ISSN:0975-9646

A Framework for Detection of Video Spam on YouTube

Niyanta Ashar, Hitarthi Bhatt, Shraddha Mehta, Prof. (Mrs.) Chetashri Bhadane

Department of Computer Engineering, D.J. Sanghvi College of Engineering
University of Mumbai
Mumbai, India

***Abstract*—YouTube is one of the largest video sharing websites (with social networking features) on the Internet. The immense popularity of YouTube, anonymity and low publication barrier has resulted in several forms of misuse and video pollution such as uploading of malicious, copyright violated and spam video or content. It has been observed that the presence of opportunistic users post unrelated, promotional, pornographic videos (spam videos posted manually or using automated scripts). A method of mining YouTube to classify a video as spam or legitimate based on video attributes has been presented. The empirical analysis reveals that certain linguistic features (presence of certain terms in the title or description of the YouTube video), temporal features, popularity based features, time based features can be used to predict the video type. We identify features with discriminatory powers and use it to recognize video response spam.**

***Keywords*—** video spam; spam detection; YouTube; TubeKit

I. INTRODUCTION

The popularity of social networking sites such as Facebook, Twitter, YouTube, Flickr has increased a lot since the last decade which specializes in micro- blogging, video sharing, photo sharing and discussion forums. In particular, video is becoming a most important part of user's daily life, the reason being video is the most usable medium to share views with others and is a medium of many type of interactions among users such as business discussion, political debates, educational tips etc.[20] YouTube is one of the most popular and widely used video sharing website (with social networking features) on the Internet.

A whooping 1 billion unique users visit YouTube every month and they watch almost 4 billion hours of video content [14]. Web is being transformed into a major channel for the delivery of multimedia. As a consequence, various services on the Web are offering video-based functions as alternative to text-based ones, such as video reviews for products, video ads and video responses. By allowing users to publicize and share videos, video social networks become susceptible to different types of opportunistic user actions. As an example, YouTube provides features that allow users to post a video in a particular category/genre/topic with any title. Although appealing as a mechanism to enrich the online interaction, these features open opportunities for users to introduce polluted content, into the system. For example, users, which we call spammers, may post an unrelated video aiming at increasing the likelihood of it being viewed by a larger number of users. Moreover, opportunistic users, namely promoters may try to gain visibility to a specific video by posting a large number of (potentially unrelated) responses to boost the rank of the video, making it appear in the top lists maintained by YouTube[21]. Promoters and spammers are motivated to pollute for several reasons, such as to spread advertises, disseminate pornography (often as an advertisement to a Web site), or just to compromise system reputation.

In summary, the main contributions of this research are:

- Find out quantitative evidence of video spamming activity (as defined above) in YouTube.
- A test collection of videos from YouTube, classified as spam or legitimate videos.
- A video spam detection mechanism based on a set of classification algorithms, for example, ID3 algorithm, Naïve Bayes, and K-nearest neighbor algorithm.
- Predict the spam from the data mining model

II. RELATED WORK

Mechanisms to detect and identify spam and spammers have been largely studied in the context of Web [12, 16] and email spamming [15]. In particular, Castillo et al. [12] proposed a framework to detect Web spamming which uses social network metrics. A framework to detect spamming in tagging systems, which is a type of attack that aims at raising the visibility of specific objects, was proposed in [18]. Although applicable to social media sharing systems that allow object tagging by users, such as YouTube, the proposed technique exploits a specific object attribute, i.e., its tags. Our approach is complementary to these efforts as it aims at detecting spam videos, using a combination of different categories of attributes of both objects and users.

The methodology used in [2] tries to identify non-cooperative users and addresses both spamming and ballot stuffing (which they call "promoting"), by analyzing parameters like tags, user profile, the user posting behavior and the user social relations. [3] Classifies videos, accordingly to their pattern of access by users, into three categories: quality (the normal ones), viral (videos which experience a sudden surge in popularity) and junk (spurious videos, like spam). Neither work use the video content itself for classification, instead, they rely on meta-data and access logs.

A survey of approaches to combat spamming in Social Web sites is presented in [17]. Many existing approaches are based on extracting evidence from the content of a text, treating the text corpus as a set of objects with associated attributes and using these attributes to detect spam. These techniques, based on content classification, can be directly applied to textual information, and thus can be used to detect spam in email, text commentaries in blogs, forums, and online social networking sites. Additionally, detection of email spam based on image content was also studied previously [11, 19].

Lee et. al propose a multi-level hierarchical system to identify offensive videos. They use multimedia content (frame, color, images), contextual metadata, hash signature (encrypted video header) as discriminatory features [5]. However, content classification is much harder to do for video objects.

