International Journal of Computational Intelligence Techniques
ISSN: 0976-0466 & E-ISSN: 0976-0474, Volume 3, Issue 2, 2012, pp.-82-85.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000221

COMPARISON BETWEEN THE PROBABILISTIC AND VECTOR SPACE MODEL FOR SPAM FILTERING

BANSAL S.*

Department of Humanities and Applied Sciences, YMCA University of Science and Technology, Faridabad-121006, Haryana, India.
*Corresponding Author: Email- soniabansal2@yahoo.com

Received: October 25, 2012; Accepted: November 06, 2012

Abstract- Spamming is the practice of sending mass mailings to large numbers of people who have no relationship with the sender and who did not ask for such emails. Growing volume of spam mails has generated a need for development of filter detecting unsolicited emails. Many spam filtering techniques have made considerable progress in recent years. The predominant approaches include data mining methods and machine learning methods. Many techniques applied to text categorization like machine learning, information retrieval but they have the difficulty of high dimensionality. Many Spam filtering techniques have been proposed in the literature. But the performance of spam filters depends on the methods applied for reducing the dimensionality is still a challenging task. In this paper, we compare the vector space model which depends on "bags of words" and probabilistic model which depends on the probability. The results possess simplicity, flexibility, performance and formalism of the probabilistic model is more as compared to the vector space model.

Keywords- Spam, Spam Filter, Information Retrieval, Vector Space Model, Probabilistic Model.

Citation: Bansal S. (2012) Comparison between the Probabilistic and Vector Space Model for Spam Filtering. International Journal of Computational Intelligence Techniques, ISSN: 0976-0466 & E-ISSN: 0976-0474, Volume 3, Issue 2, pp.-82-85.

Introduction

Email is a simple and effective method to communicate anyone and everyone in world wide. But the tremendous amount of unsolicited electronics mail, which is known as spam, has recently increased enormously and becomes a serious problem to user. Spam problem is increasing day by day as number of internet user is increasing. Mass mailing not only affect the single user account but it can create a great problem to internet servers also. By considering these problems, there exist strong requirement for a spam filter, which can protect user account as well as large mail servers. However a lot of studies have been undertaken to create and improved spam filter but in this paper, the comparison of probabilistic model and vector space model to filter the spam will be discussed. In vector space model, *tf* (term frequency) and *imf* (inverse message frequency) combined with machine learning techniques, which are commonly used to detect spam. Apart from this, the probability model for spam filtering does not rely on the arbitrary rules. Each message is assigned weight and probability that any given weight occurred in a spam is computed. By using this information it is possible to compute the overall probability that the email is spam or legitimate. Apart from Vector Space Model, the probabilistic model computes the similarity coefficient between the query and the message as probability that the message will be relevant to the query. This will reduce the relevance ranking problem. In this probabilistic model the term weight is estimated, which is based on how often the term appears or does not appear in a given message. In this paper, the use of term weights is based on probability to efficiently and effectively find the message ranking to estimate the probability of their relevance to query. The key is assign probabilities to components of the query and then use each of these as evidence in computing the final probability that the message is relevant to the query. The organization of the rest of the paper is as follows: Section 2 outlines the related work on e-mail classification techniques and Section 3 describes vector space model with example and algorithm. Section 4 presents details of probabilistic model for spam filtering and Section 5 presents the experimental results. Finally, the paper ends with conclusion in Section 6.

Related Work

E-mail classification techniques are able to control the problem in a variety of ways. Detection and protection of spam e-mails from the e-mail delivery system allows end-users to regain a useful means of communication. Over the past few years, different approaches have been presented to provide resistance against spammers. Prior studies commonly consider spam filtering a classical text categorization problem [1-6] though other approaches also are possible (e.g., blacklists of known spammers and white lists of trusted senders, hand-crafted rules that block messages containing specific words or phrases). Common classification analysis algorithms used in the context of spam filtering include Bayesian-like

approach [10-11] or a rule-based approach [3,7] and some use a cryptographic solution to protect against spamming problem [9]. There have been several studies in this application, which include keyword- based, phrase-based and character-based[17]. Naive Bayes-based [13,16] method is also another efficient approach of keyword and phrase based. It is a probabilistic classification by using features extracted from emails. Additionally, SVM [9,18] and decision tree based on ID3, C4.5, or C5 algorithm, can be identified as the representative methods to analyze keywords in email [12,15]. Finally, although unsolicited content is current affecting not only e- mail, but also search engines and blogs, this survey focuses solely on dealing with e-mail Spam.

