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

February 25, 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 94.82% to 98.53%. The classifier using the general vocabulary posted the highest accuracy while the classifier with reduced vocabulary and 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

For building the classifiers, the Python machine learning library `scikit-learn` [2] will be used. Implementing the Naïve Bayes algorithm as well as Laplace smoothing is available in the class `sklearn.naive_bayes.MultinomialNB`. Multinomial Naïve Bayes is the selected implementation as it is recommended by [2] for text classification problems. For this paper, an 80/20 training/test split will be observed.

`scikit-learn` provides metrics for model evaluation. These are accuracy, precision, recall, f1-score, and, support.

The above metrics will be employed for evaluating the four models that will be constructed, listed and will be referred to as follows.

3.1 Classifier Using the General Vocabulary (`cgv`)

For this classifier, the Naïve Bayes model will be built on the pre-processed dataset without any alterations.

3.2 Classifier with Laplace Smoothing Using the General Vocabulary `cgv_l`

For this classifier, the Naïve Bayes model will be built on the pre-processed dataset with Laplace smoothing applied. This can be achieved by simply passing an `alpha` parameter to the `MultinomialNB` class.

3.3 Classifier Using the Reduced Vocabulary (`crv`)

For this classifier, the Naïve Bayes model will be built on the pre-processed training dataset with the words not in the listed vocabulary dropped.

3.4 Classifier with Laplace Smoothing Using the Reduced Vocabulary (`crv_l`)

For this classifier, the Naïve Bayes model will be built on the pre-processed training dataset with the words not in the listed vocabulary dropped and with Laplace smoothing applied.

4 Results

The classifier using the general vocabulary garnered the following results:

	precision	recall	f1-score	support
ham	0.96	0.99	0.98	4926
spam	1.00	0.98	0.99	9697

The accuracy of the classifier was 98.52971%

The classifier using the general vocabulary with Laplace smoothing where `alpha` =1 garnered the following results:

	precision	recall	f1-score	support
ham	0.98	0.97	0.97	4926
spam	0.99	0.99	0.99	9697

The accuracy of the classifier was 98.22198%

The classifier using the reduced vocabulary consisting of 200 words garnered the following results:

	precision	recall	f1-score	support
ham	0.96	0.88	0.92	4926
spam	0.94	0.98	0.96	9697

The accuracy of the classifier was 94.91896%

The classifier using the reduced vocabulary consisting of 200 words with Laplace smoothing where `alpha` =1 garnered the following results:

	precision	recall	f1-score	support
ham	0.97	0.88	0.92	4926
spam	0.94	0.98	0.96	9697

The accuracy of the classifier was 94.82322%

References

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

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.