2016 4th International Symposium on Computational and Business Intelligence

A Study of Machine Learning Classifiers for Spam Detection

Shrawan Kumar Trivedi
BML Munjal University
Gurgaon, Haryana, India
e-mail: shrawan.trivedi@bml.edu.in

Abstract—**In the present world, there is a need of emails communication but unsolicited emails hamper such communications. The present research emphasises to build a spam classification model with/without the use of ensemble of classifiers methods have been incorporated. Through this study, the aim is to distinguish between ham emails and spam emails by making an efficient and sensitive classification model that gives good accuracy with low false positive rate. Greedy Stepwise feature search method has been incorporated for searching informative feature of the Enron email dataset. The comparison has been done among different machine learning classifiers (such as Bayesian, Naïve Bayes, SVM (support vector machine), J48 (decision tree), Bayesian with Adaboost, Naïve Bayes with Adaboost). The concerned classifiers are tested and evaluated on metric (such as F-measure (accuracy), False Positive Rate, and training time). By analysing all these aspects in their entirety, it has been found that SVM is the best classifier to be used. It has the high accuracy and the low false positive rate. However, training time of SVM to build the model is high, but as the results on other parameters are positive, the time does not pose such an issue.**

Keywords-machine learning classifier, naïve bayes, SVM, J48, ada boost, spam classification

I. Introduction

Presently, email is a crucial and necessary tool for enabling rapid and cheap communication. It is a widespread medium and an essential part of the life [1]. However, spam or unsolicited email is an inconvenience in this form of communication. They can be in the form of advertisements or similar explicit content which may contain malicious code embedded in them. A study has estimated that 70 percent of the total business emails are spam [2]. It has also been found that rapid growth of spam email has consequences like over-flowing of users' mailboxes, consumption of bandwidth and storage space, compromising of important emails and problems to the user in terms of consuming their time to sort through all emails [3]. Currently, there is growing interest in the field of spam classification because of the complexity that has been introduced by the spammers making it difficult to distinguish between spam mails (unsolicited emails) and ham mails (legitimate emails). The complexity can be due to attacks like Tokenisation (modifying or splitting a feature such as 'free' written as f r 3 3) and Obfuscation (hiding certain features by adding HTML or some other codes such as 'free' coded as FR3E or fr&#101xe), which are used to alter the information on particular features [4].

For spam classification, various spam filtering methods are used. The function of a spam filter is to identify spam email and prevents it from going to the mailbox. With the help of filters, the adverse impact of spam email is mitigated and operates like a predictable and reliable tool to eliminate unwanted emails. However, there exists a small risk of misclassification or removal of legitimate emails. In the research, various spam filtering methods have been tested but none were found to be perfect. However, filters are beneficial for an email recipient who has to go through the burden of detecting spam. The costs which are incurred in reading every email for identifying spam involves more than just time consumption [5]. It is now becoming difficult for a user to distinguish between emails merely by reading the subject or the email content, thereby increasing the necessity of spam filter. Rarely but sometimes, filters also make mistakes but are used in conjunction with users to minimize these errors.

II. Related Work

Spam classification has become a challenging area to conduct research as spammers are finding ways to change the information relating to spam words through the addition of complexities. Various machine learning classifiers have been used in the research to tackle such problems. This section talks about the literature description related to this research.

Bayesian methods are increasing in popularity and their use in text and spam classification applications are increasing. It provides benefits in a cost sensitive evaluation as it is capable of providing a higher degree of confidence relating to the classification. For the research of Sahami et al. [6], a naive bayes classifier and a bag of word representation for email dataset have been used. In this paper, an improved level of performance has been shown with the addition of features that are complex (such as FREE!, f r 3 3), domain name features and some non-alphabetic features in the bag of feature.

An ensemble of classifiers approach is also within the scope of the research. Sakkis et al. [7] has developed a combined approach for designing a better ensemble of classifiers. This design includes NB and *k*NN classifiers to achieve good classification results. In a recent study by Trivedi and Dey [8] they used ensemble of classifier techniques with respect to probabilistic classifiers and found that this technique helped in improving the performance of probabilistic classifiers even with a small subset of informative features.

