Spam Filtering using SVM with different Kernel Functions

Deepak Kumar Agarwal
M.Tech CS-IInd year
ABES Engineering College Ghaziabad, India

Rahul Kumar
M.Tech CS-IInd year
ABES Engineering College Ghaziabad, India

ABSTRACT

The growing volume of unwanted bulk e-mail (also known as junk-mail or spam) has generated a need for trustworthy anti-spam filters. Now a day, many Machine learning techniques have been used which are robotically filter the junk e-mail in much unbeaten rate. In this paper, we used one of the most popular machine learning Algorithm support vector machine (SVM) with different parameters using different kernel-functions (linear, polynomial, RBF, sigmoid) are implemented on spambase-dataset. Comparison of SVM performance for all kernels (linear, polynomial, RBF, sigmoid) using different parameters (C-SVC, NU-SVC) evaluated on spambase-dataset to get best accuracy.

General Terms

Classification using SVM

Keywords

Spam-filtering, Support Vector Machine, Kernel-functions

1. INTRODUCTION

The expanding volume of spontaneous mass e-mail (otherwise called spam) has created a requirement for dependable hostile to spam filters. Machine learning techniques now days used to naturally filter the spam e-mail in an exceptionally effective rate. As of late spontaneous business/mass e-mail otherwise called spam, turn into a major inconvenience over the internet. Spam is exercise in futility, storage room and correspondence data transmission. The issue of spam e-mail has been expanding for quite a long time. In late insights, 40% of all emails are spam which around 15.4 billion email every day and that cost internet clients about $355 million every year. Programmed e-mail filtering is by all accounts the best system for countering spam right now and a tight rivalry in the middle of spammers and spam-filtering techniques is going on. Just quite a while back the greater part of the spam could be dependably managed by blocking e-mails originating from certain addresses or filtering out messages with certain titles. Spammers started to utilize a few dubious techniques to beat the filtering strategies like utilizing irregular sender addresses and/or affix arbitrary characters to the starting or the end of the message title [1]. Knowledge engineering and machine learning are the two general methodologies utilized as a part of e-mail filtering. In knowledge engineering methodology an arrangement of principles must be indicated by emails are classified as spam or ham. An arrangement of such guidelines ought to be made either by the client of the filter, or by some other power (e.g. the software organization that gives a specific standard based spam-filtering apparatus). By applying this system, no encouraging results demonstrates on the grounds that the tenets must be always overhauled and kept up, which is an exercise in futility and it is not helpful for most clients. Machine learning methodology is more proficient than knowledge engineering methodology; it doesn't require determining any tenets [1]. Rather, an arrangement of training samples, these samples is an arrangement of pre ordered e-mail messages. A particular algorithm is then used to take in the classification rules from these e-mail messages. Electronic mail is seemingly the "executioner application" of the internet. It is utilized day by day by a huge number of individuals to communicate around the world and is a mission-critical application for some businesses. Throughout the most recent decade, spontaneous mass email has turned into a noteworthy issue for email clients. A staggering measure of spam is streaming into clients' mailboxes day by day. In 2004, an expected 62% of all email was credited to spam, as per the counter spam outfit Brightmail.1 Not just is spam frustrating for most email clients, it strains the IT base of associations and expenses organizations billions of dollars in lost efficiency. Lately, spam has advanced from a disturbance into a genuine security risk, and is currently a prime medium for phishing of touchy data, too the spread of malicious software. A wide range of methodologies for battling spam have been proposed, extending from different sender authentication protocols to charging senders unpredictably, in cash or computational assets [2]. The subject of machinelearning has been generally considered and there are loads of algorithms suitable for this task but here we are considered svm machine learning algorithm with different kernels also with different parameters.

2. RELATED WORK

In This paper [3], author to provide a complete machine learning algorithms comparison within the Web spam detection community. They use more than a few machine learning algorithms(SVM,MLP,BN,DT,RF,NB,KNN) and group meta-algorithms(AdaBoost,LogitBoost,Real,AdaBoost,Bagging,Dagging,Rotation- Forest) implements two freely available datasets (WEBSPAM-UK2006 and WEBSPAM-UK2007) as classifiers. Distribution of Features Vectors of above both dataset define by notation A(Content-based Features-24), B(Full Content-based Features-96), C(Link-based Features-41), D(Transformed Link-based Features-138) These data set in the results they show that Random Forest has proven to be a powerful classifier than most top data mining tools including SVM and MLP in Web spam detection with AUC results of 0.927 in WEBSPAM-UK2006 and 0.850 in WEBSPAM-UK2007 using both full content and transformed link-based features. With group meta-algorithm such as Real AdaBoost and Discrete AdaBoost, the performance is slightly improve with 0.937 in WEBSPAM-UK2006 and 0.852 in WEBSPAM-UK2007.

