Content-Based Spam Filtering and Detection Algorithms- An Efficient Analysis & Comparison

[1]R.Malarvizhi, [2]K.Saraswathi
[1]*Research scholar, PG & Research Department of Computer Science, Government Arts College (Autonomous), Coimbatore-641018, TamilNadu, India.*
[2]*Assistant Professor, PG & Research Department of Computer Science, Government Arts College (Autonomous), Coimbatore-641018, TamilNadu, India.*

Abstract:

Spam is one of the major problems faced by the internet community. There are many approaches developed to overcome spam and filtering is one of the important one. The Content-based filtering is also known as cognitive filtering that recommends items based on a comparison between the content of the items and a user profile items. The content of each item is represented as a set of descriptors or terms. The terms are typically, the words that occur in a document. User profiles are represented with the same terms and built up by analyzing the content of items seen by the user. In this paper, an overview of the state of the art for spam filtering is studied and the ways of evaluation and comparison of different filtering methods. This research paper mainly contributes to the comprehensive study of spam detection algorithms under the category of content based filtering. Then, the implemented results have been benchmarked to examine how accurately they have been classified into their original categories of spam.

Key words: Spam, AdaBoost, KNN, Chi-Square, Black list, White list, Bayesian filters, Cache Architecture.

I. INTRODUCTION

In this digital age, is the time of computers, one of the well-organized and easier modes of communication is the email. Reading an email is becoming a regular habit of many people. This is an efficient, fast and cheaper means of communication. Email formulates it desired both in professional and personal associations. The difficulty of undesired electronic messages is nowadays a serious issue, as spam constitutes up to 75-80% of total amount of emails [1]. The spam causes several problems may result in direct financial losses. Also causes misuse of traffic, storage space and also computational power. Spam makes the user to sort out additional email, as it wasting their time. This causes loss of work productivity, often irritate users by violating the privacy rights [2]. The Ferris Research Analyzer Information service estimates that over $50 billion financial loss has been caused by the spam worldwide. Undesired, unsolicited email is a nuisance for its recipients, it also often presents a security threat. It may contain a link to a fake website intending to capture the users login credentials (identity theft, phishing), or a link to a website that installs malicious software (malware) on the user's computer. The Installed malware can be used to capture user information, to send spam, host malware, host phish, or conduct denial of service attacks. While prevention of spam transmission would be ideal, detection allows users & email providers to address the problem today [1].

Spam is defined as the unsolicited (unwanted, junk) email for a recipient or any email that the user do not want to have in his inbox.

Daily Spam emails sent	12.4billion
Daily Spam received per person	6
Annual Spam received per person	2,200
Spam cost to all non-corporate Internet users	$255 million
Spam cost to all U.S Corporation in 2002	$8.9 billion
Email address changes due to spam	16%
Annual Spam in 1,000 employee company	2.1 million
Users who reply to Spam email	28%

Fig 1: Statistics of spam mails [15]

The spam is also defined as "The Internet Spam is one or more unsolicited messages, sent as a part of larger collection of messages, having substantially identical content" [14]. Several problems have been encountered from the spam mails such as wastage of network resources (bandwidth), wastage of time,

damage to the PC's & laptops due to viruses & the ethical issues such as the spam emails advertising pornographic sites which are harmful to the young generations [5]. The basic concepts of spam filter can be illustrated in the following diagram;

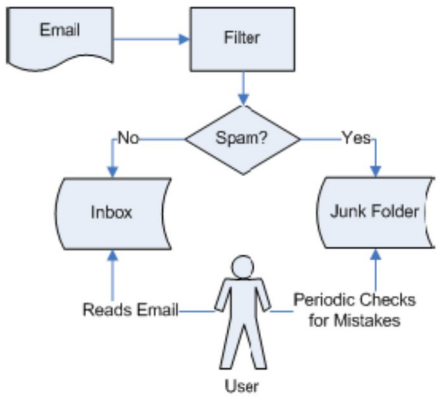

Fig 2: Spam Filter

The basic format of Electronic-mail generally consists of the following sections:

- Header section includes the sender email address, the receiver email address, the Subject of the email and
- The Content of the email includes the main body consisting of text, images and other multimedia data [3].

