The War Against Spam: A report from the front line

Bradley Taylor
Google
Mountain View, CA
bat+nips@google.com

Dan Fingal
Google
Mountain View, CA
dfingal+nips@google.com

Douglas Aberdeen
Google
Zurich, Switzerland
daa+nips@google.com

Abstract

Fighting spam is a success story of real-world machine learning. Despite the occasional spam that does reach our inboxes, the overwhelming majority of spam — and there is a lot of it — is positively identified. At the same time, the rarity with which users feel the need to check their spam box for false positives demonstrates a high precision of classification. This paper is an overview of Google's approach to fighting email abuse with machine learning, and a discussion of some lessons learned.

1 Introduction

Google is one of the world's most popular providers of free email services. While Gmail is the high profile mail product, many Google services send and receive messages. All of these services may be subject to *abuse*. This term includes not just inbound spam, but outbound spam, denial-of-service (DoS) attacks, virus delivery, and other imaginitive attacks.

We have several layers of anti-abuse mechanisms, more than one of which use machine learning approaches. This short paper gives a high-level description of these mechanisms with an emphasis on the learning components. The other contribution of this paper is to share some of the issues that Google has faced in trying to deploy machine learning to fight email abuse.

2 Google's Anti-Abuse Mechanisms

Google's anti-abuse system consists of traditional hard-wired rules, rules inferred by a learning system, clustering, and an innovative reputation based system. Google was also an early adopter of the two well known sender authentication systems: Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM) [1]. Many of these components provide overlapping protection from abuse. All outgoing, incoming and internal messages are tested against this system.

An important component of the system is the online estimation of sender reputations [3]. Briefly, statistics are kept on the amount of spam and nonspam originating from each source. Examples of sources include the sender IP address and domain. Spam and nonspam are identified by automatic classification and by users clicking on the "Report Spam" and "Not Spam" buttons. These statistics are used to compute a reputation score that is passed onto the rest of the system. Messages from very low reputation sources are immediately labelled as spam. Reputations change over time; very quickly in some circumstances. This can be viewed as a simple form of online semi-supervised learning.

We also use machine learning methods to infer new abuse classification rules. We deployed an innovative in-house algorithm that combines some features of decision trees, parallel logistic regression, and efficient optimisation methods. One of the unique features of our approach is the ability to handle huge feature sets as well as huge training sets. It is also distributed, operating across many processors. Feature engineering is important, but the learning mechanism relieves engineers of the burden of establishing how to combine features and how much trust to give each feature. Learning also infers probabilities about what is spam and nonspam, which is valuable for fine tuning.

3 Observations

Correctly classifying a single text email into spam, phishing, viral, non-spam, and other categories is already challenging. But even more difficult is trying to classify many *millions* of messages a day, where the messages can be in any language, the message can include images and other attachments, and may or may not conform to recommended protocols. This section expands on a few of these issues from a machine learning perspective.

Spam Flavours: Google's definition of spam is anything unwanted by the user. As well as unsolicited ads, this definition includes phishing messages and undesired bulk email. However, while unsolicited ads are just annoying, the consequences of being misled by a phishing attack can be severe. Thus we might want to target phishing messages more aggressively, assigning a greater cost to misclassification, and giving the user more warning. Bulk email is difficult for a different reason: some users want these messages, while others do not. This necessitates a per-user approach to classification. The core difficulty is how to do this more fine grained classification of messages.

Abuse is More Than Spam: Established email services are popular tools for sending spam because outgoing mail from these services is SPF/DKIM authenticated and comes from a reputable domain. Preventing outgoing spam is more difficult then preventing inbound spam. For example, reputation is harder to compute because the final classification of the message is not always known. Feedback from cooperative recipients and services can help alleviate this.

Spammers programmatically create thousands of accounts from which to launch spam campaigns. This is surprising given that free services require the solution of a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [5] prior to the opening an account. Spammers have a very lateral solution: CAPTCHA images are farmed out to many humans to solve. These people might be paid, or they might solve them on behalf of the spammers unwittingly, in an attempt to access some other — often dubious — content.