This paper though only focuses on the structure of the machine learning classifiers used for Web spam classification. For future work, the features for Web spam detection are intended to comprehensively compared and studied. Furthermore, the structures in this study are intended to test on other Web Spam datasets.

In this paper [4], author show the AdaBoost incorporating appropriately designed RBFSVM (SVM with the RBF kernel) component classifiers, which they call AdaBoostSVM, can perform as well as SVM. they projected AdaBoostSVM demonstrates better generalization performance than SVM on imbalanced classification problems. idea of AdaBoostSVM is that for the sequence of trained RBFSVM component classifiers, AdaBoost algorithem widely used in properly designed SVM-based component classifiers which is achieved by adaptively adjusting the kernel parameter to get a set of effective RBFSVM component classifiers. The experimental results on benchmark data sets demonstrate that proposed AdaBoost SVM according to paper performs better than other approaches of using component classifiers such as Decision Trees,Neural Networks and many more according to paper. It is found in the paper that Ada BoostSVM demonstrates good performance on imbalanced classification problems. According to the paper improved version is further developed to deal with the accuracy/diversity dilemma in Boosting algorithms, giving rising to better generalization performance.

In this paper [5], an extensive survey of late machine learning ways to deal with Spam filters was introduced. Focusing on many approach like textual- and image-based approaches etc. Despite of considering Spam filtering as a standard classification problem, In the paper we highlight the importance of considering specific characteristics spam filtering using machine learning, especially concept drift, in designing new filters. There is two particularly important aspects which is not widely recognized in research of the spam filtering using machine learning. The difficulty is arise when we need to updating a classifier based on the sack-of-words representation method and a major difference between two early naive Bayes models supposed. in this paper Author define A quantitative examination of the utilization of feature determination algorithms and datasets was led. It was checked that the information gain is the most regularly utilized strategy for feature determination, in spite of the fact that it has been proposed that others (e.g., the term-frequency change, in Koprinska et al. (2007)) may prompt enhanced results when utilized with certain machine learning algorithms. Among the few openly accessible datasets, the LingSpam and SpamAssassin corpora stand as the most mainstream, while the late TREC corpora, which endeavor to replicate a reasonable, web, setting, are tolerably prevalent at present. Regarding assessment measures, the genuine positive and negative rates, which are given, separately, by the relative number of Spam and honest to goodness messages accurately ordered, are proposed as the favored files for assessing filters, particularly as ROC bends (Fawcett, 2006). Two imperative perspectives not generally perceived in the writing were examined. Albeit most algorithms speak to messages as sack of-words, it ought to be precisely utilized, as it forces an extreme inclination in the issue. This is because of that reality that redesigning a model to consider new terms, which were not at first accessible, can be a feeble point, as it as a rule requires re-constructing the classifier starting with no outside help.

In This review paper [6], purpose of the research and describe how spam has become crucial issue in marketing communications, considering opinions of the digital marketing sector and Internet users. In depth paper and concept were organized with digital marketing experts in order to gain a profounder understanding in the complex construct of spam. Additionally, in this paper a web-based survey explored for us in this wide whether and how Internet users handle spam and privacy online. The reason of the current study was to look at people's attitudes towards online privacy and the measures taken to protect themselves against spamming. Survey results unveiled three users segments, each holding specific profiles on concern for personal information exposed online, sharing information online and attitude towards spam.

They show in Results of this survey Internet users clearly engage in various coping actions in protecting themselves from spam. but, in the experts' opinions Internet users need to be sensitized on spam to an even greater level taking into account spamming via new technologies such as blue spam or mobile spam. Experts claim that users do not fully take hold of the construct of spam and in addition state that users require to be empowered and educated on permission management and how to act upon cookies.

