Notebook Paper (Draft #2)

IBM SpamGuru on the TREC 2005 Spam Track

Richard Segal
IBM Research
Hawthorne, NY
rsegal@us.ibm.com

October 25, 2005

1 Introduction

IBM Research is developing an enterprise-class anti-spam filter as part of our overall strategy of attacking the Spam problem on multiple fronts. Our anti-spam filter, SpamGuru, mirrors this philosophy by incorporating several different filtering technologies and intelligently combining their output to produce a single spamminess rating or score for each incoming message. The use of multiple algorithms improves the system's effectiveness and makes it more difficult for spammers to attack. While a spammer may defeat any single algorithm, SpamGuru can rely on its remaining algorithms to maintain a high-degree of effectiveness.

The IBM Research submission to the TREC 2005 spam track used the SpamGuru anti-spam framework to evaluate three experimental anti-spam technologies that are currently under development. The first is LNB (Less-Naive Bayes), an extension of the ubiquitous naive-bayes text classifier that relaxes the independence assumption by modeling some of the dependencies between attributes. The second is SMTP Path Analysis, an algorithm that classifies incoming mail based on the servers used to deliver the message. The third technology evaluated here is a classifier aggregation algorithm that uses the Nelder-Mead nonlinear optimizer to dynamically select weights for combining the LNB and SMTP path analysis classifiers into a single prediction.

The next section presents an overview of the SpamGuru anti-spam framework. The following sections details each of the three algorithms used in our evaluation. The appendix summarizes our results on the TREC 2005 Spam Track datasets.

2 SpamGuru Overview

SpamGuru is an server-based framework for anti-spam filtering. The SpamGuru framework provides the plumbing needed for a complete anti-spam solution. The SpamGuru server can communicate with a variety of SMTP servers and e-mail gateways to provide anti-spam services.

The work of labelling incoming messages is done using the SpamGuru Filtering Pipeline. The filtering pipeline consists of one or more pluggable anti-spam modules. SpamGuru processes each incoming e-mail by passing it along from one anti-spam module to the next. Each module analyzes the message and optionally assigns it a score based on its prediction of how likely it is to be spam. The last module in the chain is always an aggregator who is responsible for combining the results of each individual classifier into a single score. That score is then used by SpamGuru to decide what should be done with the incoming message.

The SpamGuru Filtering Pipeline was instantiated for TREC 2005 as follows. Our primary TREC submission consisted of a MIME decoder, the LNB classifier, the SMTP Path Analysis classifier, and our optimization-based classifier aggregator. For our other submissions, we submitted each classification technology individually so that we could evaluate each individual algorithm as well as our classification aggregation technology. Thus, our second submission used just the SMTP Path analysis module. Our third submission consisted of the MIME decoder and the LNB classifier. Our second submission does not consider the contents of an e-mail and therefore does not benefit from MIME decoding.

3 Less Naive Bayesian

Traditional Naive Bayesian filters make the assumption that words are conditionally independent given the target classification and use that assumption to derive an otherwise mathematically sound formula for the probability of a document being a member of that class. However, it is manifestly true that this conditional independence assumption does not hold. The word “inkjet” appears much more often in spam documents that also contain the word “printer” than in random

spam documents. The goal of Less Naive Bayes (LNB) is to produce a classifier that is more accurate than simple naive-bayes by taking the dependencies among features into account.

The LNB algorithm is not strictly bayesian. LNB takes the approach of a discriminative classifier in which probabilities are replaced by a single weight that represents a word's relative spaminess. As is typically done with discriminant-style classifiers, the word weights are ajdusted as new training examples arrive to ensure that the new training example is categorized correctly. What is unique is that the particular way in which LNB adjusts weights makes it less sensitive to feature dependencies. Early experiments with this algorithm suggest that it consistently outperforms traditional naive-bayes on spam classification tasks.

4 SMTP Path Analysis

SMTP Path Analysis categorizes incoming spam based on the sequence of gateways that delivered the message. The intuition behind the algorithm is that mail sent through the same or similar IP addresses are likely to share the same classifications.

The SMTP protocol specifies that each SMTP relay used to send an email message must add at the beginning of the message's header list a "received" line that contains (at least) information about the SMTP server receiving the message, from where the server received the message, and a timestamp stating when the header was added. These header lines, taken together, provide a trace of the SMTP path used to deliver a message.

SMTP Path Analysis learns the spamminess or goodness of IP addresses by analyzing the past history of e-mail sent using that IP address. When training, the algorithm extracts from each message the sequence of IP addresses that mail supposedly took to get to the recipient and collects statistics about the spam and good e-mail sent through each IP address. During classification, the algorithm extracts the IP address sequence from the target message and yields a score for that message based on the IP addresses of the gateways supposedly used to deliver the message. The algorithm looks at no other information; in particular, it does not otherwise analyze the content of the message or consider any domain information.

The probability that mail passing through any previously-seen IP address is spam is estimated, when possible, based on the frequency of spam in e-mail previously sent by that host. However, due to dynamic IP addresses and other similar complexities inherent in IP addresses, a substantial amount of e-mail originates

from IP addresses for which we may have little to no data. We address this issue by combining statistics of the current IP address with those of "nearby" IP addresses whenever there is not sufficient data regarding the current IP address to make a reliable decision.

As described, SMTP Path Analysis is susceptible to spoofing. A spammer can easily add false received line headers to a message to make it appear to be sent through a realiable source. To address this problem, we establish a credibility value for each intermediate address, and if an address is not credible we partially ignore the remaining addresses.

5 Classifier Aggregation

SpamGuru employs an aggregate classifier to combine the results of each classification algorithm into a single score that can be used to decide how each incoming message should be routed. Classifier aggregation provides two benefits. First, it improves classifier accuracy by combining the best features of multiple algorithms. Second, it improves the robustness of the overall system since a spammer trying to attack the system must defeat multiple anti-spam filtering technologies to defeat the entire system.