In content based spam filtering, the main focus is on classifying the email as spam or as ham, based on the data that is present in the body or the content of the mail. However, the header section is ignored in the case of content based spam filtering. There are number of techniques such as Bayesian Filtering, AdaBoost classifier, Gary Robinson technique, KNN classifier. Combining function based on Fisher-Robinson Inverse Chi-Square Function are available which can be used for content based filtering.

This research work comprises of the analytical study of various spam detection algorithms based on content filtering such as Fisher-Robinson Inverse Chi Square function, Bayesian classifiers, AdaBoost algorithm and KNN algorithms. The algorithms have been implemented; the results were studied to draw a relative comparison on the effectiveness of a technique to identify the most accurate one. Each technique is demonstrated in the following sections with their implemented result. The paper is concluded with the benchmarking of the techniques.

II.FISHER-ROBINSON INVERSE CHI-SQUARE FUNCTION

The Chi-Square method is content based filtering technique and Robinson has proposed this technique[5]. The system uses the probability function which is also named as "Robinson's Degree of Belief". The function takes the parameters as the following: s as a tunable constant, p(w) is the Robinson's total probability function, then x is an assumed probability given to words never seen before (hapaxes), and n is the number of messages containing the token. The initial values were recommended as 1 and 0.5 for s and x, respectively.

The development of two combination functions is credited to Gary Robinson [4]. The functions have been utilized with great success in many spam filters. Robinson's geometric mean function is shown in Figure 3.

$$P = 1 - \sqrt[n]{((1-p1)*(1-p2)*\ldots*(1-pn))}$$

$$Q = 1 - \sqrt[n]{(p1*p2*\ldots.*pn)}$$

$$S = \frac{1+\frac{(P-Q)}{(P+Q)}}{2}$$

Fig 3: Robinson's Geometric Mean Function

This function is quite similar to Burton's combination function in Spam survey. Both use the nth root of products and return values between 0.0 to 1.0. He has also developed an altered token probability function [5]. He has named this function f(w), in Figure 4, a degree of belief.

$$f(w) = \frac{(s*x) + (x*p(w))}{s+n}$$

Fig 4: Robinson's Degree of Belief Function

In this function, in Graham's essay p (w) can be calculated, s is a tunable constant, x is an assumed probability given to words never seen before (hapaxes), and n is the number of messages containing this token. The initial values of 1 and 0.5 for s and x, respectively, are recommended. Hence, Robinson suggests using this function in situations where the token has been seen just a few times. There may be some case is where a token has never been seen before. For that, the value of x will be returned and the number of occurrences increases, so does the degree of belief. In Robinson's degree of belief function, the value of p (w) can be calculated as Graham's process and it includes certain modifications [5]. Fig 4 shows how instead of using the total number of occurrences of a token in a ham or spam corpus, the number of messages containing that token has been used.

$$g(w) = \frac{numHamWithToken}{numHam}$$

$$b(w) = \frac{numSpamWithToken}{numSpam}$$

$$p(w) = \frac{b(w)}{b(w)+ g(w}$$

Fig 5: Robinson's Token Probability Function

Robinson considers Graham's method performs better than Graham's counting method does not ignore any of the token occurrences data. The second combining function Robinson has proposed is based on the work of Sir Ronald Fisher. This method has been named the Fisher-Robinson Inverse Chi-Square Function [14].

$$H = C^{-1}(-2\text{In} \prod_w f(w), 2n)$$
$$S = C^{-1}(-2\text{In} \prod_w 1 - f(w), 2n)$$
$$I = \frac{H}{H+S}$$

Fig 6: Fisher-Robinson's Inverse Chi-Square Function

There are three parts to this equation, as shown in Fig 5. In this, H is the combined probability sensitive to hammy values, S is used to calculate the probability sensitive to spammy values, I is used to produce the final probability in the usual 0 to 1 range, C−1 is the inverse chi-square function, and n is the number of tokens used in the decision matrix. Jonathan Zdziarski [6] gives the C code for C−1 in Figure 2.12. Zdziarski notes the high level of uncertainty provided by this function.

SpamBayes is a free and open-source spam filter that uses the Fisher-Robinson Inverse Chi-Square Function [7]. This uncertainty allows SpamBayes to return an unsure result instead of just Ham or Spam.

```
double chi2Q(double x, int v)
{
        int i;
        double m, s, t;
        m = x /2.0;
        s = exp (-m);
        t = s;
        for(i=1;i<(v/2);i++) {
        t *=m/i;
        s +=t; }
        return (s<1.0) ? s:1.0;
}
```

Fig 7: The Inverse Chi-Square Function: C−1

Spam Bayes is noted for using a slightly different function for I, where $I = \frac{1+H-S}{2}$.

