Spam Detection for YouTube using Bayesian Method

Shreeya Jaiswal[1] Shishir S. Kurhade[2] Murtaza S. Khambaty[3] Susmit Gaikwad[4]
[1,2,3,4]Department of Computer Engineering
[1,2,3,4]SKNCOE, Vadgaon(BK), Pune 411041 - India

Abstract— In the last few years, social media websites have seen a dramatic growth in the number of users. More and more people are now communicating over the internet using social media websites, thus, these websites are very popular nowadays. Just like any other, YouTube is one popular video streaming social media website. A whooping 1 billion users visit the website every month. But it has become susceptible to different types of unwanted malicious spamming. Currently, YouTube relies on the community to flag videos which they find inappropriate, and it considers mass comments and messages as a part of spamming. The need to identify spam is more elaborate and so it should be done not only using flags and text mining of comments but also considering other attributes related to a video. Here, we have proposed a system capable of identifying spam based on Naïve Bayes Algorithm. The results of Naïve Bayes Algorithm are compared with other data mining algorithms and techniques. Also, ways in which the results generated by the system can be improved are stated.

Key words: Data Mining, Web Crawling, Spam Detection, YouTube, Bayesian Analysis

I. Introduction

YouTube is a video sharing website which is used to upload and watch videos by all its users. A large amount of videos are uploaded daily on YouTube and this has led to immense popularity. However, the spam content on this website is also increasing with increasing popularity. A spam can be referred to as irrelevant or unsolicited messages sent over the Internet, typically to large numbers of users, for the purposes of advertising, phishing, spreading malware, etc. In case of YouTube spam videos are used for this purpose. YouTube currently uses the flagging system to identify spam videos, wherein a user flags a malicious video and YouTube reviews it and takes necessary actions. However the shortcomings of such an approach are that the nature of a video cannot be decided merely by the flagging of users. Also reviewing each video to decide its nature is time consuming and has an additional overhead. Every video has certain attributes related to it, such as username, number of likes, comments, the number of views etc. Such attributes can be fetched using crawlers like Tubekit, available on the internet. By using such crawlers, a relational database can be formed which may further be used for applying Naïve Bayes Algorithm and generating a pattern. This pattern can be used to identify spam videos.

II. Related work

YouTube uses the flagging of videos to identify malicious content and deciding the course of action. However, to improve this technique, various data mining algorithms have been applied to data collected through crawlers. Different data mining algorithms which can be used for the YouTube based spam detection are: Decision Tree, K means Clustering and Naïve Bayes.

A. Decision Tree:

It is a flow-chart-like tree structure. Here each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent class distribution. The advantage of using this for classifying the data is that they are simple to understand and interpret.

However, they have disadvantages such as:

- Most of the decision tree algorithms require that the target attribute will have only discrete values.
- As decision trees use the “divide and conquer” method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present.

B. K-Means Clustering:

Let D is a data set of n objects and k is number of clusters. Partitioning algorithm distributes the objects into k clusters such that objects within the cluster are similar and object with other cluster are dissimilar. First, it arbitrarily selects k of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster.

The advantages of this approach are:

- K-Means may be computationally faster than hierarchical clustering, if K is small.
- K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

However it has some disadvantages:

- Difficulty in comparing quality of the clusters produced.
- Fixed number of clusters can make it difficult to predict what K should be and it doesn’t work well with non-globular clusters.

C. Naive Bayes:

This algorithm assumes that there are no dependencies amongst attributes. It is made to simplify the computations involved and, hence is called "naive".

The advantages of Naive Bayes are:

It uses a very intuitive technique. Bayes classifiers, unlike neural networks, do not have several free parameters that must be set. This greatly simplifies the design process.

Since the classifier returns probabilities, it is simpler to apply these results to a wide variety of tasks than if an arbitrary scale was used. It does not require large amounts of data before learning can begin.

Naive Bayes classifiers are computationally fast when making decisions.