In this paper [7], author selected most fashionable Arabic Web pages in the Middle East region according to Alexa.com ranking for the duration of 2012 fourth quarter. They evaluated those fashionable Websites against the probable usage of spam techniques. Results showed that the bulk of those Web pages use spamming techniques with different levels and approaches. also They noticed that the bulk of the popular Web pages in Arab region are either classified as entertainment or social media Web pages. Author also center of attention on those Websites and exclude Websites of possible trusted domains such as: (.edu or .gov). Nevertheless this assumption, whether such trusted Websites, may have less usage of spam should be further investigated. Visibility to entertainment and social networking Websites is very vital. Spam techniques can be then used to increase such visibility. NaiveBayes (NB) classifier is used to classify Web pages into Spam or legitimate . The performance metrics prediction, recall, F-measure, and the region under the ROC curve are measured to show the quality or accuracy of the predicted classification. author believed nevertheless that the classification of Web pages into Spam and legitimate is not yet mature, particularly for Arabic Websites. There are a few criteria that are not widely agreed upon to be considered as a spam behavior or not. in actual fact, search engines conduct some activities that are banned by themselves, if conducted by others, and therefore classified as spam techniques.

3. STATEMENT OF THE PROBLEM

Spam is very annoying problem which is being faced by almost everyone having an email account.40% of all emails are spam which around 15.4 billion email every day and that cost internet clients about $355 million every year. It is imperative to filtering of spam email before sending it to the inbox of users, indeed this has been very important and challenging task. Various Machine learning methods are being used to classify spammer's emails from legitimate emails. Now we are using machine learning algorithm Support vector machine (SVM) for solving this problem using different kernel-functions and also using different parameter, compare the performance of SVM for all different kernels and eventually we will optimize to get best result.

4. SVM CLASSIFICATION ALGORITHM

The Algorithms: Theory

This area gives a brief diagram of the basic hypothesis of the algorithms we consider. We should talk about the naive Bayesian classifier, the k-NN classifier, the neural network classifier and the support vector machine classifier. Here we will examine the support vector machine classifier.

4.1 Support Vector Machine Classification

The last algorithm considered in this article is the Support Vector Machine classification algorithm. Support Vector Machines (SVM) is a group of algorithms for classification and regression created by V. Vapnik, that is presently a standout amongst the most generally utilized machine learning techniques with heaps of utilizations [10]. SVMs have a strong hypothetical establishment—the Statistical Learning Theory that ensures great speculation execution of SVMs. Here we just consider the most straightforward conceivable SVM application—classification of linearly separable classes—and we overlook the hypothesis. See [1] for a decent reference on SVM. The thought of SVM classification is the same as that of the perceptron: locate a straight partition limit wT x + b = 0 that effectively classifies training samples (and, as it was specified, we expect that such a limit exists). The distinction from the perceptron is this time we don't hunt down any separating hyper-plane, yet for an extremely extraordinary maximal margin isolating hyper plane, for which the separation to the nearest training sample is maximal. Definition Let X = {(xi, ci)}, xi 2 Rm, ci 2 {−1, +1} indicate as ordinarily the arrangement of training samples. Assume (w, b) is an isolating hyper-plane (i.e. sign (wT xi+b) = ci for all i). Characterize the margin mi of a training sample (xi, ci) regarding the isolating hyper plane as the separation from point xi to the hyper plane: mi = |wT xi + b| kwk The margin m of the isolating hyper plane concerning the entire training set X is the littlest margin of an occurrence in the training set:

At last, the maximal margin separating hyper plane for a training set X is the isolating hyper plane having the maximal margin concerning the training set

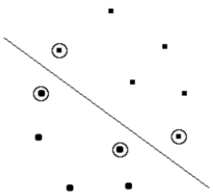


Figure 1: Maximal margin separating hyper plane and circles mark the support vectors.

Since the hyper plane given by parameters (x, b) is the same as the hyper plane given by parameters (kx, kb), we can securely bound our inquiry by just considering accepted hyper planes for which min I $|w^T x_i + b| = 1$.

5. MECHANISM OF SPAM FILTERING

Here the methodology used for spam filtering is support vector machines (SVM). SVM is concept of supervised learning. Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. Here we use LIBSVM tool which contains all the libraries of SVM (support vector machines). library for Support Vector Machines (LibSVMs)tool[9] has developed since the year 2000. The goal of developing these tool users can easily apply SVM to their applications. LIBSVM[9] has gained wide popularity in machine learning and many other areas. In LIBSVM we have used four types of kernels namely 0,1,2,3. "0" stands for linear kernel, "1" stands for polynomial "2" stands for RBF(Radial Basis Function) and "3" stands for Sigmoid kernel

A typical use of LIBSVM involves **three steps**: **first**, split spambase dataset[8] into two combination of train and test ratio like as 10:90 , 20 :80 , 30:70 , 40:60 , 50:50, 60:40, 70:30, 80:90, 90:10 respectively. The same procedure follows for rest of the three kernels. **Second**, training a data set to obtain a model and **third**, using the model to predict information of a testing data set. For SVM, LIBSVM[9] can also output probability estimates.

