International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 9, September - 2013

Behavior Analysis of SVM Based Spam Filtering Using Various Kernel Functions and Data Representations

Author :Sushama Chouhan
Author Affiliation: MTech Scholar (MIST BHOPAL)

Abstract

In today's world Internet has become very popular, and the concept of electronic mail has made it easy and cheap to communicate with many people. But, many undesired mails are also received by users. And the higher percentage of these emails is termed as spam. The goal of Spam Classification is to distinguish between spam and legitimate mail message. SVM are the classifiers giving top notch performance in classification. This paper studies the performance of SVM classifier by varying spam ham ratio in Training and Testing data set. This paper also evaluates change in performance by using different representation for the document vector like term frequency (TF), Binary, inverse document frequency (IDF) and TF-IDF. As Kernel Function Plays Important role in classification this paper also evaluate SVM Classifier using different kernel function.

Index Terms—- spam , classification , SVM , kernel function.

Introduction

Internet has become an indispensable method to communicate with each other, because of its popularization, low cost, and fast delivery of message. Along with the growth of Internet and email there has been a dramatic growth in spam in recent years. Spam can originate from any location across the globe where internet access is available. Spamming is the abuse of electronic messaging systems to send unsolicited bulk messages or to promote products or services, which are almost universally undesired. The Problem of Spam is currently of serious and escalating concern, and it is challenging to develop spam filters that can effectively eliminate the increasing volumes of unwanted mails automatically before they enter a user's mailbox. along with the growth of Internet and email there has been a dramatic growth in spam in recent years.

In order to address the spam email issue, a significant research on anti-spam techniques has been taken place and various kinds of anti-spam software have been developed and used by email users. Spam filter techniques include both manual and automatic methods. In manual methods, negative lists of spammers, list of authentic senders, and selected list of words in email content or subject are considered for developing anti-spam filter. In recent years, machine learning technique, a better technique compare to manual methods, is used to detect and classify spam emails automatically [1].

In traditional programming computer takes data and program as inputs and accordingly provides the output. On the other hand, in case of machine learning, computer takes data and output as input and then provides the program as outcome. In simple terms, these algorithms get adapted and enable the computer "learn" from given observations. The algorithms extract useful knowledge automatically from the data [2,3, 4].

SVM

Support vector machines (SVMs) [5,6,7] are a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Support Vector Machine (SVM) has been recently proposed by Dr.V.vapnik [8] as an effective statistical learning method for pattern recognition [9]. The SVM based on statistical learning theory has many advantages. SVM has

demonstrated higher generalization capabilities in high dimensional space and spare samples. Its essence is to map optimal separating hyper plane that can correctly classify all samples.

SVM has proved to be one of the most efficient kernel methods. The success of SVM [10] is mainly due to its high generalization ability. Unlike many learning algorithms, SVM leads to good performances without the need to incorporate prior information. Moreover, the use of positive definite kernel in the SVM can be interpreted as an embedding of the input space into a high dimensional feature space where the classification is carried out without using explicitly this feature space.

SVM can be used to solve Linearly Separable as well as Non Linear Separable Problems.

A. Linear Separable Problems

While Classification if we can separate the classes using Linear Decision Boundary then it is Linear Separable Problem [11]. The Main task is to find the appropriate Decision Boundary. There may exist many decision boundary but we have to choose the best one. The decision boundary [12] should be as far away from the data of both classes as possible. We should maximize the margin, *m.*

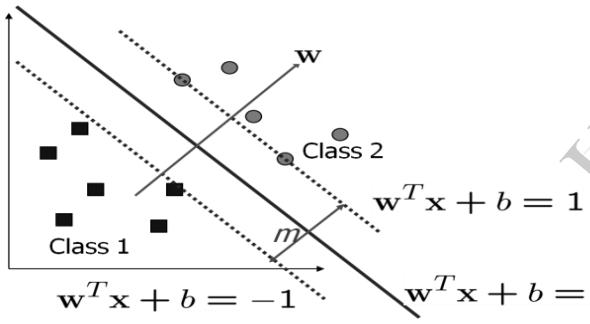


Figure 2.1 Linear Separable Problems

- Let $\{x_1, ..., x_n\}$ be our data set and let $y_i \hat{I}\{1,-1\}$ be the class label of x_i
- The decision boundary should classify all points correctly
  $Y_i (w^T x_i + b) \geq 1, \quad \forall_i$
- The decision boundary can be found by solving the following constrained optimization problem
  Minimize ½ $||w||^2$
  Subject to $Y_i (w^T x_i + b) \geq 1, \quad \forall_i$

This Optimization Problem can be solved by using Lagrange's Dual Problem [13].

