Machine Learning in Spam Filtering

John Levine, IETF 94

Spam filtering

- It’s hard
- Spammers adapt
- So filters have to adapt too

Metadata vs. contents

- Metadata
  - Connecting IP, Envelope to/from, Timing, etc.
  - Usually cheap
- Contents
  - Naughty words, statistical patterns, etc.
  - Always expensive, since it requires message receipt

Content filtering

- Originally static words or regexps
  - viagra
  - v[1i][a@]gr[a@]
  - v[.]*[1i][.]*[a@][.]*g[.]*(a[.]*)?r[.]*[a@]
- Now usually dynamic

Bayesian filtering

- Paul Graham, *A Plan for Spam*
- Bayesian filtering on word sequences

Bayesian filtering

- Tokenize the message
  - All headers? Some headers? Body? Body minus attachments? Decoded attachments?
- Look up word sequences in database, compute score
- Do filtering
- Tune filters when they’re wrong

Filter tuning

- Tune when user reports spam/not-spam
- Auto-tune as mail goes by
- Per system? Per user? Shared among multiple systems?

Spamassassin `spamassassin.apache.org`

- The de facto standard filterware
- Big perl module with plugins
- Fairly sophisticated bayesian filters
- Tuning is your problem via `sa-learn`

Bulk counting

- Characterize message as a checksum
- Razor `razor.sourceforge.net`
  - Spam reported manually or automatically
  - Shared database of checksums of spam
- Distributed Checksum Clearinghouse `www.dcc-servers.net`
  - Count them all
  - Whitelist legit bulk mail
  - IP reputation add-on