Prerequisites

Definition of conditional probability:

$$P(B|A) = P(A,B)/P(A)$$

Product rule:

$$P(A,B) = P(A|B)*P(B) = P(B|A)*P(A)$$

Chain rule with a condition:

$$P(x1,x2,\ldots xn|y) = P(w1|y)*P(w2|y,w1)*P(w3|y,w1,w2)*\ldots*P(wn|y,w1,\ldots,wn\text{-}1)$$

Spam: Unsolicited bulk emails. E.x.: emails from a retailer about their sales promotions or emails from a malicious sender that ask for sensitive information, such as SSN or credit card numbers (Phishing emails).

Naïve Bayesian Classifier for Spam Detection

Declarations:

We assume an email is a collection of words with all punctuation have been removed.

We use $X^{(i)}$ to denote the i^{th} email in the *training set*.

We say there is in total *m* emails in the *training set*.

Naïve Bayesian is a *supervised learning algorithm* also known as a *classifier*. For a training example in the *training set*, we have: $(X^{(i)}, y^{(i)})$ implying the email has a label $y^{(i)} \in \{0,1\}$,with $y^{(i)} = 0$ being a non-spam and $y^{(i)} = 1$ being a spam.

The labels are mutually exclusive, indicating if an email is a non-spam, it cannot be a spam.

Let $X^{(j)}$ denote the j^{th} email in the *testing set*.

Problem Definition:

Given an email, say $X^{(j)}$, the Naïve Bayesian classifier works as follows:

1. Computes $P(y^{(j)} = 0|X^{(j)})$ and $P(y^{(j)} = 1|X^{(j)})$
2. Compares $P(y^{(j)} = 0|X^{(j)})$ and $P(y^{(j)} = 1|X^{(j)})$, if $P(y^{(j)} = 0|X^{(j)}) > P(y^{(j)} = 1|X^{(j)})$ then it classifies $X^{(j)}$ as a non-spam; if $P(y^{(j)} = 0|X^{(j)}) < P(y^{(j)} = 1|X^{(j)})$ it classifies $X^{(j)}$ as a spam.

Probability Analysis:

According to the definition of *conditional probability*, we have:

$$P\left(y^{(j)}=0\middle|X^{(j)}\right)=\frac{P(y^{(j)}=0,X^{(j)})}{P(X^{(j)})}$$

$$P\left(y^{(j)}=1\middle|X^{(j)}\right)=\frac{P(y^{(j)}=1,X^{(j)})}{P(X^{(j)})}$$

Therefore, to compare both probabilities we just need to compute and compare the numerators, since their denominators are the same.

Applying the *product rule* to the numerators, we have:

$$P\left(y^{(j)}=0,X^{(j)}\right)=P\left(X^{(j)}\middle|y^{(j)}=0\right)*P(y^{(j)}=0)$$

$$P\left(y^{(j)}=1,X^{(j)}\right)=P\left(X^{(j)}\middle|y^{(j)}=1\right)*P(y^{(j)}=1)$$

Say that $X^{(j)}=\{w_1,w_2,\dots,w_k\}$ ($X^{(j)}$ consists of k words in total), where $k\geq 1$, we then have[1]:

$$P\left(X^{(j)}\middle|y^{(j)}=0\right)=P(w_1,w_2,\dots,w_k|y^{(j)}=0)$$

By applying the *chain rule*, we have:

$P\left(w_1,w_2,\dots,w_k\middle|y^{(j)}=0\right)$

$=P\left(w_1\middle|y^{(j)}=0\right)*P\left(w_2\middle|y^{(j)}=0,w_1\right)*\cdots*P(w_k|y^{(j)}=0,w_1,w_2,\dots w_{k-1})$

The **Naïve Bayesian assumption** claims that:

Knowing the existence of one word in a piece of information does not affect the probabilities of other words in that information[2].

By applying the Naïve Bayesian assumption, we have:

$$P\left(w_1,w_2,\dots,w_k\middle|y^{(j)}=0\right)=P\left(w_1\middle|y^{(j)}=0\right)*P\left(w_2\middle|y^{(j)}=0\right)*\cdots*P(w_k|y^{(j)}=0)$$

This is equivalent to assume all words are *conditionally independent*.

Now, to compute $P\left(X^{(j)}\middle|y^{(j)}=0\right)$, we will need to compute:
$P\left(w_1\middle|y^{(j)}=0\right),P\left(w_2\middle|y^{(j)}=0\right),\dots,P(w_k|y^{(j)}=0)$ and $P(y^{(j)}=0)$

---

[1] Since both of the spam and non-spam cases share the same analysis process, we will only focus on the non-spam case for simplicity.

[2] Despite it is a very strong assumption and we can easily find some cases that make this assumption false, it turns out that this assumption does make the Naïve Bayesian classifier work very well, especially in document classifications.

In the next two sections, we will introduce two different probabilistic models to solve the above problem.