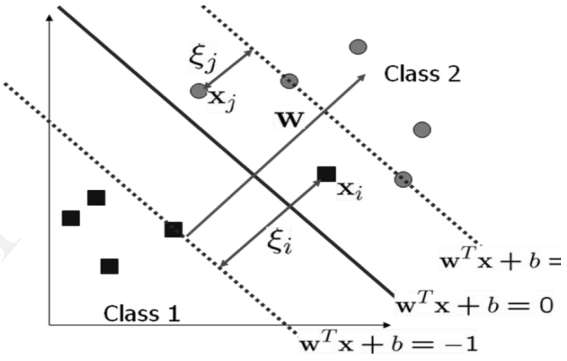
B. Non-Linearly Separable Problems

The Problem that cannot classify linearly is Non Linear Separable Problems [13]. Here we allow error $£_i$, if error is in between $0 <= £_i <= 1$, it can be properly classified but if $£_i > 1$ it is misclassified. Thus we should minimize error. So we minimize$\sum £_i$, $£_i$ can be computed by

$w^T x_i + b \geq 1 - £_i \quad Y_i = 1$
$w^T x_i + b \leq -1 + £_i \quad Y_i = -1$
$£_i \geq 0$

Thus the Optimization Problem [14] Becomes
Minimize ½ $||w||^2 + C \sum^n_{i=1} £_i$
Subject to $Y_i (w^T x_i + b) \geq 1 - £_i \quad £_i \geq 0$

Figure 2.2 Non Linear Separable Problems



Now to Obtain Non Linear Decision Boundary we transform the input space to feature space.

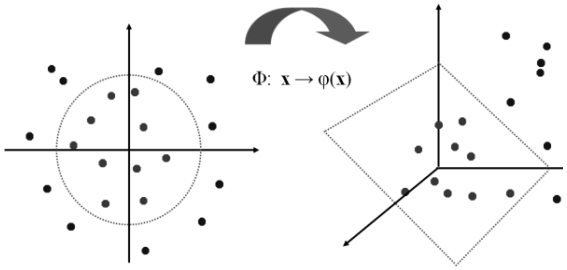


Figure 2.3 mapping from input space to feature space

To Obtain Non Linear Decision Boundary key idea is to transform xi to a higher dimensional space called feature space to make it easier.

Input space: the space the point xi is located
Feature space: the space of f (xi) after transformation

Transformation is needed as linear operation in the feature space is equivalent to non-linear operation in input space and classification can become easier with a proper transformation. In the XOR problem, for example, adding a new feature of x1x2 make the problem linearly separable.

Experiment Setup

3.1 Experiment Design

Different Variants of Enron Dataset is used by varying spam: ham ratio. To conduct experiments Support Vector Machine is used which contains basic four Kernel Functions. SVM Light tool is used for the experiment.

3.2 Data used

a The bench mark datasets used here for spam filtering are ENRON dataset[15]. Each dataset has two folders containing spam and ham messages. The various datasets have different spam: ham ratios given below.

Table 3.1 Enron Data Set folders

Folder Name	Spam	Ham	Total	Spam:Ham ratio
Enron 1	1500	3672	5975	1:03
Enron 2	1496	4361	5857	1:03
Enron 3	1500	4012	5512	1:03
Enron 4	4500	1500	6000	03:1
Enron 5	3675	1500	5175	03:1
Enron 6	4500	1500	6000	03:1
Total	**17171**	**16545**	**34519**	

These bench mark datasets are available in raw format. We have processed the dataset to obtained appropriate form.

