ISSN 1512-1232

NAIVE BAYESIAN CLASSIFIER AND PCA FOR WEB LINK SPAM DETECTION

Dr.S.K.Jayanthi[1], S.Sasikala[2]

[1]Head, Asso.Prof., Department of Computer Science,
Vellalar College for Women, Erode-12, India
E-mail: jayanthiskp@gmail.com

[2]Asst.Prof., Department of Computer Science,
KSR College of Arts and Science, Tiruchengode-637211, India
E-mail: sasi_sss123@rediff.com

Abstract

WWW is a huge information space with rapid growth. Web spam is a bad method which deceives the search engine results. Combating spamdexing is a tough task because spammers change the techniques day by day. Analyzing the properties of websites, so called features and applying classifiers differentiate the spam and nonspam sites. This paper applies the naive bayes algorithm for website features and classifies whether the given website is a spam or nonspam. WEBSPAM-UK-2007 link based features dataset is taken as base dataset. It has 44 features. It is preprocessed by applying PCA and important 10 features are selected. The naïve bayes classifier is trained. User interface is created to obtain the features of the test data. Later the test data obtained through the user interface is converted into CSV (comma separated values) file and fed into the classifier for the class determination. Results are discussed. Naive Bayesian classification seems to perform well as shown through experiments.

Keyword

Web Link Spam, Classification, naive Bayesian, Search Engine

1. INTRODUCTION

World Wide Web (WWW) is a massive collection of interlinked hypertext documents known as web pages. Users access the WWW content through internet. WWW size tends to show exponential growth. Size of the WWW becomes many folds in recent times, now it contains 2.18 billion web pages comprising 80 billion publicly accessible web documents distributed all over the world on thousands of web servers.

Searching information in such a huge collection of web pages is a difficult process. The content is not organized like books on shelves in a library and web pages are not completely catalogued at one central location. Distinguishing between desirable and undesirable content in such a system presents a significant challenge. Retrieving required information from the web needs the information retrieval system. Search engine is one such application. It is a program which retrieves the relevant information from the web with content relevancy and link trustworthiness.

Search Engine Optimization (SEO) is performed in websites to achieve the top ranks in Search Engine Results Page (SERP). SEO process is classified into two types: White-hat and Black-hat. White-hat SEO is the process of improving the website visibility, rank, reputation and user visits by improving the website content quality. Black-hat SEO is the process of improving the aforesaid website parameters by cheating the search engine ranking algorithm.

Obtaining a higher rank is strongly correlated with traffic, and it often translates as high revenue to the website owners. Spamming the web is cheap, and in many cases, it is successful. For manipulating the ranking metrics it employs two major types of techniques: Content-based Spamdexing and Link-based Spamdexing.

Spamdexing creates a bad impact on the search engine. Email spam evolves at first, followed by search engine content spam. Once it has been controlled, next category of spam arises. Symantec releases the following key findings in 2013 Internet Security Threat Report: Web-based attacks increased 30%, Targeted attacks raised in 2012 as 42%, 31% of all targeted attacks aimed at businesses with less than 250 employees. One specific attack infected 500 organizations in a single day and a single threat infected 600,000 Macs in 2012. The number of phishing and spoofing social networking sites increased 125%. Web attacks blocked per day at 2011 is 190,370 in average and in 2012 it increases to 247,350. New unique web domains identified in 2010 is 43,000 and in 2011 is 57,000 and it is raised to 74,000 [1].

Symantec intelligence report released in August 2013 states that: The global spam rate is 65.2 % in August 2013. The top-level domain (TLD) of Poland, .pl, has topped the list of malicious. Sex/Dating spam continues to be the most common category, at 70.4%. Weight loss spam comes in second at 12.3% [1].

Addressing web spam is an important issue right now as witnessed from the reports. Researchers proposed many methods for combating the spamdexing. Machine learning techniques are proved to effective in spam classification for over a long while. This paper addresses the problem of the link spamdexing with the 10 new features and naïve Bayesian classifier. Working method adopted in this paper is portrayed in Fig. 1.