Support Vector Machine (SVM) is also popular in this domain. Drucker et al. [9] have carried out a comparative study of SVM with various machine learning classifiers and

978-1-5090-3488-8/16/$31.00 ©2016 IEEE

have identified that SVM and boosted decision tree are the classifiers which are promising in terms of speed as well as accuracy. However, the training speed of SVM was faster than for the Boosted decision tree. Trivedi and Dey [10] tested the effect of various kernel functions on the performance of SVM and found that the best way to enhance the learning capability of SVM was through Normalized Poly Kernel.

III. Structure of the Spam Classifier

Enron email corpus [11]-[13,] has been used in this study. Concerned machine learning classifiers are tested on Enron corpus with the help of F-value, false positive rate and training time.

Enron email corpus is the main focus of this study for testing concerned classifiers where 6000 email files are selected with a 50% spam rate from Enron version 5& 6. The rationale behind this selection is that the complexities which are present in the form of attacks that have been identified in the features/words of email files of these versions, help in evaluating the actual strength of classifiers.

Pre-Processing of the Corpus

The information of an Email file is divided into Header (general information such as subject, sender and recipient) and Body (main content of the email). However, this research focuses only on the body of the email.

Pre-Processing is performed on the selected corpuses to transform the email files where strings of characters are transformed to a representation suitable to the classification algorithm. The words/features are extracted from the email files through the feature extraction process.

In the process, the information from email files is extracted by string-to-word-vector transformation method for developing an associated feature dictionary. It includes, HTML (or other) tags removal, stop-word removal (some words which appear frequently such as prepositions, articles and conjunctions etc.) and lemmatization (reducing words to their basic form, e.g. from improving to improve). Selected feature set is further taken for dimensionality reduction.

A major problem in spam classification is high dimensionality of the feature space where one dimension of a unique word is found in various files. This large set of the feature space leads to difficulty in standard classification methods as a result of unreliable classification results and computation cost. This large feature space is reduced through dimensionality reduction method and done through the process of feature selection.

Feature selection is used to obtain the informative feature subset to reduce the feature space. This research incorporates greedy Stepwise subset evaluation method [12, 13] to obtain informative feature subset.

Greedy stepwise feature subset search:

In this process, features are evaluated iteratively to search a single informative feature to add in the model. A stepwise regression method is used for the evaluation process. Feature selection is done through three different process i.e. forward selection (addition of good features), backward selection (deletion of worst features) and mixed selection (forward and backward selection simultaneously). Some methods such as P-value are offered to decide termination process which informs, whether all the good features have been added or not in the model or none of the feature is left which can add value.

Let us assume f^s feature set is taken for search process and f^e is the number of features which participate for evaluation according to their fitness. Hence, best feature set can be produced with the following equation.

$$f_*^b = \arg\max_{f^e \notin f^s} fit(f^s \cup \{f^e\}) \tag{1}$$

After getting the informative features subset, feature representation process is applied where words/features of email files are represented by binary representation method. In this method, email files and words together form a binary matrix called Term-Document Matrix (*TDM*) and this method is referred to as the term weighting method. This binary matrix contains binary values (1 and 0), where 1 is an indicator of the presence of the particular feature/word in a specific email file and 0 is for otherwise.

Machine Learning Classifier of this study:

Bayesian Classifiers

The Bayesian classifier was proposed by Lewis in 1998 [14]. He introduced the term $P\left(\frac{c_i}{d_j}\right)$ which is defined as the probability of a document recognized by a vector $d_j = w_j^1, w_j^2, ..., w_j^n$ of terms falling within a particular category c_i . This probability is calculated by the Bayes theorem -

$$P\left(\frac{c_i}{d_j}\right) = \frac{P(c_i) * P\left(\frac{d_j}{c_i}\right)}{P(d_j)} \tag{2}$$

where, $P(d_j)$ symbolizes the probability of arbitrarily selected documents represented by the documents vector d_j and $P(c_i)$ is the probability of arbitrarily selected documents d_j falling in a particular class c_i . This classification method is usually known as “Bayesian Classification”.

Bayesian method is popular but is seen as challenging in the case of high dimensional data vector d_j . This challenge is well accepted and tackled by an assumption that any two arbitrarily selected coordinates of document vector d_j (tokens) will be kept independent to each other. This assumption is well described by the given equation -

$$P\left(\frac{d_j}{c_i}\right) = \prod_{l=1}^{n} P\left(\frac{w_j^l}{c_i}\right) \tag{3}$$

This assumption is captured by the classifier named “Naïve Bayes” which is a well-known classifier in the Text Mining domain.