We used Enron 1, 2 and 3 to create training data set and Enron 4, 5 and 6 to create test data set from unprocessed data so training data is different from test data.

3.2.1 Creating Training Dataset

In this paper create training data set (In raw form)Train 1:1, 1:1 is ham ratio with respect to spam.

Table 3.2 Train 1:1 Dataset Composition

Datasets	Training		Spam: Ham ratio	Total
	Spam	Ham		
Train 1:1	4496	4361	1:1	8857

3.2.1 Creating Test Dataset:

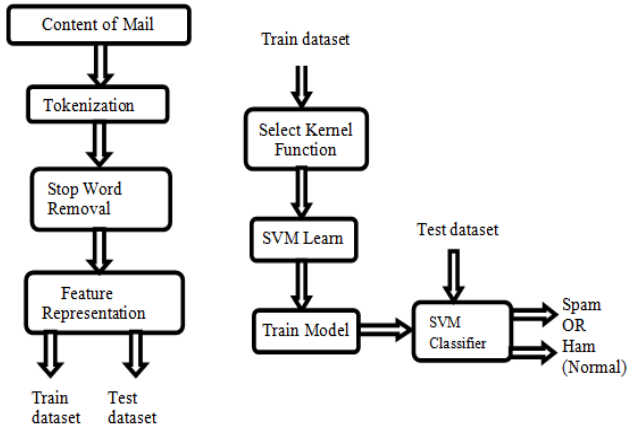
Create three test data set(In raw form) namely Test 1:1, Test 2:1 and Test 3:1 with changing spam ratio.

Table 3.3 Test 1:1, Test 2:1 and Test 3:1 Dataset Composition

Datasets	Training		Spam: Ham ratio	Total
	Spam	Ham		
Test 1:1	4500	4500	1:1	9000
Test 2:1	3675	1500	2:1	5175
Test 3:1	4500	1500	3:1	6000

3.3 Proposed model of spam classification

Following diagram illustrates the approach used for spam classification



3.4 Data Set preprocessing:

This section we shows how to document (mail) convert into vector form and what is possible weight of vector.

3.4.1 Steps for data preprocessing:

Following steps used in preprocessing of data set

Step 1 Tokenization: Read all word from the every message (For this task write a program in c).

Step 2 Removing stop word: Removing stop word is a method of simplifying the task of text classification. A stop list [16] is a collection of words that are not used in feature selection. A stop list may include words such as 'a', 'as', 'the', 'for', etc. that are not useful in classification because of their high appearance frequency in all documents.

Step 3: Feature Representation: Use one of the feature representation technique.

3.4.2 Feature Representation:

A feature is a word that present in document. Any word in document is called feature if it is satisfies some predefine constraint. Every mail represent by vector. There is various ways to represent weight of a vector[17].

Term Frequency (TF): Term frequency tf_{ij} is the number of occurrences of term ***tj*** in document ***Di***
Note: Different author and research paper used different definition of **TF** some of given below

$$f(tf_{ij}) = tf_{ij}$$
$$f(tf_{ij}) = tf_{ij} / l(Di)$$

Where ***l(Di)*** is the length of document ***Di*** ,means total number of term occurrences in document ***Di***

$$f(tf_{ij}) = \sqrt{tf_{ij}}$$
$$f(tf_{ij}) = 1+\log(tf_{ij})$$

We can say that tern frequency refers as a local and I am using TF using

$$f(tf_{ij}) = tf_{ij}$$

Binary: Binary representation which indicates whether a particular term ***tj*** occurs in a particular document or not. In this representation weight of term ***tj*** define as

$$Wij=1 \text{ if } tj \in Di$$
$$Otherwise\ Wij=0$$

Document Frequency (DF): Document Frequency df_j is the number of documents in the collection (*Di* where $1 \le i \le n$) that term **Tj** occurs in. Document Frequency refers as global. In DF we consider only term occurs or not ignore whatever value of **Wij** hold.