After this the accuracy is estimated for all the kernels at all the combinations of train files and test files i.e. from "10 train files and 90 test files" to "90 train files and 10 test files" respectively. Now there find out accuracies for all 4 kernel. Here we get different results (accuracy) on different kernel while using different parameter.

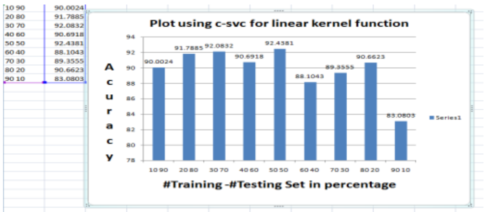
6. RESULTS OF SPAM FILTERING

The following graphs show the accuracies of various combinations used in all the four types of kernels.

6.1 Accuracy for all kernels using C-SVC

6.1.1 *C-SVC for Linear kernel*

we get maximum 92.4381% accuracy for train test ratio (50:50)

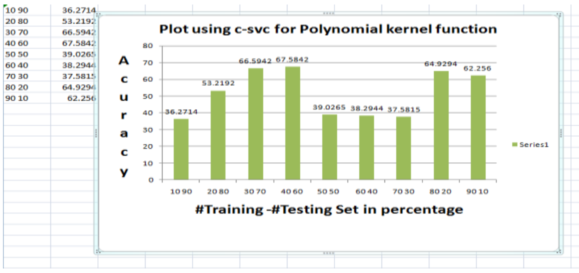


10 90	90.0024
20 80	91.7885
30 70	92.0832
40 60	90.6918
50 50	92.4381
60 40	88.1043
70 30	89.3555
80 20	90.6623
90 10	83.0803

Figure 6.1: Accuracy on c- svc using linear kernel

6.1.2 *C-SVC for polynomial kernel:*

we get maximum 67.5842% accuracy for train test ratio (40:60).

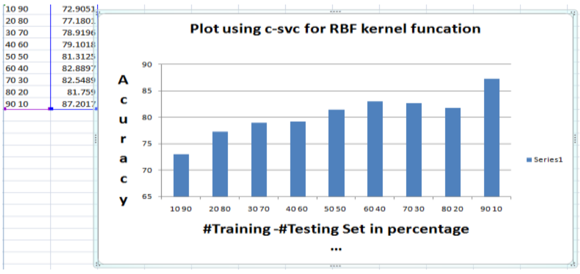


10 90	36.2714
20 80	53.2192
30 70	66.5942
40 60	67.5842
50 50	39.0265
60 40	38.2944
70 30	37.5815
80 20	64.9294
90 10	62.256

Figure 6.2: Accuracy on c- svc using polynomial kernel

6.1.3 *C-SVC for RBF kernel*

we get 82.8897% accuracy for train test ratio (60:40).



10 90	72.9051
20 80	77.1801
30 70	78.9196
40 60	79.1018
50 50	81.3125
60 40	82.8897
70 30	82.5489
80 20	81.759
90 10	87.2017

Figure 6.3: Accuracy on c- svc using RBF kernel

6.1.4 *C-SVC for Sigmoid kernel*

we get maximum 82.8897% accuracy for train test ratio (10:90)

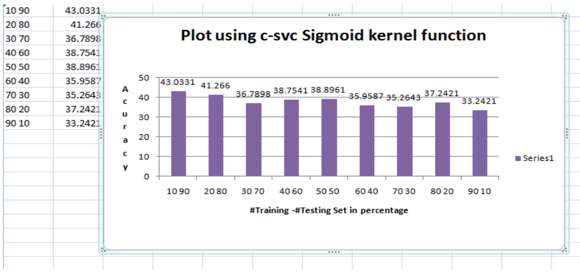


Figure 6.4: Accuracy on c-svc sigmoid kernel function

6.2 Accuracy for all kernels using NU-SVC

6.2.1 *NU-SVC for Linear kernel*

we get maximum 88.7722% accuracy for train test ratio (40:60).