Each classifier in SpamGuru rates the spamminess of incoming messages by returning a score from 0 to 1000. A score of 0 indicates that the message is almost certainly good, while a score of 1000 indicates that the message is almost certainly spam. The output of most classifiers can be scaled to fit this range as needed.

The scores of several classifiers are combined by computing a single score from the scores of each individual classifier. There are several methods for combining scores. One option is to return the minimum score output by any of the classifiers. This method tags a message as spam only if all the classifiers return a score over a threshold provided by the user. That is, all the classifiers agree that the message is spam. The minimum score aggregator produces a very low false-positive rate since a good message can only be misclassified if all the algorithms incorrectly label the message as spam. On the other hand, its spam detection rate can be no better than the least effective classifier.

By experimentation, we found that the most successful way to combine classifiers was to use their unthresholded output scores as input to a super-classifier; a linear one typically worked well in practice. The linear super-classifier's score was a weighted sum of the scores of the constituent classifiers. The optimal values of the weights were established by using the Nelder-Mead nonlinear optimizer

to minimize a penalty function that emphasized the relative importance of false positives and false negatives in the anti-spam domain, the optimization being performed over a window of several thousand most recently labeled emails.

6 Acknowledgements

The authors want to thank the many in IBM who have helped in the development of SpamGuru and the development of many of the ideas presented in this paper. Those involved include Bill Arnold, Nathaniel Borenstein, Jason Crawford, Mike Halliday, Shlomo Hershkop, Tien Huynh, Barry Leiba, Jeff Kephart, Joel Ossher, V. T. Rajan, Isidore Rigoutsos, Mark Wegman, Ian Whalley. We also give special thanks to the track organizers for all their hard work in putting together this workshop and their patience in addressing some techical difficulties that arose with our initial submission.

A Results

Table 1: Spam misclassification rate on "full" dataset.

Ham	Spam Misc %			
Misc %	LNB	SMTP	Aggregate	Best
0.01	24.73	28.87	15.10	9.89
0.02	17.69	28.87	11.99	5.72
0.05	12.09	28.86	6.00	2.73
0.10	7.02	28.85	3.63	1.78
0.20	4.10	28.83	2.02	0.73
0.50	1.62	28.75	0.94	0.38
1.00	0.72	28.60	0.47	0.23
2.00	0.29	28.25	0.24	0.12
5.00	0.10	27.32	0.09	0.03
10.00	0.04	26.14	0.04	0.01
20.00	0.02	24.92	0.02	0.00
50.00	0.02	13.32	0.01	0.00

Table 2: Spam misclassification rate on "Mr. X" dataset.

Ham Misc %	Spam Misc %			
	LNB	SMTP	Aggregate	Best
0.01	17.01	94.19	14.16	14.16
0.02	10.36	87.04	12.13	10.36
0.05	6.21	76.30	8.05	6.21
0.10	4.55	59.16	5.71	3.56
0.20	3.59	34.44	3.96	2.42
0.50	3.20	25.08	3.14	0.86
1.00	2.99	24.24	2.96	0.34
2.00	2.86	24.08	2.81	0.17
5.00	2.76	22.01	2.69	0.05
10.00	2.72	21.93	2.65	0.02
20.00	2.71	15.13	2.63	0.00
50.00	2.70	2.63	2.62	0.00

Table 3: Spam misclassification rate on "sb" dataset.

Ham Misc %	Spam Misc %			
	LNB	SMTP	Aggregate	Best
0.01	80.00	83.23	70.84	53.81
0.02	48.90	80.65	70.06	19.61
0.05	22.19	74.06	18.45	14.19
0.10	17.16	61.94	15.48	11.23
0.20	9.94	55.48	11.87	9.29
0.50	7.61	54.84	8.52	5.81
1.00	6.45	39.35	5.81	3.48
2.00	5.03	13.29	5.03	2.19
5.00	4.13	3.48	3.87	0.77
10.00	3.10	3.10	2.84	0.39
20.00	2.58	2.84	2.32	0.13
50.00	2.32	2.32	2.19	0.06

Table 4: Spam misclassification rate on "tm" dataset.

Ham Misc %	Spam Misc %			
	LNB	SMTP	Aggregate	Best
0.01	40.27	90.87	33.86	33.86
0.02	18.35	71.23	18.74	18.35
0.05	9.11	54.96	8.08	8.08
0.10	6.15	47.89	5.42	5.42
0.20	4.05	41.38	3.67	3.67
0.50	2.29	40.83	1.99	1.99
1.00	1.55	39.98	1.28	1.28
2.00	1.00	33.01	0.79	0.79
5.00	0.66	13.25	0.38	0.38
10.00	0.48	7.56	0.26	0.20
20.00	0.34	1.21	0.18	0.12
50.00	0.24	0.49	0.09	0.04

Table 5: Corrected Spam misclassification rate on "Mr. X" dataset. These results use an updated version of SpamGuru that corrects a bug that caused all messages larger than 100,000 characters to be classified as ham.

Ham Misc %	Spam Misc %			
	LNB	SMTP	Aggregate	Best
0.01	–	–	13.46	14.16
0.02	–	–	12.82	10.36
0.05	–	–	7.49	6.21
0.10	–	–	3.99	3.56
0.20	–	–	1.60	2.42
0.50	–	–	0.70	0.86
1.00	–	–	0.45	0.34
2.00	–	–	0.26	0.17
5.00	–	–	0.12	0.05
10.00	–	–	0.07	0.02
20.00	–	–	0.05	0.00
50.00	–	–	0.04	0.00