Assignment-3
AV489 Machine Learning for Signal Processing

Subrahmanya V Bhide (SC18B030)
Indian Institute of Space Science and Technology
Department of Aerospace Engineering
(Dated: 02 April 2021)

This document is a report based on the implementation of Naive Bayes algorithim for Spam Ham classification of SMS texts. A breif introduction is provided about the Naive Bayes algorithims and their implementation in the begininng, and at the end results from the implememtation are provided.

NAIVE BAYES CLASSIFIER

Naive Bayes classifier is not a single algorithim but a set of different algorithims, all based on the Bayes Rule. The Bayes rule can be stated as follows:

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Where the LHS represents the probability that, given $\mathbf{x}$, it belongs to the class C_k which is estimated using the knowledge of $P(C_k)$ i.e. probability of a given class and the probability of $\mathbf{x}$ occuring given that the sample is drawn from C_k i.e $P(\mathbf{x} \mid C_k)$.
The Classifier is reffered to as Naive due to the two following assumptions made by the classifier with respect to the features in the feature vector $\mathbf{x}$:

1. Independant, in the Naive bayes classifier we assume that the features are independant of each other.
2. Equal, the features exert an equal influence, i.e. none of the features are irrelevant and are assumed to be contributing equally to the outcome.

Since the denominator of the Bayes theorem expression p($\mathbf{x}$) is only dependant on the x_i we can consider it be a constant as a consequence of which we can write

$$p(C_k \mid \mathbf{x}) \propto p(\mathbf{x} \mid C_k)p(C_k) = p(\mathbf{x}, C_k)$$

$$p(\mathbf{x}, C_k) = p(x_1, x_2, \dots x_n, C_k) = p(x_1 \mid x_2, x_3 \dots, C_k) \times p(x_2 \mid x_3, \dots, C_k) \dots p(x_n \mid C_k) \times p(C_k)$$

Now using the independance assumption of the Naive bayes theorem we can write

$$p(C_k \mid \mathbf{x}) \propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Using this expression based on the Bayes classifier we choose the C_k having the maximum posterior probability.
Given the pre-labelled data we can estimate the $p(C_k)$ and using density estimation methods such as MLE we can obtain the likelihoods. Based on the parametric likelihoods chosen the Naive Bayes classifier is divided into different algorithims.

Gaussian Naive Bayes

This is used for continuous data where we assume that the attributes for each class vary according to a gaussian distribution which is given as:

$$p(x \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

The mean and variance can be found using methods such as maximum likelihood estimation or other methods. The Naive assymptions greatly simplify the problem because since we assume that the attributes are independant, the gaussian distribution which we deal would always remain 1 dimensional.

Multinomial Naive Bayes

While using the multinomail model we assume that the attribute x is some frequnecy of occurence of something. This model is used quite often in document classification where the number of times i.e. frequency of a word is used as an attribute for a class, (x_i). Therefore if $\mathbf{x} = (x_1, x_2, \ldots x_n)$ and $p = (p_1, p_2, \ldots p_n)$ i being an event such as an occurence of a word then we obatin an expression for probability as :

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

which is obtained using the naive assumptions made.
One poosible drawback of this method is that the probability depends on the frequency of occurence and if it is zero the probability is exactly zero. To avoid this we add a minimum probability for attributes, usually corresponding to a single word occurence, which is known as Laplace smoothing.

Bernoulli Naive Bayes

This is quite similar to the multinomail method but here we use the bernoulli variates which signify the occurence or absence of a certain term rather than its frequency. The likelihood is given by :

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Unlike the Multinomail Naive Bayes where non occurence of a word is not directly taken into account, but rather by some corrections, in the Bernoulli naive bayes it can take into account the absense of any attribute/word.

NAIVE BAYES CLASSIFIER FOR TEXT CLASSIFICATION

As discussed above the Gaussian Naive Bayes is more suited for continuous data of the attributes, for instance the IRIS floral dataset which consists of data about 3 categories of a flower based on metrics of the petal, sepal lengths and radius of the flower ..etc. Here the attributes can be considered to be continuous values and hence Gaussian Naive Bayes can be applied here to classifiy the flowers.
For Text classification, such as the given problem of Spam/Ham classification, the multinomial Naive Bayes is more suited. The methodology followed to implement the Multinomail Naive Bayes is charted out below:

1. For each of the class we make a set of all the words that occur in the messages and also keep count of the frequency of occurence of each word.
2. Using the total number of words that are present in the vocabulary of each class we convert the word freqencies to probabilities
3. To avoid the occurence of zero probabilities we use the Laplace smoothing as explained previously.
4. Thus we have the data regarding the prior and the class probability and we can use this to get the posterior of each class which can be used to assign a class for a new data.
5. As a preprocessing step we can either remove the words which are of very less significance such as 'the', 'and', 'or' . . . etc. Or else we can keep them as it is as if they are common they would be common for all the classes and hence not affect the classifiaction or if there is any bias in their occurences they may improve our classification.