Vector Space Model for Spam Filtering

The vector space model procedure can be divided in to three stages. The first stage is the message indexing where content bearing terms are extracted from the document text [8,14]. The second stage is the weighting of the indexed terms to enhance retrieval of message as spam. The last stage ranks the message with respect to the query according to a similarity measure. After defining the measure for the similarity between messages the design corresponding clustering algorithm has been done. The clusters are so formed on the basis of similarity used for further processing. The VSM framework is presented in [Fig-1].

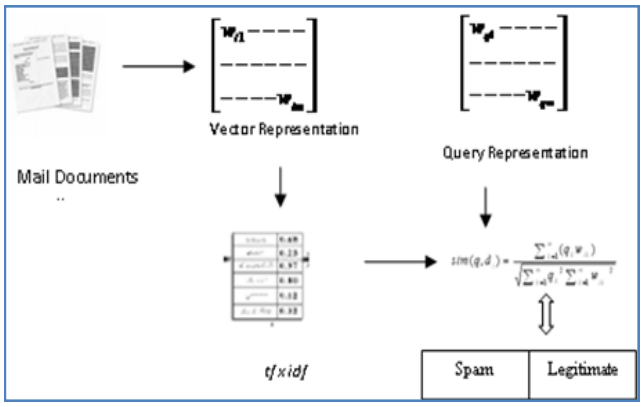

Fig. 1- Vector Space Model framework for Spam Detection

This method constructs the spam detection model by contents of various kind of mail and finds spam and legitimate from the corpus. The algorithm has been proposed as shown in [Fig-2] that can be applied on bulk of the messages to detect the spam mails.

Step 1: Collection of emails
Step 2: Split e-mail and query into matrix
Step 3: Calculate term frequency and inverse message frequency for each term ie *tf* and *imf*
Step 4: Calculation of Similarity Coefficient

- $SC(Q,M) = \sum_{j=1}^{t} w_{qj} \times m_{ij}$

  Where $M(m_{i1}$---------$m_{it})$ collection of messages with t term and $Q(w_{q1}, w_{q2}$--------$w_{qm})$ terms found on the query

Step 5: do it for all messages
Step 6: collect the spam and discard.

Fig. 2- Vector Space Model based Spam Filter Algorithm

Consider a case of query and message collection consisting of three mail

- Q: "Free ticket click"
- M1: "Guaranty of free shirt in a mall"
- M2: "Delivery of ticket movie in a ticket click"
- M3: "Guaranty of free movie in a click"

In the collection there are three mail documents n=3. If the term appears in only one of the three mail document, its *imf* is $\log(n/m_{fj})$ $=\log(3/1)=0.477$. If the term appears in two of the three mail document, its *imf* is $\log(2/1)=0.176$ and it appears in all the three documents it has an *imf*= $\log(3/3)=0$. The imf for the terms in the three mail documents are as:

$imf_a = 0$

$imf_{click} = 0.176$

$imf_{delievry} = 0.477$

$imf_{free} = 0.176$

$imf_{guaranty} = 0.176$

$imf_{in} = 0$

$imf_{mall} = 0.477$

$imf_{movie} = 0.176$

$imf_{of} = 0$

$imf_{shirt} = 0.477$

$imf_{ticket} = 0.176$

Mail document vectors now be constructed. Since eleven terms appear in the mail document collection, an eleven-dimensional mail document vector is constructed as shown in [Table-1]. The alphabetical ordering given above is used to construct the mail document vector. The weight for the term *i* in the vector *j* is computed as the *imf* × tf_{ij}.

Table 1- Terms Appear in the Collection

Message	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11
M1	0	0	0	0.176	0.176	0	0.477	0	0	0.477	0
M2	0	0.176	0.477	0	0	0	0	0.176	0	0	0.352
M3	0	0.176	0	0.176	0.176	0	0	0.176	0	0	0
Q	0	0.176	0	0.176	0	0	0	0	0	0	0.477

The mail document vectors are

SC(Q,M1) = (0)(0) + (0)(0.176) + (0)(0) + (0.176)(0.176) + (0)(0.477 + (0)(0) + (0)(0) + (0)(0.477)= $(0.176)^2 = 0.031$

Similarly,

SC(Q,M2) = $(0.352)(0.477) + (0.176)^2 = 0.519$

SC(Q,M3) = $(0.176)^2 + (0.176)^2 = 0.062$

Hence the mail document M2 having weight more than M1 and M3.