III.ADABOOST CLASSIFIER:

AdaBoost(Adaptive Boosting) proposed by Yoav Freund and Robert Schapire which is a machine learning algorithm. Adaboost is one of the meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is sensitive to noisy data and outliers. However, it is less susceptible to the overfitting problem than most learning algorithms. The ADABoost classifier works with the concept of active learning using confidence based label sampling. The variance is used to train a classifier and obtain a scoring function which can be used to classify the mail as spam or ham [8]. This technique needs label data for training its classifier. This indicates that the data has originally been classified as spam or ham. Initially the data's were trained for classifier and produce necessitate functions to classify the spam messages.

This data can initially train the classifier which can generate the required functions for classifying spam messages. This algorithm is used to improve the training process. AdaBoost is one of the most widely used boosting techniques. This uses a classifier recursively in a series of rounds n = 1,. . . ,N. For each call a distribution of weights D (n) is updated that indicates the importance of each record in the data corpus for the classification. The weight of each wrongly classified record is increased in iteration. That is, the importance correctly classified record is decreased hence making the new classifier more sensitive to the incorrectly classified records. The examples are initially identified by the user to train the classifier manually. Additionally *k* records are identified as hard records to train the classifier to the hard examples, as a result that the efficiency of the classifier can be improved which will be used to classify the unlabelled data. [9]

The Active learning technique used is,

Given data corpus C, categorized into;

- Unlabeled data corpus C (unlabeled),
- Labeled data corpus C (labeled).

Recursively iterate;

- Using the labeled data corpus C, trains the classifier.
- Using the above generated classifier, test the C (unlabeled) corpus and the scoring functions will generate scores.
- Each record is associated with the corresponding score, which is previously generated.
- The records with lowest scores are labeled.
- Includes the newly labeled data records into C (labeled) corpus.
- Remove the newly labeled records from the C (unlabeled) corpus. This criteria (scoring) used to find the *k* hard records is Boosting in which the choice is

based on the weighted majority vote. Training is carried out by using AdaBoost algorithm.

Input: Instance distribution D;
Base learning algorithm L;
Number of learning rounds T.

Process:

1. $D_1 = D$. % Initialize distribution
2. For t = 1,…,T:
3. $h_t = L(D_t)$; %Train a weak learner from distribution D_t
4. $€_t = P_{rx\sim Dt,y}I[h_t(x)\neq y]$; % Measure the error of h_t
5. D_{t+1} = Adjust_Distribution(D_t,$€_t$)
6. End

Output: H(x) = Combine_Outputs({$h_t(x)$})

Fig 8: A general Boosting algorithm

Input: Data set D = {$(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)$};
Base learning algorithm L;
Number of learning rounds T.

Process:

1. $D_1(i) = 1/m$. % Intialize the weight distribution
2. For t=1,…,T:
3. $h_t = L(D,D_t)$; % Train a learner h_t from D using % distribution D_t
4. $€_t = P_{rx\sim Dt,y}I[h_t(x)\neq y]$; % Measure the error of h_t
5. If $€_t > 0.5$ then break
6. $\alpha_t = \frac{1}{2} \ln(\frac{1-€t}{€t})$; % Determine the weight of h_t
7. $D_{t+1}(i) = \frac{Dt}{zt} \times \begin{cases} \exp(-\propto t) if\ ht\ (xi) = yi \\ \exp(\propto t)\ \ if\ ht(xi) \neq yi \end{cases}$
$= \frac{Dt\ (i)\exp(-\propto t\ yi\ ht\ (xi))}{zt}$ % Update the % distribution, where Z_t is a normalization %factor which enables D_{t+1} to be a distribution
8. End

Output: H(x) = sign $(\sum_{t=1}^{T} \propto t\ ht\ (x))$

Fig 9: The Ada boost algorithm

IV.Bayesian Classifiers

The Bayesian approach is fundamentally an important DM technique. The Bayes classifier can provably achieve the optimal result when the probability distribution is given. Bayesian method is based on the probability theory. A Bayesian filter learns a spam classifier from a set of manually classified examples of spam and legitimate (or *ham*) messages i.e. Training collection. This training collection is taken as the input for the learning process, this consists of the following steps [11];

Preprocessing: The preprocessing is the deletion of irrelevant elements (e.g. HTML), and selection of the segments suitable for processing (e.g. headers, body).

