Spam Classification using new kernel function in Support Vector Machine

Surendra Kumar Rakse
M.Tech CSE, MANIT
Bhopal, INDIA
surendra.rakse@gmail.com

Sanyam Shukla
Asstt. Prof. MANIT
Bhopal, INDIA
sanyamshukla@manit.ac.in

***Abstract*—Due to the increase in internet users, there is a rapid growth in spam e-mails. In recent years, kernel function have received major attention, particularly due to the increased popularity of Support Vector Machine. It is a best classifier for binary classification. Kernel functions are used to map data into high dimensional feature space. In this paper a new kernel function is developed called Cauchy kernel, and performance is evaluated over ECML-PKDD dataset and compared with the predefined kernel functions. The experiment shows that the results are better than predefined kernel functions.**

Keywords- Spam Filtering, SVM, Kernel Functions.

I. INTRODUCTION

Internet is the fast growing technology, and one of its finest services is Electronic Mail. It is the cheapest and the fastest medium of communication. All the internet users are not untouched of this service. Most of us using the Internet e-mail service face almost daily unwanted messages in our mailboxes. We have never asked for these e-mails, and often do not know the sender, and puzzle about where the sender got our e-mail address from. The types of those messages vary: some contain advertisements, others provide winning notifications, and sometimes we get messages with executable files, which finally emerge as malicious codes, such as viruses and Trojan horses. Apparently, the Internet e-mail infrastructure is widely used, as well as misused, as an efficient medium for information distribution [1]. Spam mails are different for different users like if a user likes to receive advertising e-mails having various lucrative offers but some other user do not want to receive these types of advertising e-mails. Spammers flood the internet with thousands of unwanted emails at negligible cost, the actual cost is distributed among the maintainers and users of the internet. Spam filters can be implemented either on server side or on client side. E-mail users spend a lot of time reading messages and deciding whether they are spam or non-spam and categorizing them into folders. These spam mails produce unnecessary network congestion, and waste huge amount of network Bandwidth.

This paper described that support vector machines (SVMs) have attracted much attention as a new classification technique with good generalization ability. The basic idea of SVMs is to map input vectors into a high-dimensional feature space and linearly separate the feature vectors with an optimal hyperplane in terms of margins. Kernel functions are the root of support vector machine, since if we properly choose kernel function can make a lot of difference in classification problems.

The rest of the paper is organized as follows: Section II contains the basics of spam filtering, classification, and text categorization. Section III describes the Support Vector Machine, and kernel functions. Section IV comprises of dataset, ECML-PKDD dataset. In section V results are shown for the experiments done. Section VI concludes the results.

II. LITERATURE SURVEY

A. Spam Filtering

Spam Filtering is a binary classification, with the categories spam and ham (legitimate mail). Spam mails are broadly categories as UCE and UBE.

UCE – Unsolicited Commercial E-Mail, that is E-Mail containing commercial information that has been sent to a recipient who did not ask to receive it.

UBE – Unsolicited Bulk E-Mail, that is E-Mail with substantially identical content sent to many recipients who did not ask to receive it. [2]

B. Spam Categories

Spam can be categorized according to the spammer’s goal. Many spammers send out their bulk e-mail for advertising reasons, for example, they send commercial ads or participate in political campaigns, whereas others have some kind of criminal fraud in mind or distribute malicious software, such as viruses or Trojan horses.

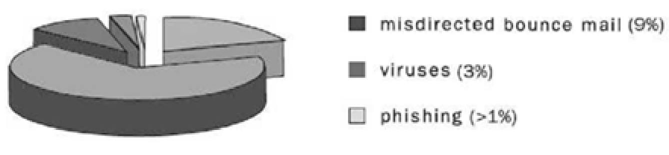


Fig – 1 [1]

C. Classification

Databases are rich with hidden information that can be used for intelligent decision making. Classification is the form of data analysis that can be used to extract models describing important data classes. Classification predicts categorical labels and these categories can be represented by discrete values, where the ordering of values has no importance. Classification is two-step process. In first step, classifier is built describing predetermined set of data classes, this is "learning" step, where a classification algorithm builts the classifier by analyzing or "learning from" a "training set" made up of database tuples and their associated class labels. Spam Filtering is a binary classification, with the categories spam and ham. [3]

