International Journal of Computer and Communication Technology

Volume 1 | Issue 4 Article 6

October 2010

Spam Filtering using Support Vector Machine

Priyanka Chhabra
*MTech CSE *1, Asst.Professor MANIT, Bhopal, MP, India*, priyankachhabda@gmail.com

Rajesh Wadhvani
*MTech CSE *1, Asst.Professor MANIT, Bhopal, MP, India*, wadhvani_rajesh@rediffmail.com

Sanyam Shukla
M.Tech (C.S.E) 1, Assistant Professor M.A.N.I.T. . Bhopal ,India, sanyamshukla@manit.ac.in

Spam Filtering using Support Vector Machine

Priyanka Chhabra[*1] Rajesh Wadhvani[*2], Sanyam Shukla[*3]
MTech CSE [*1,] Asst.Professor [*2, 3]
MANIT, Bhopal, MP, India
[1]priyankachhabda@gmail.com , [2]wadhvani_rajesh@rediffmail.com ,[3]sanyamshukla@manit.ac.in

Abstract: **The traditional anti-spam techniques like Black and White List is not up to the mark in current scenario. The goal of Spam Classification is to distinguish between spam and legitimate mail message. But with the popularization of the Internet, 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. Many researchers have been trying to separate spam from legitimate emails using machine learning algorithms based on statistical learning methods. In this paper, we evaluate the performance of Non Linear SVM based classifiers with various kernel functions over Enron Dataset.**

Key words: *Machine learning, spam, SVM, kernel*

I. 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.

It is evident that people's work efficiency and their emotions will be influenced, if they have to spend time and efforts on identification E-mail every day. So to auto distinguish spam has important meaning and applying value. To classify the technologies of spam filtering, they can be classified into two types: server spam filtering and client spam filtering according to different places the filter is executed. One popular solution is an automated email filtering using Machine Learning (ML). Support Vector Machine is one of the most used techniques as the base classifier to overcome the spam problem [1]. Some studies developed spam filtering in a batch mode [2].

The remainder of this paper is organized as follows. Section II gives brief review of Spam filtering, Section III explores the evaluate standards for spam filtering, Section IV focuses on SVM, tool used for classifying spam and ham Section V elaborates the pre-processing done on the datasets before performing experiments, Section VI gives the Experimental Results whereas in Section VII we analyse the experimental results and finally conclusion and future work is given in Section VIII.

II. BRIEF REVIEW OF SPAM FILTERING

Spam filtering is the processing of e-mail to organize it according to specified criteria. Most often this refers to the automatic processing of incoming messages, but the term also applies to the intervention of human intelligence in addition to anti-spam techniques, and to outgoing emails as well as those being received.

Spam filtering software inputs email, for its output, it might pass the message through unchanged for delivery to the user's mailbox, redirect the message for delivery elsewhere, or even throw the message away. Some mail filters are able to edit messages during processing.

There are various Benchmark Datasets available for researchers related to spam filtering. There has been significant effort to generate public benchmark datasets for anti-spam filtering. One of the main concerns is how to protect the privacy of the users whose ham messages are included in the datasets. The first approach is to use ham messages collected from freely accessible newsgroups, or mailing lists with public archives.
Like
1. Ling-Spam
2. The SpamAssassin
3. The Spambase

4. Ecml-pkdd 2006 challenge dataset
5. PU corpora dataset
6. Enron dataset
7. Trec 2005 dataset

Some of these dataset are in processed form for e.g. Ecml-pkdd and data available in Spambase UCI repository. Authors [3 - 11] have applied various classification Algorithms like Naive Bayes, NPE, 2v-SVM n the abve mentioned dataset.

The major Problem faced is that no single Benchmark is used by all the Researchers, so the results cannot be compared. This Paper uses Enron Dataset which is publicly available in raw format; we have preprocessed the dataset and applied Nonlinear SVM classifier.