Our approach to detect video spam consists on classifying users, their videos, and relies on a set of attributes associated with the user actions and social behavior in the system as well as attributes of their videos. Towards the end, this paper presents a characterization of user, social and video attributes that can be used to distinguish spam from legitimate videos on YouTube.

III. LITERATURE SURVEY

A. *Tubekit*

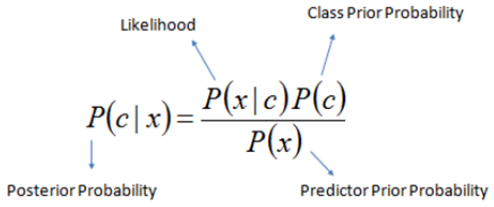
TubeKit is a PHP based program runs from a web server. It is a toolkit for creating customized YouTube crawlers. It allows one to build one's own crawler that can crawl through YouTube based on a set of seed queries and collect up to 16 different attributes. In addition to creating crawlers, TubeKit also provides several tools to collect a variety of data from YouTube, including video details and user profiles.

B. *Naïve Bayes Classification Algorithm*

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes (1702-1761), who proposed the Bayes Theorem. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Naïve Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all Naïve Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A Naïve Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features. For some types of probability models, Naïve Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naïve Bayes models uses the method of maximum likelihood; in other words, one can work with the Naïve Bayes model without accepting Bayesian probability or using any Bayesian methods.

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naïve Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig. 1 Computation for Naïve Bayes Classifier

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Advantages of Naïve Bayes Classifier

- A Naïve Bayes classifier is mathematically easy to follow.
- It is efficient in terms of time needed to train, and the time needed to classify an unknown video
- The classifier is easy to update as more training data is accumulated.

C. *K-Nearest Neighbor Algorithm*

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

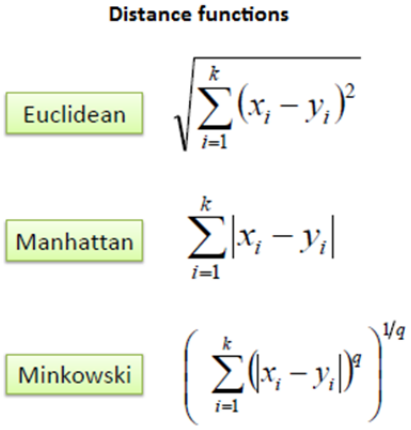


Fig. 2 Distance Functions for K-Nearest Neighbor Classification

It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the data set.

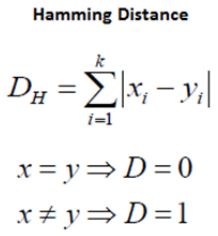


Fig. 3 Hamming Distance for K-Nearest Neighbor Classification

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent data set to validate the K value. Historically, the optimal K for most data sets has been between 3-10. Basically what we do is that we try to find the k nearest neighbor and do a majority voting. Typically k is odd when the number of classes is 2. Let us say k = 5 and there are 3 instances of video being spam and 2 instances of video not being spam. In this case, KNN says that new point has to labeled as spam as it forms the majority. We follow a similar argument when there are multiple classes.

One of the straight forward extensions is not to give 1 vote to all the neighbors. A very common thing to do is weighted kNN where each point has a weight which is typically calculated using its distance. For e.g. under inverse distance weighting, each point has a weight equal to the inverse of its distance to the point to be classified. This means that neighboring points have a higher vote than the farther points.

It is quite obvious that the accuracy might increase when you increase k but the computation cost also increases.

The algorithm can be summarized as:

1. A positive integer k is specified, along with a new sample

2. We select the k entries in our database which are closest to the new sample

3. We find the most common classification of these entries

4. This is the classification we give to the new sample

Advantages of K-Nearest Neighbor Algorithm

- The cost of the learning process is zero
- No assumptions about the characteristics of the concepts to learn have to be done
- Complex concepts can be learned by local approximation using simple procedures

D. ID3 Decision Tree Algorithm

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from the dataset.[22] To model the classification process, a tree is constructed using the decision tree technique. Once a tree is built, it is applied to each tuple in the database and this results in classification for that tuple.

The following issues are faced by most decision tree algorithms [23]:

- To choose splitting attributes
- Order of splitting attributes
- Number of splits to be taken
- Balance of tree structure and pruning
- The stopping criteria

The decision tree algorithm is based on Entropy, its main idea is to map all examples to different categories based upon different values of the condition attribute set; its core is to determine the best classification attribute from condition attribute sets. The algorithm chooses information gain as attribute selection criteria; usually the attribute that has the highest information gain is selected as the splitting attribute of the current node. Branches can be established based on different values of the attributes and the process above is recursively called on each branch to create other nodes and branches until all the samples in a branch belong to the same category. To select the splitting attributes, the concepts of Entropy and Information Gain and Gain Ratio are used.