D. Text Categorization

The goal of text categorization is the classification of documents into a fixed number of predefined categories. Each document can be multiple, exactly one, or no category at all. The first step in text categorization is to transform document, which typically are string of characters, into a representation suitable for the learning algorithm and the classification task. Information retrieval research suggests that word stems work well as representation units and that their ordering in a documents is of mirror importance for many tasks. This leads to an attribute value representation of text. Each distinct word w_i corresponds to a feature, with the number of times word w_i occurs in the document as its value to avoid unnecessarily large feature vector, words are considered as features only if they occur in the training data at least three times and if they are not "stop words" (like "and", "or" etc). This representation scheme leads to very high dimensional feature space. This leads the need of feature selection to make the conventional learning methods possible to improve generalization accuracy, and to avoid "overfitting". [4]

E. Text Categorization approaches

1) Rule based learning:

This is one of the earliest approaches attempted. Recently automatic rule induction techniques have also been used to avoid the laborious construction of the rule base. Dimension reduction is always necessary in order to reduce the search space. One of the main advantages of these rule-based approaches is that the resulting rules are usefully readily interpreted by humans.

2) K-Nearest Neighbour

Variants of the k-NN classifiers have shown very good performance in text categorization. However, on categorizing a new document, a large amount of computational power is required for calculating its similarity with every training document.

3) Neural Network

Because the decision boundary in a text categorization problem may not be linear, nonlinear classifiers such as artificial neural networks may produce better results than linear models. To alleviate the problem of high dimensinality use both weigh elimination and early stopping in addition to dimension reduction. However, experimental results suggest that the nonlinearity in neural networks does not yield substantial gain over linear models.

4) Neural Probablistic Models

They are very widely used in many text categorization experiments Examples include the bayes classifier. [5] Bayesian filters, employ the laws of mathematical probability to determine which messages are legitimate and which are spam. The Bayes theorem is –

$$P(S|M) = \frac{P(M|S) * P(M)}{P(S)}. \quad (1)$$

A simple example is used to illustrate its application in a Bayesian filter. Let S be the event "message is spam" and M be the event "message contains the token 'mortgage' ". Then, P(S|M) denotes the probability that a message which belongs to the historical data and which contains the token "mortgage" is categorized as spam. [1]

III. SUPPORT VECTOR MACHINE

Support Vector Machine was first heard in 1992, introduced by Boser, Guyon, and Vapnik in 1992. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. In another terms, SVM is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machines can be defined as systems which use hypothesis space of linear function in a high dimensional feature space. The formulation uses the Structural Risk Minimization (SRM) principle, which has been shown to be superior, to traditional Empirical Risk Minimization (ERM) principle, used by conventional neural networks. A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. SVM can be used to solve Linearly Separable as well as Non Linear Separable Problems. [6]

A. Linearly seperable problems

Let M m-dimensional training inputs x_i ($i = 1, \ldots, M$) belong to Class 1 or 2 and the associated labels be $y_i = 1$ for Class 1 and −1 for Class 2. If these data are linearly separable, the decision function is given by

$$y_i\,(w^T x_i + b) \geq 1 \quad \text{for } i = 1, \ldots, M. \quad (2)$$

The decision boundary can be found by solving the following constrained optimization problem