Viruses embedded in attachments are another form of abuse that email providers must check for in the final classification of messages.

Abuse is a (highly) Non-Stationary Process: As spam filters become more sophisticated, so do spammers. Restating the example above, spammers have begun to use reputable free email services to avoid being caught by increasing use of SPF and DKIM authentication. This is a slow form of spam evolution. However, when a dedicated campaign is launched — such as a phishing attack targeting customers of a particular bank — the spam source and content can evolve hourly. A concerted campaign has the appearance of a game, involving a sequence of moves made by the spammer, and counter-moves by service providers. Effective learning methods must be online and able to adapt quickly to a relatively small number of samples.

Algorithms Must Scale: Consider a very simple machine learning algorithm such as ID3 for decision trees. The learning algorithm is relatively cheap with $O(n.f^2)$ operations, where n is the number of training instances and f is the number of spam features (assuming they are all discrete) [4]. However, for even a modest email service there may be millions of training messages and almost as many features. It is obvious that a single machine cannot hold every training message in memory. But a single machine might not even be able to hold the entire list of potential *features*. For example, we might have a feature for every n-gram in the message, across many languages. Additionally, many useful features are expensive to compute. For example, performing image to text conversion on embedded images. Parallelization helps, but this requires development of learning algorithms that permit distributed models, distributed data, and asynchronous feature evaluation.

Generating Training Data is Difficult: The reputation calculator and rule learning mechanisms rely heavily on users declaring spam using the feedback mechanism in the web interface. Unfortunately, many users use this infrequently or not at all (consider POP access). This can be a severe problem when a new campaign starts. Consider what happens if less than 50% of recipients label a new form of spam. Because users only label false negatives (misclassified as nonspam) and false positives (misclassified as spam), we have the problem of selecting reasonable true positive and true negative examples for training. If we do not do this correctly the training true negatives could include the *majority* of the new form of spam messages labelled as nonspam! The problem is worse for the positive class. A large number of users hardly ever look in their spam folder to report false positives. When sampling messages for training we employ a number of heuristics to mitigate such problems. Because we have a large amount of data, with a relatively small number of labels, semi-supervised learning [2] is very useful.

Understanding the Results: A well known barrier to deploying statistical machine learning methods in industry is the difficulty of interpreting the results. To generate trust, and to help debug algorithms, we need the ability to understand what the classifiers have learned. We should be able to explain concisely *why* any particular message was given a spam classification. The multi-layer perceptron is a traditional example of a classifier that fails to meet this criteria.

Internationalization: We are trying to detect abuse in any language, or even a mixture of languages. This limits the effectiveness of language specific features. We want our abuse system to work equally well for Japanese as it does in English. This can be surprisingly difficult. For example, with some languages it is not even clear how to compute word boundaries.

Make Reverse Engineering Difficult: We illustrate this point by example. To encourage users to find and label false positives we could order their spam folder by decreasing spaminess. But this would be very useful to spammers who regularly test if their messages are delivered. Similarly, it would be useful for POP users to have access to spam scores computed by Google in the headers.

4 Conclusion

To conclude, Google *does* successfully use machine learning technology to filter out the vast majority of spam and other abuse. However, the occasional appearance of spam in our inboxes reveals that there are remaining challenges, particularly in responding rapidly to new campaigns. There are many potential improvements to be made in algorithms for scalable online semi-supervised learning.

References

[1] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. Domainkeys identified mail (DKIM) signatures. (rfc 4871)., 2007. `http://www.ietf.org/rfc/rfc4871.txt`.

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[3] Bradley Taylor. Sender reputation in a large webmail service. In *Third Conference on Email and Anti-Spam (CEAS 2006)*, 2006.

[4] Paul E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.

[5] Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Communiciations of the ACM*, 47(2):56–60, 2004.