RESULTS OBTAINED FOR THE SPAM-HAM CLASSIFICATION PROBLEM

A spam ham classifer for SMS text messages was implemented in the methodology explained in the previous section, using the frequency approach along with laplace smoothing. The confusion matrix obtained is shown in Tab. I.
The Performance measures for the classifier are tabulated in Tab. II
The histograms for Spam and Ham classes are shown as Figs. 1 & 2

TABLE I: Confusion matrix for the Spam Ham Classifier

Total N = 4108	Prediction of Ham	Prediction of Spam
Actual Ham	2895	659
Actual Spam	32	522

TABLE II: Performane Parameters for the Spam Ham Classifier

Performance Parameters	Values
Accuracy	0.83179
Precision	0.9890
Recall	0.81457
F1	0.89437

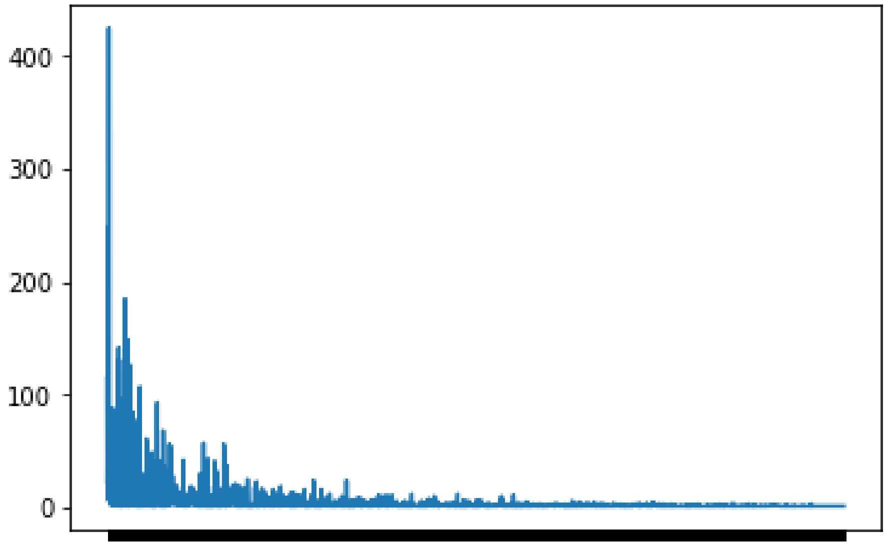

FIG. 1: Histogram for the Spam Class

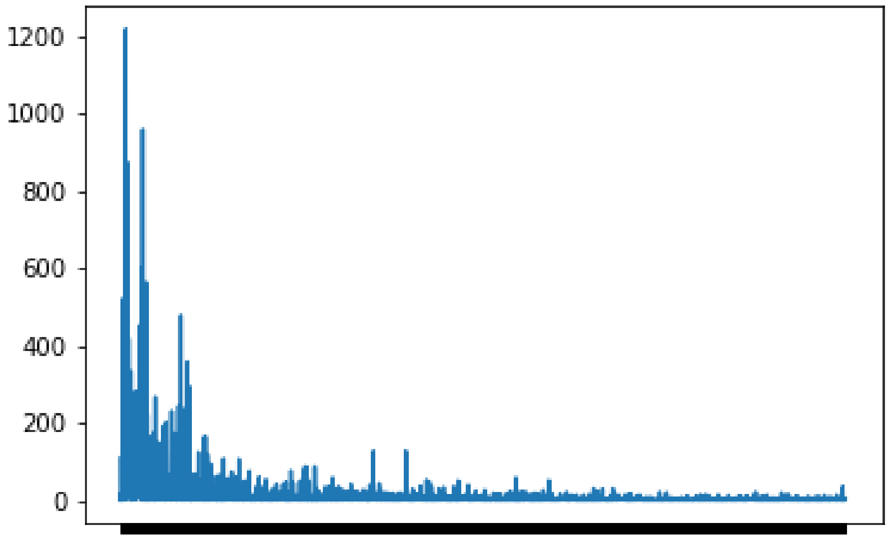

FIG. 2: Histogram for the Ham Class

APPENDIX

The words are not mentioned in the histogram directly but are tabulated in Tab. III which shows top 20 words based on their frequency of occurence.

TABLE III: Word Frequencies

Spam Word	Frquency	Ham Word	Frequency
to	425	i	1218
a	248	you	958
call	185	to	874
your	149	the	601
you	142	a	563
for	131	u	521
the	126	in	478
free	115	and	451
2	114	is	419
ur	107	my	402
have	96	me	334
is	93	of	295
u	89	for	283
txt	88	that	267
and	84	have	257
from	78	it	244
of	68	but	238
text	64	at	230
with	62	your	223
reply	61	on	222

Observing the word frequencies we can observe that Spam classes do not have certain words such as 'my', 'mine', 'me' ...etc, which are specific to Ham classes and vice versa.

[1] https://www.geeksforgeeks.org/naive-bayes-classifiers/
[2] https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[3] https://web.stanford.edu/~jurafsky/slp3/slides/7_NB.pdf
[4] https://github.com/shashank136/Spam-SMS-Classifier