Facebook Page Spam detection using Support Vector Machines based on n-gram model

Himani Chawla[1]

[1] **Student, College of Engineering and Technology, Department of Computer Science**
Bikaner, Rajasthan, India

Abstract

With social networks like Facebook, twitter reaching to the common masses, these have become the best target for spammers. The newest way to mislead and fraud viewers is Page Spam [1]. Viewers are deceived to click on links to spam their connections, redirect to a fraudulent business or spread wrong information about famous figures, organizations and causes.

This research aims to categorize such pages from authentic fan pages using support vector machines [2] and n gram models. Further an attempt has been made to improve our findings by some optimizations.

Keywords: *Spam Detection, SVM, n-grams, Natural Language Processing, Page Spam*

1. Introduction

As of July 2014, there are 1.32 billion active Facebook users [3]. That's a huge market for anyone seeking attention and to their comfort Facebook provides FanPage feature as one of core elements. Organizations, Celebrities, Politicians, etc. use this to reach out to this massive crowd. These fan pages help people stand together in times of distress and tragedies. But at the same time this opens the doors for spammers. A recent study by mashable [4] has found out that Facebook spam is a 200 billion US dollar business in itself. This revenue is generated from users' time, people getting fooled clicking on misdirected links, etc.

To address this concern, Facebook introduced concept of "verified" by placing a blue badge on verified pages [5]. But this process is manual and takes time.

This experiment deals with classification of pages which can be marked verified and which can't be using SVM as classifier based on n gram modeling.

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

2. Data Collection and Processing

Facebook's Graph API was used to mine thousands of fan pages in different categories, e.g. Hollywood celebrities, singers, politicians, global brands etc. 50% of these pages were verified while other half was handpicked to be spam page of each of the verified profile. For example, if there was a verified profile for brand "Coke", spam page for coke e.g., CokeClub was handpicked.

For every fan page, we analyzed their posts, page description, occurrences of hyperlinks in description and posts.

We chose to represent the updates and description in form of unigrams and bigrams. N-gram model[7] is a probabilistic model of text that uses some limited amount of history, or word dependencies, where n refers to the number of words that participate in the dependence relation.

n-grams are important to model some of the structural usage of natural language, i.e., the model uses word dependencies to assign a higher probability to ''how are you today'' than to ''are how today you,'' although both phrases contain the exact same words. If used in information retrieval, simple unigram language models (n-gram models with n ¼ 1), i.e., models that do not use term dependencies, result in good quality retrieval in many studies. The use of bigram models (n-gram models with n ¼ 2) would allow the system to model direct term dependencies, and treat the occurrence of ''New York'' differently from separate occurrences of ''New'' and ''York,'' possibly improving retrieval performance. The use of trigram models would allow the system to find direct occurrences of ''New York metro,'' etc.

Other data preprocessing steps:

• Removed the stop words from the posts and page description

• Lower case the posts and bio description.

• Then unigrams and bigrams are created from the posts and bio description.

• Average length of posts for verified and not verified profiles may be different. For this we consider the average length of posts as a feature.
• From these five features – unigrams, bigrams, number of public likes, Number of verified likes and average post length – we create twenty four different combinations of features. Refer to Table 1.
• Input feature vector is normalized to unit length.

3. Experiment

• We then trained the SVM using SVM Light [11], once for each feature combination and attempt to find the most appropriate feature combination for detecting non verified profiles. We trained the SVM using various combinations of features and use 10 fold cross validation[6] to find the feature set which gives maximum accuracy.

For this experiment, training involves the minimization of the error function [12]:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

Subject to the constraints:

$$y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \ldots, N$$

Where C is the capacity constant, w is the vector of coefficients, b is a constant, and ξ_i represents parameters for handling non separable data (inputs). The index i labels the N training cases. Note that $y \in \pm 1$ represents the class labels and xi represents the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

For each feature combination, we train the SVM to find the optimal C value using the technique of 10 fold cross validation [6].
In 10-fold cross validation, the original data is randomly partitioned into 10 equal subsamples. One of these 10 samples is retained as validation data for testing the model and the remaining 9 subsamples are used as training data. This process is then repeated 10 times (the folds) and each of the 10 subsamples acts as the validation data during one of the folds in experiment. All the results from 10 folds are averaged to provide a single estimate for the whole experiment iteration. The advantage of this method over repeated random sub sampling is that all observations are used for both training and validation and observation is used for training exactly once.

