CSE 537. Artificial Intelligence
Project Report
Naive Bayes Spam Filtering

Yingtao Tian, ID: 109761013
Konstantin Dmitriev, ID: 109770677

December 6, 2014

1 Introduction

Email spam is one of the most important problem with electronic correspondence nowadays. Most people are spending incredible amour of their time on distinguishing unwanted messages from useful ones. There are a number of software systems that perform spam filtering based on a set of rules that are predefined by users. However, the bottleneck of such systems is that they rely heavily on the assumption that this set of rules is robustly adjusted for every possible spam mail.

Nevertheless, there are a lot of different automated approaches that aims on spam prevention as well, and filtering is one of them. Most filtering techniques take advantage of machine learning, which can improve the their accuracy in comparison to manual methods. Methods based on the machine learning do not require a specified set of rules, instead, they require a set of pre-classified emails, that are called training samples. Using these samples, method "learns" the classification rule that can be applied to an email from outside of training sample, to successfully categorize it.

We have developed a spam identification software that distinguishes junk messages from the valuable messages. The process of identification is based on the Bayesian classification technique.

Approach that is based on the Naive Bayes has several benefits comparing to the others approaches, for example, that uses SVM. One of the reason in the difference of performances of those two approaches is that SVM is based on the iterative evaluation, whereas Naive Bayes classifier can be created in a single pass and the process of classification itself requires only one lookup table.

2 Implementation

2.1 Bayesian Classification Technique

To recognize a mail that can be considered as spam, we have built the probabilistic classifier based on Bayesian networks. All of the mails from the testing dataset have been grouped in two classes:

1. Spam (S);
2. Ham (H).

The probability $P(C = c_k|\mathbf{X}{=}\mathbf{x})$ of a particular message, described by the vector of features $\mathbf{x}$, belonging to a particular class c_k, has been calculated based on the given mails from the training dataset. This was done by via Bayes theorem,

$$P(C = c_k|\mathbf{X}{=}\mathbf{x}) = \frac{P(\mathbf{X}{=}\mathbf{x}|C = c_k)P(C = c_k)}{P(\mathbf{X}{=}\mathbf{x})}, \tag{1}$$

where

$$P(\mathbf{X}{=}\mathbf{x}|C = c_k) = \prod_i P(X_i = x_i|C = c_k). \tag{2}$$

The visually simplified version of the eq. (1) can be written as

$$P(L_1|W) = \frac{P(W|L_1)}{P(W|L_1) + P(W|L_2)}, \tag{3}$$

where $P(L_1|W)$ is the probability that a message is $Label_1$, knowing that a word W is in it; $P(W|L_1)$ is the probability that the word W appears in $Label_1$ messages (approximated by its frequency); $P(W|L_2)$ is the probability that the word W appears in $Label_2$ messages (approximated by its frequency).

Following the approach presented in the paper, we represent mail message as a feature vector so as to make such Bayesian classification methods directly applicable. Each individual message was represented as a binary vector denoting which words are present and absent in the message.

2.2 Dataset

For the evaluation purposes the 2005 TREC Public Spam Corpus was used. It contains a training set (3837 messages labeled as ham; 5163 messages labeled as spam) and a testing set (420 messages labeled as ham; 580 messages labeled as spam). Both types of messages, that are represented as a line, have the same structure:

ID Label Word1 Frequency1 ... WordN FrequencyN

2.3 Email's representation

Each email either from Train database or Test database is denoted by the element in the list ($emails$) property of the instance of the class EmailData. This element is represented by the 3-tuple: $(ID, Label, \{Word1 : Frqn1; ...; WordN : FrqnN\})$. To create the list of emails class EmailData has a method ($load\_from\_file$ ($filename$: str)) that reads emails from the file, creates respective 3-tuples and add them to the list. UML diagram of EmailData class is presented on the Fig. 1.

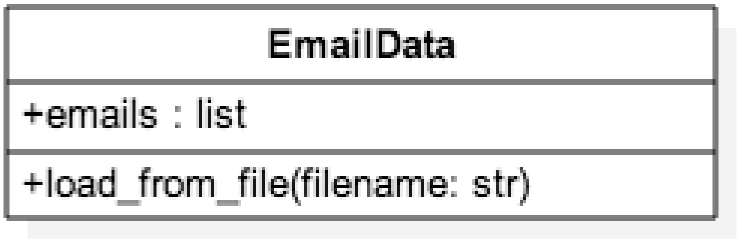

Fig. 1 UML Class diagrams for EmailData class.

