BAYES vs NAIVE BAYES CLASSIFIER IN SPAM FILTERING

NITHIN SYLESH
DEPARTMENT OF
COMPUTER SCIENCE
AND ENGINEERING
AMRITA VISHWA VIDHYAPEETHAM
AMRITAPURI

YADHUKRISHNAN J
DEPARTMENT OF
COMPUTER SCIENCE
AND ENGINEERING
AMRITA VISHWA VIDHYAPEETHAM
AMRITAPURI

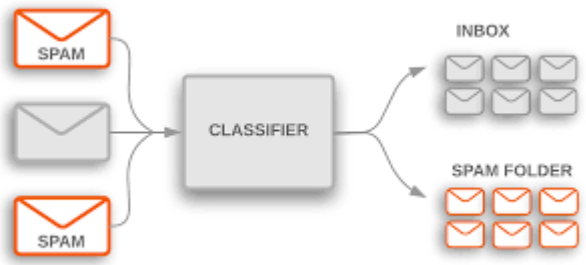
AKASH
DEPARTMENT OF
COMPUTER SCIENCE
ENGINEERING
AMRITA VISHWA VIDHYAPEETHAM

ADITHYA D
DEPARTMENT OF
COMPUTER SCIENCE
AN ENGINEERING
AMRITA VISHWA VIDHYAPEETHAM
AMRITAPURI

***Abstract*– Electronic mail (email or e-mail) is a method of exchanging messages ("mail") between people using electronic devices**

Email spam is nothing but junk email or unsolicited bulk emails sent through the email system. It refers to the use of an email system to send unsolicited emails especially advertising emails to a group of recipients. Unsolicited emails mean the recipient did not grant permission for receiving those emails.In this paper we apply naïve bayes classifier to filter spam emails

From ham emails. we apply different criteria of filters for reducing spam as well as to detect spam



1. INTRODUCTION

A **spam filter** is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. Like other types of **filter**ing programs, a **spam filter** looks for certain criteria on which it bases judgments.

Mailbox providers can also install mail filters in their mail transfer agents as a service to all of their customers. Anti-virus, anti-spam, URL filtering, and authentication-based rejections are common filter types.

Corporations often use filters to protect their employees and their information technology assets. A catch-all filter will "catch all" of the emails addressed to the domain that do not exist in the mail server - this can help avoid losing emails due to misspelling.

Users, may be able to install separate programs (see links below), or configure filtering as part of their email program (*email client*). In email programs, users can make personal, "manual" filters that then automatically filter mail according to the chosen criteria.

As technology around spam filtration increases, it will be more difficult for your legitimate email to make it to the recipient's inbox. One small mistake can lead to a snowball of issues down the road. This is why it's so important to learn about what spam filters are and how they work. Making spam filters a tool to be used to your advantage will help you build a successful email platform that out-performs your competitors.

2. BAYESIAN FILTER

A bayesian filter is a filter that learns your spam preferences. When you mark emails as spam, the system will note the characteristics of the email and look for similar characteristics in incoming email, filtering anything that fits the formula directly in to spam for you. A bayesian filter is one of the most intelligent types of spam filter because it is able to learn and adapt on its own.

If a word never appears in spam but often in the legitimate email you receive, the probability that word indicates spam is near zero. For example, say you receive many legitimate messages that contain the word *Cartesian.* That fact decreases the likelihood that email messages you receive containing the word *Cartesian* are spam. On the other hand, say you rarely or ever receive legitimate messages that contain the word *toner.* If you receive a message that does contain the word *toner,* it's likelier to be spam.

A. HOW BAYESIAN FILTER EXAMINES AN EMAIL

Message characteristics a Bayesian spam filter looks at include:

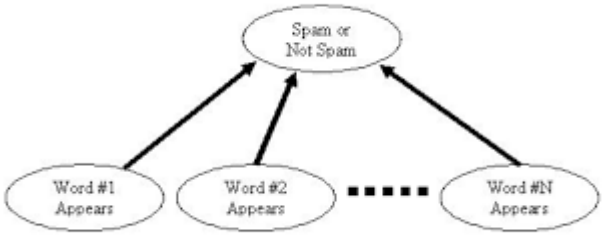
- Words in the body of the message
- Words in the message header (such as the sender and message path)
- Other elements such as HTML/CSS code (such as colors and other formatting)
- Word pairs and phrases
- Meta information (such as where a particular phrase appears)

When a new message arrives, the Bayesian spam filter analyzes it and calculates the probability of it being spam according to these attributes.