Tokenization: This is the process of dividing the message into semantically coherent segments (e.g. words, other character strings).

Representation: The representation is the conversion of a message into an attribute-value pairs' vector [10], where the attributes are the previously defined tokens, and their values can be binary, (relative) frequencies, etc.

Selection: The selection process includes the Statistical deletion of less predictive attributes (using e.g. quality metrics like Information Gain).

Learning: The learning phase automatically building a classification model (the classifier) from the collection of messages. The shape of the classifier depends on the learning algorithm used, ranging from decision trees (C4.5), or classification rules (Ripper), to statistical linear models (Support Vector Machines, Winnow), neural networks, genetic algorithms, etc.

Naïve Bayesian Classifiers

Naive Bayes can often outperform more with sophisticated classification methods. [12] The following example shows the Naïve Bayes classifier demonstration. Here, the objects can be classified as either GREEN or RED. The task is to classify new cases as they arrive (i.e., decide to which class label they belong).

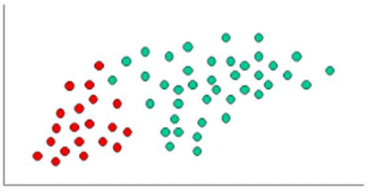

Fig 10: Objects are classified to GREEN or RED.

The calculation of the priors is (i.e. the probability of the object among all objects) based on the previous knowledge [13]. Therefore:

Prior probability for GREEN $\alpha \frac{Number\ of\ GREEN\ objects}{Total\ number\ of\ objects}$

Prior probability for RED $\alpha \frac{Number\ of\ RED\ objects}{Total\ number\ of\ objects}$

There is a total of 60 objects, 40 of which are GREEN and 20 RED, the prior probabilities for class membership are:

Prior probability for GREEN α $\frac{40}{60}$

Prior probability for RED α $\frac{20}{60}$

Having formulated the prior probability, the system is ready to classify a new object (WHITE circle in Figure 10).

As the objects are well clustered, assume that the more GREEN (or RED) objects in the vicinity of X, more likely that the new cases belong to that particular color. Then a circle is drawn around X to measure this likelihood, which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then the number of points in the circle is calculated.

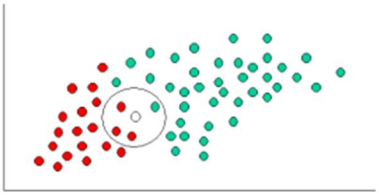

Fig 11: classify the WHITE circle.

Then the likelihood is calculated as follows:

Likelihood of X given GREEN α $\frac{Number\ of\ GREEN\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ GREEN\ cases}$

Likelihood of X given RED α $\frac{Number\ of\ RED\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ RED\ cases}$

In Figure 2, it is clear that Likelihood of X given RED is larger than Likelihood of X given GREEN, as the circle encompasses 1 GREEN object and 3 RED ones. Thus:

Probability of X given GREEN α $\frac{1}{40}$

Probability of X given RED α $\frac{3}{40}$

In the Bayesian analysis, the final classification is produced by combining both sources of information (i.e. the prior and the likelihood) to form a posterior probability using Bayes Rule.

Posterior probability of X being GREEN α

Prior probability of GREEN X Likelihood of X given GREEN

$= \frac{4}{6} X \frac{1}{40} = \frac{1}{60}$

Posterior probability of X being RED α

Prior probability of RED X Likelihood of X given RED

$= \frac{2}{6} X \frac{3}{40} = \frac{1}{40}$

Finally, classify X as RED since its class membership achieves the largest posterior probability.

V.KNN CLASSIFIER

The K Means is used to partition the objects in such a way that the intra cluster similarity is high but inter cluster similarity is comparatively low. Simply, kNN classification classifies instances based on their similarity to instances in the training data. The set of n objects are classified into k clusters by accepting the input parameter k. As an alternative of assigning to a test pattern the class label of its closest neighbor, the K Nearest Neighbor classifier finds k nearest neighbors on the basis of Euclidean distance.