Inverse Document Frequency (IDF): Inverse Document Frequency idf_j calculate as follow

$$idfj=\log(N/df_j)$$

N: Total number of document

Term frequency–Inverse document frequency (TF-IDF): Term frequency multiply by inverse document frequency is called **TF-IDF.**

$$(tf\text{-}idf)_{ij} = tf_{ij} * idf_j$$

Performance Measure

In order to measure the performance of supervised machine learning (Classification)methods various performance measures are used such as recall, precision, false positive rate and accuracy etc. These measures can easily be derived from the confusion matrix of the model[18].

Table 4.1 Confusion Matrix for Spam and Ham class

- **True positive (TP)**: Correct classifications, spam documents (positive class) classified as spam (positive class)
- **True negative (TN):** Correct classifications, ham documents (negative class) classified as ham (negative class)
- **False positive (FP):** Incorrect classification, FP occurs when the outcome is incorrectly predicted as spam (or positive) when it is actually ham (negative).
- **False negative (FN):** Incorrect classification, FN occurs when the outcome is incorrectly predicted as ham (or negative) when it is actually spam (positive).
- **Accuracy (AC):** accuracy is ratio of correct classification and total number of predictions

$$\text{Accuracy} = \frac{\mathbf{TN + TP}}{\mathbf{TN + FP + FN + TP}}$$

Precision

Precision for a class is the ratio of true class (same class in actual belong to same class in prediction) and total number of item belong for that class in prediction. In other word we can say precision is accuracy of our classification for this class[19].

$$\text{Precision for spam documents} = \frac{\mathbf{TP}}{\mathbf{FP + TP}}$$

$$\text{Precision for ham documents} = \frac{\mathbf{TN}}{\mathbf{FN + TN}}$$

Recall:

Recall for a class is the ratio of true class (same class in actual belong to same class in prediction) and total number of item belong for this class in actual. In other word recall is completeness our classification for this class[19].

$$\text{Recall for spam documents} = \frac{\mathbf{TP}}{\mathbf{FN + TP}}$$

$$\text{Recall for ham documents} = \frac{\mathbf{TN}}{\mathbf{FP + TN}}$$

		predicted class	
		ham (-1)	spam (+1)
Actual Class	ham (-1)	TN	FP
	Spam +1)	FN	TP

EXPERIMENTAL RESULTS

As discussed above in all experiment use train data Train 1:1. There are three test data namely Test 1:1,Test 2:1 and Test 3:1. In all result feature is represent in **Binary, Term frequency, Inverse document frequency, Term frequency-Inverse document frequency** and apply different kernel function.

Table 5.1 Result with Binary representation

Train	Kernel Fun.		Test 1-1	Test 2-1	Test 3-1
			Spam	Spam	Spam
Train 1-1	Linear	Prec	75.17	94.23	82.83
		Recall	99.62	100.0	99.62
		Accu	83.36	95.65	84.23
	Polynomial d=2	Prec	65.33	88.22	78.89
		Recall	99.56	99.86	99.56
		Accu	73.37	90.43	79.68
	Radial Basis	Prec	93.43	99.10	98.70
		Recall	23.71	53.88	23.71
		Accu	61.02	66.9	42.55
	Sigmoid	Prec	59.61	84.03	72.83
		Recall	55.76	56.00	55.76
		Accu	58.99	61.2	51.22

Table 5.2 Result with Term Frequency

Train	Kernel Fun.		Test 1-1	Test 2-1	Test 3-1
			Spam	Spam	Spam
Train 1-1	Linear	Prec	73.69	91.97	84.51
		Recall	99.64	99.76	99.64
		Accu	82.03	93.64	86.03
	Polynomial d=2	Prec	60.31	82.46	78.88
		Recall	99.60	99.65	99.60
		Accu	67.03	84.7	79.7
	Radial Basis	Prec	94.91	99.39	98.86
		Recall	21.13	48.82	21.13
		Accu	60	63.44	40.67
	Sigmoid	Prec	58.59	74.97	86.07
		Recall	87.22	85.99	87.22
		Accu	62.79	69.66	79.83

