**A Spam Filtering System with Machine Learning Algorithm using Bayes Theorem**

This week’s study focuses on how Machine learning help finding the legitimate email vs spam email. This is really big problem for end user to find the spam vs ham email. It is very difficult to separate the unwanted emails and delete every day. It kills our time. I found multiple algorithms to solve this problem. I will explain about the **Bayes Theorem**.

**Bayes Theorem-**

Bayes theorem is stated as the probability of the event **A** given **B** is equal to the probability of the event **B** given **A** multiplied by the probability of **A** upon the probability of **B**.

***P(A|B) = P(B|A) \* P(A)/P(B)***

One key condition the above equation must satisfy is that neither (A) nor (B) should be equal to zero. This condition thus must be met for all circumstances when the Bayes Theorem is applied.

Besides, the alternative form of Bayes Theorem is generally encountered when looking at two competing statements or hypotheses:

***P(A|B) = P(B|A) \* P(A)/ P(B|A) \* P(A) + P (B|A’)\*P(A’)***

**P(A)** is the corresponding probability of the initial degree of belief against **A**, where **P(A’) = 1 — P (A)**.

For some partition {**Ai** } of the sample space, the extended form of Bayes Theorem is:

**How Naive Bayes Algorithms works?-**

let’s take some trained datasets:

D1: “send us your password” **Spam**

D2: “send us your review” **Spam**

D3: “review your password” **Ham**

D4: “review us” **Ham**

D5: “send us password” **Spam**

D6: “send us your account” **Spam**

So, we have some data some are Spam and some are Ham(which is not Spam)

Here, the probability of Spam and Ham:

P(Spam) = 4/6

P(Ham) = 2/6

Here, the probability of every word in the Spam and Ham:

|  |  |  |
| --- | --- | --- |
| **Spam** | **Ham** | **Word** |
| 24/ | ½ | password |
| ¼ | 2/2 | review |
| ¾ | ½ | send |
| ¾ | ½ | us |
| ¾ | ½ | your |
| ¼ | 0/2 | account |

Let’s assume New email is “*review us****now***”(now is the new word). now, let’s check that the above email is Spam or Ham.

so, here we calculate the conditional probability :

P(“review us now” | Spam) = P(0,1,0,1,0,0) = (1-2/4)(1/4)(1-3/4)(3/4)(1-3/4)(1-1/4) = 9/2048

P(“review us now” | Ham) = P(0,1,0,1,0,0) = (1-1/2)(2/2)(1-1/2)(1/2)(1-1/2)(1-0/2) = 1/16

Now, we apply Bayes theorem:

here, you see the probability of email happening to be Spam is very less{**0.1229(approx.)**}.

So, the Algorithm predicts that the email {“review us now”} is **not Spam(Ham).**

**Advantages -**

● Very simple, and easy to use.

● Need less training data.

● Makes Probabilistic predictions.

● Handles continuous and discrete data.

● It is a generative model, i.e, it can make predictions even if some feature is missing by altering decision rules.

**Disadvantages -**

● A subtle issue with Naive-Bayes Classifier is that if you have no occurrences of a class label and a certain attribute value together then the frequency-based probability estimation will be zero.

● A big data set is required for making reliable predictions of the probability of each class. We can use this with small data sets but the precision will be altered.

● Attribute independence.

**Other Applications -**

● Multi-class Prediction

● Real-time Prediction

● Recommendation System