Naive Bayes Email Classifier

Weiting Zhan

November 9, 2018

We are trying to find the probability that email is spam given the text in the email. Bayes Theorem notation:

$$p(Hypothesis|Evidence) = \frac{p(Evidence|Hypothesis)p(Hypothesis)}{p(Evidence)} \tag{1}$$

In this email classification problem,

$$p(Spam|\omega_1, ..., \omega_n) = \frac{p(\omega_1, ..., \omega_n|Spam)p(Spam)}{p(\omega_1, ..., \omega_n)} \tag{2}$$

$$p(Ham|\omega_1, ..., \omega_n) = \frac{p(\omega_1, ..., \omega_n|Ham)p(Ham)}{p(\omega_1, ..., \omega_n)} \tag{3}$$

In order to simplify the math, we make Naive Bayes assumption that each word is independent of all other words.

By making the Naive Bayes, we can break down the numerator into the following.

$$p(Spam|\omega_1, ..., \omega_n) = \frac{p(\omega_1|Spam)p(\omega_2|Spam)...p(\omega_n|Spam)p(Spam)}{p(\omega_1, ..., \omega_n)} \tag{4}$$

$$p(Ham|\omega_1, ..., \omega_n) = \frac{p(\omega_1|Ham)p(\omega_2|Ham)...p(\omega_n|Ham)p(Ham)}{p(\omega_1, ..., \omega_n)} \tag{5}$$

Our decision rule is :

$$p(Spam|\omega_1, ..., \omega_n) \begin{cases} > p(Ham|\omega_1, ..., \omega_n) & \text{Spam,} \\ = p(Ham|\omega_1, ..., \omega_n) & \text{Unable to predict} \\ < p(Ham|\omega_1, ..., \omega_n) & \text{Ham} \end{cases} \tag{6}$$

We can get rid of the denominator since its only purpose is to scale the numerator.
So we get:

$$p(Spam|\omega_1, ..., \omega_n) \propto p(\omega_1|Spam)p(\omega_2|Spam)...p(\omega_n|Spam)p(Spam) \tag{7}$$

$$p(Ham|\omega_1, ..., \omega_n) \propto p(\omega_1|Ham)p(\omega_2|Ham)...p(\omega_n|Ham)p(Ham) \tag{8}$$

Since the value of probability is in between [0,1], multiply probability 100 times, for example, the calculation of $p(Spam|\omega_1, ..., \omega_n)$ and $p(Ham|\omega_1, ..., \omega_n)$,have the risk of to end up to zero. To prevent this, we use log probability. We have the property of logarithm:

$$\log(xy) = \log(x) + \log(y) \tag{9}$$

Apply (9) to equation (7) and (8) by taking the log of each side of the equation:

$$\log p(Spam|\omega_1, ..., \omega_n) \propto \log p(Spam) + \sum_{i=1}^{n} \log p(\omega_i|Spam) \tag{10}$$

$$\log p(Ham|\omega_1, ..., \omega_n) \propto \log p(Ham) + \sum_{i=1}^{n} \log p(\omega_i|Ham) \tag{11}$$

For the prior probability of spam $P(Spam)$ and ham $P(Ham)$ are calculated by counting how many messages are spam/ham, dividing by the total number of messages.

$$p(Spam) = \frac{N_{spam}}{N_{Spam} + N_{Ham}} \tag{12}$$

$$p(Ham) = \frac{N_{Ham}}{N_{Spam} + N_{Ham}} \tag{13}$$

N_{Spam}: the total number of Spam in the data set.
N_{Ham}: the total number of Spam in the data set.
Define the condition probability, $P(\omega_i|Spam)$ and $p(\omega_i|ham)$:

$$p(\omega_i|Spam) = \frac{T_{\omega_i+1}}{\sum_{\omega \in spam} T_{\omega_i+1}} \tag{14}$$

$$p(\omega_i|Ham) = \frac{T_{\omega_i+1}}{\sum_{\omega \in Ham} T_{\omega_i+1}} \tag{15}$$

T_{ω_i} : the number of occurrences of word i ω_i in the training data from class Spam.

Part (a) Decisions

Algorithm:

1. Compute Spam and Ham priors by using equation 12 and 13.

2. For each (email , label) pair, get rid of the punctuation, and stop words ,tokenize the document into words.

3. From the training data, for each word, either add it to the vocabulary for spam/ham dictionary , and add the word to the global dictionary.And count the frequency of the word. After iterate the training data, we get a Spam dictionary, Ham dictionary and global dictionary.

4.For the test data, apply Naive Bayes as described above. For example, given a document, we need to iterate each of the words and compute $\log p(\omega_i|Spam)$ and sum $\log p(\omega_i|Spam)$

all up. We also compute $\log p(\omega_i|Ham)$ and sum $\log p(\omega_i|Ham)$ all up. Then we add the log Spam priors and log Ham priors to get the score.

5 . use the decision rule to predict the label is Spam or Ham:

Our decision rule is :

$$p(Spam|\omega_1, ..., \omega_n) \begin{cases} > p(Ham|\omega_1, ..., \omega_n) & \text{Spam,} \\ = p(Ham|\omega_1, ..., \omega_n) & \text{Unable to predict} \\ < p(Ham|\omega_1, ..., \omega_n) & \text{Ham} \end{cases} \tag{16}$$

Part (b) Accuracy on the test set

$$Accuracy = \frac{TP + TN}{TP + FP + FN + FN} \tag{17}$$

TP: True positive,prediction is Spam and it's spam
TN: True negative, prediction is Spam and it's ham
FP: false positive,prediction is Ham and it's ham
FN: False Negative, prediction is Ham and it's Spam
Corpus size = 15525 emails;
Collected 15525 feature sets;
Training set size = 12420 emails;
Test set size = 3105 emails;
Accuracy on the training set = 0.7981481481481482;
Accuracy of the test set = 0.7716586151368761;

Part (c) Prior probabilities for Spam and Ham

For the prior probability of spam $P(Spam)$ and ham $P(Ham)$ are calculated by counting how many messages are spam/ham, dividing by the total number of messages.

$$p(Spam) = \frac{N_{spam}}{N_{Spam} + N_{Ham}} \tag{18}$$

$$p(Ham) = \frac{N_{Ham}}{N_{Spam} + N_{Ham}} \tag{19}$$

N_{Spam}: the total number of Spam in the data set.
N_{Ham}: the total number of Spam in the data set.

Part (d) Without and with Laplace smoothing

Without Laplace Smoothing: Since log of 0 is undefined. The code will have errors when running the algorithm. For example, if the world "Happy" never appeared in the spam of training data, then the

$$p(\omega_i|Ham) = 0 \tag{20}$$

which leads to:

$$\log p(\omega_i|Ham) = \log 0 \tag{21}$$

There will be error by running this way. ValueError: math domain error

With Laplace Smoothing: Define the condition probability, $P(\omega_i|Spam)$ and $p(\omega_i|ham)$: known as Laplace Smoothing, there will not be situation like $\log 0$:

$$p(\omega_i|Spam) = \frac{T_{\omega_i+1}}{\sum_{\omega \in spam} T_{\omega_i+1}} \tag{22}$$

$$p(\omega_i|Ham) = \frac{T_{\omega_i+1}}{\sum_{\omega \in Ham} T_{\omega_i+1}} \tag{23}$$

Part (e) the most discriminate words based on the learned probabilities

The discriminative words are those that have very different probabilities in the two classes: [scale = 0.4]1.png

So our discriminative words are:kaminski, shirley, hpl, vince, valium, melissa.