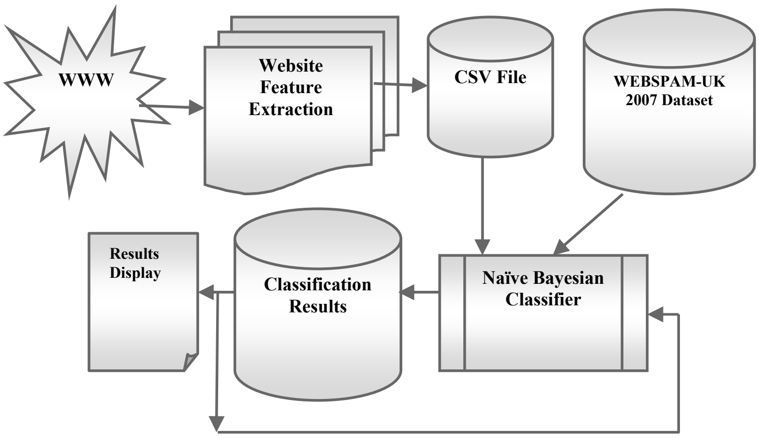


Figure 1. Working Method of the proposed system

2. RELATED WORK

Researchers proposed many methods for combating the spamdexing. Machine learning techniques are proved to effective in spam classification for over a long while. Standard datasets used in existing literature are WEBSPAM-UK datasets (Content and link features), Clueweb Datasets (Content features) and TREC Datasets (Content features). Some authors propose their own datasets crawled and compiled from publicly available sources such as Dmoz and yahoo directories. This paper utilizes the link based dataset of WEBSPAM-UK 2007. In addition manually collected features based on recent SEO advances are incorporated. Summary of machine learning techniques proposed by various researchers along with the description are offered in table 1.

Table 1 Summary of methods using Machine Learning Techniques

	Author	Proposed	Task Carried Out	Mode	Based on	Spam Type	Machine Learning Technique	Dataset
1	Egele et al. 2009 [2]	New Features and Eight classification Techniques	Feature exploration/ applied in SERP. New dataset in existing methods	Offline	Naive Bayesian, Fuzzy lattice reasoning, SVM-SMO, J48, Best-first decision tree, Locally weighted learning, Conjunctive Rule and Clustering	Link Spam	Classification and Clustering	Manually classified dataset
2	Chung et al. 2010[3]	Spam link GeneratorIdentification	New Features/ applied in SER	Online	Online Learning Algorithm	Link Spam	Classification	Three yearly snapshots of Japanese Web archive
3	Erdelyi et al. 2011[4]	Ensemble based methods	New methods on existing dataset	Offline	Bagged LogitBoost, J48, Bagged Cost-sensitive Decision Trees, Naive Bayes, Logistic Regression, and RandomForests	Link Spam	Classification	WEBSPAM-UK2007 and the ECML/ PKDD DC2010 dataset
4	Tian et al.[5]	Semi supervised machine learning	New features in existing dataset and methods	Offline	ADTree, SMO and Bayes	Link Spam	Pre processing/ Classification	ECML/PKDD 2007
5	Silva et al. 2012[6]	Classification models (Neural networks, bagging and boosting)	New methods in existing dataset and methods	Offline	Multilayer perceptron in neural networks, SVM, J48, random forest, bagging/adaptive boosting of trees, and k-nearest neighbor	Link Spam	Classification	WEBSPAM-UK2007
6	Karimpour et al. 2012[7]	Impact of feature selection	Feature selection and classification in existing dataset	Offline	Feature selection (imperialist competitive algorithm and genetic algorithm)/ Classification(SVM, Bayesian network and Decision trees)	Link Spam	Pre processing, Feature Selection and Classification	WEBSPAM-UK2007
7	Geng et al. 2008[8]	Re-extracted features (spamicity, clustering, propagation and neighbor details)	New method and features on existing dataset	Offline	Stack graph learning (Sgl)	Link Spam	Preprocessing and Classification	WEBSPAM-UK 2006
8	Benczur et al. 2007[9]	New Features (OCI, MindSet, Adwords, google Adsense and Pagecost)	New Features on existing dataset	Offline	J48 Decision Tree	Link Spam	Pre processing and Classification	WEBSPAM-UK 2006
9	Gan and Suel 2007[10]	Re-labeling two-stage approach and heuristics usage	New dataset and features used in existing methods	Offline	J48 Decision Tree and SVM	Link+ Content Spam	Pre processing and Classification	Manually classified dataset (Swiss web sites crawled using the PolyBot crawler)
10	Castillo et al.[11]	Notion of spamicity and unsupervised classification	New features in existing dataset and method	Online/ Offline	J48 Decision Trees	Link+ Content Spam	Preprocessing and Classification	WEBSPAM-UK 2006
11	Jayanthi. S.K., Sasikala.S [12]	GAB_CLIQDET: Genetic algorithm for Spam sites classification	New methods in existing dataset and methods	Offline	Genetic Algorithm	Link Spam	Classification	WEBSPAM-UK 2006
12	Jayanthi. S.K., Sasikala.S [13]	Perceiving LinkSpam based on DBSpamClust: spam page	New methods in existing dataset and methods	offline	Fuzzy C-means Clustering	Link Spam	Clustering	Own dataset

3. PROBLEM DESCRIPTION

3.1 Spamdexing and Naive Bayes

Spamdexing subvert the search engine results through manipulating the content, link or meta tags of a website. Content spamdexing is achieved through the interpretation of the title text, anchor text or body text of a webpage. One example is stuffing a popular keyword in any part of webpage. Link spamdexing refers manipulation of the links (inlinks and outlinks). Thus spamdexing of a website W is referred as:

$$\mathrm{Spam}(W) = \forall_{WP \in W} \left(\sum_{i=1}^{N} CS'(W) + \sum_{i=1}^{N} LS'(W) + \sum_{i=1}^{N} MS'(W)\right. . \tag{1}$$