Entropy

It is a measure in the information theory, which characterizes the impurity of an arbitrary collection of

examples. If the target attribute takes on 'c' different values, then the entropy S relative to this c-wise classification is defined as

Entropy(s) =∑- Pi log2Pi

Where Pi is the probability of S belonging to class i. Logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits. For e.g. if training data has 14 instances with 5 positive (spam) and 9 negative (not spam) instances of videos, the entropy is calculated as

Entropy ([5+,9-])=-(5/14)log(5/14)-(9/14)log(9/14)=0.9402

A key point to note here is that the more uniform the probability distribution, the greater is its entropy. If the entropy of the training set is close to one, it has more distributed data and hence, considered as a good training set.

Information Gain

The decision tree is built in a top-down fashion. ID3 chooses the splitting attribute with the highest gain in information, where gain is defined as difference between how much information is needed after the split. This is calculated by determining the differences between the entropies of the original data set and the weighted sum of the entropies from each of the subdivided data sets. The motive is to find the feature that best splits the target class into the purest possible children nodes - pure nodes with only one class. This measure of purity is called information. It represents the expected amount of information that would be needed to specify how a new instance of an attribute should be classified. The formula used for this purpose is:

G(D, S) = H(D) - ∑P(Di)H(Di)

The attribute with highest value of information gain is used as the splitting node thereby constructing the tree in top down fashion.

IV. PROPOSED SYSTEM

Methodology

The goal is to design a mechanism to classify the video on social video sharing systems into legitimate and spam, using a set of video attributes like YouTube ID, Username, upload time, duration, category, video url, video count, view count, rating average, rating count, comment count, spam. A test collection, including a sample of the crawled data, can be built and used to evaluate the effectiveness of our classification approach. Next section describes our crawling strategy followed by presenting the criteria used to select users for the test collection.

TubeKit [10] can be used in all the phases of this process starting database creation to finally giving access to the collected data with browsing and searching interfaces. The working design of the crawler is shown in Fig. 5 and was first presented in [10].

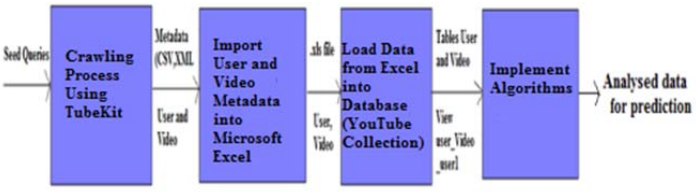


Fig. 4 Working Flow of Proposed System

Crawling Process

I. A set of seed queries will be provided to the monitor.

II. The system uses these queries to go out and search on YouTube.

III. A set of meta-data is extracted from a subset of the results returned from YouTube. Meta-data is defined to be the information about the given video which are provided by the author of that video, and are usually static in nature, for instance, the genre of the video.

IV. The video downloader component checks the meta-data table to see which videos have not been previously downloaded and collects those videos in ash format from YouTube.

V. The video converter component checks which videos are downloaded and not converted, and converts them into mpeg format.

VI. The context capturing component goes out to YouTube and captures various contextual information about the video items for which the meta-data is already collected. Each time such social context is captured, a time-stamp is recorded. We define social context as the data contributed by the visitors to a video page. This would include fields such as ratings and comments. Note that other types of social context in blogs and other sources could also be harvested with different components. The context capturing component runs periodically and updates time-sensitive data such as new comments or video postings, thus capturing temporal context.

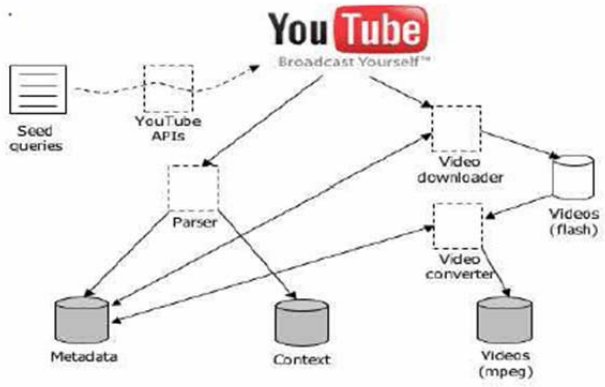


Fig. 5 Scheme for query based YouTube crawling

The excel file shall then be connected with Microsoft SQL server 2012. Data mining operations can be run with the help of SQL Server Tools and excel data mining add-in for SQL server 2012. For deploying the data mining techniques a new Analysis Services database shall be

created. We will then add a data source and data source view, and prepare the new database to be used with data mining. After creating the data source from the extracted excel file, we shall create a data source viewer to see the tables and views of the database.