Support Vector Machine (SVM):

Support Vector Machine (SVM) is a well-known classifier in the Text classification research. This classifier works by taking the concept of “statistical learning theory and structural maximization principal” [9]. It is a popular and well accepted classifier in the research due to its strength to deal with high dimensional data by the help of distinctive Kernel function.

SVM works to separate classes (Positive/Negative) with the help of maximum margin created by hyper-plane. Let us assume a training set $X = \left\{ x^i, y^i \right\}$, where $x^i \in R^n$ and $y^i \in \left\{ +1, -1 \right\}$, which indicates the a unique class for i^{th} training sample. This research takes +1 as Spam or unsolicited class and −1 as Ham or legitimate class. The final classifiers output is measured by the following equation:

$$y = w.x - b \qquad (4)$$

where y is the final classifier output, w indicates normal vector comparable to those in the feature vector x , and b is known as bias parameter which is decided by the training process. The separation between the classes is maximized by the equations given bellow:

$$\text{minimize} \qquad \tfrac{1}{2}\left\| w \right\|^2 \qquad (5)$$

$$\text{subject to} \qquad y^i \left(w.x - b \right) \geq 1, \forall i \,. \qquad (6)$$

Decision Tree (J48):

J48 [12, 13] is a simple decision tree classification technique which is based on C4.5 algorithm. It is a open source JAVA implementation of C4.5 algorithm which uses the concept of Entropy for developing a decision tree with the help of training data. At the each node, C4.5 selects a most informative feature from features subset. The selection mechanism is done by normalising the information gain (i.e. difference of entropy). Some of the base cases are necessary to understand for C4.5 algorithm

I. If entire sample of the list belongs to the same class, it creates a leaf node in the decision tree to opt that class.

II. If none of the features provide information gain, it develops a decision node, placed at higher up the tree with the help of expected value of the class.

III. If the instance of earlier unseen class comes upon, again it develops a decision node, situated at higher up the tree with the help of expected value of the class.

Algorithm for C4.5:

(a) *Check the above base cases*

(b) *For each feature x^i , find normalise information gain from splitting capability on x^i .*

(c) *If the x_b^i is a best feature with higher normalise gain, develop a decision node that split on x_b^i .*

(d) *Repeat above on the sub lists created by splitting on x_b^i .*

Boosting with AdaBoost

The basis of the boosting methods [8] is bootstrapping. The fundamental principle of bootstrapping is to re-assess the accuracy of some estimate. It is a statistical sample based method that consists of drawing randomly with replacement from a data set. In the classification domain, some of the boosting methods have shown their potential in terms of strengthen the classifiers’ accuracy.

Adaptive boosting works to re-weight the data rather random sampling. This method develops a concept of building ensembles for performance improvement of the classifiers. AdaBoost learns with the collection of output M_x of weak classifiers $G_t^{m_x}$ and then predict decision, which forms the final classifier G_t^x .

Algorithm for boosting classifiers

Input: Training set $T_r = t_{1,} t_2, t_3 ... t_n$ with $t_i = \left(x^i, y^i \right)$.
Number of sample version of training set B .
Output: An appropriate classifier for the training set G_t^x

Initialise the weights $w_i^t = \frac{1}{N}, \qquad i \in \left\{ 1, 2, 3, ... N \right\}$

From $m = 1, 2, 3, ... M_x$

a) *Train the weak classifier $G_t^{m_x}$ with the training outset using weights w_i^t*

b) *Calculate the error term* $E_{rror}^m = \frac{\sum_{i=1}^N w_i^t I(\mathrm{y}_i \neq G_t^{m_{xi}})}{\sum_{i=1}^N w_i^t}$

c) *Calculate weight contribution* $\theta_m = 0.5 \log\left(\frac{1 - E_{rror}^m}{E_{rror}^m} \right)$

d) *Substitute* $w_i^t \leftarrow w_i^t Exp\left(-\theta_{(m)} I\left(y_i \neq G_t^{m_{xi}} \right) \right)$ *and Renormalize* $\sum_i w_i^t = 1$

The final classifier is :–

$$G_t^x = \theta_m sign\left(\sum_{m=1}^{M_x} G_t^{m_x} \right) \qquad (7)$$

IV. Experiment Design

Instruments for Evaluation

Three measures are used for evaluating the performance: F-value and False Positive rate and Training Time.

