

Daniel Choy
2/16/22

Spam or Not Spam Email Classification

Special notes before running program:

- This program was done in the python coding language
- The interpreter used is Python 3.9
- Make sure to have installed the package numpy using the following command:
pip3 install numpy
- Make sure the data folder and naive_bayes.py are in the same folder

Instructions to compile/run program:

Step 1 - Open a terminal window

Step 2 - Enter the following command: python3 naive_bayes.py

Screenshot of the expected output of my program:

```
(venv) (base) thevillage-100-64-9-213:Project II data and code-1 danielchoy17$ python3 naive_bayes.py
Train Accuracy: 0.9508267124758428
Test Accuracy: 0.936
```

Why my program works:

This program is a machine learning program that takes in an email (a string type) and returns either that it is "SPAM" or "HAM." We accomplish this by using the Naive Bayes classifier. Due to the fact that a particular email is either spam or ham, the probabilities

$$\mathbb{P}(\text{spam}|\{w_1, \dots, w_k\}) :$$

$$\mathbb{P}(\text{ham}|\{w_1, \dots, w_k\}) :$$

must sum to 1. Because they both sum to 1, we can just compute one of them and predict SPAM if $\mathbb{P}(\text{spam}|\{w_1, \dots, w_k\}) > 0.5$ and HAM otherwise. We can calculate $\mathbb{P}(\text{spam}|\{w_1, \dots, w_k\})$ using Bayes Theorem with the Law of Total Probability which gives us the following:

$$\mathbb{P}(\text{spam}|\{w_1, \dots, w_k\}) = \frac{\mathbb{P}(\text{spam}) \prod_{i=1}^k \mathbb{P}(w_i|\text{spam})}{\mathbb{P}(\text{spam}) \prod_{i=1}^k \mathbb{P}(w_i|\text{spam}) + \mathbb{P}(\text{ham}) \prod_{i=1}^k \mathbb{P}(w_i|\text{ham})}$$

Because of the fact that in an email there may be duplicate words, a lot of punctuation, and some words are capitalized and some not, this program cleans the email by reducing it into a set of lowercase words and nothing else.

The following is how you calculate all the probabilities needed to be able to get $P(\text{spam}|\{w_1, \dots, w_k\})$:

$P(\text{spam}) = \# \text{ of training spam emails} / \# \text{ of total training emails}$

$P(\text{ham}) = \# \text{ of training ham emails} / \# \text{ of total training emails}$

*To avoid zeros occurring when calculating the probability of certain words, we apply Laplace Smoothing to all these probabilities.

So instead of doing

$P(\text{word} | \text{spam}) = \# \text{ of training spam emails with word} / \# \text{ of training spam emails}$

We will do

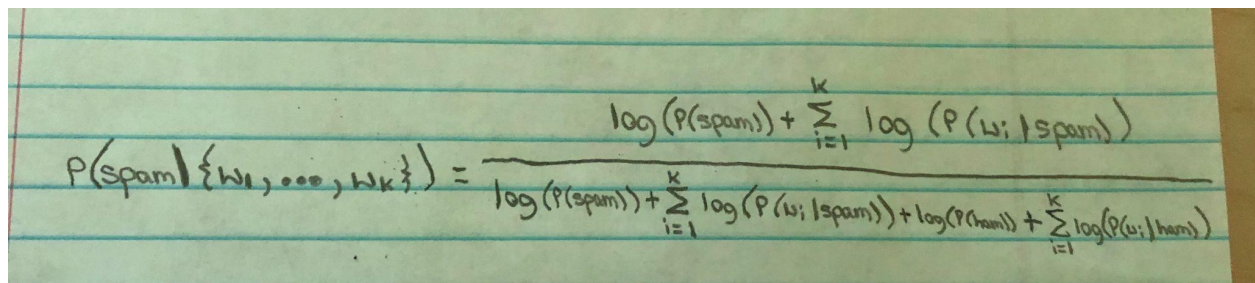
$P(\text{word} | \text{spam}) = (\# \text{ of training spam emails with word} + 1) / (\# \text{ of training spam emails} + 2)$

In order to prevent underflow from occurring while calculating $P(\text{spam}|\{w_1, \dots, w_k\})$, we will use a log trick.

Instead of doing

$$P(\text{spam}|\{w_1, \dots, w_k\}) = \frac{P(\text{spam}) \prod_{i=1}^k P(w_i|\text{spam})}{P(\text{spam}) \prod_{i=1}^k P(w_i|\text{spam}) + P(\text{ham}) \prod_{i=1}^k P(w_i|\text{ham})}$$

We will do



A photograph of a handwritten formula on lined paper. The formula is:

$$P(\text{spam}|\{w_1, \dots, w_k\}) = \frac{\log(P(\text{spam})) + \sum_{i=1}^k \log(P(w_i|\text{spam}))}{\log(P(\text{spam})) + \sum_{i=1}^k \log(P(w_i|\text{spam})) + \log(P(\text{ham})) + \sum_{i=1}^k \log(P(w_i|\text{ham}))}$$

Lastly, to test the accuracy of my program we do the following:

$$\text{accuracy} = \frac{\# \text{ of emails calssified correctly}}{\# \text{ of total emails}}$$