Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 12 Issue 13 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Approach to Email Classification Using Bayesian Theorem

By Denil Vira, Pradeep Raja & Shidharth Gada

K.J.Somaiya College of Engineering, Mumbai, India

Abstract - Email Classifiers based on Bayesian theorem have been very effective in Spam filtering due to their strong categorization ability and high precision. This paper proposes an algorithm for email classification based on Bayesian theorem. The purpose is to automatically classify mails into predefined categories. The algorithm assigns an incoming mail to its appropriate category by checking its textual contents. The experimental results depict that the proposed algorithm is reasonable and effective method for email classification.

Keywords : Bayesian, Email classification, tokens, text, probability, keywords.

GJCST-C Classification: E.5

An Approach to Email Classification Using Bayesian Theorem

Denil Vira[α], Pradeep Raja[σ] & Shidharth Gada[ρ]

Abstract - Email Classifiers based on Bayesian theorem have been very effective in Spam filtering due to their strong categorization ability and high precision. This paper proposes an algorithm for email classification based on Bayesian theorem. The purpose is to automatically classify mails into predefined categories. The algorithm assigns an incoming mail to its appropriate category by checking its textual contents. The experimental results depict that the proposed algorithm is reasonable and effective method for email classification.

Keywords : Bayesian, Email classification, tokens, text, probability, keywords.

I. Introduction

Internet e-mail is an essential communication method for most computer users and has been treated as a powerful tool intended to idea and information exchange, as well as for users' commercial and social lives. Globalization has resulted in an exponential increase in the volume of e- mails. Nowadays, a typical user receives about 40-50 email messages every day. For some people hundreds of messages are usual. Thus, users spend a significant part of their working time on processing email. As the popularity of this mean of communication is growing, the time spent on reading and answering emails will only increase. At the same time, a large part of email traffic consists of non-personal, non time critical information that should be filtered. Irrelevant emails greatly affect the efficiency and accuracy of the aimed processing work. As a result, there has recently been a growing interest in creating automated systems to help users manage an extensive email flow.

Consider the following scenario –

You have just returned from a relaxing two week vacation. There has been no phone, no email for two wonderful weeks, and now you are back. You open your inbox and ... Wow! There are 347 new messages! How could you manage to read all of them? Probably, you will spend the whole day trying to sort out all this mail. Having done this burdensome work, you feel like you need a vacation again. What is worse is that most of those messages are out of your interest or out of date.

Here comes the need for automatic email classification system that would sort the important and the unimportant mails, thus saving a much precious time of the users.

The rest of the paper is organized as follows. The next section describes the generic Bayesian filtering logic. Section 3 introduces the proposed algorithm for email classification. Section 4 presents experiments and results. The final section consists of the conclusion.

II. Bayesian Filter

Bayesian filter has been used widely in building spam filters. The Naïve Bayes classifier is based on the Baye's rule of conditional probability. It makes use of all the attributes contained in the data, and analyses them individually as though they are equally important and independent of each other. The rule for conditional probability is as follows

$$P(H \mid E) = P(E \mid H)P(H) / P(E) \qquad \text{.. Eq.} \quad (1)$$

Where *P(H/E)* is the conditional probability that hypothesis H is true given an evidence E; *P(E/H)* the conditional probability of E given H, *P(H)* the prior probability of H, *P(E)* the prior probability of E. In the case of spam classification, hypothesis H can be defined as spam or legitimate given an email E.

In applying this theorem, an email needs to be tokenized, and the extracted *n* tokens (i.e. words or phrases) are then used as the evidences E{e1, e2,…,en}, and their probability of spam or non-spam is calculated from previous emails that have been classified. Using past classified emails to estimate the probabilities of the tokens belonging to spam or no spam is a learning process and then they are used to predict the spam probability of a new incoming email. Assume that n key-words are extracted from the content of an email as evidences, then the probability that the email is spam can be calculated by:

$$P(S|E(\,e1,e2..\,en)) =$$
$$P(E(e1,\,e2,..\,en)|\,S)*P(S)\,/\,P(E(\,e1,e2..\,en) \qquad \text{… Eq.} \quad (2)$$

Where, *P(S/E)* is the probability that an email E is spam S. In practice, an assumption is commonly made naïvely for simplifying calculation, that is, all the evidences are independent from each other. For example, if an email contains four evidences: e1, e2, e3 and e4, then the joint and conditional probabilities given S can be easily calculated by,

Author α σ ρ : Department of Computer Engineering, K.J.Somaiya College of Engineering, Mumbai, India.
E-mail α : denilvira@gmail.com
E-mail σ : pradeepraja13@gmail.com
E-mail ρ : siddgada@gmail.com

$$P(e1,e2,e3,e4) = P(e1)P(e2)P(e3)P(e,),$$
$$P(e1,e2,e3,e4|S) = P(e1|S)P(e2|S)P(e3|S)P(e4|S)$$

Thus the probability that a given email is considered spam when evidences (e*1, e2, e3* and *e4*) appear in the given email is calculated by Equation (2), and a decision can be made if it is higher than a pre-set decision threshold, 0.5, usually.