F-Value

F-measure is a measure of the accuracy of a test. It takes into consideration both precision *(P)* and recall (*R)* to compute the score. Precision *(P)* is the number of correct positive results divided by the number of all positive results, and recall (R) is the number of correct positive results divided by the number of positive results that should have been returned. The final value is interpreted as a weighted average of the precision and recall. The accuracy is highest when the value is 1 (100%) and lowest when the value is 0 (0%). Mathematically, it can be calculated using the following formula:

$$F^{Value}_{H,S} = \frac{2 * P^{precision}_{H,S} * R^{recall}_{H,S}}{P^{precision}_{H,S} + R^{recall}_{H,S}} \quad (8)$$

False Positive Rate

FP rate is said to be the probability of falsely rejecting a null hypothesis study. Due to this, this value should be as low as possible as it refers misclassification of the model, which is something one should try and avoid. Mathematically, false positive rate is calculated using the following formula:

$$FP_{rate} = \frac{H^{mis}_{am}}{H^{mis}_{am} + H^{correct}_{am}} \quad (9)$$

where H^{mis}_{am} is denoted as total misclassified Ham Emails and $H^{correct}_{am}$ represented as total correctly classified Ham Emails.

Software:

Microsoft Excel 2010 and JAVA based implementation on the WINDOW 7 operating system with 4GB RAM has been preferred for this study. After Pre-processing (feature extraction and feature search), 49 most informative features for Enron corpus are selected for training of the classifiers. Whole corpus is split to obtain 66% files for training and 34% for testing.

V. Results and Analysis

The analysis section is divided in three segments. The first segment demonstrates F-Value (Table. 1 & Fig. 1) of all concerned machine learning classifiers. Second part presents analysis of False Positive (FP) Rate (Table 1 & Fig. 2) of the classifiers. In the last segment, all the classifiers are tested with the help of training time (Table 1 & Fig. 3).

The following results have been obtained from different metrics.

Table 1. Evaluation Results of Classifiers

	FP rate	**F-measure**	**Training Time**
Bayes	8.1%	92.0%	0.51 sec
NaiveBayes	6.9%	92.8%	**0.25 sec**
SVM	**6.5%**	**93.3%**	10.54 sec
J48	7.6%	92.1%	4.73 sec
With Boosting			
BayesNet	8.1%	92.1%	7.18 sec
NaiveBayes	**6.5%**	93.2%	17.35 sec

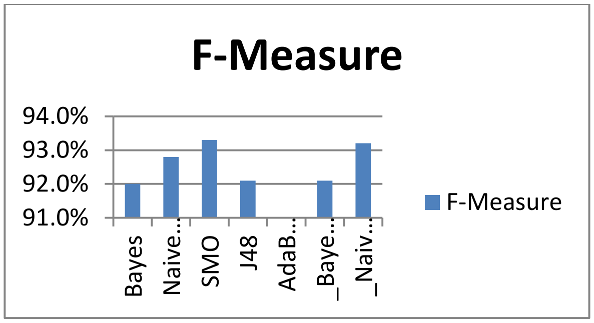

Figure 1. F-Measure of Classifiers

From the Table 1 and based on the values obtained for the different classifiers, it is seen that SVM has given the highest F measure value. This indicates that the accuracy is highest for this method. It is also seen that NaiveBayes classifier is also quite effective and gives an even more positive result when used though the AdaBoost classifier. Bayes has the lowest accuracy as compared to other classifiers, which makes it the least desirable in terms of this aspect.

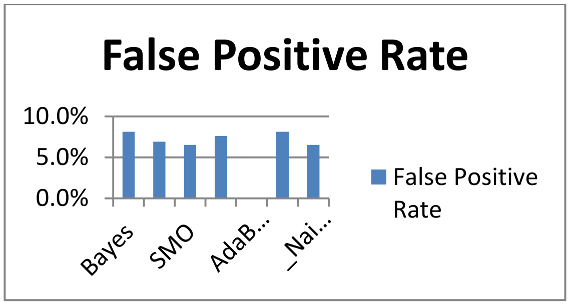

Figure 2. False Positive Rate of Classifiers

In this study, FP rate is less for SVM and Naïve Bayes with boosting and it is 6.5%. This indicates that the fall-out rate is the lowest for these methods, which makes these classifiers desirable in this study.

Bayesian Classifier has given the FP rate of 8.1% that is higher than other methods. This rate remains the same even when Bayesian classifier is used with AdaBoost classifier.