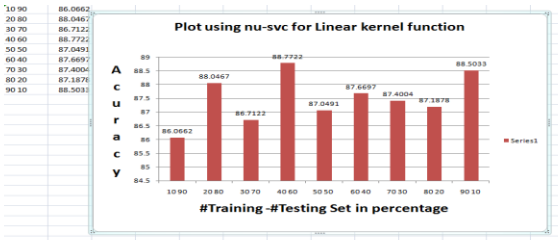


Figure 6.5: Accuracy on nu- svc using linear kernel

6.2.2 *NU-SVC for Polynomial kernel*

we get maximum 78.305% accuracy for train test ratio (70:30).

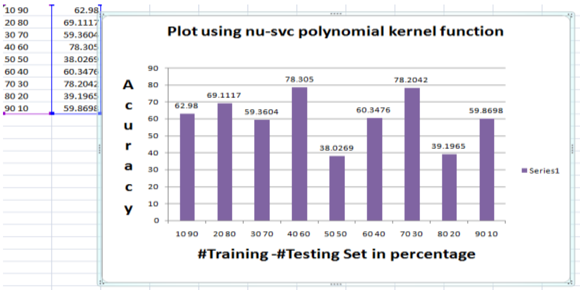


Figure 6.6: Accuracy on nu- svc using Polynomial kernel

6.2.3 *NU-SVC for RBF kernel*

we get maximum 87.8897% accuracy for train test ratio (90:10).

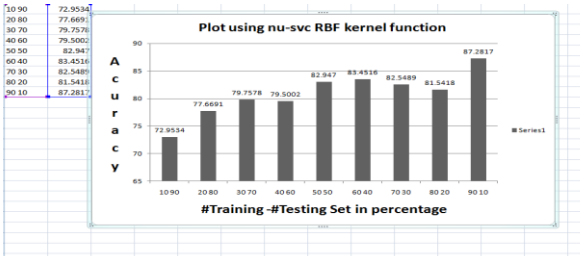


Figure 6.7: Accuracy on nu- svc using RBF kernel

6.2.4 *NU-SVC for Sigmoid kernel*

we get maximum 66.5944% accuracy for train test ratio (90:10).

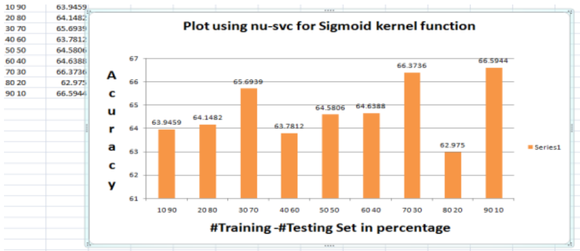


Figure 6.8: Accuracy on nu- svc using sigmoid kernel

6.3 Performance of SVM

Here we express the performance of SVM; we will have Comparison of SVM performance through following process:

6.3.1 Performance of SVM, when we considered different parameters.
6.3.2 Performance of SVM, when we considered different kernel functions.
6.3.3 Performance of SVM, when we considered different train: test combinations.

6.3.1 *Performance of SVM, when we considered different parameters*

here we considered two different parameters namely C-SVC and NU-SVC. So we compared the performance of SVM.

6.3.1.1 *Analysis the Performance of SVM using C-SVC parameter*

In this step we used C-SVC parameter for all different kind of kernels and after operation we can observe some result (See Table 6.1)

Table 6.1: Performance of SVM, when we considered C-SVC parameter for all different kernels

Train : test combination	**Kernels on C-SVC**	**Accuracy%**
50:50	Linear kernel	92.4381
40:40	Polynomial kernel	67.5842
60:40	RBF kernel	82.8897
10:90	Sigmoid kernel	43.0331

6.3.1.2 *Analysis the Performance of SVM using NU-SVC parameter*

In this step, we used NU-SVC parameter for all different kind of kernels and observe some unique result which is explain below (See Table 6.2)

Table 6.2: Performance of SVM, when we considered NU-SVC parameter for all different kernels

Train : test combination	**Kernels on NU-SVC**	**Accuracy%**
40:60	Linear kernel	88.7722
40:60	Polynomial kernel	78.305
90:10	RBF kernel	87.2817
90:10	Sigmoid kernel	66.5944

In this Section, We are focus on different parameters(C-SVC, NU-SVC), doesn't matter what we have in training or test ratio? And don't care about the kernel which we are using.