2.4 Modifications

After implementing the plain Naive Bayes classifier, we also decided to add some modifications to it, Laplace Smoothing in particular.

One of the reason to use smoothing is to handle cases when, for example, classifying a particular email, classifier encounters a word that wasn't in the training data.

To so, we implemented the following modification:

$$P(W|L_i) = \frac{P(W, L_i) + \alpha}{P(L_i) + n\alpha}, \tag{4}$$

where α is a smoothing parameter.

2.5 Results

In this section we present the results of the using the plain Naive Bayes approach and the approach with Laplacian Smoothing (with different smoothing parameters). Several important number are reported in the results. The most important one is the number of ham messages classified as spam messages. Mistakenly blocking a legitimate message (classifying as spam) is generally more severe than letting a spam message pass the filter (classifying a spam message as legitimate). This scenario can cause the loss of the potentially important information.

We conducted a set of experiments using a small training database (490 ham messages, 683 spam messages), a medium training database (2150 ham messages, 2082 spam messages) and a large training database (3837 ham messages, 3876 spam messages).

The results for each test case were evaluated by the precision and recall values for spam and ham messages. Precision is the fraction of retrieved instances that are relevant. Considering the true positive (TP) and false positive (FP) for classifying spam messages (spam messages were classified as spam; ham messages were classified as spam, respectively), precision is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall is defined as:

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

where FN denotes false negative results (spam messages were classified as ham).

The overall accuracy (total precision) of classification was evaluated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

2.5.1 Plain Naive Bayes

At first, we tested the implemented algorithm using plain Naive Bayes approach.

Large Training Database

Spam as Ham	10
Ham as Spam	52
Spam as Spam	570
Ham as Ham	368

Table 1: Total: precision = 0.938 Junk: precision = 0.916, recall = 0.983 Legitimate: precision = 0.974, recall = 0.88

Medium Training Database

Spam as Ham	38
Ham as Spam	50
Spam as Spam	530
Ham as Ham	382

Table 2: Total: precision = 0.912 Junk: precision = 0.933, recall = 0.914 Legitimate: precision = 0.884, recall = 0.91

Small Training Database

Spam as Ham	83
Ham as Spam	15
Spam as Spam	497
Ham as Ham	405

Table 3: Total: precision = 0.902 Junk: precision = 0.971, recall = 0.857 Legitimate: precision = 0.830, recall = 0.96

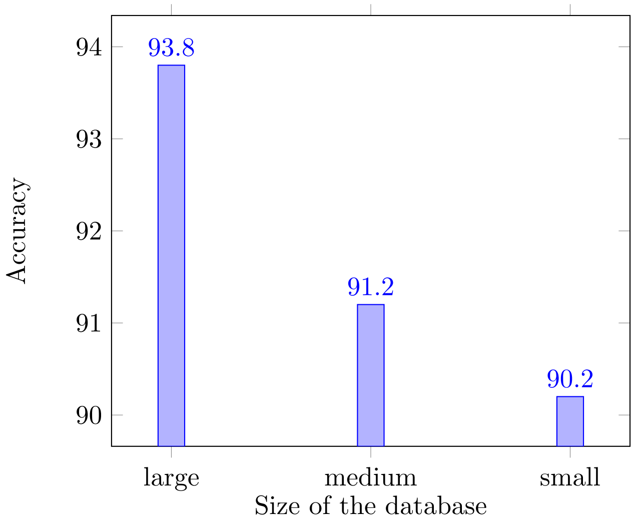

Figure 1: Comparison of the accuracy of the plain Naive Bayes classifications on different training datasets

2.5.2 Smoothing

Then, we tested the implemented algorithm on three training datasets using three different laplacian smoothing parameters.