III. EVALUATION METRICS FOR SPAM FILTERING

The performance evaluation on spam filtering often makes use of some related indexes in text classification. The standard, which can decide whether text classification is mature or not, is the accuracy and speed. And the speed is decided by the complexity of arithmetic; the accuracy is evaluated by information retrieval evaluation. The followings are the definitions about two common indexes: Recall Ratio and Precision Ratio of information retrieval in spam filtering field [12]

Def 1: Recall Ratio is the ratio of the amount of spam that has been filtered to the amount of E-mails that should be filtered. The computing formula of Recall Ratio is:

$$\text{Recall} = \frac{\text{Amount of filtered spam}}{\text{Amount of E mails that should be filtered}}$$

Def 2: Precision Ratio is the ratio of the amount of spam that has been filtered to the amount of E-mails that have been filtered. The computing formula of Precision Ratio is:

$$\text{Precision} = \frac{\text{Amount of filtered spam}}{\text{Amount of E mails that have been filtered}}$$

IV. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) [13, 14] has been recently proposed by Dr.V.vapnik as an effective statistical learning method for pattern recognition. The SVM based on statistical learning theory has many advantages. Different from previous nonparametric techniques such as nearest-neighbors and neural network that are based on the minimization of the empirical risk, SVM operates on another induction principle, called structural risk minimization, which can overcome the problem of over fitting and local minimum and gain better generalization capability. Kernel function method is applied in SVM, which doesn't increase the computational complexity, furthermore overcomes the curse of dimensionality problem effectively. 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.

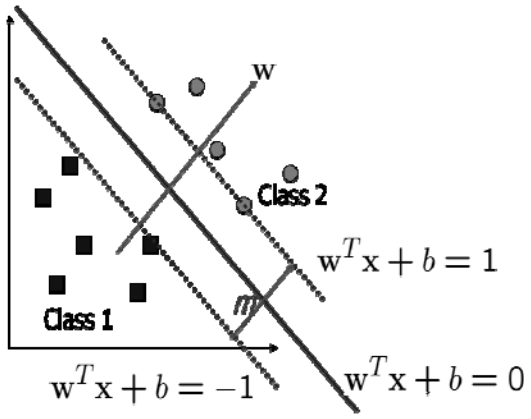
SVM has proved to be one of the most efficient kernel methods. The success of SVM [15] is mainly due to its high generalization ability. Unlike many learning algorithms, SVM leads to good performances without the need to incorporate prior information. Moreover, the use of positive definite kernel in 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. Hence, the problem of choosing architecture for a neural network application is replaced by the problem of choosing a suitable kernel for a Support Vector Machine.

Support Vector Machine has shown power in Binary Classification. It has Good Theoretical Foundation and Well Mastered learning algorithm. It shows Good Results in static data classification. The only disadvantage is; it is time and memory consuming when size of data is enormous.

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

4.1.1 Linear Separable Problems

While Classification if we can separate the classes using Linear Decision Boundary then it is Linear Separable Problem [16]. 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 [17] should be as far away from the data of both classes as possible. We should maximize the margin, *m.*



- 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

$$\text{Minimize } ½ \|w\|^2$$
$$\text{Subject to } Y_i (w^T x_i + b) \geq 1, \quad \forall_i$$

This Optimization Problem can be solved by using Lagrange's Dual Problem [18].

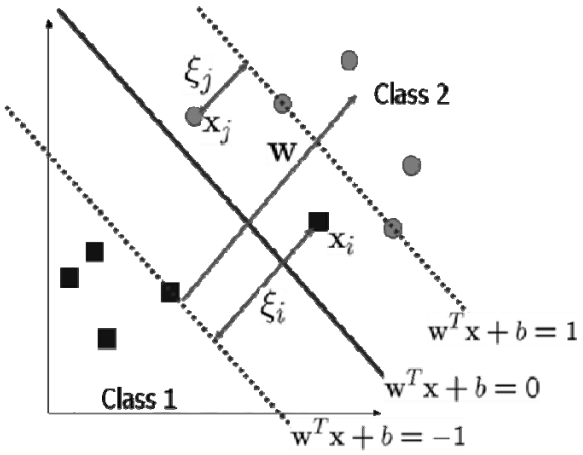
4.1.2 Non-Linearly Separable Problems

The Problem that cannot classify linearly is Non Linear Separable Problems. 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