Continuing with the examples above, suppose a message contains both words, *Cartesian* and *toner*. From these words alone it's not clear whether the message is spam or legitimate email. Following the classification into "spam" or "legitimate email," the filter can use that determination to further train

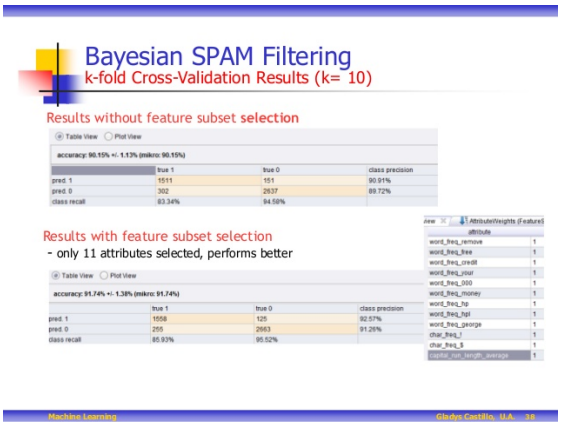
Using this auto-adaptive technique, Bayesian filters can learn from both their own and users' (if they manually correct wrongly evaluated messages) decisions. The adaptability of this system ensures these filters are most effective for individual email users because, while most people's spam may have similar characteristics, legitimate mail is characteristically different for each person.

B. CAN SPAM MAILS GET PASS THE BAYESIAN FILTER?



Spammers' messages only make it past well-trained Bayesian filters if the tricksters make their spam look like a perfectly ordinary email.
But spammers don't usually send such ordinary messages because they don't work well to serve their purposes (i.e. convince you to buy something or click a link).

As good as a Bayesian filter might be, one word or characteristic that frequently appears in good mail can be so significant as to prevent a message that contains it from being rated as spam. Therefore, if spammers could find a way to determine your sure-fire good-mail words they could include one of them in a junk mail and reach you even through a well-trained Bayesian filter. But, according to researchers who have tried this method, it's time-consuming and complex enough that it's not likely to be used very frequently.



2. NAÏVE BAYES SPAM FILTER

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use bag of words features to identify spam e-mail, an approach commonly used in text classification.

Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayes' theorem to calculate a probability that an email is or is not spam.

Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the email needs of individual users and give low false positive spam detection rates that are generally acceptable to users. It is one of the oldest ways of doing spam filtering

A. HOW NAÏVE BAYES FILTER AN EMAIL

Bayesian email filters utilize Bayes' theorem. Bayes' theorem is used several times in the context of spam:

- a first time, to compute the probability that the message is spam, knowing that a given word appears in this message;
- a second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them);
- .
- sometimes a third time, to deal with rare words.

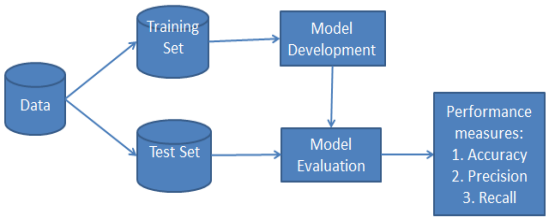
The formula used by the software to determine that, is derived from Bayes' theorem

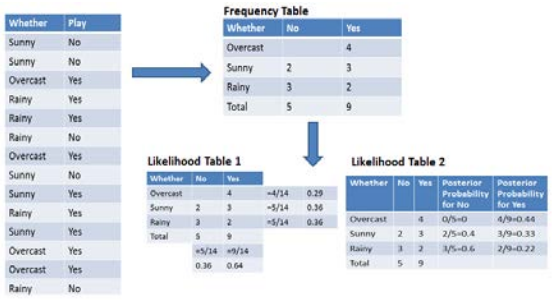
$$Pr(S|W)=Pr(W|S)*Pr(S)/Pr(W|S)*Pr(S)+Pr(W|H)*Pr(H)$$

where:

- is the probability that a message is a spam, knowing that the word "replica" is in it;
- is the overall probability that any given message is spam;
- is the probability that the word "replica" appears in spam messages;
- is the overall probability that any given message is not spam (is "ham");
- is the probability that the word "replica" appears in ham messages

"Neutral" words like "the", "a", "some", or "is" (in English), or their equivalents in other languages, can be ignored. More generally, some bayesian filtering filters simply ignore all the words which have a spamicity next to 0.5, as they contribute little to a good decision. The words taken into consideration are those whose spamicity is next to 0.0 (distinctive signs of legitimate messages), or next to 1.0 (distinctive signs of spam).