$\sqrt{((x}2\text{-}x1)^2 - (y2\text{-}y1)^2)$

The value of k is very crucial because the right value of k will help in better classification. [6]

KNN selection strategies:

$y' = argmax_v \quad \Sigma_{(xi,yi)\,\epsilon\,Dz}\ \delta(v, y_i)$

$y' = argmax_v \quad \Sigma_{(xi,yi)\,\epsilon\,Dz}\ w_i\ \delta(v, y_i)$

where $w_i = 1/d(x', x_i)^2$

KNN classification algorithms:

K= number of nearest neighbors

Foreach test example $z = (x', y')do$

Compute d(x,x') foreach (x,y) $\in D$

Select Dz $\subseteq D$, the set of k

Closest training examples

$y' = argmax_v \quad \Sigma_{(xi,yi)\,\epsilon\,Dz}\ \delta(v, y_i)$

The notion of a distance/similarity measure is essential to the kNN approach. There are numerous distance/similarity measures.

$d(x, x^2) = \sum i = 1^n |x_i\text{-}x_i'|$ (Manhattan distance)

$sim(x, x^2) = \sum_{i=1}^{n} x^1 x'_i / \sqrt{\sum_{1=1}^{e} x_1^2} \sqrt{\sum_{i=1}^{e} x'_1{}^2}$ (cosine similarity)

$sim(x, x^2) = 2\sum_{i=1}^{n} x^1 x'_i / \sum_{1=1}^{e} x_1^2 + \sum_{i=1}^{e} x'_2{}^2$ (dice's coefficient)

VI.BENCHMARKING OF TECHNIQUES

The major techniques illustrated in the previous sections have been implemented and the results are shown in the table [14]. The mails are categorized as:

- The Spam mails that were incorrectly classified as ham mails.

- Spam mails that were correctly classified as spam mails.
- Ham mails that were correctly classified as ham mails.
- Ham mails that were incorrectly classified as spam mails.

Name of Technique	AdaBoost Classifier (%)	KNN classifier (%)	Chi square function (%)	Bayesian classifier (%)
Spam as ham	8	5	2	2
Spam as spam	54	57	59	65
Ham as ham	17	42	45	49
Ham as spam	32	7	4	1
Correctly Classified	78%	89%	92%	96.5%
Incorrectly Classified	22%	11%	2.8%	1.2%

VII. Conclusion

Some of the content based filtering techniques are studied and analyzed. The better technique is decided with the implementation result as shown in the tabular representation. The efficient technique among the discussed techniques is chosen as Bayesian method to create a spam filter. This gives effective outcomes rather than other methods.

REFERENCES

[1] MAAWG. Messaging anti-abuse working group. Email metrics repost. Third & fourth quarter 2006. Available at http://www.maawg.org/about/MAAWGMetric 2006 3 4 report.pdf Accessed: 04.06.07, 2006.

[2] Mikko Siponen and Carl Stucke. Effective anti-spam strategies in companies: An international study. In Proceedings of HICSS '06, vol 6, 2006.

[3] Khorsi A., "An Overview of Content-Based Spam Filtering Techniques", Informatica (Slovenia), pp. 269-277, 2007.

[4] Robinson, G. Gary Robinson's Rants. Available: http://www.garyrobinson.net.

[5] Robinson, G. A Statistical Approach to the Spam Problem. Linux J. 2003, 107 (2003), 3.

[6] Zdziarski, J. A. Ending Spam: Bayesian Content Filtering and The Art of Statistical Language Classification. No Starch Press, San Francisco, CA, USA, 2005.

[7] SpamBayes Development Team. Spambayes. Available: http://spambayes.sourceforge.net.

[8] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, Cavtat-Dubrovnik, Croatia, 2003.

[9] D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proceedings of the 14th International Conference*

[10] Salton, G. 1989. Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley.

[11] Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47.

[12] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos. Learning to ¯lter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *Proceedings of Workshop on Machine Learning and Textual Information Access*, pages 1{13, 2000.

[13] Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Augmenting naïve bayes classi¯ers with statistical language models. *Information Retrieval*, 7:317{345, 2004.

[14] Heron S., "Technologies for spam detection",Network Security, 2009, 1, pp. 11-15, 2009.

[15] M. Basavaraju, Dr. R. Prabhakar, "A Novel Method of Spam Mail Detection using Text Based Clustering Approach" ,International Journal of Computer Applications (0975 – 8887) Volume 5– No.4, August 2010