Time taken to build the model:

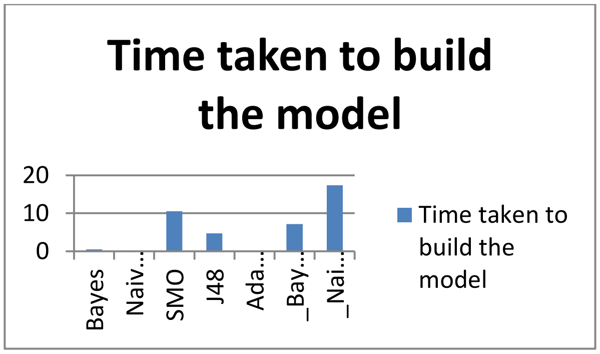

Figure 3. Training Time of Classifiers

This refers to the time taken to build the model. The time taken to formulate the model in NaiveBayes was less, i.e. 0.25 second, while NaiveBayes through AdaBoost took the longest time of more than 17 seconds. SMO, too took a long time of 10.54 seconds to give the results, which is also quite high.

Looking at all the parameters collectively, it can be said that SVM is the best classifier to be used. This is because even though the time taken to build the model is more than that for the other classifiers, the FP rate is the lowest and the F-measure is the highest. Therefore, due to the desirable results on other parameters, even if it takes a little longer to build the model, it is acceptable.

VI. Conclusion

It is important that spam mails do not reach the inbox of the users as this reduces efficiency of operations. But more importantly it is necessary that no ham mail goes to the spam folder as those lead serious problems to the user.As the analysis and results section has revealed, looking at all the parameters collectively, it is found that SVM is the best classifier of this study. The False Positive Rate (FP Rate) is the lowest for this classifier and the F-measure, which represents the accuracy, is highest for SVM. Due to these reasons, the SVM classifier should be used as it will yield positive and desirable results. It will help in classifying the spam and ham mails in their respective folders. On the other hand, it is also seen that ensemble of classifiers technique helps in learning capability of the classifiers.The time taken to build the model for the best classifier found for this study, which is SVM, is high. However, as already established, even though the time taken to build the model is more, the results are more effective and thus even if it takes a little longer to achieve more desirable results, it is acceptable.

References

[1]. Whittaker, S., Bellotti, V., & Moody, P. (2005). Introduction to this special issue on revisiting and reinventing e-mail. Human-Computer Interaction, 20(1), 1-9.

[2]. Aladdin Knowledge Systems, Anti-spam white paper, <http://www.eAladdin.com>.

[3]. Lai, C. C. (2007). An empirical study of three machine learning methods for spam filtering. Knowledge Based Systems, 20(3), 249-254

[4]. Goodman, J., Cormack, G. V., & Heckerman, D. (2007). Spam and the ongoing battle for the inbox. Communications of the ACM, 50(2), 24-33.

[5]. Caliendo, M., Clement, M., Papies, D., & Scheel-Kopeinig, S. (2008). The cost impact of spam filters: Measuring the effect of information system technologies in organisations.

[6]. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105).

[7]. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2001). Stacking classifiers for anti-spam filtering of e-mail. arXiv preprint cs/0106040.

[8]. Trivedi, S. K., & Dey, S. (2013). Interplay between Probabilistic Classifiers and Boosting Algorithms for Detecting Complex Unsolicited Emails. Journal of Advances in Computer Networks, 1(2).

[9]. Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. Neural Networks, IEEE Transactions on, 10(5), 1048-1054.

[10]. Trivedi, S. K., & Dey, S. (2013). Effect of Various Kernels and Feature Selection Methods on SVM Performance for Detecting Email Spams." *International Journal of Computer Applications* 66.21 (2013)

[11]. Trivedi, S. K., & Dey, S. (2013, December). An Enhanced Genetic Programming Approach for Detecting Unsolicited Emails. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*(pp. 1153-1160). IEEE.

[12]. Trivedi, S. K., & Dey, S. (2014, October). A study of ensemble based evolutionary classifiers for detecting unsolicited emails. In *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems* (pp. 46-51). ACM.

[13]. Trivedi, S. K., & Dey, S. (2014). Interaction between feature subset selection techniques and machine learning classifiers for detecting unsolicited emails.*ACM SIGAPP Applied Computing Review*, *14*(1), 53-61.

[14]. Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In Machine learning: ECML-98 (pp. 4-15). Springer Berlin Heidelberg.