The Naive Bayes algorithm uses a probabilistic classifier that represents a supervised learning and statistical method for classification. It allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Naive Bayes classifiers can be trained efficiently using supervised learning, depending on the precise nature of the probability model. It is mostly used to solve diagnostic and predictive problems.

Let us consider the mathematical modeling of Naïve Bayes:

Let S be a system such that,
S = {I, O, Fail, Success}
Where,
Inputs: *I*= Data set
Output: *O*= Patterns used for further predictions
Fail: a) If ambiguity exist, and system is unable to predict outcome
b) If system gives wrong prediction
Success: Expected result is obtained

To find if a tuple belongs to a particular class, the Naive Bayes Algorithm is given by the formula,

Where, X is the data sample or tuple and it is represented by *X= (x1, x2, x3, . . . , xn)*
m= Number of Classes from *C1, C2 , .. Cm.*
P(X—Ci) = P(Ci) 1
If predictable class is true, then,
P(Ci) = (No. of ways the predictable class is true) /Total no. of outcomes. 2
If predictable class is false, then,
P(Ci) = (No. of ways the predictable class is False) /Total no. of outcomes. 3
Where,
P(x1—Ci) = (No. of ways x1 happens)/Total outcome for the predictable class as true.
P(x1—Ci) = (No. of ways x1 happens)/Total outcome for the predictable class as false.

Thus considering each algorithm for our application we find out the performance parameters as follows:

– Naïve Bayes outperforms Decision Tree and k-means. It is the best in all performance parameters but precision, they are: recall, F-measure, accuracy, and AUC (Area under curve).
– Although it is simple, Naive Bayes can outperform more sophisticated classification methods. It displays high accuracy and speed when applied to large database. Moreover, it is very fast for both learning and predicting. Naïve Bayes classifier is also fast, consistent, easy to maintain and accurate in the classification of attribute data. And from computation point of view, Naïve Bayes is more efficient both in the learning and in the classification task than Decision Tree.
– For less number of test cases, k-means gives better result. But if we increase the test size, Decision tree and Naïve Bayes outperforms K-means.

For example at 40% of total population Naïve Bayes has a predict probability of 99.75%,decision tree 98.98%.But, at 85% of total population Naïve Bayes has a predict probability of 80.20%, decision tree has 82.11% and clustering has 65.79%. Therefore, we could conclude that, for higher number of test cases Naïve Bayes and Decision tree are more accurate for spam prediction. Here is a comparison of different algorithms:

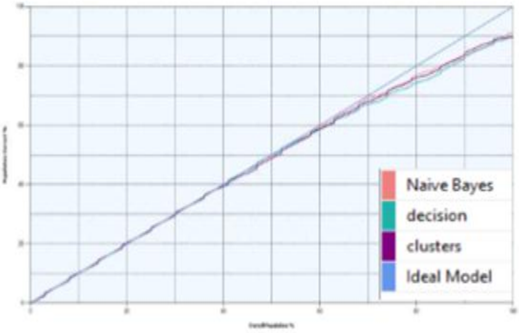

Fig. 1: Comparison Chart

III. PROPOSED WORK

The proposed system uses the Naive Bayes algorithm to generate a spam detection pattern. This pattern is used to build upon a spam detection system that is capable of identifying spam and legitimate videos. The pattern developed is dependent on the attributes of videos. We first create a dataset of videos and their corresponding attributes by using a crawler and giving seed queries to the crawler. The crawler collects all of the data related to user as well as video. The crawled data is then imported into our SQL database. Manual classification of videos as spam or not spam is done after downloading each of them. The results of the manual classification are added in a column manually in the tables of metadata i.e. user and Video data. Now assigning this manually classified column as dependent column and user name as the primary column, we run the Naive Bayes Algorithm on the database. This would generate a profile view which gives the range of probabilities of the attributes. This pattern is used to build spam detection models.