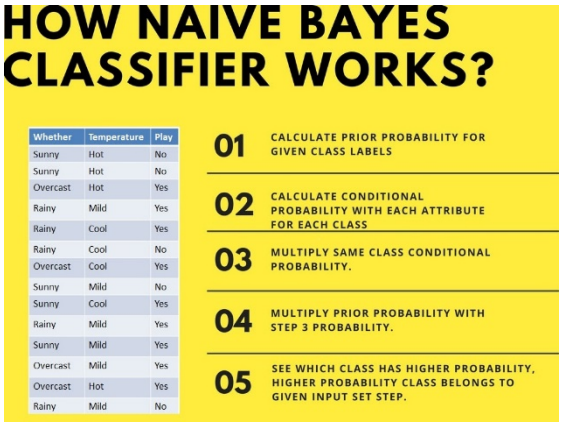




SECOND APPROACH

FIRST APPROACH

- ***Step 1: Calculate the prior probability for given class labels***
- ***Step 2: Find Likelihood probability with each attribute for each class***
- ***Step 3: Put these value in Bayes Formula and calculate posterior probability.***
- ***Step 4: See which class has a higher probability, given the input belongs to the higher probability class.***



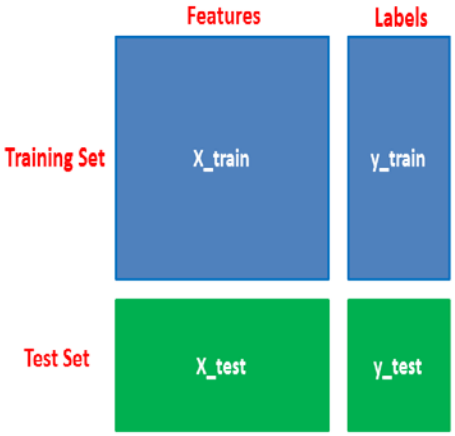
Generating Model

Generate a model using naive bayes classifier in the following steps:

- **Create naive bayes classifier**
- **Fit the dataset on classifier**
- **Perform prediction**

SPLITTING DATA

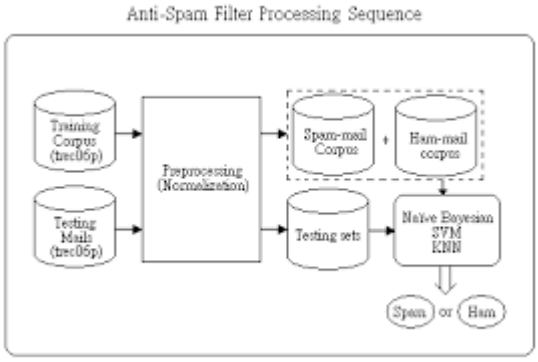
First, you separate the columns into dependent and independent variables(or features and label). Then you split those variables into train and test set.



After splitting we need to generate a random forest model of training set and perform prediction on test set

EVALUATING MODEL

After model generaton check accuracy using actual and predicted values



DATASET AND METHODOLOGY

There has been significant effort to generate public benchmark datasets for anti-spam filtering .one of the main concern is how to protect privacy of users whose ham messages are included in dataset the first approach is to use ham message collected from freely accessible newsgroups,or mailing list with public archive

An alternative solution to this privacy problem is to distribute information of each message.the spambase collection follows this approach,it consist of vecors representing single message with each vector containing value of pr selected attributes

The third approach is to release benchmark each consisting of of message received by a particular user after replacing each token by a particular number . the mapping between token and number is not releasedMaking it extremely difficult to recover orginal message thus bypass privacy problems

One of our main goal of evaluation was to emulate the situation that a new user of a personalised learning based on anti spam filter face:

The user starts with a small amount of training message and retrains filter dor a new message,this incremental retraining and evaluation differ significantly from cross validation experiments that are used to measure performance of learning algorithm

There are many reasons for this including varying reasons of training set,the

increasingly more sophisticated tricks used by spam sender over time,varying proportion of of spam to ham message which makes estimation of prior difficult and topic shift of spam over time

Hence an incremental retraining and evaluation procedure that also takes into account of characteristics of spam is essential when comparing different learning levels of spam filtering

In order to understand the incremental procedure whit the use of our dataset we needed to order the message of each dataset in a way that preserves the orginal order of arrival of each message in each category i.e each spam mail must be preseded by spam that received earlier

The same applies to ham messages

The varying ham-spam ratio over time also had to be emulated

This was achieved by using following algorithm in each dataset

1. Let S and H be the set of spam and hammessages of dataset.
2. Order messages of h by time of arrival
3. Insert spam slots between the ordered messages of H by independent random draws with replacements ,if the outcome of the draw is I,a new spam slot is inserted after the I th node .a ham message may thus be followed by several solts
4. Fill the spam slots with message of S by iteratively filling the earliest empty spam slot with the oldest message of S that has not been placed to a slot.