Different values of C chosen for 10- fold cross validation were 0.001, 0.01, 0.1, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024. We obtained an optimal C using the same experiment. We then create 10 random training and test set for each feature combination and find the accuracy for it using the optimal C. We average these 10 values to find the test accuracy for each feature combination. We find the test accuracy 10 times to smooth the result and avoid any peaks.

Table 1: Different feature combination

N	Unigrams	Bigrams	Likes	Verified likes	Post length
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	
3	✓	✓	✓		✓
4	✓	✓		✓	✓
5	✓	✓	✓		
6	✓	✓		✓	
7	✓	✓			✓
8	✓	✓			
9	✓		✓	✓	
10	✓		✓		✓
11	✓			✓	✓
12	✓		✓		
13	✓			✓	
14	✓				✓
15	✓				
16		✓	✓	✓	
17		✓	✓		✓
18		✓		✓	✓
19		✓	✓		
20		✓		✓	
21		✓			✓
22		✓			
23	✓		✓	✓	✓
24		✓	✓	✓	✓

4. Results

Feature combination number	Test accuracy
1	83.724
2	82.37
3	89.6
4	77.31
5	88.9
6	72.946

Feature combination number	Test accuracy
7	77.37
8	76.97
9	84.129
10	88.946
11	75.4
12	88.467
13	72.962
14	73.93
15	75.25
16	80.87
17	85.112
18	68.399
19	81.60
20	62.34
21	59.92
22	49.28
23	84.5
24	78.4

5. Conclusion

Our analysis shows that SVMs prove to be good classifiers for filtering out page spam because they use overfitting protection in a very high dimensional space. The ability to represent pages in a high dimensional space helps in this regard.

Further, we analyzed different combinations of feature set. Of different combinations, unigrams give the best accuracy in classification of pages. Combination of unigrams and bigrams has slightly lower accuracy while bigrams alone have a drastically lower accuracy. This is supported by the fact that while creating bigrams, we create a lot of random bigrams and a very handful of meaningful ones which help in classification. The difference in number is so large that effects of discriminating bigrams is eclipsed by random bigrams.

References

[1] Page Spam - https://www.facebook.com/help/116053525145846/
[2] SVM: http://link.springer.com/chapter/10.1007%2FBFb0026683?LI=true#
[3] Facebook 2014 Q2 report - http://files.shareholder.com/downloads/AMDA-NJ5DZ/3349478089x0x770377/abc6b6d4-df03-44e1-bb4d-7877f01c41e0/FB%20Q2
[4] http://mashable.com/2013/08/29/facebook-fan-pages-spam-200-million-business/
[5] What's a verified profile or Page? https://www.facebook.com/help/196050490547892
[6] Sharma Sumant, Arora Amit "Adaptive Approach for Spam Detection". IJCSI, Vol. 10, Issue 4, No 1, July 2013
[7] Yew Choong Chew,, Yoshiki Mikami, Robin Lee Nagano "Language Identification of Web Pages Based on Improved N-gram Algorithm" IJCSI, Vol. 8, Issue 3, No 1, May 2011
[8]Stemming: http://www.infoautoclassification.org/public/articles/Ikonomakis-et.-al._Text-Classification-Using-Machine-Learning-Techniques.pdf
[9] Stemming: http://nlp.stanford.edu/IR-book/html/htmledition/features-for-text-1.html
[10] Bigrams: http://people.cs.umass.edu/~ronb/papers/bigrams.pdf
[11] SVM Light - http://svmlight.joachims.org
[12] SVM Classfication - http://www.statsoft.com/Textbook/Support-Vector-Machines