Assignment 1: Email Classification

Barbara Engelhardt, Princeton University

out 02/03/2015; due 02/26/15

Background

People send and receive an incredibly large number of emails every day. In order to effectively manage this flow of emails, we must find ways to focus our attention on emails that are valuable and ignore emails that are not useful. One way to do this is via *spam detection*, or classifying unhelpful emails as *spam*. *Spam* is defined by wikipedia as "the use of electronic messaging systems to send unsolicited messages (spam), especially advertising, as well as sending messages repeatedly on the same site." It is an inexpensive way to advertise, as there is little cost to sending millions of emails. Many groups have developed *spam filters* to identify and filter spam from email inboxes. These systems are imperfect and always requiring updates, as spammers continuously adapt their spamming approaches to circumvent the most recent filters.

Project definition

Your goal in this homework project is to use a data set consisting of 90,000 spam emails and 90,000 non-spam emails (from the Enron email data set [Klimt and Yang, 2004]) to build a spam filter, or software that classifies an email as *spam* or *not spam*. We have supplied you the training and test email data sets on the Piazza website and also a python script to identify a simple dictionary of words from the emails, creating, for each email, a bag-of-words representation using the dictionary words. Your first step in the process is to download the script and the data and run the script on the training data to build a vocabulary and create a bag-of-words feature representation for each email; see the `readme.txt` file in Piazza/homework1/ folder for the initial steps in the process. Feel free to extend the feature set in interesting and well-motivated ways (see *Extensions*, below).

Then, you should build (multiple) classifiers that take in the feature sets and the classifications of those feature sets and fits a classifier. Feel free to use the classifiers we have or will discuss in class as well as others mentioned in our text books, described in the scientific literature, or implemented in software. You may also use more sophisticated classifiers (see *Extensions*). Because of the large number of possible features, we recommend using some type of feature selection to reduce the number of features. Finally, you should evaluate the classifiers you apply to this problem according to (at a minimum) the Receiver Operating Characteristic (ROC) curves on the test data set, which consists of 10,000 held out spam emails and 10,000 held out non-spam emails.

Essential to any data analysis task is the interpretation of the results. What features were most important for classification, and what do these features tell us about the problem? What is worse: classifying a non-spam email as spam, or classifying a spam email as non-spam? What types of emails were easy to classify for all approaches, and on what types of emails did they disagree? Simply building a machine learning approach to solve the problem does not constitute a data analysis; recovering and characterizing signal from these results does.

Deliverables

Your deliverables for this project include:

- A four page (not including citations) summary of the project work, which should contain (as described in the Example project write up on Piazza):
  - A title, authors' names, and abstract for the project;
  - an introduction to the problem being addressed;
  - a description of the data;
  - a description of the methods developed and used, and how they were fitted using training data;
  - a presentation of the results of the methods applied to the test data;
  - a discussion of the results, including specific examples of emails and features that highlight the behavior of the classification models;
  - a short summary and conclusion, including extensions that you believe would be particularly valuable based on the results;
  - a *complete* bibliography to support the email databases, feature selection, classifiers, code bases, and related work that are relevant to your project.
- If you develop new methods for this project, please include a link to a GitHub repository with your software available there.

Please put your PDF write up of the project into
`https://dropbox.cs.princeton.edu/COS424_S2015/Assignment1_Spam_Detection`
by 5pm on the assignment due date, with the file name `<author1PUID>_<author2PUID>_hw1.pdf`. Please only submit one PDF per pair of authors.

We strongly recommend *writing as you go* in the project, which means starting to write the project report as you are downloading and analyzing the data. That said, you should avoid speculative writing, and only write results once you have them.

Extensions

If you would like to extend this assignment to more interesting ground after first completing the basic deliverables for the project, you might consider the following:

- *Extend the data set*: The emails compiled here represent less than 1/5 of the emails in both of these data sets. We have uploaded the full sets of data if you would like to try to take advantage of them. Moreover, there are a number of publicly available spam and email corpora. Processing and incorporating other data sets—including ones you personally compile—and releasing these data with appropriate permissions would be worthwhile. We have separated out training and test data, but feel free to use K-fold cross validation on the larger data sets if you would like.
- *More interesting features*: while we have only asked you to work with a simple word dictionary, there are many extensions to this to consider, including features involving:
  - bigrams, punctuation, proper nouns
  - email length and distribution of length
  - analysis of the URLs included in the emails
  - email headers: email addresses, IP addresses, timestamps
- *More complex classifiers*: there are a number of exciting classifiers that might be used for this task, including, e.g., supervised topic models [Zhu et al., 2009, Mcauliffe and Blei, 2008], conditional tensor factorization [Yang and Dunson, 2013], or something of your own design that might identify latent structure in the data that is predictive of spam/no-spam. *Ensemble classifiers* that combine email classifications from a number of classifiers to improve results may be built from a number of the more simple classifiers used in your basic analyses.
- *More classes of emails*: Google unveiled their new classification of emails to include *Primary*, *Social*, *Promotions*, and *Spam*. You might consider this or something related that would be useful for filtering emails.
- *Better evaluation metrics*: what are better metrics that you might use to evaluate these classifiers? Time wasted reading spam? Critical emails lost in the spam filter? Can you improve model evaluation using cross validation instead of our training and test sets?
- *Additional types of problems*: What about emails that do not have class labels? You might consider developing an active learning method that will ask users to classify emails as spam/not-spam that will, in expectation, reduce uncertainty maximally across all unlabeled emails. What about adaptive spam filters that can be refitted as new types of spam arise?

Resources

There is a large literature on spam filters, including by Princeton Professor Nick Feamster [Ramachandran and Feamster, 2006, Hao et al., 2009, Hao, 2014, Jones et al., 2014, Hao et al., 2013]. Many involve fairly simple classification methods and large numbers of features or reference data sets. There are also a number of reviews available. Review some of this literature to get ideas on ways to create a really great classifier.

References

Shuang Hao. Early detection of spam-related activity. 2014.

Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G Gray, and Sven Krasser. Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine. In *USENIX Security Symposium*, volume 9, 2009.

Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. Understanding the domain registration behavior of spammers. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 63–76. ACM, 2013.

Ben Jones, Tzu-Wen Lee, Nick Feamster, and Phillipa Gill. Automated detection and fingerprinting of censorship block pages. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 299–304. ACM, 2014.

Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *CEAS*, 2004.

Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. *ACM SIGCOMM Computer Communication Review*, 36(4):291–302, 2006.

Yun Yang and David B Dunson. Bayesian conditional tensor factorizations for high-dimensional classification. *arXiv preprint arXiv:1301.4950*, 2013.

Jun Zhu, Amr Ahmed, and Eric P Xing. Medlda: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1257–1264. ACM, 2009.