   The actual dates of the message are then discarded and we assume that the messages of each dataset arrive in order produced by above algorithm
   The below fig shows the resulting fluctuation of ham spam ratio

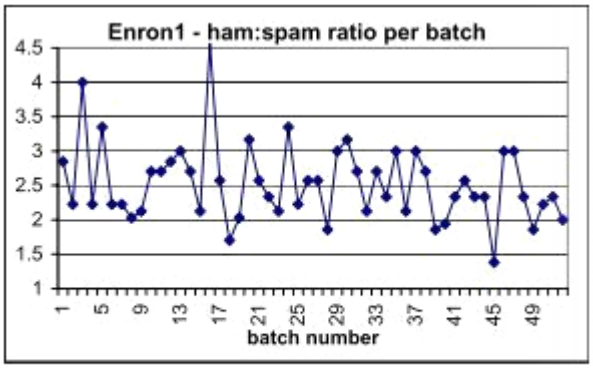


Figure 1: Fluctuation of the ham-spam ratio.

In each ordered dataset the incremental retraining and evaluation process was implemented as follows

1. Split sequences of message into batches of adjacent message each .preserving the order of arrival
2. Train the filter (including attribute selection) on the batch of message and test it on message of bi+1.

At the end of the evaluation each message of the dataset have been classified exactly once. The number of true positive (TP)is number of spam message that have been classified as spam and similarly for false positive (FP),ham misclassified as spam;true negative (TN) correctly classified as hamand false negative (FN) spam misclassified has ham

In all our evaluation we used spam recall(TP/TP+PTN) and ham recall as (TN/TN+FP).spam recall is the proportion of spam message that the filter managed to identify correctly whereas the ham recall is the

proportion of ham message that passed the filter

p

an NB-based filter can easily be retrained on line immediately after receiving each new mail

EXPERIMENTAL RESULTS

SIZE OF ATTRIBUTE

We first examined the impact of number of attributes . the difference in effectiveness across different number of attributes are rather insignificant .in operational filter the difference in effectiveness may not justify the increased computational cost that larger attribute set require even though the increase in computational cost is linear in number of attributes

CONCLUSION

We have discussed and evaluated experimentally in a spam filtering context .we emulated the situation faced by a new user of personalise learning based spam filter adopting an incremental retraining and evaluation procedure .the dataset we udes are publically available

In the website csmining.org

REFERENCES

1. *Brunton, Finn (2013). Spam: A Shadow History of the Internet. MIT Press. p. 136. ISBN 9780262018876. Archived from the original on 2019-03-23. Retrieved 2017-09-13.*
2. ^ *M. Sahami; S. Dumais; D. Heckerman; E. Horvitz (1998). "A Bayesian approach to filtering junk e-mail" (PDF). AAAI'98 Workshop on Learning for Text Categorization. Archived (PDF) from the original on 2013-06-26. Retrieved 2007-08-15.*
3. ^ Paul Graham (2003), Better Bayesian filtering Archived 2010-06-21 at the Wayback Machine
4. ^ Brian Livingston (2002), Paul Graham provides stunning answer to spam e-mails Archived 2010-06-10 at the Wayback Machine
5. ^ *"Junk Mail Controls". MozillaZine. November 2009. Archived from the original on 2012-10-25. Retrieved 2010-01-16.*
6. ^ *"Installation". Ubuntu manuals. 2010-09-18. Archived from the originalon 29 September 2010. Retrieved 2010-09-18. Gary Robinson's f(x) and combining algorithms, as used in SpamAssassin*
7. ^ *"Background Reading". SpamBayes project. 2010-09-18. Archivedfrom the original on 6 September 2010. Retrieved 2010-09-18. Sharpen your pencils, this is the mathematical background (such as it is).* The paper that started the ball rolling: Paul Graham's A Plan for Spam.* Gary*