Table 5.3 Result with Inverse Document Frequency

Train	Kernel Fun.		Test 1-1	Test 2-1	Test 3-1
			Spam	Spam	Spam
Train 1-1	Linear	Prec	76.70	95.79	83.41
		Recall	99.56	99.76	99.56
		Accu	84.66	96.71	84.82
	Polynomial d=2	Prec	62.89	85.74	78.33
		Recall	99.31	99.84	99.31
		Accu	70.36	88.1	78.88
	Radial Basis	Prec	50.38	71.58	75.03
		Recall	100	100	100
		Accu	50.7	71.81	75.03
	Sigmoid	Prec	50.07	71.11	75.01
		Recall	100	99.97	100
		Accu	50.13	71.13	75.02

Table 5.4 Result with TF-IDF

Train	Kernel Fun.		Test 1-1	Test 2-1	Test 3-1
			Spam	Spam	Spam
Train 1-1	Linear	Prec	74.34	93.66	83.83
		Recall	99.64	99.76	99.64
		Accu	82.62	95.03	85.32
	Polynomial d=2	Prec	57.57	80.28	77.01
		Recall	99.58	99.59	99.58
		Accu	63.09	82.34	77.38
	Radial Basis	Prec	50.36	71.55	75.03
		Recall	100	100	100
		Accu	50.72	71.77	75.03
	Sigmoid	Prec	60.22	88.54	72.58
		Recall	57	63.92	57.60
		Accu	59.78	68.5	51.88