$$\text{Minimize } \frac{1}{2}\|W\|^2. \quad (2)$$

$$\text{Subject to } y_i\,(w^T x_i + b) \geq 1. \quad (3)$$

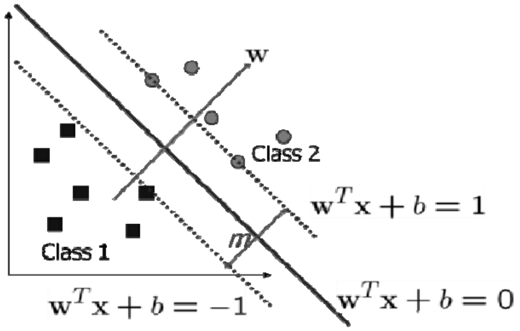


Fig - 2

B. *Non Linearly seperable problems*

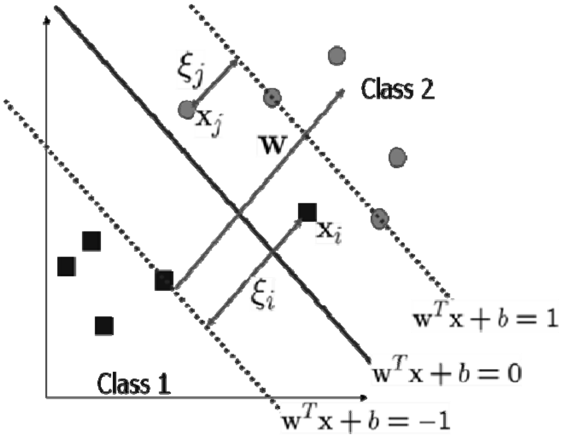


Fig – 3

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

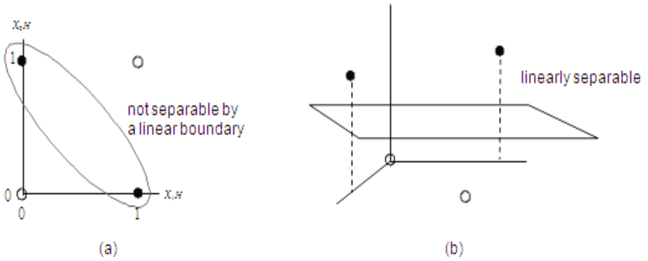


Fig – 4

In non linearly seperable problem we introduce a slack variable ξ. If error is between 0 ≤ ξ ≤ 1, then data can be properly classified, but if ξ ≥ 1 then the data is misclassified. So we should minimize ξ. The hyperplanes are computed as

$$y_i (w^T x_i + b) \geq 1 - \xi_i, \qquad \text{for } y_i = 1. \qquad (4)$$

$$y_i (w^T x_i + b) \leq -1 + \xi_i, \qquad \text{for } y_i = -1. \qquad (5)$$

the decision boundary can be found by the following optimization problem

$$\text{Minimize } \frac{1}{2} \|W\|^2 + C\sum_1^M \xi_i. \qquad (6)$$

$$\text{Subject to } y_i (w^T x_i + b) \geq 1 - \xi_i, \text{ for } \xi_i > 0. \qquad (7)$$

[7]

C. *Kernel Functions*

Training vectors xi are mapped into a higher (may be infinite) dimensional space by the function Φ. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimension space .C > 0 is the penality parameter of the error term.

Furthermore, K(xi , xj) ≡ Φ(xi)T Φ(xj) is called the kernel function. There are many kernel functions in SVM, so how to select a good kernel function is also a research issue. However, for general purposes, there are some popular predefined kernel functions:

Linear kernel: K (xi , xj) = xi ^{T} xj.
Polynomial kernel: K (xi , xj) = (α xi ^{T} xj + r)^d , γ > 0
Radial basis kernel: K (xi , xj) = exp(-γ ‖xi - xj‖2) , γ > 0
Sigmoid kernel: K (xi , xj) = tanh(γ xi ^{T} xj + r)
Here d and γ are kernel parameters. [8]

D. *Mercers condition*

Kernel functions must be continuous, symmetric, and most preferably should have a positive (semi-) definite Gram matrix. Kernels which are said to satisfy the Mercer's theorem are positive semi-definite, meaning their kernel matrices has no non-negative Eigen values. The use of a positive definite kernel insures that the optimization problem will be convex and solution will be unique. Since any positive semi definite function can be a kernel function, we can make a new kernel function.

1. K(x, y) = K1(x, y) + K2(x, y)
2. K(x, y) = cK1(x, y) + K2(x, y) for c Є R_+
3. K(x, y) = K1(x, y) + c for c Є R+
4. K(x, y) = K1(x, y).K2(x, y)
5. K(x, y) = (K1(x, y) + c)^d for d Є N
6. K(x, y) = exp(K1(x, y)/σ^2) for σ Є R
7. K(x, y) = exp(−(K1(x, x) − 2K1(x, y) + K1(y, y))/2σ^2)
8. K(x, y) = K1(x, y)/$\sqrt{K1(x,x)K1(y,y)}$. [9]

