Email Classification with Naïve Bayes

Mitchell Della Marta, Shu Han Bor

April 30, 2014

1 Introduction

1.1 Aim

To implement text classification of emails using the Naïve Bayes classifier for spam detection.

1.2 Importance

As email is one of the main forms of communication, spam detection to remove spam is important. Email spam is an unsolved global issue that negatively affects productivity and uses up valuable resources. Spam emails are usually sent with malicious intent to users for some monetary gain. Email users spend a large amount of time regularly deleting these spam emails, which needlessly occupy storage space and consumes bandwidth. Hence the development of classifiers that are able to separate legitimate from spam emails is required. Classification of emails to separate and remove spam from other emails, aims to save the user from having to spend time and effort sorting through and deleting spam, and protect unaware users from malware.

Traditional non-machine learning based techniques are easily fooled by spammers and require periodic manual updates. Therefore, machine learning based methods are required, which are able to automatically analyse email contents and dynamically update themselves to cope with new spamming techniques.

2 Data Preprocessing

Before classification can be performed, we must represent the files in our sample set *Lingpspam-mini600* appropriately. Using the "bag-of-words" model, words are extracted from the file and treated as features. There are two main characteristics in an email; the subject and the body. Thus, we will construct two "bags-of-words", one for each component. To determine which corpus a feature will belong to, we if the line begins with "Subject:". If it does the words within the line will be added to the subject corpus, otherwise they will be added to the body corpus.

To each file, we performed the following steps:

1. Replace all punctuation and special symbols with a space.
2. Remove stop words, using a list of stop words "english.stop".
3. Remove numbers or words containing digits.

The tokenisation schemes used in the example data are different to that used in our stop words list. The data detaches clitics and contraction affices, whereas the stop words list is tokenised based on whitespace and punctuation characters. Therefore, we chose to replace punctuation with a space (e.g. "n't" becomes "n", "t") instead of simply removing them. This meant that the stop word list would remove clitics as it removes single characters, instead of keeping it as a feature as ("nt" isn't in the stop words list).

The *document frequency* feature selection method was then used to select the 200 words which occured in the most documents, and used to build a classifier for the given documents. The following steps were taken to accomplish this:

1. Counter is created for each email, which keeps track of the number of times each word appears in that email.
2. For each word that appeared in the counter for an email, it is added once to the counter for the corresponding subcorpus.
3. In each corpus the words with the top 200 document frequency score are selected as features to represent that corpus.

Note that different runs of our preprocessing script can result in different classifier accuracies. As the counter is stored in a hashmap, words with the same frequency before and after the 200 word cutoff are chosen arbitrarily. Depending on the words chosen, the accuracy of the classifier can be affected.

Each selected feature was then weighted using its *td-idf* score [1].

$$tfidf(t_k, d_j) = \#(t_k, d_j) \times \log \frac{|Tr|}{\#Tr(t_k)}$$

Then the scores were normalised, using *cosine normalisation* [1].

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|}(tfidf(t_s, d_j))^2}}$$

2.1 Data Characteristics

Table I: Characteristics of dataset

	Subject	**Body**
# Features before removing stop words	1074	19886
# Features after removing stop words	915	19386
# $Class_{nonspam}$	600	600
# $Class_{spam}$	200	200

The “bag-of-words" model produced 19886 and 1074 features in the body and subject corpora respectively. After the removal of stop words, there were 19386 unique words remaining in the body corpus, and 915 words in the subject corpus (see Table I).

3 Feature Selection

Feature selection is performed using document frequency. This method involves computing the number of documents a word occurs in, for every word, and selecting the top 200 words with the highest scores to build a classifier.

Shown in Table II and Table III are the top 100 words for the subject and body corpora respectively and their document frequency score. Removing stop words filters out extremely common words that have little value in classification. It is beneficial in text processing, as it removes low quality features, allowing more significant features to have precedence. Most resulting words are sensible, although there are some aren’t (e.g. “qs”, “ll”, “ca”). However, removing these words results in a lower accuracy. Therefore, we have chosen to retain these as some of them may be help the classifier distinguish between spam and non-spam emails. Thus, given our task to process natural language text, the selection of words shown makes sense as it gives a better representation of the contents of the emails, and helps improve the accuracy of the classifier.

A comparison of Table II and Table III shows significant disparities in the document frequency of features and word distribution. The frequencies for the subject is significantly lower than that of the body. Whilst some features are shared between subject and body, most features selected are different.