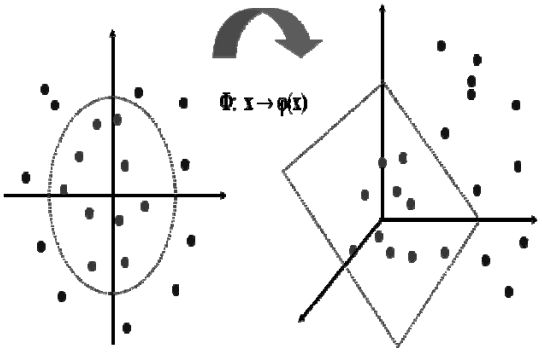
$$w^T x_i + b \geq 1 - £_i \qquad Y_i = 1$$
$$w^T x_i + b \leq -1 + £_i \qquad Y_i = -1$$
$$£_i \geq 0$$

Thus the Optimization Problem [19] Becomes

$$\text{Minimize } ½ \|w\|^2 + C \sum^n_{i=1} £_i$$
$$\text{Subject to } Y_i (w^T x_i + b) \geq 1 - £_i \qquad £_i \geq 0$$



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



To Obtain Non Linear Decision Boundary key idea is to transform $\mathbf{x}^i$ to a higher dimensional space called feature space to make it easier.

Input space: the space the point $\mathbf{x}^i$ is located

Feature space: the space of f ($\mathbf{x}^i$) 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 x^1x^2 make the problem linearly separable.

4.2 Kernel Functions:

An SVM is largely characterized by the choice of its kernel, and SVMs thus link the problems they are designed for with a large body of existing work on kernel-based methods. Now the main forms of the

kernel function are [5]: linear kernel function, polynomial kernel function, radial basis function and sigmoid function:

4.2.1 Linear kernel function

The Linear kernel is the simplest kernel function. It is given by the inner product <x,y> plus an optional constant **c**. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts

$$K(x,y) = x^T y + c$$

4.2.2 *Polynomial kernel function*

The Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$K(x,y) = (\alpha x^T y + c)^d$$

Adjustable parameters are the slope **alpha**, the constant term **c** and the polynomial degree **d**.

4.3.3 Radial Basis Function (RBF) or Gaussian Kernel

The Gaussian kernel [20] is an example of radial basis function kernel.

$$K(x,y) = \exp\left(-\gamma \|x-y\|^2 \right)$$

The adjustable parameter **sigma** plays a major role in the performance of the kernel

4.3.4 Hyperbolic Tangent (Sigmoid) Kernel

The Hyperbolic Tangent Kernel is also known as the Sigmoid Kernel. The Sigmoid Kernel comes from the Neural Networks field, where the bipolar sigmoid function is often used as an activation function for artificial neurons.

$$K(x,y) = \tanh(\alpha x^T y + c)$$

There are two adjustable parameters in the sigmoid kernel, the slope **alpha** and the intercept constant **c**.

The Above mentioned four Kernel Functions are the basic Kernel functions already implemented in svm_light tool. We have implemented Logarithmic Kernel.

4.3.5 Logarithmic Kernel The Logarithmic kernel [21] is log function implementation of the Euclidean distance between the two points.

$$K(x,y) = -\log(\|x-y\|^d + 1)$$

The x, y are vector representation of the data points in a N-dimensional space.

V. DATA PREPROCESSING

The datasets used here for spam filtering are ENRON dataset. Each dataset has two folders containing spam and ham messages. The various datasets have different spam: ham ratios

Enron-1	3672:1500
Enron-2	4361:1496
Enron-3	4012:1500
Enron-4	1500:4500
Enron-5	1500:3675
Enron-6	1500:4500

These Datasets are publicly available in raw format. We have processed the dataset to obtain it in svm light format.