Large Training Database

Spam as Ham	9
Ham as Spam	59
Spam as Spam	571
Ham as Ham	361

Table 4: Smoothing parameter - 0.00005. Total: precision = 0.932 Junk: precision = 0.906, recall = 0.984 Legitimate: precision = 0.976, recall = 0.86

Spam as Ham	6
Ham as Spam	61
Spam as Spam	574
Ham as Ham	361

Table 5: Smoothing parameter - 0.005. Total: precision = 0.933 Junk: precision = 0.904, recall = 0.990 Legitimate: precision = 0.984, recall = 0.85

Spam as Ham	4
Ham as Spam	67
Spam as Spam	576
Ham as Ham	359

Table 6: Smoothing parameter - 1. Total: precision = 0.929 Junk: precision = 0.896, recall = 0.993 Legitimate: precision = 0.989, recall = 0.84

Medium Training Database

Spam as Ham	37
Ham as Spam	48
Spam as Spam	543
Ham as Ham	372

Table 7: Smoothing parameter - 0.00005. Total: precision = 0.915 Junk: precision = 0.919, recall = 0.936 Legitimate: precision = 0.910, recall = 0.89

Spam as Ham	27
Ham as Spam	58
Spam as Spam	553
Ham as Ham	362

Table 8: Smoothing parameter - 0.005. Total: precision = 0.915 Junk: precision = 0.905, recall = 0.953 Legitimate: precision = 0.931, recall = 0.86

Spam as Ham	8
Ham as Spam	73
Spam as Spam	572
Ham as Ham	347

Table 9: Smoothing parameter - 1. Total: precision = 0.919 Junk: precision = 0.887, recall = 0.986 Legitimate: precision = 0.977, recall = 0.83

Small Training Database

Spam as Ham	59
Ham as Spam	29
Spam as Spam	521
Ham as Ham	391

Table 10: Smoothing parameter - 0.00005. Total: precision = 0.912 Junk: precision = 0.947, recall = 0.898 Legitimate: precision = 0.869, recall = 0.93

Spam as Ham	48
Ham as Spam	37
Spam as Spam	532
Ham as Ham	383

Table 11: Smoothing parameter - 0.005. Total: precision = 0.915 Junk: precision = 0.935, recall = 0.917 Legitimate: precision = 0.889, recall = 0.91

Spam as Ham	16
Ham as Spam	68
Spam as Spam	564
Ham as Ham	352

Table 12: Smoothing parameter - 1. Total: precision = 0.916 Junk: precision = 0.892, recall = 0.972 Legitimate: precision = 0.957, recall = 0.84

Fig. 1 shows the comparison of the accuracy of classifications emails from testing database, using classifier trained on databases with different sizes.

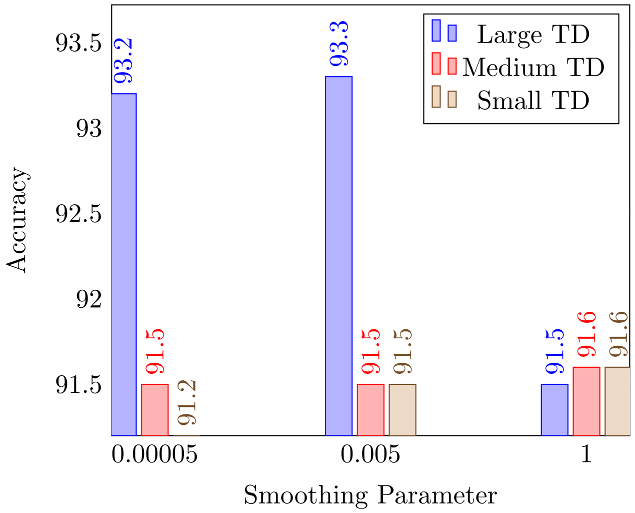

Figure 2: Comparison of the accuracy of the classifications on different training datasets with different smoothing parameters

3 Conclusion

The naive Bayes spam classification model based on the proposed algorithm in [1] has been implemented. The implementation has been tested on different training databases (small training database (490 ham messages, 683 spam messages), a medium training database (2150 ham messages, 2082 spam messages) and a large training database (3837 ham messages, 3876 spam messages)). The implementation has been also tested with different smoothing parameters. After the examination of several parameters, we noticed:

1. Accuracy is dramatically affected by the size of the training database;
2. The results are also affected by the smoothing parameter.

The best result for classifying emails was 93.8% of accuracy, which is comparable to the results in the paper [1].

References

[1] Sahami M., Dumais S., Heckerman D., Horvitz E., *A Bayesian Approach to Filtering Junk E-Mail.* AAAI Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin. AAAI Technical Report WS-98-05.