III. Proposed Algorithm for Email Classification

The entire email classification process is divided into two phases.

a) First phase is the Training phase/Learning phase

During this phase, the classifier will be trained to recognize attributes for each category. So that later on when a new mail arrives, it compares the attributes of the mail with attributes of each category and the mail is classified into the category having most similar attributes as that of the mail. To build the attribute list (also referred as keywords database) for each category, the emails will be classified manually in to different categories by the user. For better understanding of the working, we create two categories for emails say, *work* and *personal*.

The user manually specifies whether each mail belongs to *work* or *personal* category based on its contents including the subject and the body.

For each email classified manually by the user, the algorithm extracts keywords from the mail and stores them into the keyword database for that particular category along with count of the number of occurrences. For eg. If the user classifies the mail as *work,* the keywords extracted from that mail will be stored in *work_database.* Every category will have its own set of keywords database.

Also, before the mail is processed for keywords, it is first filtered. Mail is divided into set of tokens separated by blank space or any other punctuation marks of English language. Proverbs, articles, html tags, noise words and other unnecessary contents are removed and then the keywords are extracted. Thus the training phase is responsible for building the keyword database for each predefined category.

b) The second is Classification phase

Once the learning is sufficiently done, the algorithm is ready to move to next phase. The new mails that arrive should to be automatically assigned their categories. Basically, here we compare the contents of the mail with those of all category keyword databases using Bayesian theorem and look for best matching category for the mail. The new incoming mail (also referred as unclassified mail) is broken into tokens and filtered. The tokens are then compared with keyword databases of each category. The probability that the mail belongs to a category is found out for every category. The category for which the probability outcome is highest, it is then compared with the threshold of same category and classified into that category if following condition holds true else the mail stays unclassified.

P_n= 1 , if P(category | E) > threshold
P_n= 0 , if P(category | E) < threshold
Where E is the newly arrived mail.

IV. Algorithm

a) Training /Learning

1. *For each mail, specify its category manually.*
2. *Divide the mail into tokens (both subject and body)*
3. *Filter out stop words such as html tags, articles, proverbs, noise words.*
4. *Extract keywords and store them along with frequency count in to keywords database of the selected category.*

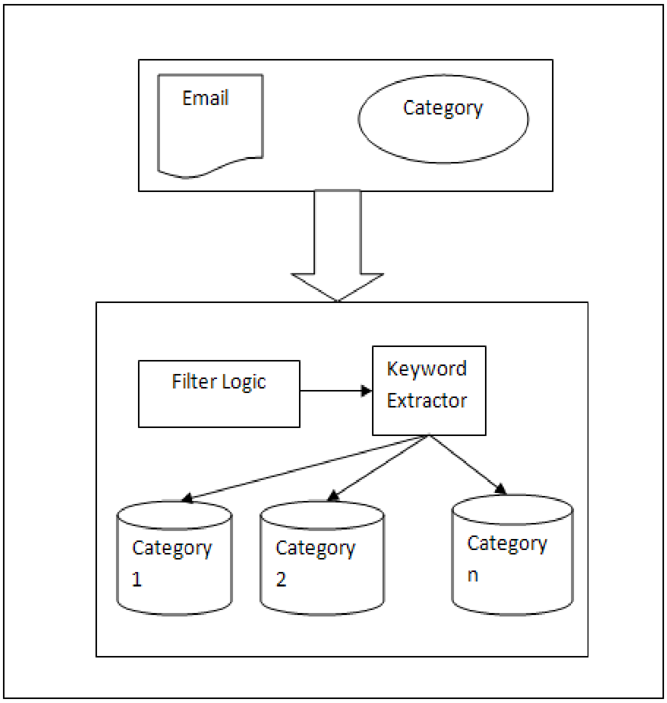

Figure 1 : Training process

b) Classification

1. *For each newly arrived mail, divide the mail into set of tokens. (Consider both, the subject and the body)*
2. *Filter out stop words such as html tags, articles, proverbs, noise words and extract the keywords, say E {e1,e2, e3…en} is the list of extracted keywords.*
3. *Find P(category| E) = P(E | category)*P(category)/ P(E) for all categories where , P (E | category) = P(e1|category)*P(e2|category)*…..*P(en | category)*
4. *Find the category for which the value of P(category | E) is highest.*

5. Compare the value with threshold value of that category.

If $P(category \mid E) > threshold$, the mail is classified into that category.