Probability Based Model for Spam Filtering

A probabilistic model based on likelihood that a term will appear in a relevant message is computed for each term in the collection. For terms that match between query and the message, the similarity measure is computed as the combination of the probabilities of

each of the matching terms. This method constructs the spam detection model based on term weight assignment which is further used to determine the total message weight as shown in [Fig-3]. Only term is taken as consideration to avoid lengthy processing.

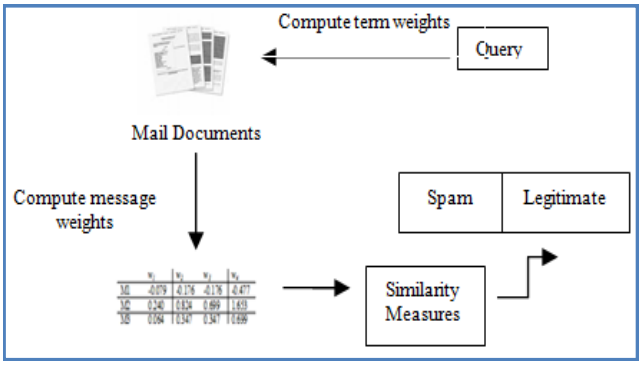

Fig. 3- Vector Space Model framework for Spam Detection

To detect the spam from the corpus the similarity measure of messages has to calculate on the bases of these term weight. The algorithm has been proposed that can be applied on bulk of the messages to detect the spam mails and shown in [Fig-4].

The terms in the query are assigned weights which correspond to the probability that a particular term, in a match with a given query, will retrieve a spam message. The weights for each term in the query are combined to obtain a final measure of relevance.

Step 1: Collection of emails
Step 2: Calculate the frequency and weight of each query term
Step 3: Message weight is computed on bases of term weight
Step 4: Similarity measures by adding the total term in the message is calculated to detect the spam
Step 5: do it for all messages
Step 6: collect the spam and discard.

Fig. 4- Probabilistic Model based Spam Filter Algorithm

Four weights are then derived based on different combinations of these ordering principles and independence assumptions [18] are shown in equation 1. Given a term *t*, consider the following quantities:

- M: number of messages in the collection
- S: number of relevant messages for a given query *q*
- m: number of messages that contain the term *t*
- s: number of relevant messages that contain term t

$$w_1 = \log\left(\frac{\frac{s}{S}}{\frac{m}{M}}\right); \quad w_2 = \log\left(\frac{\frac{s}{S}}{\frac{m-s}{M-S}}\right); \quad w_3 = \log\left(\frac{\frac{s}{S-s}}{\frac{m}{M-m}}\right); \quad w_4 = \log\left(\frac{\frac{s}{S-s}}{\frac{m-s}{(M-m)-(S-s)}}\right) \quad \ldots\ldots(1)$$

When in complete relevance information is available, 0.5 is added to the weights to account for uncertainty involved in estimating relevance. Consider the same example as previously used in vector space model to compute the similarity coefficient, assign term weights to each term in the query, then sum the weights of matching terms. Frequency of each term in query is shown in [Table-2]. Now using the equation 1, weights w_1, w_2, w_3 *and* w_4 have for each term has been computed and the results are summarized in [Table-3].

Table 2- Frequencies for Each Query Term

	free	click	ticket
M	3	3	3
m	2	2	1
S	2	2	2
s	1	2	1

Table 3- Term weight

	w_1	w_2	w_3	w_4
click	0.143	0.523	0.523	1.176
free	-0.079	-0.176	-0.176	-0.477
ticket	0.097	0.301	0.176	0.477

On the basis of term weights, the message weight has been calculated as shown in [Table-4].