Where wp-webpages in a particular website W, n -number of pages, CS' – content spammed, LS'-link spammed, MS'-meta spammed. Naive bayes theorem is a classifier based on the bayes theorem with strong independence assumptions. Web features pertaining to link of a website is extracted with an user interface and it is converted into a CSV file. The CSV file is fed into the naïve bayes classifier. The link spam is detected with the help of the feature inference as shown in Eqn. 2.

$$LS'(W) \rightarrow (\text{naive bayes}(C_{\text{Spam/Nonspam}} \mid F_{1......}F_N)) . \tag{2}$$

4. METHODS AND MATERIALS

4.1 Naive Bayesian Considerations

4.1.1 Bayes Theorem

Let B_1,B_2,… B_n be an exhaustive and mutually exclusive events and A be a related event to Bi. Now consider the equation 3.

$$P(B_i/A) = \frac{P(B_i)\ P(\frac{A}{B_i})}{\sum_{i=1}^{N} P(B_i)\ P(\frac{A}{B_i})} \equiv \text{posterior} = \frac{\text{prior}\times\text{likelihood}}{\text{evidence}} \tag{3}$$

Naïve bayes theorem is a probabilistic classifier based on the bayes theorem with independence assumptions. Each and every feature presence or absence doesn't reflect any change in other feature based on the naïve bayeisan theorem [15]. Each feature is independent and not related to other one. Here the link based features of the website is used. All features will be independently inferred and as a result the best discriminate probability for each feature could be obtained. It easily classifies spam website from the genuine one. By conditional probability, the classifier is denoted as:

$$P(C|F_1,F_2,..F_N) \tag{4}$$

where C (spam/nonspam) is the class and F is the features ranging from 1 to N.

4.2 Parameter Estimation

The model parameters is approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class prior is calculated by assuming equiprobable classes or by calculating an estimate for the class probability from the training set as in Eqn 5.

$$\text{priors} = \frac{1}{\text{number of classes}}$$

$$\text{prior for a given class} = \frac{\text{number of samples in the class}}{\text{total number of samples}} \quad (5)$$

The training data contain a continuous attribute, say x. Segment the data by the class and then compute the mean and variance of x in each class. Let $\mu_c$be the mean of the values in x associated with class c, and let σ_c^2 be the variance of the values in x associated with class c. Then, the probability of some value given a class, $P(x = v|c)$, can be computed by plugging $v$into the equation for a Normal distribution parameterized by $\mu_c$and σ_c^2. That is,

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}} \quad (6)$$

4.3 PREPARATION OF DATASET - COMPUTING PRINCIPAL COMPONENTS

4.2.1 Iterative Computation

The pseudocode for finding the principal component is given in this section. For a data matrix X^T with zero mean, without ever computing its covariance matrix [16].

```
Pseudocode: PCA
P= a random vector
do c times:
    t=0; (a vector of length m)
    for each row $\mathbf{x} \in \mathbf{X}^T$
        $\mathbf{t} = \mathbf{t} + (\mathbf{x} \cdot \mathbf{p})\mathbf{x}$
    $\mathbf{p} = \frac{\mathbf{t}}{|\mathbf{t}|}$
return p
```

Subsequent principal components can be computed by subtracting component p from X^T and then repeating this algorithm to find the next principal component. This is how the process is repeated. Initially 44 link based features of the website are given into PCA and after 2-fold validation 10 features are obtained. The 10 features are used for training the naïve Bayesian classifier. The settings used for the PCA are given in the table 2. 3998 instances with 44 attributes are provided for the PCA. Among them selected principal components are listed in table 3. The eigenvectors created for the selected principal components are listed in table 4. With these 10 features classifier is trained. Weka [14] is used for leveraging the performance of the naïve bayes classifier.

Table.2. PCA settings and specifications

=== Run information ===
Evaluator: weka.attributeSelection.PrincipalComponents -R 0.95 -A 5
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation: .\uk-2007-05.link_based_features.csv
Instances: 3998
Attributes: 44
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:Attribute ranking.
Attribute Evaluator (unsupervised):Principal Components Attribute Transformer

Table.3. Feature used from WEBSPAM-UK-2007

Feature	Eigenvalue	Proportion	Cumulative
Truncated pagerank	14.32993	0.33325	0.33325
siteneighbors	7.91094	0.18398	0.51723
reciprocity	2.01891	0.04695	0.56418
trustrank	1.92246	0.04471	0.60889
outdegree	1.82547	0.04245	0.65134
avgout_of_in	1.70571	0.03967	0.69101
prsigma	1.52949	0.03557	0.72658
avgin_of_out	1.40187	0.0326	0.75918
assortativity	0.94015	0.02186	0.88104
indegree	0.42629	0.00991	0.93217

