4. PROPOSED MODEL

The spam classification algorithm we will be using is Support Vector Machine, popularly known as SVM.

4.1 INTRODUCTION TO SVM

SVM is one of the most popular algorithm used widely in industry as well as academia. It gives a cleaner and more powerful picture for solving nonlinear functions as compared to neural networks or logistic regression.
SVMs are based on the concept of decision boundaries, which separates a set of objects having different class memberships. The SVM algorithm finds an optimal boundary having maximum margin to separate the two classes, which requires solving the optimization objective given in next section.

4.2 OPTIMIZATION OBJECTIVE

The optimization objective is to find a solution for the following equation such that it minimizes 'Θ' :

$$\min_{\theta} C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

where,

C=Regularization parameter,
$cost_1(\Theta^T x^{(i)}) = -\log(h_\Theta x^{(i)})$,
$cost_0(\Theta^T x^{(i)}) = -\log(1-h_\Theta x^{(i)})$,
Θ=Parameter vector,
$h_\Theta x$=Hypothesis and

$h_\Theta x = 1$, if $\Theta^T x >= 0$ and
$= 0$, otherwise

We have used **SMO** or the Sequential Minimal Optimizatin algorithm, which is an approximate of SVM software package- *libsvm* to solve for the parameter Θ.

4.3 DECISION BOUNDARY

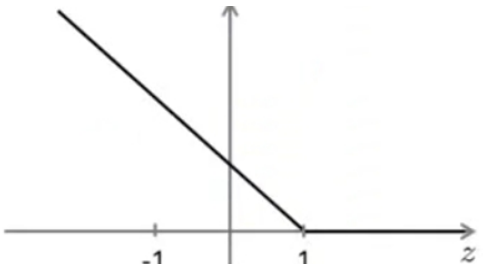


Fig 4.1 : $cost_1 (\Theta^T x^{(i)})$

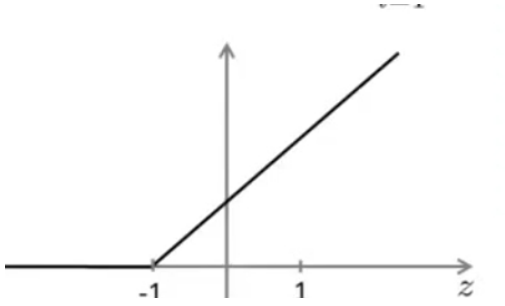


Fig 4.2 : $cost_0 (\Theta^T x^{(i)})$

As we can see from Fig 4.1 and Fig 4.2 ,

Whenever we want $y^{(i)} = 1$, $(\Theta^T x^{(i)}) > 1$
and if we want $y^{(i)} = 0$, $(\Theta^T x^{(i)}) < -1$

So, by above calculations, we get a decision boundary which is a large margin classifier between the different classes of datasets. The decision boundary can be linear as well as non-linear.

4.4 SPAM CLASSIFICATION ALGORITHM

KERNEL
We have used linear kernel , or no kernel in our implementation of SVM.

REGULARIZATION PARAMETER
We have chosen C=0.1 after experimenting with many other C values.

VOCABULARY LIST
We have created our vocabulary list by choosing all words which

occur at least a 100 times in the spam corpus, resulting in a list of 1899 words. This list of words will be used to judge whether an email contains spam words or no.

LANGUAGE/SOFTWARE USED

We have used Octave to code our spam classification algorithm.

THE ALGORITHM :

1. Preprocess or normalize each email

While many emails would contain similar types of entities (e.g.,numbers, other URLs, or other email addresses), the specific entities (e.g.,the specific URL or specific dollar amount) will be different in almost every email. Therefore, one method often employed in processing emails is to “normalize” these values, so that all URLs are treated the same, all numbers are treated the same, etc.

1.1 Convert entire email to lowercase
1.2 Strip all HTML tags
1.3 Normalize URLs
1.4 Normalize email addresses
1.5 Normalize numbers
1.6 Remove non-alphanumeric characters

2. Create word index for each email

Given the vocabulary list, we now map each word in the preprocessed emails into a list of word indices that contains the index of the word in the vocabulary list.

3. Construct feature vector

We now implement the feature extraction that converts each email into a vector in R^n, where n=number of words in vocabulary list. Specifically, the feature $x^i \in \{0, 1\}$ for an email corresponds to whether the i-th word in the dictionary occurs in the email. That is, $x^i = 1$ if the i-th word is in the email and $x^i = 0$ if the i-th word is

not present in the email.