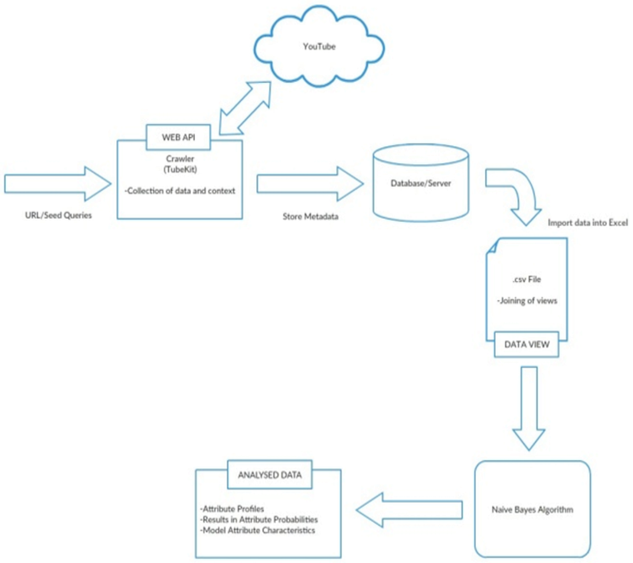

Fig. 2: System Architecture

IV. IMPROVEMENTS

The existing algorithm can improved by following methods:

– Pre-processing input data: Transforming raw data into an understandable form. Real-world data is often incomplete, inconsistent, or lacking in certain aspects, and is likely to contain many errors.

- Correct feature selection: Selecting a subset of relevant features to enable simplification of models making them easier to interpret, shorter training times and enhanced generalization.
- Adjusting classifier's tunable parameters: Parameters of model are adjusted upwards or downwards to achieve an improved or specified result.
- Refine data fed into algorithm: The input data is filtered to remove irrelevant and useless data, thus reducing time complexity.
- Normalizing input probabilities: Transforming all variables in the data to a specific range.

The most effective way of improving the accuracy of Naive Bayes is to train it using multiple data sets. Also the accuracy of this algorithm is noticeably better on large data sets.

REFERENCES

[1] Rashid Chowdury,Md. Nuruddin Monsur Adnan, G.A.N. Mahmud,Rashedur M Rahman "Data Mining Based Spam Detection System For YouTube". IEEE - 2013.

[2] Ritesh Kumar, Shital Ghadge,G.S. Navale "Spam Detection Using Approach of Data Mining for Social Networking Sites" International Journal of Computer Applications, Dec-2014.

[3] Nikita Spirin, Jiawei Han "Survey on Web Spam Detection: Principles and Algorithms", Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801, USA.

[4] Suman Arora and Vipin Arora "A Study of Video Response Spam Detection on YouTube" International Journal of Engineering, Applied and Management Sciences Paradigms, Vol. 18, Issue 01, August 2014.

[5] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, Dawn Song, "Design and Evaluation of a Real Time URL Spam Filtering Service" University of California, Berkeley, International Computer Science Institute.

[6] Sahil Puri, Dishant Gosain, Mehak Ahuja "Comparison and Analysis of Spam Detection Algorithm ", International Journal of Application or Innovation in Engineering and Management (IJAIEM), Volume 2, Issue 4, April 2013.

[7] Ahmad Ashari, Iman Paryudi and A Min Tjoa, "Performance Comparison between Nave Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool" International Journal of Advanced Computer Science and Applications (IJACSA), Vol 4, No. 11, 2013.

[8] Nadir Omer Fadl Elssied and Othman Ibrahim, "K-Means Clustering Scheme for Enhanced Spam Detection "Research Journal of Applied Sciences, Engineering and Technology, 15 march 2014.

[9] Chirag Shah "Supporting Research Data Collection from YouTube with TubeKit" School of Information and Library Science University of North Carolina.

[10] Ashish Sureka "Contextual Feature Based One-Class Classifier Approach for Detecting Video Response Spam on YouTube" Eleventh Annual Conference on Privacy, Security and Trust (PST). 2013