USING THE NAÏVE BAYES ALGORITHM FOR CLASSIFYING E-MAIL AS HAM OR SPAM

CMSC 191 - MACHINE LEARNING

Harold R. Mansilla
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
hrmansilla@up.edu.ph

Virgilio M. Mendoza III
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
vmmendoza1@up.edu.ph

March 5, 2019

ABSTRACT

The Naïve Bayes algorithm is a well-known classification algorithm based on the Bayes theorem. In this paper, four (4) classifiers were developed employing specific techniques in preprocessing and model building to classify e-mail contents as ham or spam. Model accuracy ranged between 33.93504% to 99.000171%. The classifier using the general vocabulary with Laplace smoothing posted the highest accuracy while the classifier with reduced vocabulary without Laplace smoothing had the lowest.

Keywords First keyword · Second keyword · More

1 Introduction

1.1 The Naïve Bayes Learning Algorithm

The Naïve Bayes learning algorithm is a simple technique for the creation of classifiers, models capable of assigning class labels, obtained from a finite data set, to data instances represented as a feature vector. The class label, C_i, given to the instance, X, is the class which has the highest probability given the probabilities of classes and the data of the instance. To compute for the said probability, the Bayes' Theorem is given as:

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)}$$

The theorem takes on the assumption that each attribute of a class is independent. This assumption simplifies computing for $P(X \mid C_i)$ since it can now be written as

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

where x_k is a component of X. This makes resulting classifier less computationally expensive when compared to the formula without the assumption. The classifier for a feature x_i can then be written as:

$$P(C \mid x_i) = \frac{P(x_i \mid C)\, P(C)}{\sum P(x_i \mid C) P(C)}$$

1.2 Laplace Smoothing

Laplace smoothing is a technique for smoothing categorical data [1]. To implement this technique, a smoothing parameter α is introduced to the classifier. The value is added such that:

$$P(x_i \mid C) = \frac{count(x_i \mid C) + \alpha}{\sum count(x_i|C) + \alpha|X|}$$

This prevents the denominator from reaching 0 in the extreme case where none of the words in training set appear in the test set.

2 The Dataset and Preprocessing

2.1 Dataset

The dataset, a collection of emails which are either spam emails or legitimate emails (ham emails), was retrieved from the 2007 TREC Public Spam Corpus.

2.2 Preprocessing

The python libraries `pandas` and `nltk` were used for preprocessing.

First, the `index` from the dataset was read to identify which emails were ham or spam. The emails' content were then read and saved to a csv file along with their label.

Next, the emails' contents were tokenized, removing punctuation marks and stop words in the process.

3 Bayesian Classifier Construction

The Naïve Bayes Classifier was made from scratch using Python. The implementation of Laplace Smoothing was done by adding in a value $alpha$ to the original Naïve Bayes formula. To obtain a reduced vocabulary of 200 words per each class, the researchers found the most discriminating words per class. That is, the most frequent words that are found in each class, exclusive to them, are kept in the vocabulary. In this study, the top 200 words per class are retained to construct the vocabulary the classifier will use.

An 80-20 train-test split was used to split the dataset. To obtain the accuracy of the model, the number of correctly classified test cases over the total number of test cases in the dataset was be used.

4 Results

The classifier resulted in the following:

4.1 General Vocabulary without Laplace Smoothing ($alpha = 0$)

Implementing the classifier using this setup garnered an accuracy of 79.52137% with a vocabulary of 226426 words. This low accuracy can be attributed to the division by 0 that may occur during the training of the model.

4.2 General Vocabulary with Laplace Smoothing

The results for this experiment is summarized in the table below.

$alpha$	$Accuracy$
1.00	98.69402%
0.95	98.70769%
0.90	98.72137%
0.85	98.72821%
0.80	98.74188%
0.75	98.76923%
0.70	98.77607%
...	...
0.05	99.00171%

Table 1: Accuracy with varying $alphas$ and a general vocabulary

Based on the table above, it can be seen that as $alpha$ tends closer to 0, the accuracy increases. This is because the higher the value used for smoothing, the farther it is from the actual value. Thus, a smaller $alpha$ yields a better accuracy.

4.3 Reduced Vocabulary of 200 words each without Laplace Smoothing ($alpha = 0$)

Similar to the first experiment, this experiment yielded a low accuracy of 33.93504%. Again, this can be attributed to the possibility of division by 0 occurring. Another factor is the reduced vocabulary. It is possible that the words retained fail properly discriminate between classes, leading to such a low accuracy.

4.4 Reduced Vocabulary of 200 words each with Laplace Smoothing

$alpha$	$Accuracy$
1.00	89.28547%
0.95	89.29231%
0.90	89.30598%
0.85	89.34017%
0.80	89.38803%
0.75	89.41538%
0.70	89.44274%
...	...
0.05	90.19487%

Table 2: Accuracy with varying $alphas$ and a reduced vocabulary

Just as the second experiment, the accuracy increases as $alpha$ decreases to 0. This, as is in the second experiment, is due to the value of $alpha$ altering the actual value used in classification.

References

[1] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.