Table II: Top 100 words in subject corpus and corresponding document frequency (DF) scores

Rank	Word	Score	Rank	Word	Score	Rank	Word	Score
1	sum	30	35	business	6	69	great	4
2	summary	26	36	workshop	6	70	summer	4
3	english	24	37	speaker	6	71	home	3
4	language	21	38	german	6	72	people	3
5	free	20	39	linguist	5	73	profit	3
6	disc	19	40	list	5	74	linguists	3
7	query	18	41	native	5	75	decimal	3
8	linguistics	15	42	resources	5	76	line	3
9	sex	13	43	grammar	5	77	credit	3
10	comparative	13	44	time	5	78	verbal	3
11	opposites	12	45	research	5	79	video	3
12	words	12	46	word	5	80	millions	3
13	email	10	47	needed	5	81	lists	3
14	book	10	48	languages	5	82	dental	3
15	method	9	49	software	5	83	life	3
16	call	9	50	jobs	5	84	resolution	3
17	job	9	51	spanish	5	85	comparison	3
18	japanese	8	52	fwd	4	86	offer	3
19	correction	8	53	web	4	87	latin	3
20	announcement	7	54	phonetics	4	88	change	3
21	syntax	7	55	american	4	89	uniformitarianism	3
22	million	7	56	request	4	90	check	3
23	qs	7	57	banning	4	91	hey	3
24	chinese	7	58	intuitions	4	92	reference	3
25	program	7	59	pig	4	93	debt	3
26	lang	6	60	books	4	94	synthetic	3
27	internet	6	61	unlimited	4	95	world	3
28	slip	6	62	programs	4	96	chomsky	3
29	armey	6	63	systems	4	97	site	3
30	money	6	64	address	4	98	released	3
31	mail	6	65	secrets	4	99	mac	3
32	part	6	66	information	4	100	sites	3
33	conference	6	67	www	4			
34	dick	6	68	read	4			

Table III: Top 100 words in body corpus and corresponding document frequency (DF) scores

Rank	Word	Score	Rank	Word	Score	Rank	Word	Score
1	information	205	35	message	91	69	full	74
2	language	192	36	ll	89	70	business	74
3	mail	183	37	phone	88	71	today	73
4	university	179	38	check	88	72	ac	73
5	time	178	39	receive	88	73	questions	72
6	list	171	40	good	87	74	interest	72
7	address	165	41	day	86	75	remove	72
8	english	159	42	year	86	76	john	71
9	http	156	43	interested	86	77	found	70
10	linguistics	156	44	case	85	78	related	70
11	people	146	45	working	85	79	site	69
12	send	146	46	include	85	80	usa	69
13	free	144	47	based	84	81	linguist	69
14	make	140	48	ve	84	82	text	68
15	email	133	49	made	83	83	read	68
16	work	128	50	home	83	84	point	68
17	number	128	51	note	83	85	week	67
18	www	122	52	part	83	86	ago	67
19	languages	119	53	including	81	87	book	67
20	find	118	54	mailing	81	88	cost	66
21	fax	116	55	type	80	89	dear	66
22	order	108	56	place	79	90	making	66
23	call	103	57	give	79	91	question	65
24	form	101	58	web	79	92	simply	65
25	research	100	59	program	79	93	offer	63
26	state	99	60	special	78	94	general	63
27	linguistic	99	61	date	78	95	received	63
28	subject	98	62	line	78	96	important	62
29	years	98	63	days	77	97	data	62
30	world	98	64	back	76	98	ca	61
31	contact	97	65	internet	76	99	summary	61
32	de	96	66	service	75	100	long	61
33	money	94	67	american	75			
34	word	91	68	system	74			

4 Subject vs Body Analysis

4.1 Results

We ran out preprocessed data against a variety of Weka classifiers (ZeroR, OneR, 1-NN, 3-NN, NB, DT and MLP) tested with their 10 fold cross validation, as well as against our Naïve Bayes classifier (MyNB) tested with our 10 fold cross validation method (Table IV).

Table IV: Various classifiers tested with 10 fold cross validation for both the subject and body corpora

	Accuracy (%)	
Classifier	**Subject**	**Body**
ZeroR	66.67	66.67
OneR	70.00	81.50
1-NN	75.50	87.50
3-NN	71.17	84.33
NB	69.67	95.00
DT	66.67	91.50
MLP	79.00	96.83
MyNB	66.17	94.83

4.2 Discussion

4.2.1 Comparison of Classifiers

The ZeroR classifier adheres to no rules and performs poorly for both subject and body, returning only an accuracy of 66.67% for both. This is because it returns the majority of the class given in the training data. As our dataset always contains more non-spam than spam emails (twice as many), it will always return non-spam.

