Eric Chen
CS445 Hw1

Data Mining Applied to Email SPAM Detection and Filtering

What is spam? Spam, also known as Unsolicited Commercial Email, is generated by sending unsolicited commercial messages to many recipients without their permission. The spammers use a computer program to check almost every website on the internet. The program looks at the code of every web page, it looks for an email address and it collects and save your email address to the spammers database of millions of harvested addresses. Or you can be victim of dictionary spam. The spammers know that there is going to be a mikesmith@gmail.com. They setup computers to spam billions of names at that mail service simply by targeting every person names [1]. If you have an email address I bet you're fed up with spam you receive. The increase in spam has virtually relegated email as an adult only facility. Spam is a waste of our time. Many spam emails are obscene; many are offensive or insulting to one's intelligence, like

> *" Dear Lucky Winner, RE: BONUS LOTTERY PROMOTION PRIZE AWARDS WINNING NOTIFICATION 2008: The result of our computer draw (#978) selected your name and ... won you the lottery in the 2nd category i.e. match 5 plus bonus. You have therefore been approved to claim a total sum of £ 600,000.00 in cash...".*

Spam is a costly problem and many experts agree it is only getting worse because of the economics of spam and the difficulties inherent in stopping it. It is unlikely to go away soon. Here is a little fact about the spam that tells why we need to filter the spam; $730/year in lost productivity for every employee; $8,900,000,000/year total cost to US corporations [4]. We can view spam filter as immune system, it needs to distinguish between good and harmful elements and it is impossible to produce all the existing "antibodies" [4]. There is no such spam filter would claim that it able to detect the spam 100% without blocking any ham. Many data mining and machine learning researchers have worked on spam detection and filtering, which can be seen as a specific text categorization task. What makes this task so difficult to accomplish? Spam likes computer viruses it keeps mutating in response to the latest "immune system" response [4]. A further complication of in spam filtering is the asymmetry in error costs. Judging

a legitimate email to be spam (a false positive error) is usually far worse than judging a spam email to be legitimate (a false negative error). A false negative simply causes slight irritation, i.e., the user sees an undesirable message. A false positive can be critical. If spam is deleted permanently from a mail server, a false positive can be very expensive since it means a (possibly important) message has been discarded without a trace. I experienced the pain myself, my potential future employer sent me a hiring application few days ago that needs me to fill up and send them back as soon as possible. But somehow Microsoft Outlook classified that email as a spam and I do not check spam every day. Fortunately I found out today and able to meet the deadline. In an essay on developing a Bayesian spam filter, Paul Graham [3] describes the different errors in an insightful comment:

> False positives seem to me a different kind of error from false negatives. Filtering rate is a measure of performance. False positives I consider more like bugs. I approach improving the filtering rate as optimization, and decreasing false positives as debugging.

Spam filtering can be applied to different level of communication architecture. At the network level, routers keep a list of the IP addresses of known spammers (black list) and block emails from those addresses. At the server level, the similar message sent in high volume should be detected as spam [4]. The spam filtering at the user level also known as content-based filtering. In general, there are two approaches to do filtering. The first relies on hard-coded rules, which are periodically updated for and by the user. Each email is given a certain amount of spam points based on some rule set. If the email exceeds some threshold score, it is typically quarantined for later deletion by the user. The second approach uses machine learning models, leveraging work done on text classification and natural language processing. A training set of emails is created with both ham and spam emails, and a machine learning technique is chosen to classify the emails [1]. More advanced versions of this technique allows the user to update the learner to give it feedback by clicking "this is not spam move to inbox" or "mark this as spam" and continuously train the classifier using new examples [2].

The most popular of spam filtering technique is Bayesian filtering, it is a simple, probabilistic method based on Bayes' rule of combining evidence. In the email context, it simply means the

probability of spam given a certain word or token can be calculated if you know the probability of unwanted emails and how often this word appears in them, divided by how often the word appears in general in any email body [3]. (i.e. if the words like click, Viagra, hottest, girls occur couple times in the message, it will be more likely to be classified as spam.) Most reports have shown that Bayesian filters works correctly over 99 percent for one user [4].

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\Pr(\text{spam})}{\Pr(\text{words})} \quad [3]$$

To circumvent simple filtering, spammers began to employ content obscuring techniques such as inserting spurious punctuation, using bogus HTML tags and adding HTML comments in the middle of words know as Bayesian poisoning [1]. It is now common to see fragments such as these: v.ia.g.ra, v1agra, 100% Mo|ney Back Guaran|tee!, C<!--7udspp6-->lic<!--yan1nbecx2-->k he<!--ewu-->re. When rendered, these are recognizable to most people but they foil simple word and phrase filtering. To counter this, some filters remove embedded punctuation and bogus HTML tags before scanning, and consider them to be additional evidence that the message is spam. Spammers have also realized that filters use Bayesian word analysis and content hashing. In response, they often pepper their messages with common English words and nonsense words to foil these techniques [2]. Messages are designed so these words are discarded when the text is rendered, or are rendered unobtrusively. Graham-Cumming [3] maintains an extensive catalog of the techniques used by spammers to confuse filters. Furthermore, spammer display image with text of message in the email, this makes those spam messages are undetectable by using text based spam filter like Bayesian filtering [1].

Whatever new filtering capabilities arise, it is just a matter of time before spammers find ways to evade them. Because of this text distortion and image spam, spam filtering are not just simple text classification and information retrieval problems anymore. We will have to develop new techniques based on Bayesian filtering that can adapt to new distortion patterns and figure out a new way to solve image spam. The war between spam and spam filter would never end…