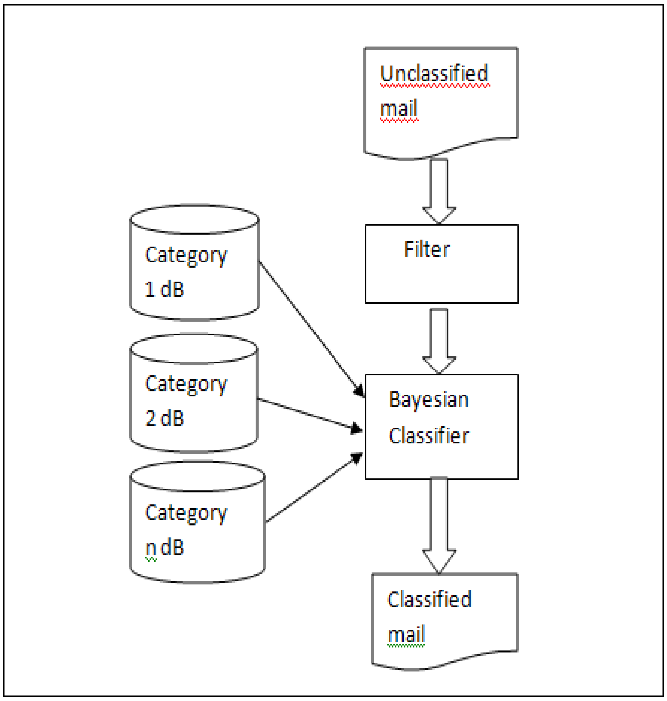

Figure 2 : Classification process

V. Implementation and Results

The implementation of the algorithm was done in Java. Authors' working email accounts were used for fetching the mails. A total of 5175 e-mails are used for the purpose of experiment.

In classification task, the performance of algorithm is measured in terms of accuracy.

Accuracy = N_c / N

N_c : Number of correctly classified mails

N : Number of total mails classified

The accuracy of classification is dependent on several parameters such as number of mails considered during training phase, number of categories defined, threshold value of each category , size of mail etc.

We initially created two categories, *work* & *personal* and then gradually added three more categories. The test was repeated for different values of parameters. Through repeated tests, we choose the best-performing configuration for our algorithm. 10-fold cross-validation was used in our test: the corpus was partitioned randomly into ten parts, and the experiment was repeated ten times, each time reserving a different part for testing and using the remaining nine parts for training. All the results were then averaged over the ten iterations.

Table 1 : Performance results of Email classifier based on proposed algorithm

Size of training data	No. of categories	Accuracy
100	2	0.841
200	2	0.925
600	2	0.992
1000	2	0.991
200	3	0.936
500	3	0.973
1000	3	0.995
200	4	0.762
500	4	0.836
1000	4	0.861
200	5	0.846
500	5	0.921
1000	5	0.967

The size of training data is the number of mails manually classified during learning phase.

Note that the accuracy is also dependent on number of overlapping words among different categories. We observed that when two different categories have many keywords in common, the classifier accuracy is low. With distinct keywords for each category, with minimum overlapping, the classifier accuracy was above 0.9. The results also show that larger the size of training data better is the accuracy.

VI. Conclusion

Considering the requirements of improving efficiency in processing emails, this paper introduced an approach to classify mails based on Bayesian theorem. The algorithm was implemented and the experimental results were studied. The results show that our approach to classify mails is a reasonable and effective one. However, there is lot of work to be done in future. The future research includes improving accuracy of the classifier ,working with languages other than English, making the keyword extraction more efficient, classifying based on attachments, understanding the semantics of email text and so on.

VII. Acknowledgements

The authors would like to acknowledge the contribution of Mr.Dhruven Vora and Mr.Jenish Jain for helping us in coding of the algorithm and testing of results. We would also like to acknowledge Ms.Prof. Kavita Kelkar for valuable suggestions and guidance. Lastly, we would like to thank the entire Department of Computer Engineering, K.J.Somaiya College of Engineering, Mumbai for providing us with all the resources needed for the research.

References Références Referencias

1. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail,” Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pp. 55-62, 1998.
2. I.Androutsopoulos, G. Paliouras, E. Michelakis, “Learning to Filter Unsolicited Commercial E-Mail,” Technical Report of National Centre for Sciential Research “Demokritos”, 2004.
3. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras and C.D. Spyropoulos, “An Evaluation of Naïve Bayesian Anti-Spam Filtering,” Proc. of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, 2000, pp. 9-17.
4. S. Youn and D. McLeod, “Efficient Spam e-mail Filtering using Adaptive Ontology,” International Conference on Information Technology (ITNG'07), pp.249-254, 2007.
5. S. Hershkop, and J. Stolfo, “Combining e-mail models for false positive reduction,” Proc of KDD’05 of ACM. Chicago : [s.n.], pp. 98—107, 2005.
6. Andrew McCallum, Kamal Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” AAAI-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05, 1998, pp. 41-48.
7. Wang, W. et al (2000): Diversity between neural networks and decision trees for building multiple classifier systems. *Multiple Classifier Systems*: pp 240-249.
8. R. Beckermann, A. McCallum, and G. Huang. Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora. Technical report IR-418, University of Massachusetts Amherst, 2004.
9. F. Peng, D. Schuurmans, and S. Wang. Augmenting naive bayes classifiers with statistical language models. Information Retrieval, 7:317–345, 2004.
10. David D. Lewis and William A. Gale. A sequential algorithm for training text classfiers. In Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, 1994.