IV DATASET

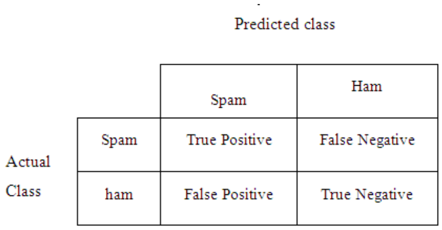
I have use the datasets available from the ECML-PKDD Discovery Challenge website. The training dataset is a general corpus containing 4000 e-mails collected from several users' inboxes. The evaluation datasets consist of three different users' inboxes each containing 2500 e-mails. These evaluation datasets are identified as Eval-00, Eval-01, and Eval-02. Each e-mail in the datasets is represented by a word (term) frequency vector. Each word in an e-mail is identified by an ID and its frequency count in the e-mail. An additional attribute identifies the label of the e-mail as either spam or non-spam.

The emails are in a bag-of-words vector space representation. Attributes are the term frequencies of the words. They removed words with less than four counts in the data set resulting in a dictionary size of about 150,000 words. The data set files are in the sparse data format used by SVM light. Each line represents one email, the first token in each line is the class label (+1=spam; -1=non-spam; 0=unlabeled-evaluation-data). The tokens following the label information are pairs of word IDs and term frequencies in ascending order of the word IDs

To give an example, the first line in task_a_labeled_train.tf starts like this:
1 9:3 94:1 109:1 163:1
This line represents a spam email (starting with class label "1") with four words. The word ID of the first token is 9 and the word occurs 3 times within this email, indicated by ":3".
Furthermore, all messages were already fully pre-processed and represented in SVM light format, so no specific additional pre-processing was possibly. As proper pre-processing can be essential for learning success, this setup somewhat limited the possibilities, but also ensured a level playing field for all learning approaches. Generally in text classification, and especially in Spam classification, smart preprocessing can greatly simplify the problem, e.g. one can include attributes representing meta-information like sender names and servers, or other header-fields, or make a distinction between text coming from the subject and text coming from the message body and so on. None of that was possible for the fully pre-processed data given here. [10]

A. Performance Criteria

- True positive rate (TP): fraction of spam e-mails correctly classified as spam
- False positive rate (FP): fraction of non-spam e-mails incorrectly classified as spam.
- Accuracy: fraction of all e-mails that are correctly classified.
- Precision: TP / (TP + FP).
- Recall: TP / (TP + FN), where false negative rate (FN) is the fraction of spam e-mails that are incorrectly classified as non-spam.
- AUC: area under the receiver operating characteristics (ROC) curve. ROC curve (Receiver Operating Characteristics) is a curve plotted between true positive rate and false positive rate. The area under this curve has the nice property that it specifies the probability that, when we draw one positive and one negative example at random, the decision function assigns a higher value to the positive than to the negative example. [11]



		Predicted class	
		Spam	Ham
Actual Class	Spam	True Positive	False Negative
	ham	False Positive	True Negative

Fig – 5

V EXPERIMENT AND RESULT

In this paper experiments are done on ECML-PKDD dataset with the predefined kernel functions in support vector machine, by varying parameter C (tradeoff between margin and error penalty). Then compared the results (AUC values) with the new kernel function called Cauchy kernel and it is found that the cauchy kernel gives better result than the predefined kernel functions in SVM. Cauchy kernel comes from the Cauchy distribution. It is a long-tailed kernel and can be used to give long-range influence and sensitivity over the high dimension space. [12]

$$K\,(xi,\,xj) = \frac{1}{1+\frac{\|xi - xj\|^2}{\sigma}} \tag{8}$$

Table I AUC values for linear kernel function

Linear	C=0.001	C=0.01	C=0.1	C=1.0	C=10.0	C=100.0
Eval00	0.68855	0.71246	0.70128	0.72545	0.72545	0.72544
Eval01	0.48515	0.48279	0.48213	0.48199	0.48199	0.48199
Eval02	0.48817	0.48753	0.48620	0.48628	0.48628	0.48626

Table II AUC values for polynomial kernel function

Poly	C=0.001	C=0.01	C=0.1	C=1.0	C=10.0	C=100.0
Eval00	0.58522	0.58625	0.58638	0.58638	0.58638	0.58638
Eval01	0.48472	0.48227	0.48221	0.48221	0.48221	0.48221
Eval02	0.49122	0.49237	0.49236	0.49236	0.49236	0.49236