Table.4. Feature and their Eigen vector

Feature	Eigenvectors									
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
truncatedpagerank	0.2217	0.1692	0.0573	0.0334	0.0512	0.0348	0.0036	0.0347	0.0304	0.0198
siteneighbors	0.1926	0.0839	0.0941	0.0251	0.0648	0.042	0.1307	0.1052	0.3168	0.1588
Reci procity	0.007	0.0004	0.4881	0.2448	0.0885	0.1532	0.1733	0.2854	0.0878	0.0883
Trust rank	0.0301	0.0147	0.2701	0.6329	0.152	0.0032	0.0055	0.0024	0.0139	0.0101
Out degree	0.0623	0.0255	0.0086	0.1257	0.5481	0.0897	0.3487	0.0856	0.1115	0.0898
avgout_of_in	0.032	0.0766	0.0854	0.0305	0.2676	0.5675	0.2261	0.0918	0.0929	0.1761
prsigma	0.0297	0.0888	0.0379	0.0405	0.2176	0.3223	0.4189	0.3827	0.1242	0.0013
avgin_of_out	0.0777	0.0293	0.3152	-0.116	0.0346	0.1075	0.1879	0.4311	0.2667	0.1475
assortativity	0.1585	0.146	0.0188	0.013	0.0158	0.022	0.0302	0.0098	0.0648	0.0879
indegree	0.2017	0.0942	-0.056	0.0007	0.094	0.0835	0.1049	0.0862	0.0716	-0.107

5. NAIVE BAYES CLASSIFIER FOR SPAMDEXING

The Naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule [9]. The corresponding classifier is the function classify defined as follows:

$$\mathrm{classify}(f_1,\dots,f_n) = \underset{c}{\mathrm{argmax}}\; p(C=c)\prod_{i=1}^{n} p(F_i = f_i \mid C=c). \tag{7}$$

5.1 Spamdexing Classification

Consider the problem of classifying website by their link based features, into spam and non-spam site. Probability that the i-th feature of a given website occurs in a feature from class C can be written as

$$p(w_i \mid C) \tag{8}$$

Then the probability that a given website D contains all of the features w_i, given a class C, is

$$p(D \mid C) = \prod_i p(w_i \mid C) \tag{9}$$

The goal is to find: "what is the probability that a given website D belongs to a given class C?" In other words, what is $p(C|D)$?
It is defined as:

$$p(D|C) = \frac{p(D \cap C)}{p(C)} \tag{10}$$

and

$$p(C|D) = \frac{p(D \cap C)}{p(D)} \tag{11}$$

Bayes' theorem manipulates these into a statement of probability in terms of likelihood.

$$p(C|D) = \frac{p(C)}{p(D)} p(D|C) \tag{12}$$

Assume for the moment that there are only two mutually exclusive classes, S and ¬S (spam and not spam), such that every element (website) is in either one or the other;

$$p(D|S) = \prod_i p(w_i|S) \text{ and } p(D|\neg S) = \prod_i p(w_i|\neg S) \tag{13}$$

Using the Bayesian result above, it can be written as:

$$p(S|D) = \frac{p(S)}{p(D)} \prod_i p(w_i|S) \tag{14}$$

$$p(\neg S|D) = \frac{p(\neg S)}{p(D)} \prod_i p(w_i|\neg S) \tag{15}$$

Dividing one by the other gives:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S) \prod_i p(w_i|S)}{p(\neg S) \prod_i p(w_i|\neg S)} \tag{16}$$

Which can be re-factored as:

$$\frac{p(S|D)}{p(\neg S|D)} = \frac{p(S)}{p(\neg S)} \prod_i \frac{p(w_i|S)}{p(w_i|\neg S)} \tag{17}$$

Thus, the probability ratio p(S | D) / p(¬S | D) can be expressed in terms of a series of likelihood ratios. The actual probability p(S | D) can be easily computed from log (p(S | D) / p(¬S | D)) based on the observation that p(S | D) + p(¬S | D) = 1. Taking the logarithm of all these ratios, it is possible to obtain the results:

$$\ln \frac{p(S|D)}{p(\neg S|D)} = \ln \frac{p(S)}{p(\neg S)} + \sum_i \ln \frac{p(w_i|S)}{p(w_i|\neg S)} \tag{18}$$

Finally, the website can be classified as follows. It is spam if $p(S|D) > p(\neg S|D)$(i.e., $\ln \frac{p(S|D)}{p(\neg S|D)} > 0$), otherwise it is not spam.

6. RESULTS AND DISCUSSION

6.1 Evaluation Metrics Used

Evaluation metrics and confusion matrix specifications used in this paper are listed in table 6 and 5 respectively. The results are also given in this section for all the specified metrics. Detailed accuracy of the spam/nonspam classes are given in table 7. Confusion matrix generated by the naïve bayes classifier is given in table 8. Feature inference of the naïve bayes classifier for the selected 10 features is given in the table 9. The training data and testing data are taken as 60% and 40% for 3998 instances. Results are compared with the standard Decision Stump classifier for the performance measurement. The comparison shows that naive bayes seems to perform well than the Decision Stump. Overall performance comparison chart is given in Fig 5. Evaluation metrics are compared in Fig. 6. Results shows that naïve bayes classifier have less incorrectly classified instance leading to correct classification. The classification accuracy of the naive Bayesian classifier is 98.07%.