V. CONCLUSION

We came to a conclusion that we will make use of Naïve Bayes, K-nearest neighbor and ID3 algorithm for the purpose of spam detection in YouTube videos. Using these algorithms we hope to overcome the disadvantages faced by the existing systems and achieve results that are more efficient and accurate in the classification of a given video as a spam.

ACKNOWLEDGMENTS

We express our sincere gratitude towards our guide Prof. Chetashri Bhadane who assisted us throughout our work. We thank her for directing us to the right tract and for the motivation and prompt guidance she has provided whenever we needed it.

REFERENCES

[1] Antonio da Luz1,2, Eduardo Valle3, Arnaldo Araujo1, "CONTENT-BASED SPAM FILTERING ON VIDEO SHARING SOCIAL NETWORKS", Cornell University Online Library

[2] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida and M. Gonçalves, "Detecting Spammers and Content Promoters in Online Video Social Networks", In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 620627, 2009

[3] R. Crane and D. Sornette, "Robust Dynamic Classes Revealed by Measuring the Response Function of a Social System", In: National Academy of Sciences, 105(41):15649-15653, 2008.

[4] Antonio DA luz, Arnaldo Ariyo," Content based spam filtering on video sharing systems",

[5] Lee, S., Shim, W., & Kim, S. (2009). "Hierarchical system for objectionable video detection." Consumer Electronics, IEEE Transactions on, 55 (2), 677-684.

[6] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints", In: International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[7] I. Laptev. "On Space-Time Interest Points", In: International Journal of Computer Vision, vol 64, number 2/3, p.107-123, 2005.

[8] Rashid Chowdury, Md. Nuruddin Monsur Adnan, G.A.N. Mahmud, Rashedur M Rahman, 'A Data Mining based Spam Detection System for YouTube', IEEE 2013

[9] Y.C. Shah, "Supporting Research Data Collection from YouTube with TubeKit". Proceedings of YouTube and 2008 Election Cycle in the United States, Amherst, MA: April 16-17, 2009.

[10] TubeKit, http://www.tubekit.org/

[11] H. Aradhye, G. Myers, and J. Herson. "Image analysis for efficient categorization of image-based spam e-mail." In Proc. of the Int'l Conf. on Document Analysis and Recognition (ICDAR), volume 2, pp.914918, 2005.

[12] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your neighbors: Web spam detection using the web topology", In Int'l ACM SIGIR, pp. 423–430, 2007.

[13] S. Avila, A. Lopes, A. da Luz Jr., and A. de A. Araujo. "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method". Pattern Recogn. Lett. 32, 1, 56-68, 2011.

[14]] A. Thomason, "Blog spam: A review", In Proc. Of Conf. on Email and Anti-Spam (CEAS)", paper no.85, 2007.

[15] L. Gomes, F. Castro, V. Almeida, J. Almeida, R. Almeida, and L. Bettencourt, "Improving spam detection based on structural similarity", In USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), pp.85-91, 2005.

[16] Z. Gy¨ongyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank", In International Conference on Very Large Data Bases (VLDB), pp. 576–587, 2004.

[17] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges", IEEE Internet Computing, 11(6):36–45, 2007.

[18] G. Koutrika, F. Effendi, Z. Gy¨ongyi, P. Heymann, and H. GarciaMolina, "Combating spam in tagging systems", In Proceedings of the 3rd international workshop on Adversarial information retrieval on the web(AIRWeb), pp. 57-64, 2007.

[19] C. Wu, K. Cheng, Q. Zhu, and Y. Wu, "Using visual features for antispam filtering", In Proc. of 4th IEEE Int'l Conf. on Image Processing (ICIP), 2005.

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

[21] Fabrico Benevenuto, Tiago Rodrigues, Jussara Almeida, Marcos Gonc¸alves and Virgılio Almeida, "Detecting Spammers and Content Promoters in Online Video Social Networks",INFOCOM Workshops 2009,IEEE

[22] Kalpesh Adhatrao, Aditya Gaykar, Amiraj Dhawan, Rohit Jha and Vipul Honrao, " Predicting Students' Performance using ID3 and C4.5 classification algorithm", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.5, September 2013

[23] Dunham, M.H., (2003) Data Mining: Introductory and Advanced Topics, Pearson Education Inc [Online] http://www.saedsayad.com/naive_bayesian.htm