5.1 Chart on the basis of test dataset

5.1.1 Accuracy

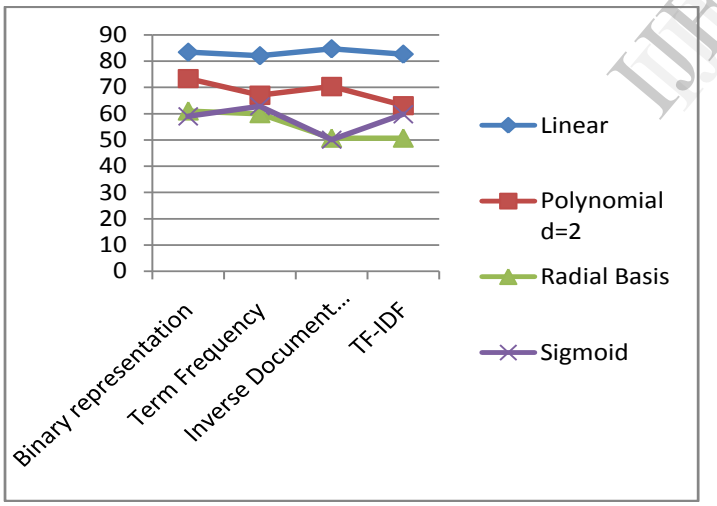


Figure 5.1 Accuracy on Test dataset Test 1:1

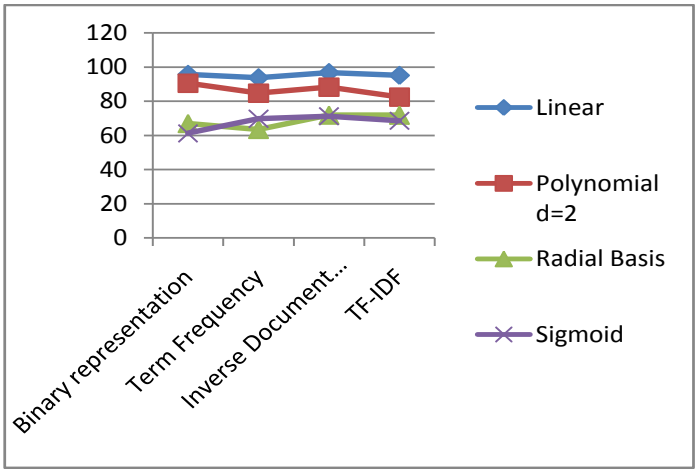


Figure 5.2 Accuracy on Test dataset Test 2:1

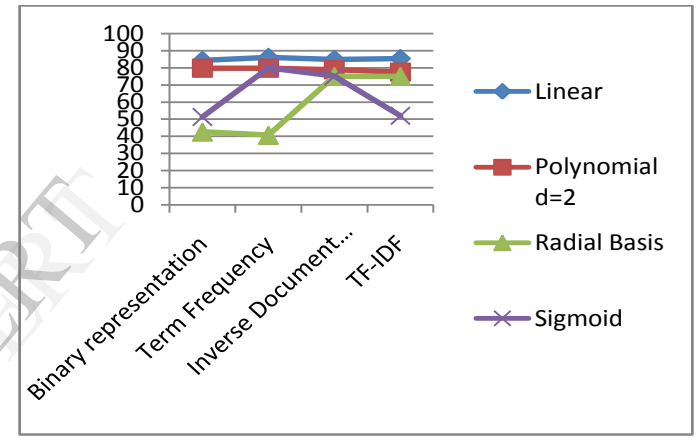


Figure 5.3 Accuracy on Test data Test 3:1

5.1.2 Precision Chart

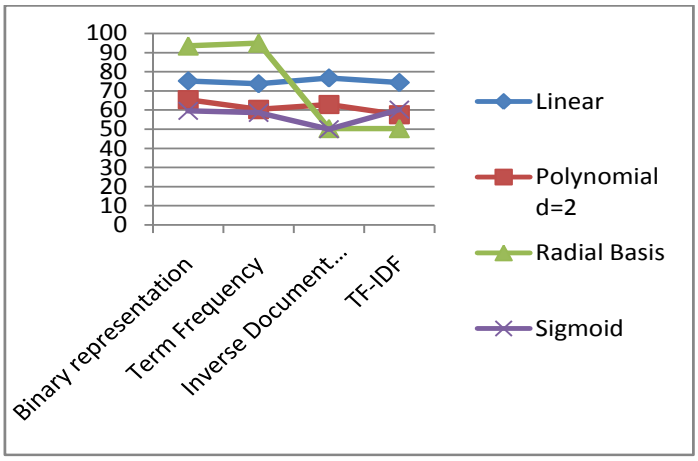


Figure 5.4 Precision of Spam on Test dataset Test 1:1

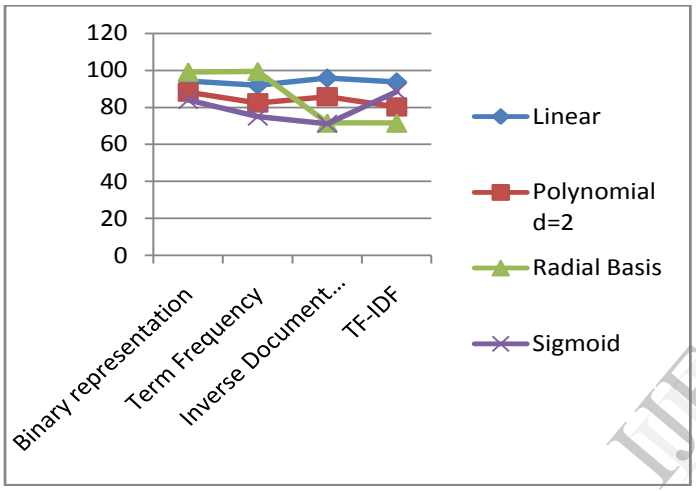


Figure 5.5 Precision of Spam on Test dataset Test 2:1

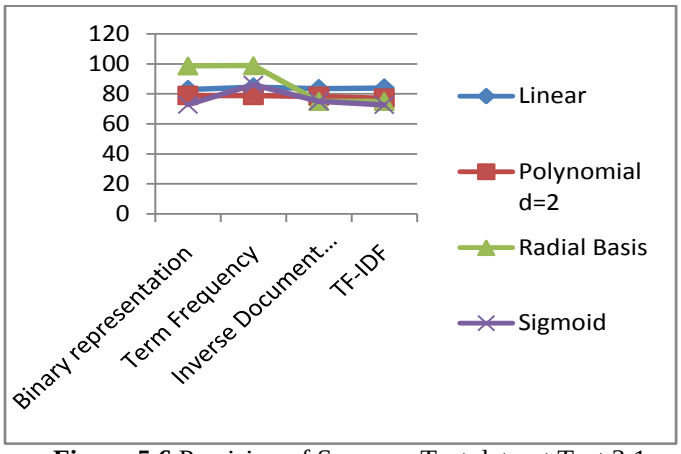


Figure 5.6 Precision of Spam on Test dataset Test 3:1