Table.5. Confusion Matrix Specification

Confusion Matrix		**Actual outcome**			
		Positive	**Negative**		
Test outcome	**Positive**	a	b	Positive Predictive Value-PPV	a/(a+b)
	Negative	c	d	Negative Predictive Value-NPV	d/(c+d)
		Sensitivity(α)	**Specificity(β)**	Accuracy(ACCU)= (a+d)/(a+b+c+d)	
		a/(a+c)	**d/(b+d)**		

Table.6. Evaluation Metrics

Evaluation Metrics	**Equations**
True Positive Rate	TPR=d/(c+d) TPR(S)- Spam, TPR(N)- Normal
False Positive Rate	FPR=b/(a+b) FPR(S)- Spam, FPR(N)- Normal
Precision (γ)	γ=d/(b+d) γS – Spam, γN-Normal
Recall(δ)	δ=d/(c+d) δS-Spam, δN-Normal
F-Measure(F)	F=2*((γ*δ)/ (γ +δ)) FS-Spam, FN-Normal

Classifier Results:
TPR(S)=0.955
TPR(N)=0.982
FPR(S)= 0.018
FPR(N)=0.45
γS=0.76
γN=0.997
δS=0.955
δN=0.982
FS=0.846
FN=0.99
ROC=0.981	

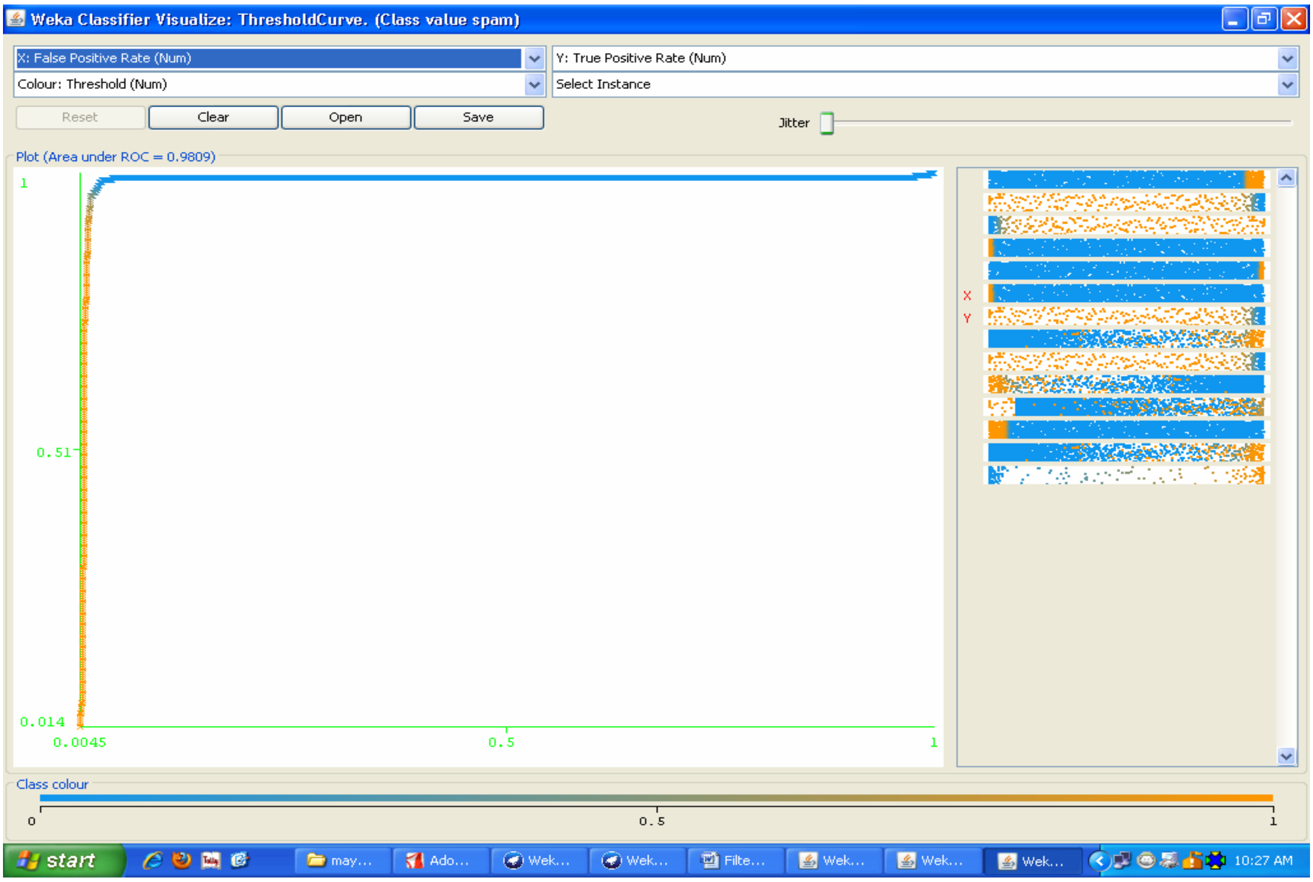
The cost curves for spam and nonspam classes are given in the Fig.3. Cost/benefit analysis for applying the naïve bayes classifier to the spamdexing application is given in Fig. 4.