5.2 SVM Light Format

- The input file to SVM light tool contains the emails in a particular format. Each line of the file represents a mail of the form

<line> .=. <target> <feature>:<value> <feature>:<value> ...
<target> .=. +1 | -1 | <integer>
<feature> .=. <integer>
<value> .=. <integer>

- The target value and each of the feature/value pairs are separated by a space character. Feature/value pairs MUST be ordered by increasing feature number. Features with value zero can be skipped.

In classification mode, the target value denotes the class of the example. +1 as the target value marks a positive example, -1 a negative example respectively.

5.3 Methodology

5.3.1 Dataset Preparation

All six datasets are mixed to form 5 new data sets each having training and testing modules. Training module contains 70% of the total messages and testing module has 30% of the total messages and also varying spam ham ratios.

The spam-ham ratio in the 5 datasets were

Dataset1-1:1
Dataset2- 1:2
Dataset3- 2:1
Dataset4- 1:3
Dataset5- 3:1

5.3.2 Algorithm for processing of Dataset

- Step1: The raw mails (both training and testing) are converted to .CSV format. In which each row corresponds to one email

and each cell contains the word and the next cell contains its frequency of that word in that mail.

- Step2: Calculate most frequent words of the training module (both ham and spam)
- Step3: Take 5000 most frequent words from both spam and ham mails and mixing them to form around 7000 most frequent words.
- Step4: Assign a unique integer for each word these will act as our features for classification.
- Step5: converting the .csv format into svm_light format using the above features.
- Step6: Generate a model by applying svm_learn for the above training file.
- Step7: Applying svm_classify to test module using the model generated by svm_learn and noting the recall and precession.

The above algorithm is repeated by changing the kernel function and we also observed the results by changing the value of d (degree of the polynomial) and this is repeated for all the five datasets with varying spam: ham ratio.

VI. EXPERIMENTAL RESULT

Experiment 6.1:
In this we used a dataset1 having a spam: ham ratio of 1:1

S.NO	Kernel Function	Recall	Precision
1	Linear Kernel	86.25	99.16
2	Sigmoid Kernel	66.55	65.75
3	Radial Basis Function	1.99	100
4	Polynomial Kernel with D=1	86.25	99.16
5	Polynomial Kernel with D=2	27.78	98.70
6	Polynomial Kernel with D=3	0.18	92.36
7	Logarithmic Kernel	31.65	72.64

Experiment 6.2:
In this we used a dataset2 having a spam: ham ratio of 1:2

S.No	Kernel Function	Recall	Precision
1	Linear Kernel	95.31	97.86
2	Sigmoid Kernel	13.56	53.33
3	Radial Basis Function	99.87	51.67
4	Polynomial Kernel with D=1	95.31	97.86
5	Polynomial Kernel with D=2	99.02	54.31
6	Polynomial Kernel with D=3	10.88	0.56
7	Logarithmic Kernel	31.56	73.82

Experiment 6.3:
In this we used a dataset3 having a spam: ham ratio of 2:1

S.No	Kernel Function	Recall	Precision
1	Linear Kernel	72.31	99.75
2	**Sigmoid Kernel**	100	49.99
3	Radial Basis Function	0.18	100
4	Polynomial Kernel with D=1	72.31	99.75
5	Polynomial Kernel with D=2	18.23	99.44
6	Polynomial Kernel with D=3	0.36	100.00
7	Logarithmic Kernel	33.46	85.25

Experiment 6.4:
In this we used a dataset4 having a spam: ham ratio of 1:3

S.No	Kernel Function	Recall	Precision
1	Linear Kernel	97.56	90.28
2	Sigmoid Kernel	5.49	79.72
3	Radial Basis Function	99.91	51.17
4	Polynomial Kernel with D=1	97.56	90.28
5	Polynomial Kernel with D=2	99.39	52.42
6	Polynomial Kernel with D=3	0.29	99.33
7	Logarithmic Kernel	28.8	70.28