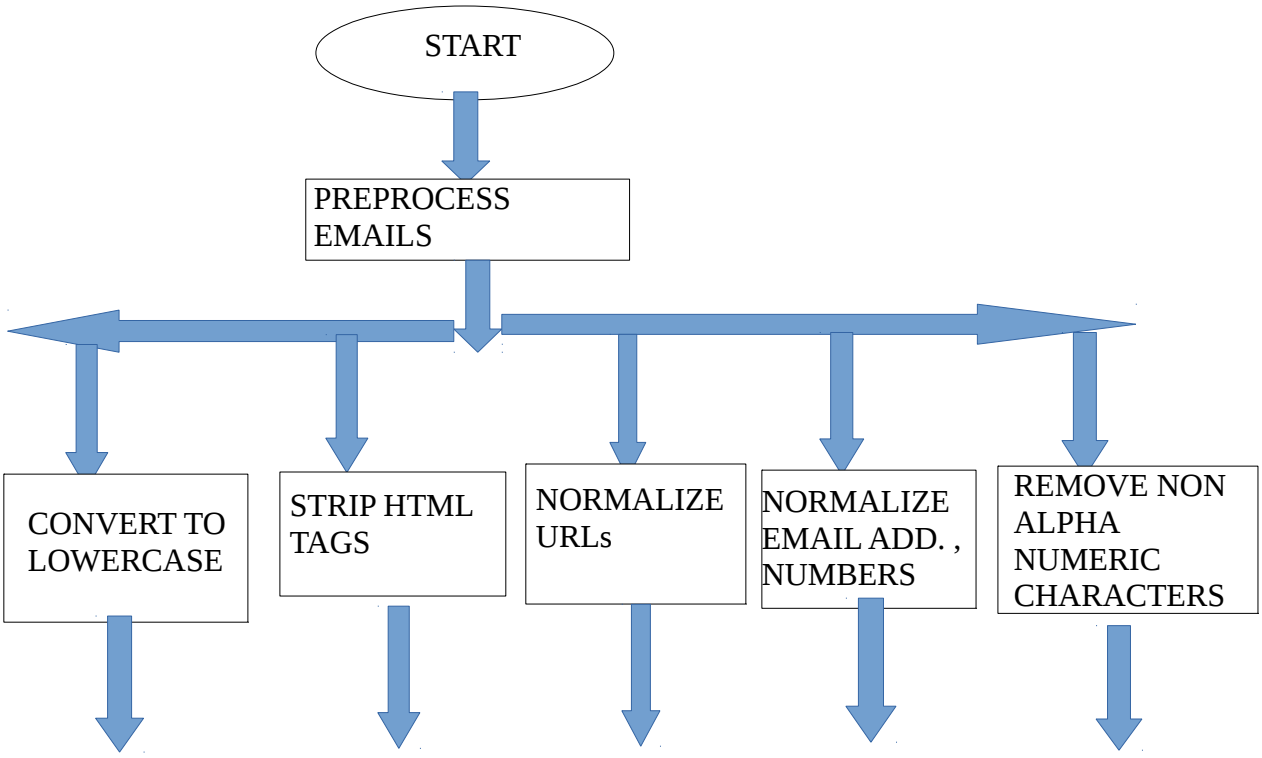
4. Train SVM for spam classification algorithm