Table.7.Detailed Accuracy by Class

TPR	**FPR**	**γ**	**δ**	**F**	**ROC**	**Class**
0.955	0.018	0.76	0.955	0.846	0.981	spam
0.982	0.045	0.997	0.982	0.99	0.981	nonspam
0.981	0.044	0.984	0.981	0.982	0.981	Weighted Avg.

Table.8.Naive Bayes Confusion Matrix

NaiveBayes		Actual			
		P	N		
Test outcome	P	212	10	PPV	0.9549
	N	67	3709	NPV	0.9822
		α	β	ACCU= 0.98074	
		0.75	0.996		



Weka Classifier Visualize: ThresholdCurve. (Class value spam)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.9809)

Figure 2. ROC Curve of Naïve Bayes Classifier

Table.9.Classifier values for all features

Attribute	Metrics	Class	
		Spam	Nonspam
Truncatedpagerank	Mean	0.3462	0.0201
	Std. Deviation	7.6372	3.4253
	Weighted sum	222	3776
	Precision	0.0198	0.0198
siteneighbors	Mean	1.0117	0.0594
	Std. Deviation	5.0901	2.6052
	Weighted sum	222	3776
	Precision	0.0172	0.0172
indegree	Mean	0.0871	0.0052
	Std. Deviation	0.741	0.715
	Weighted sum	222	3776

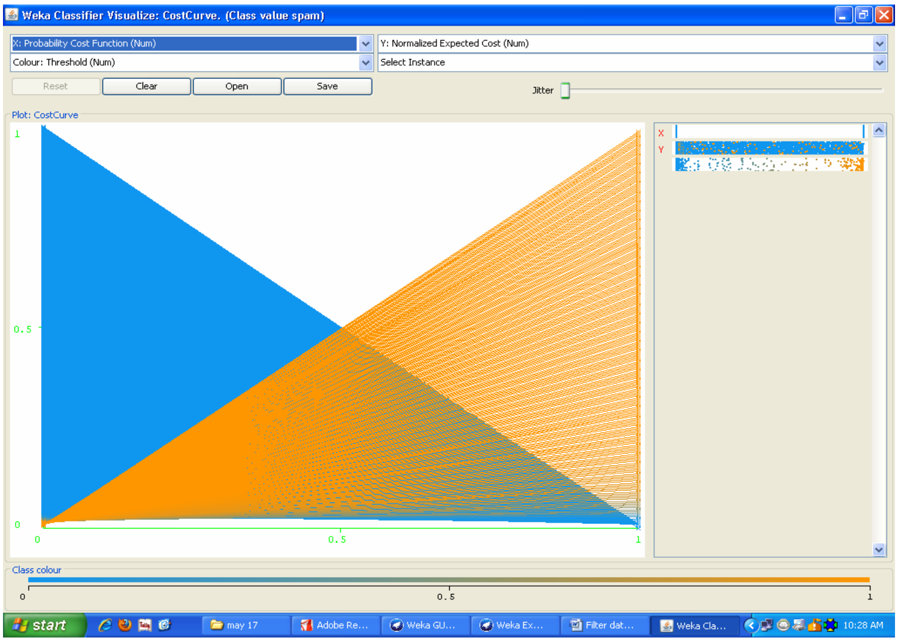
	Precision	0.0044	0.0044
reciprocity	Mean	0.0119	0.0007
	Std. Deviation	0.4367	0.5993
	Weighted sum	222	3776
	Precision	0.0038	0.0038
outdegree	Mean	0.2521	0.0149
	Std. Deviation	1.5258	1.3385
	Weighted sum	222	3776
	Precision	0.0122	0.0122
trustrank	Mean	0.5153	0.0303
	Std. Deviation	0.823	1.4067
	Weighted sum	222	3776
	Precision	0.0062	0.0062
avgout_of_in	Mean	0.2377	0.014
	Std. Deviation	1.666	1.2802
	Weighted sum	222	3776
	Precision	0.0051	0.0051
avgin_of_out	Mean	0.048	0.0029
	Std. Deviation	1.1675	1.1848
	Weighted sum	222	3776
	Precision	0.0086	0.0086
assortativity	Mean	1.2609	0.0741
	Std. Deviation	1.5658	0.9131
	Weighted sum	222	3776
	Precision	0.0098	0.0098
prsigma	Mean	0.3365	0.0198
	Std. Deviation	1.4088	1.2228
	Weighted sum	222	3776
	Precision	0.0099	0.0099