Model No.1 (The Multi-variate Bernoulli Event Model)

Let w_l be the l^{th} word in $X^{(j)}$, where $l \in [1, k]$. We have:

$$P\left(w_l \middle| y^{(j)} = 0\right) = \frac{P(w_l, y^{(j)} = 0)}{P(y^{(j)} = 0)}$$

By applying the *Maximum Likelihood Estimation* (MLE) using the emails in the *training set*, we have:

$$P\left(w_l, y^{(j)} = 0\right) = \frac{\sum_{i=1}^{m} 1\{w_l = 1, y^{(i)} = 0\}}{m} \text{ and } P\left(y^{(j)} = 0\right) = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}{m}$$

Where $1\{a\}$ is a function that returns a binary value. If a is true, the function returns 1, 0 otherwise. By combining the above two equations, we have:

$$P\left(w_l \middle| y^{(j)} = 0\right) = \frac{\sum_{i=1}^{m} 1\{w_l = 1, y^{(i)} = 0\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

To elaborate the above equation, we say:

$$P\left(w_l \middle| y^{(j)} = 0\right) = \frac{\text{the number of emails in the training set, where the emails are non-spams and contain the word } w_l}{\text{the number of non-spam emails in the training set}}$$

Notice:

1. The Multi-variant Bernoulli event model only cares the appearance of a word in an email. It ignores how many times the word occurs.
2. If there exists a list of vocabulary: [a, aardvark, aardwolf, …, course, …, IT 340, …], one may think the model actually treats an email like a list of binary values, like [1,0,0, …, 1, …, 1, …] if the email is sent out by me.

The smooth technique:

The model will always work as long as all the words in $X^{(j)}$ have been seen from the *training set*. Now, let us assume the 20th word in $X^{(j)}$ does not appear in any of the training emails. Naturally, this will lead to: $\sum_{i=1}^{m} 1\{w_l = 1, y^{(i)} = 0\} = \sum_{i=1}^{m} 1\{w_l = 1, y^{(i)} = 1\} = 0$, which further leads to $P\left(w_l \middle| y^{(j)} = 0\right) = P\left(w_l \middle| y^{(j)} = 1\right) = 0$, which, in turn, leads to $P\left(X^{(j)} \middle| y^{(j)} = 1\right) = P\left(X^{(j)} \middle| y^{(j)} = 0\right) = 0$

Therefore, it is a bad decision to assume the probability for a word that has not yet been seen from the *training set* as zero. A better approach to handle this situation is to apply the *Laplace Smooth*:

$$P(w_l|y^{(j)}=0)=\frac{(\sum_{i=1}^{m}1\{w_l=1,y^{(i)}=0\})+1}{(\sum_{i=1}^{m}1\{y^{(i)}=0\})+2}$$

Model No.2 (The Multinomial Event Model)

Let n_i be the total number of words in $X^{(i)}$, we have:

$$P(w_l|y^{(j)}=0)=\frac{\sum_{i=1}^{m}(1\{y^{(i)}=0\}*\sum_{n=1}^{n_i}1\{w_l=1\})}{\sum_{i=1}^{m}1\{y^{(i)}=0\}*n_i}$$

$$P(w_l|y^{(j)}=1)=\frac{\sum_{i=1}^{m}(1\{y^{(i)}=1\}*\sum_{n=1}^{n_i}1\{w_l=1\})}{\sum_{i=1}^{m}1\{y^{(i)}=1\}*n_i}$$

To compute $P(w_l|y^{(j)}=0)$, the model basically checks through all the emails in the *training set* that are labelled as non-spam and computes the total number of times that w_l occurs in those emails. This number will be further divided by the total number of words of those emails.

After applying the *Laplace Smooth*, we have:

$$P(w_l|y^{(j)}=0)=\frac{(\sum_{i=1}^{m}(1\{y^{(i)}=0\}*\sum_{n=1}^{n_i}1\{w_l=1\}))+1}{(\sum_{i=1}^{m}1\{y^{(i)}=0\}*n_i)+V}$$

$$P(w_l|y^{(j)}=1)=\frac{(\sum_{i=1}^{m}(1\{y^{(i)}=1\}*\sum_{n=1}^{n_i}1\{w_l=1\}))+1}{(\sum_{i=1}^{m}1\{y^{(i)}=1\}*n_i)+V}$$

Where V is the number of words in the vocabulary.

Evaluations:

Every machine learning algorithm should be evaluated. Here are some criteria that are commonly applied for the evaluation process.

Positive/Negative: For a binary classification problem, such as spam detection, training and testing emails are always labelled as positive (spam) and negative (non-spam).

True Positive (TP): Data is labelled as positive and classified as positive.

True Negative (TN): Data is labelled as negative and classified as negative.

False Positive (FP): Data is labelled as negative but classified as positive.

False Negative (FN): Data is labelled as positive but classified as negative.

Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$, the number of successfully classified testing examples over the total testing examples.

Precision: $\frac{TP}{TP+FP}$; **Recall**: $\frac{TP}{TP+FN}$; **True Negative Rate**: $\frac{TN}{TN+FP}$; **F1 score**: $\frac{2*Precision*Recall}{Precision+Recall}$