Table III AUC values for rbf kernel function

Rbf	C=0.001	C=0.01	C=0.1	C=1.0	C=10.0	C=100.0
Eval00	0.49836	0.49796	0.49798	0.49483	0.49483	0.49483
Eval01	0.50398	0.50399	0.50359	0.50753	0.50753	0.50753
Eval02	0.49959	0.50038	0.49959	0.50038	0.50038	0.50038

Table IV AUC values for sigmoid kernel function

sigmoid	C=0.001	C=0.01	C=0.1	C=1.0	C=10.0	C=100.0
Eval00	0.49002	0.63000	0.49114	0.68211	0.62424	0.51364
Eval01	0.38456	0.52311	0.38595	0.53775	0.49567	0.38307

Eval02	0.36458	0.52931	0.36394	0.57504	0.48980	0.34787

Table V AUC values for Cauchy kernel function

cauchy	**C=0.001**	**C=0.01**	**C=0.1**	**C=1.0**	**C=10.0**	**C=100.0**
Eval00	0.68864	0.68864	0.70705	0.72343	0.72343	0.72343
Eval01	0.75743	0.75743	0.76565	0.77703	0.77703	0.77703
Eval02	0.85816	0.85816	0.89118	0.88932	0.88932	0.88932

VI CONCLUSION.

Table VI Best AUC values for all kernel functions for eval01 dataset

	Lin	**Poly**	**Rbf**	**Sig**	**Cauchy**
	C = 10.0	**C =0.01**	**C= 0.001**	**C = 1.0**	**C = 10.0**
Eval00	0.72545	0.58638	0.49836	0.68211	0.72343

Table VII Best AUC values for all kernel functions for eval02 dataset

	Lin	**Poly**	**Rbf**	**Sig**	**Cauchy**
	C =0.001	**C = 0.001**	**C = 1.0**	**C = 1.0**	**C =1.0**
Eval01	0.48515	0.48472	0.50753	0.53775	0.77703

Table VIII Best AUC values for all kernel functions for eval03 dataset

	Lin	**Poly**	**Rbf**	**Sig**	**Cauchy**
	C =0.001	**C = 0.01**	**C =1.0**	**C =1.0**	**C =0.1**
Eval02	0.48817	0.49237	0.50038	0.57504	0.89118

Experiments shows that the Cauchy kernel function gives better AUC values for eval02 and eval03 dataset. And almost equal AUC values for eval01 dataset. So we conclude that the new Cauchy kernel gives better AUC values than the predefined kernel functions in support vector machine.

REFERENCES

[1] Guido Schryen, “Anti-Spam Measures Analysis and Design”, Springer Berlin Heidelberg New York, ISBN 978-3-540-71748-5.

[2] Izabella Miszalska, Wojciech Zabierowski, Andrzej Napieralski, “Article Selected Methods of Spam Filtering in Email”, CADSM 2007, Publisher, Polyana, UKRAINE, February 20-24, pp. 1-6.

[3] Jiawei Han, Micheline Kamber, “Data Mining Concepts and Techniques”, second edition , Morgan Kaufmann publisher.

[4] Thorsten Joachims, “Text Categorization with Support Vector Machine: Learning with many releavant features”.

[5] James Tin-Yau Kwok, “ Automated Text Categorization using Support Vector Machine”, Hong Kong Baptist University.

[6] Vikramaditya Jakkula, “Tutorial on Support Vector Machine (SVM)”, School of EECS, Washington State University, Pullman 99164.

[7] Shigeo Abe, “ Support Vector Machines for pattern Classifications ”, Springer-Verlag London Limited.2005.

[8] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, “A Practical Guide to Support Vector Classification*”,*Department of Computer Science, National Tiwan University, Taipei 106, Taiwan. of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[9] Kazushi Ikeda, ”Effects of Kernel Function on Nu Support Vector Machines in Extreme Cases”, IEEE Transactins on Neural Networks, VOL. 17, NO. 1.2006.

[10] http://www.ecmlpkdd2006.org/challenge.html.

[11] Ian H Witten, Eibe Frank, “ Data Mining Practical Machine Learning Tools and Techniques ”, second edition, Morgan Kaufmann publisher.

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