Table.10.Error rate of the Naïve Bayes Classifier

Time taken to build model: 0.14 seconds		
=== Stratified cross-validation ===		
=== Summary ===		
Correctly Classified Instances	3921	98.074 %
Incorrectly Classified Instances	77	1.926 %
Kappa statistic	0.8362	
Mean absolute error	0.0216	
Root mean squared error	0.1334	
Relative absolute error	20.5865 %	
Root relative squared error	58.2658 %	
Coverage of cases (0.95 level)	98.6493 %	
Mean rel. region size (0.95 level)	51.063 %	
Total Number of Instances	3998	

Error rate incurred in naive bayes classifier is listed in Table 10. It shows that naive bayes classifier yields relatively low error rate than the Decision Stump classifier.

ISSN 1512-1232



Weka Classifier Visualize: CostCurve. (Class value spam)

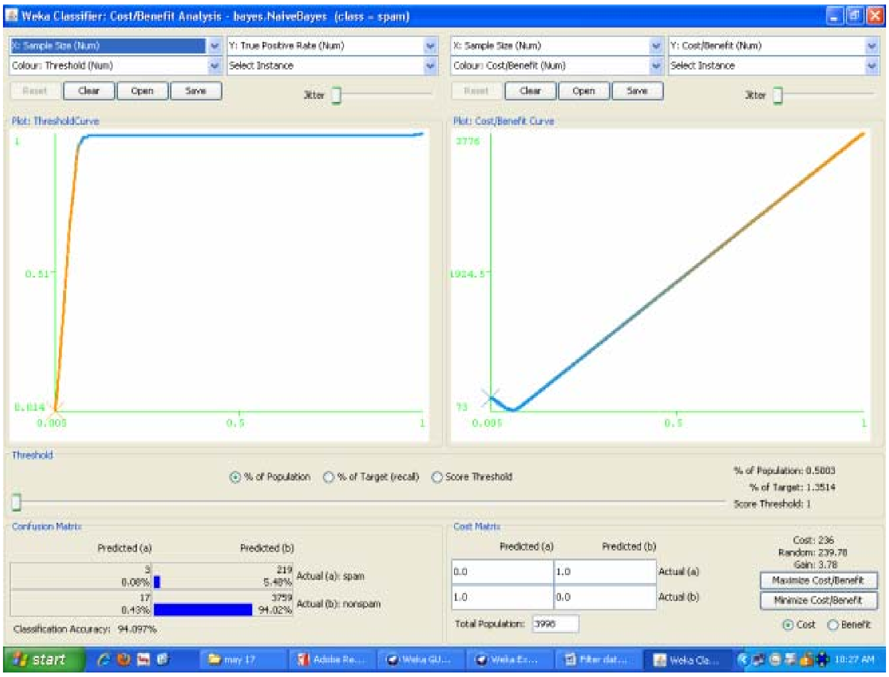
X: Probability Cost Function (Num)

Y: Normalized Expected Cost (Num)

Colour: Threshold (Num)

Select Instance

Figure 3. Cost curve for Spam/Nonspam Threshold



Weka Classifier: Cost/Benefit Analysis - bayes.NaiveBayes (class = spam)

X: Sample Size (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

X: Sample Size (Num)

Y: Cost/Benefit (Num)

Colour: Cost/Benefit (Num)

Select Instance

Threshold

◉ % of Population ○ % of Target (recall) ○ Score Threshold

% of Population: 0.5003

% of Target: 1.3514

Score Threshold: 1

Confusion Matrix

Predicted (a)	Predicted (b)	
3 0.08%	219 5.48%	Actual (a): spam
17 0.43%	3759 94.02%	Actual (b): nonspam

Classification Accuracy: 94.097%

Cost Matrix

Predicted (a)	Predicted (b)	
0.0	1.0	Actual (a)
1.0	0.0	Actual (b)

Total Population: 3998

Cost: 236

Random: 239.78

Gain: 3.78

◉ Cost ○ Benefit

Figure 4. Cost/Benefit Analysis

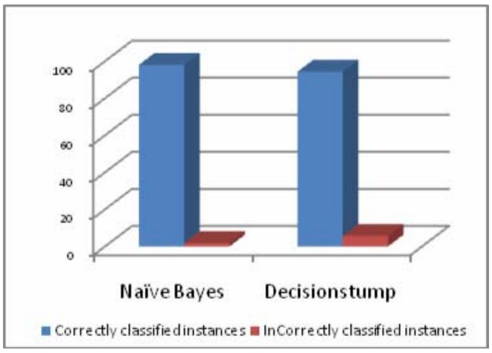


Figure 5. Overall Classification Performance of the naive bayes and Decisionstump