5.1.3 Recall Chart

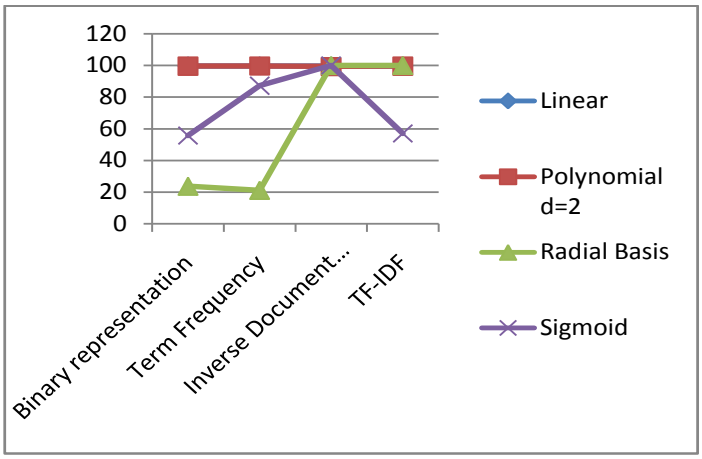


Figure 5.7 Recall of Spam on Test dataset Test 1:1

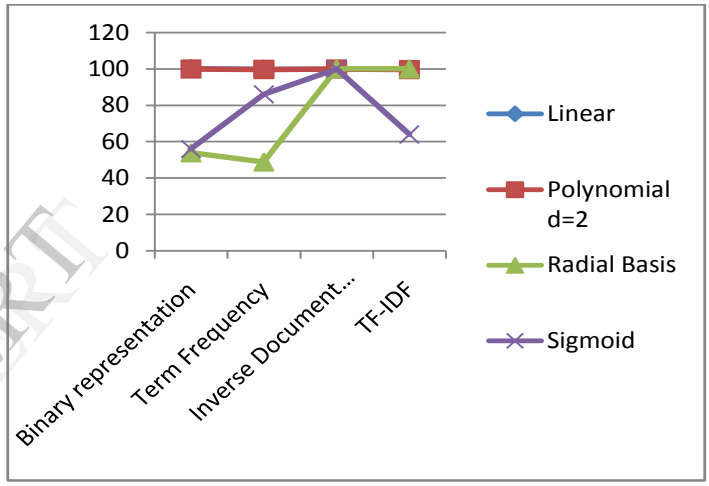


Figure 5.8 Recall of Spam on Test dataset Test 2:1

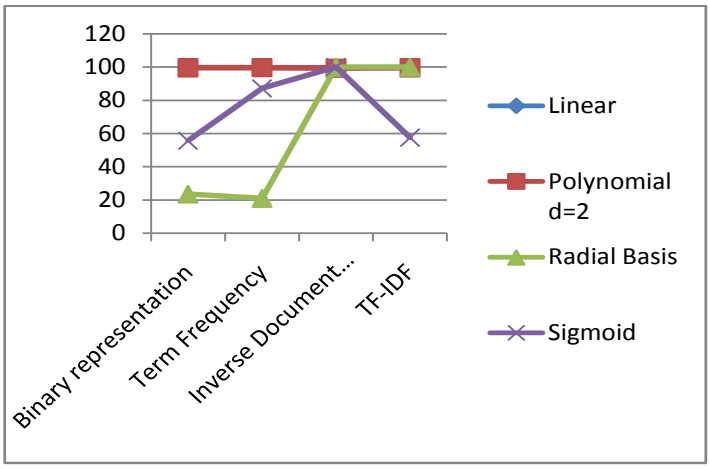


Figure 5.9 Recall of Spam on Test dataset Test 3:1

Analysis

After analyzing the results Obtained by performing train on one dataset Tain1-1 and test on different test datasets Test1-1, Test2-1 and Test3-1 having different spam: ham ratio , different feature representation and different kernel function following result is out come