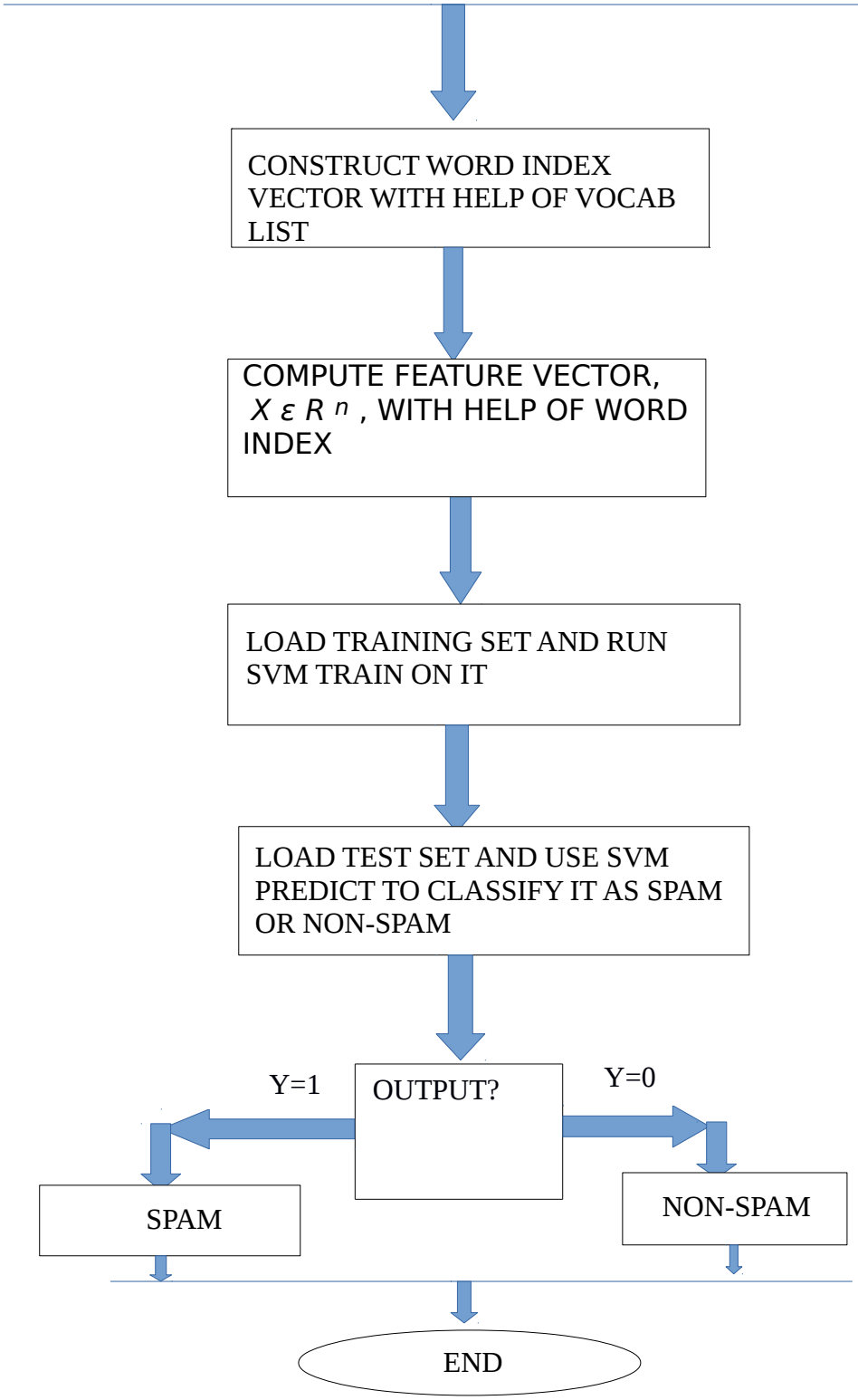
We have divided our dataset into training set and test set. The training set contains 4000 examples of emails while the test set contains 1000 examples.Each original email was pre-processed using the processEmail and emailFeatures functions and converted into a vector
$x^{(i)} \in R^{1899}$.
We run our SMO algorithm on our training set. Then we make observations on our test set. We calculate mean (accuracy) for both.

4.5 RESULT

Training accuracy/mean= 99.87%
Test accuracy/ mean = 98.5%

4.6 FLOWCHART




CONSTRUCT WORD INDEX VECTOR WITH HELP OF VOCAB LIST
COMPUTE FEATURE VECTOR, $X \varepsilon R^n$, WITH HELP OF WORD INDEX
LOAD TRAINING SET AND RUN SVM TRAIN ON IT
LOAD TEST SET AND USE SVM PREDICT TO CLASSIFY IT AS SPAM OR NON-SPAM
OUTPUT?
Y=1
Y=0
SPAM
NON-SPAM
END

5.CONCLUSION AND FUTURE SCOPE

The main purpose of this study was to show how can we construct a simple,yet efficient spm classifier using the supervised algorithm in Machine Learning- Support Vector Machine , which can classify a given email as spam (y=1) or non-spam (y=0) .

We have used a small dataset and a vocabulary list of 1899 words. In practice the vocabulary list is of atleast 4000-5000 words.
We have obtained an accuracy of 98.5%, which means 98 out of 100 emails can be correctly classified. This accuracy can be further improved using better SVM training algorithms and software packages.

In general, no algorithm can be implemented to obtain 100% accuracy. Due to small spam corpus, there is unknown word problem and many spam emails cannot be correctly classified. In future, a bigger spam corpus
can be used.

Also, in future we can use a combination of spam filtering algorithms like the Bayesian spam filter and the SVM filter, which can further increase the accuracy.

6.REFERENCES

1.http://spamassassin.apache.org/publiccorpus/
2.https://class.coursera.org/ml-006
3.