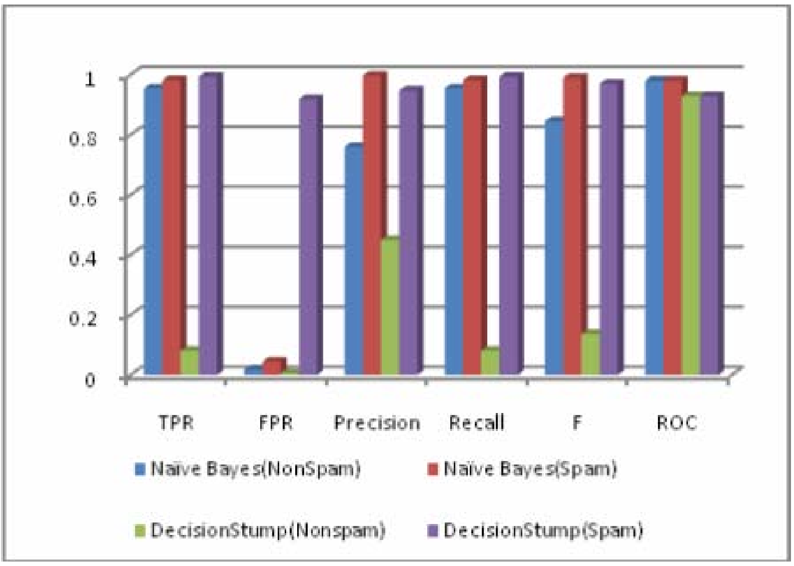


Figure 6. Performance comparison on different metrics

7. CONCLUSION

WWW had an intense impact in the past decades and acts as a vital source for information. Economic returns and a mind of play with disruptive content makes the web spam more prevalent. The consequence is reduced precision for search engines and junk time for end users. Spamdexing potentially degrades the quality of the results produced by the search engines. The usage of data mining in spamdexing data analysis is motivated by the following facts:

1. Data becomes information when it is effectively analyzed.
2. Information becomes knowledge when it is effectively interpreted.

The performance comparison of the various metrics such as true positive rate, false positive rate, F-measure, ROC is given in Fig.6. Spamdexing destructs the quality of the ranking algorithm used by the search engines. This paper addresses a naïve bayes classification to determine the link spam. This paper addresses link based features alone and content based features when combined with the link features would add more credit to the classifier. When both features are combined then it could be possible to achieve more accurate results and this will be the future scope of the research.

REFERENCES

1. Symantec Intelligence Report, b-intelligence_report_08-2013.en-us
2. Egele M, Kolbitsch C and Platzer C, 2009, Removing Web Spam Links from Search Engine Results, Journal of Computational Virology, Springer-Verlag, France, 2009.
3. Chung Y, Toyoda M and Kitsuregawa M, 2010, Identifying Spam Link Generators for Monitoring Emerging Web Spam, WICOW'10, North Carolina, USA.,pp:51-58.
4. Erdelyi M, Garzo A and Benczur A, 2011, Web spam classification: a few features worth more, WICOW/AIRWeb Workshop on Web Quality, India, pp:27-34.
5. Tian Y, Weiss G M and Ma Q, A Semi-Supervised Approach for Web Spam Detection using Combinatorial Feature-Fusion.
6. Silva R M, Yamakami A and Almeida T A, An Analysis of Machine Learning Methods for Spam Host Detection.
7. Karimpour J, Noroozi A and Abadi A, 2012, The Impact of Feature Selection on Web Spam Detection, I.J. Intelligent Systems and Applications, pp:61-67.

8. Geng G, Wang C H and Dan Li Q, 2008, Improving Web Spam Detection with Re-Extracted Features, WWW 2008, Beijing, China. ACM, pp:1119-1120.
9. Benczur A, Bıro I, Csalogany K, and Sarlos T, 2007, Web spam detection via commercial intent analysis, 3rd International Workshop on Adversarial Information Retrieval on the Web, AIRWeb'07.
10. Gan Q and Suel T, 2007, Improving Web Spam Classifiers Using Link Structure, AIRWeb '07, Canada.
11. Castillo C, Donato D and Gionis A, Scalable online incremental learning for web spam detection.
12. Jayanthi.S.K, Sasikala.S, GAB_CLIQDET: - A diagnostics to Web Cancer (Web Link Spam) based on Genetic algorithm, In Proc. Obcom'11,Vellore Institute of Technology(VIT), Chennai, Springer LNCS series, 2011, pp:524-523
13. Jayanthi.S.K, Sasikala.S, Perceiving LinkSpam based on DBSpamClust+, In Proc. 2011 International Conference on Network and Computer Science (ICNCS 2011), IACSIT, Kanyakumari, India, IEEE Xplore, pp: 31—35, Apr 2011
14. www.cs.waikato.ac.nz/ml/weka/
15. http://en.wikipedia.org/wiki/Naive_Bayes_ classifier
16. en.wikipedia.org/wiki/Principal_component_ analysis

Article received: 2012-11-01