a. In all cases linear function give highest accuracy as compare to other. highest accuracy is **96.71% on test data set Test 2-1** with **IDF.**
b. In average case test data set **Test 2-1** give highest accuracy in compare to other test data set.
c. In average case test data set **Test 2-1** give highest Precision and test data set **Test 1-1** give lowest Precision.
d. In case of Precision of spam **Radial Basis function** work well with **Binary** and **Term Frequency** representation.
e. In case of Precision of spam, **IDF and TFIDF** representation work well with **Linear function.**
f. In average case test data set **Test 2-1** give highest Precision and test data set **Test 1-1** give lowest Precision.
g. In average case test data set **Test 2-1** give highest Recall and test data set **Test 3-1** give lowest Recall.
h. sigmoid function give lower accuracy on test dataset **Test 1-1 and Test 2-1**
i. Both sigmoid and radial Basis perform approximate same.

Conclusion and Future Scope

The electronic mail (e-mail) concept makes it possible to communicate with many people in an easy and cheap way. But, many spam mails are received by users without their desire. As time goes on, a higher percentage of the e-mails are treated as spam. To efficiently solve the above problems, this work suggest which feature representation perform well with kernel function while changing in spam ham ratio in train and test dataset.

In Future, we plan to explore Other variants of SVM and Other feature selection algorithms (Information gain and $\chi 2$ statistic) can be used to reduce redundancy of dataset

References

[1] Prabin Kumar Panigrahi," A Comparative Study of Supervised Machine Learning Techniques for Spam E-Mail Filtering",2012 Fourth International Conference on Computational Intelligence and Communication Networks

[2] Ackoff, R. L. (1989). From data to wisdom. *Journal of applied systems analysis, 16*(1), 3-9.

[3] Chen, M., Ebert, D., Hagen, H., Laramee, R. S., Van Liere, R., Ma, K. L., et al. (2009). Data, information, and knowledge in visualization. *Computer Graphics and Applications, IEEE, 29*(1), 12-19.

[4] Sharma, N. (2008). The origin of the data information knowledge wisdom hierarchy. *The origin of the" Data Information Knowledge Wisdom" hierarchy*.

[5] B.Scholkopf and A. Smola. "*Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond"*, MIT Press, 2001.

[6] A. Kolcz, and J. Alspector. "SVM-based filtering of e-mail spam with content-specific misclassification costs". In Proceedings of the *Workshop on Text Mining*, pp. 123-130, California, USA, 2001.

[7] N. Cristianini and J. S. Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.

[8] V. vapnik. , "The Nature of Statistical Learning Theory", Springer, New York, 1995.

[9] Anil.K Jain, Robert P.W, Jianchang Mao "Statistical Pattern Reorganization: A Review"

[10] Kuan-Ming Lin, Chih-Jen Lin, "A Study on Reduced Support Vector Machines" *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 6, NOVEMBER 2003*

[11] K.P.Soman, R.Loganathan, V.Ajay "Machine learning with svm and other kernel methods"

[12] Jiancheng Sun, Chongxun Zheng, Xiaohe Li, Yatong Zhou "Analysis of the Distance Between Two Classes for Tuning SVM Hyperparameters" *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 21, NO. 2, FEBRUARY 2010, 305*

[13] Boyd & Vandenberghe "Convex Optimization"Available at: http://stanford.edu/class/ee364a/lectures/duality.pdf

[14] Shigeo Abe, "Support Vector Machine for Pattern Classification"

[15] http://www.aueb.gr/users/ion/data/enron-spam/

[16] http://www.webconfs.com/stop-words.php

[17] Salton G, Buckley C (1988). "Term-weighting approaches in automatic text retrieval". *Information Processing and Management* 24 (5): 513–523

[18] Stehman, Stephen V. (1997). "Selecting and interpreting measures of thematic classification accuracy". *Remote Sensing of Environment* 62 (1): 77–89

[19]Powers, David M W (2007/2011). "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation". *Journal of Machine Learning Technologies* 2 (1): 37–63