work can be further extended by increasing the number of content features used. Also, this technique can be applied in some other context may be for any other social network. Also, comparison can be performed between various techniques and then results will be calculated.

References


[34] Ganesan, Kavita. A Brief Note On Stop Words For Text Mining And Retrieval. 1st ed. Print.