On C-SVC, Analysis the accuracy 92.4381% on linear kernel in which we have train (50): test (50) ratio which is best result in all the training set : Test set ratio (See Table 6.1)

ON NU-SVC, Analysis the accuracy 88.7722% on linear kernel in which we have training set(40): Testing set(60) ratio which is best result in all the training set : Test set ratio (See Table 6.2)

6.3.2 Performance of SVM, when we considered different kernel functions

Libsvm contain four different kernels linear kernel, polynomial kernel, RBF, sigmoid kernel. We implemented SVM algorithm for all kernel which are define in libsvm tool. So we compared the performance of SVM (See Table 6.3).

Table 6.3: **Performance of SVM, when we considered all different kernels**

Train : test combination	**Kernels**	**Accuracy%**
50:50	Linear kernel	92.4381
40:60	Polynomial kernel	78.3050
90:10	RBF kernel	87.2817
90:10	Sigmoid kernel	66.5944

In this Section, We are focus on different kernel-functions (linear, polynomial, RBF and sigmoid), in this we don't care about the training and testing ratio. And its parameters (C-SVC and NU-SVC) also.

After this observation we can analysis and declare that the linear kernel perform better in the term of comparison with other reaming kernels (polynomial, RBF, sigmoid) this gives 92.4381% Accuracy which is best result.

6.3.3 Performance of SVM, when we considered different train: test combinations

We have taken spamebase-dataset from uci-repository; we have divided this spambase-dataset into combination of train files and test files. The 9 combinations include 10 train files and 90 test files, 20 train files and 80 test files, 30 train files and 70 test files, 40 train and 60 test files, 50 train files and 50 test files, 60 train files and 40 test files, 70 train files and 30 test files, 80 train files and 20 test files, 90 spam files and 10 test files respectively. After this, accuracy is estimated s at all the combinations of train files and test files i.e. from "10 train files and 90 test files" to "90 train files and 10 test files" respectively. Here we compared the performance of SVM on different train: test file combination, doesn't matter what is kernel-function (linear, polynomial, RBF, sigmoid)? And what is parameter(C-SVC, NU-SVC)? Which are showing below:

For training: testing ratio (10:90), linear kernel provides 90.0024(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (20:80), linear kernel provides 91.7685(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (30:70), linear kernel provides 92.0832(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (40:60), linear kernel provides 90.6918(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (50:50), linear kernel provides 92.4380(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (60:40), linear kernel provides 88.1043(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (70:30), linear kernel provides 89.3555(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (80:20), linear kernel provides 90.6923(C-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

For training: testing ratio (90:10), linear kernel provides 88.5033(NU-SVC) Accuracy which is better result than other remaining kernel (polynomial, RBF, sigmoid).

According to this result for all defined train: test combination, linear kernel give best result than other remaining kernels (polynomial, RBF, sigmoid) for C-SVC, but in last result for train (90): test (10) combination, linear kernel give best result than other remaining kernel (polynomial, RBF, sigmoid) for NU-SVC.

So Finally, We can say linear kernel give better performance on different train: test ratio which I have already defined previously.

According to above result and also compare the performance of svm on the basis of three different way **using different parameters, using different kernel function** and **different train: test ratio** , we can say that linear kernel give more accurate result in each case. But that result only for two class dataset (spambase), that's not enough, we couldn't speak strongly linear kernel provide best result in each case.

To verify the experimental result of my thesis we used three more dataset with different classes to verify my result. So we are taking 3 different dataset with different classes to perform my proposed system.

1. Iris dataset with 3 classes

2. pendigit dataset with 10 classes

3. News20 dataset with 20 classes

For all above dataset (multi-class) dataset divided into two train: test ratio eg. 40: 60, 60 40, and also Find out SVM performance for all kernels using different parameters for each above dataset.

Iris dataset with 3 classes: iris dataset which is more popular dataset, iris dataset has been taken from uci repository there present 3 classes, 50 instances, and each class contain 50 instances and also contain some Attributes(4 numeric, predictive attributes and the class). Iris dataset doesn't contain any missing value.

SVM performance for all kernels using C-SVC: SVM performance for iris dataset (3 class) iris dataset is divided into two certain train: test sets. Eg. 40: 60, 60: 40 respectively. Using C-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results (See Table 6.4).