Table 4- Message weight

	w_1	w_2	w_3	w_4
M1	-0.079	-0.176	-0.176	-0.477
M2	0.24	0.824	0.699	1.653
M3	0.064	0.347	0.347	0.699

The similarity coefficient for a given messages is obtained by summing the weights of term present. [Table-4] gives the similarity coefficient for each of four different weighing schemes. For M1, free is only term appear so the weight for message M1 is just the weight of free, which is -0.079. For, ticket and click which appear in M2 have total weight 0.240. For Message M3, free and click appear so the total weight of M3 is 0.064. On the basis of similarity coefficient basis the message M2 is more relevant to the query and declared spam. The ranking is same as in vector space model.

Experimental Result

To test the efficiency of our results, Initial tests have been done on 20 messages. Both the filters performed well. But in this comparison, Probabilistic model for spam detection is more efficient than the vector space model. The spam precession is (approx. 90%) more in probabilistic model as that of vector space model (87%). Apart from the query frequency and term weight calculating scheme is better than the *tf.idf* lengthy scheme. The results predict that probability model is more fast and effective than the vector space model with the less complexity of index and positioning in the vector space model.

Conclusion

In the vector space model, the messages and query are represented as vectors. Similarity among the messages and between message and the query is defined in terms of distance between two vectors. As one of common similarity measures is the cosine similarity, in which the difference between two documents or document and a query as the cosine of angle between these two vectors. In the probabilistic model, the individual term estimates can be combined into a total estimate of relevance of spam, it is necessary to

describe a means of estimating the individual term weights. The main advantage to the probabilistic model is that it is entirely based on probability theory. The inference is that other models have a certain arbitrary characteristic and perform well experimentally, as they are lack in theoretical basis, the reason is that, the parameters are not easy to estimate. The probabilistic model uses the basic probability theory with some key assumptions to estimate the probability of relevance. Here the term weight issued to calculate the real spam from the set of emails.

References

[1] Androutsopoulos I., Koutsias J., Chandrinos K.V., Paliouras G. and Spyropoulos C.D. (2000) *11th European Conference on Machine Learning, Barcelona*.

[2] Carreras X., Márquez L. (2001) 4*th International Conference on Recent Advances in Natural Language Processing, Bulgaria*, 58-64.

[3] Cohen W.W., Singer Y. (1996) 19*th ACM International conference on Research and Development in Information Retrieval*.

[4] Drucker H., Wu D., Vapnik V. (1999) *IEEE Transactions on Neural Networks,* 10(5), 1048-1054.

[5] Fdez-Riverola F., Iglesias E.L., Dı´az F., Me´ndez J.R., Corchado J.M. (2007) *Expert Systems with Applications,* 33(1), 36-48.

[6] Goodman J., Cormack G.V., Heckerman D. (2007) *Communications of the ACM,* 50(2), 24-33.

[7] Han J. and Kamber M. (2001) Data Mining Concepts and Techniques., 284-287.

[8] Hidalgo J.M., Bringas G.C., Sanz E.P. and F.C. (2006) *ACM Symposium on Document Engineering, Amsterdam, The Netherlands,* 107-114.

[9] Joe I. and Shim H. (2010) *Future Generation Information Technology,* 6485, 577-584.

[10] Liu P., Jian-she Dong and Zhao W. (2007) *Advances in Intelligent and Soft Computing, Theoretical Advances and Applications of Fuzzy Logic and Soft Computing*, 42, 527-534.

[11] Mehran Sahami, Susan Dumais, David Heckerman, Eric Horvitz (1998) *AAAI Tech Rep*., WS-98-05.

[12] Pantel P., Lin D. (1998) *Workshop on Learning for Text Categorization, Madison, WI.*

[13] Robertson and Sparcj Jones (1976) *Journal of American Scoiety for information Science*, 27(3), 129-146.

[14] Sebastiani F. (2002) *ACM Computing Surveys,* 34(1), 1-47.

[15] Takashita T., Itokawa T., Teruaki Kitasuka and Aritsugi M. (2008) *Knowledge-Based Intelligent Information and Engineering Systems,* 5178, 774-781.

[16] Wang J., Ke Gao, Jiao Y. and Gang Li (2009) *Advanced Data Mining and Applications*, 5678, 314-325.

[17] Weiss S.M., Apte C., Damerau F.J., Johnson D.E., Oles F.J., Goetz T. and Hampp T. (1999) *IEEE Intelligence Systems,* 14 (4), 63-69.

[18] Xie C., Ding L.D., Xin Du (2009) *Advances in Computation and Intelligence,* 821, 49-357.