SVM and Email Classification

Samuel Schetterer

May 6, 2019

1 Email Processing

Email processing takes a few different steps:

- Obtain a dataset of spam and not-spam emails to test the classifier against
- Clean undesireable content (actual emails, headers, etc) from the emails
- Tokenize and stem the emails
- Transform into some mathematical representation, here word vectors

1.1 Dataset

For our email dataset, I acquire emails from the pulib corpus at https://spamassassin.apache.org/old/publiccorp These are easily available for download with little work.

The script I used for this can be found in the data/download_email.sh folder

1.2 Cleaning

The emails are not immediately useful without some cleaning. They contain things like:

- Email headers: "Received: from localhost (jalapeno [127.0.0.1])"
- Urls: www.google.com
- Email addresses
- Punctuation
- Non-letter characters
- Stop words (and, the, ...)

We either strip fo replace this content from the email. Urls are replaced with httpaddr, emails are replaced with emailaddr, and the rest is stripped.

The set of regexes and list of stop words used to clean the emails is in python/process_email.py

1.3 Tokenize

We now tokenize and stem the emails. The stemming is nontrivial, since we want words like running and runs to be map to the same thing. For that, we use the porter-stemmer algorithm. There is an implementation provided with the Matlab source; however, since I don't have access to Matlab, I used the implementation from NLTK (https://github.com/nltk/nltk)