Experiment 6.5:
In this we used a dataset5 having a spam: ham ratio of 3:1

S.No	Kernel Function	Recall	Precision
1	Linear Kernel	67.49	99.66
2	Sigmoid Kernel	3.13	87.10
3	Radial Basis Function	0.09	100
4	Polynomial Kernel with D=1	67.49	99.66
5	Polynomial Kernel with D=2	17.55	99.51
6	Polynomial Kernel with D=3	0.38	100.00
7	Logarithmic Kernel	27.72	80.02

VII.ANALYSIS

After analyzing the results Obtained by performing different experiments on datasets having different spam: ham ratio we can say that Result Obtained for Dataset3 having spam: ham ratio 1:3 gives the appropriate result i.e. the classification is appropriate for more legitimate mails. The experiment also shows that the same performance for Linear Kernel and Polynomial Kernel for degree =1which should be same because they imply the same kernel function. The performance decreases with increase of degree of polynomial. It cannot compare two kernels as we need both Recall and Precision to be good.

VIII. CONCLUSION AND FUTURE WORK

Classifiers like Decision Tree Classifier have large memory requirement. The number of features for spam filtering is more than 7000, and may vary from 7000 to large number. In order to evaluate these many attributes SVM has proved to be good classifier because of its sparse data format and acceptable Recall and Precision Value. Also SVM is regarded as an important example of "kernel methods", one of the key areas in machine learning.

In Future, we plan to propose and implement a new Kernel Function and will analyze its performance over any of the available dataset.

REFRENCES

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

[2] 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.

[3] Vangelis Metsis, Ion Androutsopoulos, Georgios Paliouras,"Spam Filtering with Naive Bayes – Which Naive Bayes? "*CEAS 2006 Third Conference on Email and AntiSpam* July 27-28,2006,Mountain View, California USA

[4] Shu-wei chih, Tsan Ying Yu "Email Spam Filtering using SVM with selected Kernel Parameters ", 2009 fourth international conference on innovative computing, information and control.

[5] Zi Qiang Wang, Xia Sun, Xin Li, De Xian Zhang "An Efficient SVM based Spam Filtering Algorithm", 19th IEEE International Conference on tools with artificial intelligence,2006.

[6] Kurum Nazir Junego, Asim Karim "Automatic Personalized Spam Filtering Through Significant Word Modeling", 19th IEEE International Conference on tools with artificial intelligence,2006.

[7] Mio Ye, Qiu Jiang Xiang, Fan Jin Mai "The Spam Filtering Technology Based On SVM and D-S Theory", 2008 workshop on knowledge discovery and datamining.

[8] Mingqing Hu, Yiqiang chen, James Tin-Yau Kwok "Building Sparse Multiple Kernel SVM Classifiers", IEEE Transaction on Neural Networks, Vol.20,No 5, May 2009.

[9] A.Mathur, G.M Foody "Multiclass and Binary SVM Classification: Implecation for Training and Classification users" IEEE Geoscience and Remote sensing letters, Vol 5, No2, April 2008.

[10] Ziqiang Wang, Xia Sun "An Efficient Spam Filtering Algorithm Based on NPE" IEEE International Symposium on *Knowledge Acquisition and Modeling Workshop,21-22Dec 2008* pp 1102 – 1104.

[11] Anna Wang1 Weihao Fan2, Jie Wu1, Yongyue Shi1 and Dan Li1 "SPAM FILTERING SYSTEM STUDY BASED ON 2v-SVM" Proceedings of the 7th World Congress on Intelligent Control and Automation June 25 - 27, 2008, Chongqing, China © 2008 IEEE.

[12] C.J. van Rijsbergen. (1979). *Information Retrieval* (2nd edition), Butterworths, London, 1979.
Available at: http://www.cs.cmu.edu/~enron/

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

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

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

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

[17] 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*

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

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

[20] Bernhard Scholkopf, Kah-Kay Sung, Chris J. C. Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers" *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11, NOVEMBER 1997*

[21] http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html