Table 6.4: Result for C-SVC (iris dataset with 3 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	96.6667	74.4444	96.6667	96.6667
60 : 40	98.3333	75	98.3333	98.3333

SVM performance for all kernels using NU-SVC: Using NU-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results (See table 6.5)

Table 6.5: Result for NU-SVC (iris dataset with 3 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	96.6667	92.2222	96.6667	96.6667
60 : 40	98.3333	95	98.3333	98.3333

Now we can say that, SVM performance for linear, RBF, sigmoid give same accuracy for both C-SVC, NU-SVC. This is better result than polynomial kernel. These three kernel (linear, RBF,sigmoid) perform well on iris(3 class) dataset.

Pendigit dataset with 10 classes: I have been taken pendigit dataset form uci repository. There is no missing value and pendigit dataset have 10 different classes with some different attributes. There are 16 input attributes and 1 class attributes. These All input attributes are integers define in the range 0..100 and The last attribute is the class attribute define as 0..9.

SVM performance for all kernels using C-SVC: SVM performance for pendigit dataset (10 class) dataset it is divided into two certain train : test combination. Eg. 40 : 60 , 60 : 40 respectively.

Using C-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results (See table 6.6).

Table 6.6: Result for C-SVC (pendigit dataset with 10 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	98.0649	99.2883	11.0765	10.387
60 : 40	97.8652	99.3996	12.3082	10.4069

SVM performance for all kernels using NU-SVC: Using NU-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results.

Table 6.7: Result for NU-SVC (pendigit dataset with 10 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	87.1664	88.2562`	11.4769	Error: fscan fail to read modelfile
60 : 40	87.525	88.9927	12.9886	Error: fscan fail to read modelfile

Now we can say that, SVM performance for linear, polynomial kernel offer excellent result for both C-SVC, NU-SVC parameters.

news20 dataset with 20 classes: this dataset contain 20 classes source of this dataset Ken Lang. Newsweeder: Learning to filter netnews. In Proceedings of the Twelfth International Conference on Machine Learning, pages 331-339, 1995**.**

SVM performance for all kernels using C-SVC:

SVM performance for news20 dataset (20 class) dataset it is divided into two certain train : test combination. Eg. 40: 60, 60: 40 respectively. Using C-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results (See table 6.8).

Table 6.8: Result for C-SVC (news20 dataset with 20 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	79.9812	20.0816	36.1155	36.1364
60 : 40	81.8324	18.3088	20.9758	20.9915

SVM performance for all kernels using NU-SVC: Using NU-SVC parameter for all kernels (linear, polynomial, RBF, sigmoid) we obtain the following results(See table 6.9).

Table 6.9: Result for NU-SVC (news20 dataset with 20 classes)

Train : test ratio	Accuracy % for Linear kernel	Accuracy % for Polynomial	Accuracy % for RBF	Accuracy % for Sigmoid kernel
40 : 60	78.8725	37.5065	65.7786	65.7358
60 : 40	80.1537	48.337	64.0257	64.0414

According to above both table (See 6.8, 6.9), linear kernel gives best result than other kernel. It give approximate 80% accuracy which is highest in above both table.

6.4 Analysis the Performance of SVM for different classes dataset

In this section, analysis the performance of svm for different classes dataset including both C-SVC and NU-SVC parameter.

6.3.4 Performance of SVM for different classes dataset using C-SVC

using c-svc, we observed some different result for different which are showing in below (See table 6.10)

Table 6.10: Performance of SVM for different classes dataset using C-SVC

Dataset with class	Train : test combination	Kernel	Accuracy (%)
Spambase with 2 classes	40 : 60	Linear kernel	90.6918
	60 : 40	Linear kernel	88.1043
Iris with 3 classes	40 : 60	Linear, RBF, Sigmoid	96.6667
	60 : 40	Linear, RBF, Sigmoid	98.3333
Pendigit with 10 classes	40 : 60	polynomial	99.2883
	60 : 40	Polynomial	99.3996
News20 with 20 classes	40 : 60	Linear	79.9812
	60 : 40	Linear	81.8324

.