In contrast, Multilayer perceptron (MLP) returns the best accuracy in both cases.

OneR generates one rule for each item in the dataset. After calculating rules for each, it will select the rule that gives the smallest total error. Although it is simple, it only performs slightly worse than more complicated algorithms such as 1-NN or 3-NN.

1-NN and 3-NN are both types of the k-nearest neighbour algorithm, where $k = 1$ and $k = 3$ respectively. Although they use similar algorithms, 1-NN performs better than 3-NN. 1-NN takes one neighbour, whereas 3-NN takes three. Therefore, this disparity is probably because in most cases the closest neighbour correctly classifies the new example, but the next two closest do not. Hence 1-NN gets a greater accuracy than 3-NN.

4.2.2 Comparison of myNB vs Weka's NB

Both our and Weka's Naïve Bayes classifiers return similar accuracies for both subject and body. The discrepancies between the accuracies of the two classifiers for the subject and body are $\pm 3.5\%$ and $\pm 0.17\%$ respectively. To find out if this difference is significant, we conduct a paired t-test:

1. Calculate differences between the folds in our Bayes classifier and Weka's Bayes classifier.
2. Calculate variance of the difference.

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{k} (d_i - d_{mean})^2}{k-1}$$

3. Calculate the confidence interval Z with a confidence level $(1-\alpha)$ of 95%.

$$Z = d_{mean} \pm t_{(1-\alpha)(k-1)} \hat{\sigma}$$

4. If the interval Z contains 0, the difference is not significant. Otherwise, it is.

Table V: Comparison of fold accuracy between our Naïve Bayes clssifier (C1) and Weka's (C2) for subject corpus, and difference $d_i = |C1_i - C2_i|$ between the two classifiers at each fold

Fold Number	**C1 (%)**	**C2 (%)**	**Difference**
Fold 1	71.67	58.33	13.34
Fold 2	66.67	56.67	10.00
Fold 3	58.33	66.67	8.34
Fold 4	53.33	71.67	18.34
Fold 5	63.33	65.00	1.67
Fold 6	70.00	78.33	8.33
Fold 7	66.67	80.00	13.33
Fold 8	71.67	65.00	6.67
Fold 9	70.00	76.67	6.67
Fold 10	70.00	78.33	8.33
Mean	66.17	69.67	9.5

Table VI: Comparison of fold accuracy between our Naïve Bayes clssifier (C1) and Weka's (C2) for body corpus, and difference $d_i = C1_i - C2_i$ between the two classifiers at each fold

Fold Number	**C1 (%)**	**C2 (%)**	**Difference**
Fold 1	98.33	100.00	1.67
Fold 2	98.33	98.33	0.00
Fold 3	100.00	93.33	6.67
Fold 4	98.33	91.67	6.66
Fold 5	91.67	93.33	1.66
Fold 6	96.67	95.00	1.67
Fold 7	90.00	91.67	1.67
Fold 8	95.00	98.33	3.33
Fold 9	93.33	93.33	0.00
Fold 10	86.67	95.00	8.33
Mean	94.83	95.00	3.17

Given the differences calculated in Table V, we get that for the subject corpus:

$$\begin{aligned} \hat{\sigma} &= 10.38 \\ Z &= 9.5 \pm 2.26 \times 4.59 \\ &= 9.5 \pm 10.37 \end{aligned}$$

Similarly, given the differences calculated in Table VI, we obtain for body corpus:

$$\begin{aligned} \hat{\sigma} &= 4.47 \\ Z &= 3.17 \pm 2.26 \times 2.99 \\ &= 3.17 \pm 6.76 \end{aligned}$$

For the subject, $Z = [-0.87, 19.87]$ and for body, $Z = [-3.59, 9.93]$. In both cases the interval Z contains 0. Therefore, the difference between the two classifiers for both subject and body are not significant.

4.2.3 Comparison Between Subject and Body Corpora

The overall performance of the classifiers regardless of type, perform significantly better on the body than the subject corpus. This suggests that the email body is a better indicator of content.

5 Challenge Analysis

5.1 Results

5.2 Discussion

6 Conclusion and Future Work

The tokens used in our classifier are formed from single words. Therefore, it will not analyse common consecutive words that are found in spam emails, leading to the failure of our classifier from detecting these emails.

By taking into account permutations of consecutive words, or words that appear within a specified distance of each other, the accuracy of our Bayesian classifier could be increased.

7 Reflection

8 Instructions: How to run code

References

[1] Fabrizio Sebastiani, *Machine learning in automated text categorization.* ACM Computing Surveys, 34(1):1-47, 2002.