6.3.5 Performance of SVM for different classes using NU-SVC

Using nu-svc, we observed some different result for different which are showing in below(See table 6.11)

Table 6.11: Performance of SVM for different classes using NU-SVC

Dataset with class	Train : test combination	Kernel	Accuracy (%)
Spambase with 2 classes	40 : 60	Linear kernel	88.7722
	60 : 40	Linear kernel	87.6697
Iris with 3 classes	40 : 60	Linear, RBF, Sigmoid	96.6667
	60 : 40	Linear, RBF, Sigmoid	98.3333
Pendigit with 10 classes	40 : 60	Polynomial	88.2562
	60 : 40	Polynomial	88.9927
News20 with 20 classes	40 : 60	Linear	78.8725
	60 : 40	Linear	80.1537

According to above experimental result for different class datasets spambase(2 class), iris (3 class), pendigit (10 class) and news20 (20 class) we found that our proposed work is fine and also we define following some term which may be useful for future work.

For 2 classes dataset considered linear kernel with C-SVC parameter get most excellent result for train: test ratio (50:50).

For 3 class dataset, choose any one kernel (linear, RBF, and sigmoid kernel) with any parameters(C-SVC,NU-SVC) get excellent result for train:test ratio (60:40) .

For 10 classes dataset, considered only polynomial kernel with NU-SVC parameter achieve excellent result for train: test ratio (60:40).

For 20 class dataset, considered linear kernel with NU-SVC parameter get excellent result for train: test ratio (60:40).

7. CONCLUSION

Spam filtering has been done by making use of the support vector machines. A pre-defined spambase dataset was taken from public domain website(UCI respiratory). In the data source website having some documentation where which i get some knowledge about the containing all the spam and non-spam messages in given dataset. The tool used for this methodology is the LIBSVM. In LIBSVM there are 4 type of kernels namely linear kernel, polynomial RBF (Radial Basis Function) and sigmoid kernel. We have taken our spambase dataset for each kernel into consideration in order to justify the result and work done. As per the machine learning algorithm need we divide these data into the training and testing and pass to the machine. After this the accuracy is estimated for all the kernels using different parameter (c-svc, nu-svc) at all the combinations of train file and test files. We proudly declare the accuracy obtained is **92.4381**% using c-svc parameter for linear kernel. Linear kernel with C-SVC perform well on spambase dataset than other kernel (RBF, polynomial, sigmoid).

We also validate our result by using 3 new dataset with different classes(3, 10 and 20) And we obtain a satisfactory output which is demonstrate that my proposed work working successfully on this domain too. In this way we can say my proposed result is correct and its can be used in further research on the same field of research with any filter data.

8. REFERENCES

[1] Cormack, Gordon. Smucker, Mark. Clarke, Charles " Efficient and effective spamfiltering and re-ranking for large web datasets" Information Retrieval, Springer Netherlands. January 2011

[2] Muhammad N. Marsono, M. Watheq El-Kharashi, Fayez Gebali “Targeting spam control on middleboxes: Spam detection based on layer-3 e-mail content classification” Elsevier Computer Networks, 2009.

[3] KwangLeng Go h, Ashutosh Kumar Singh, “Comprehensive Literature Review on Machine Learning structures for Web Spam Classification”, 4thInternational Conference o n Eco-friendly Computing and Communication Systems (ICECC S),Procedia Computer Science 70 (2015) 434 – 441

[4] Xuchun Li , Lei Wang, Eric Sung,” AdaBoost with SVM-based component classifiers”, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore Engineering Applications of Artificial Intelligence 21 (2008) 785–795.

[5] Thiago S. Guzella *, Walmir M. Caminhas,” A review of machine learning approaches to Spam filtering”,

Department of Electrical Engineering, Federal University of Minas Gerais, Ave. Antonio Carlos, 6627, Belo Horizonte (MG) 31270-910, Brazil, Expert Systems with Applications 36 (2009) 10206–10222.

[6] Kim Janssensa,*, NicoNijstena, Robrecht Van Goolena, “Spam and Marketing Communications”, Enterprise and the Competitive Environment 2014 conference, ECE 2014, 6–7 March 2014, Brno, Czech Republic, Procedia Economics and Finance 12 (2014) 265 – 272

[7] Mohammed N. Al-Kabi a, Izzat M. Alsmadi b,*, Heider A. Wahsheh c,” Evaluation of Spam Impact on Arabic Websites Popularity”, Journal of King Saud University – Computer and Information Sciences (2015) 27, 222–229

[8] http://www.ics.uci.edu/~mlearn/MLRepository.html

[9] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvmWu, C. ”Behavior-based spam detection using a hybrid method of rule-based techniques andneural networks” Expert Syst., 2009