Content-based Web Spam Detection

Gordon V. Cormack

University of Waterloo
When you have a hammer ...

Everything looks like a nail!

Hammer:

Content-based *email* spam filters
  Dynamic Markov Compression (DMC)
  Orthogonal Sparse Bigrams (OSBF-Lua)
  Stacking multiple filter results
  Combining results with logistic regression

Nail:

The Web Spam Challenge
Detailed results Task B:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Task B</th>
<th>Average AUC</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.946469</td>
<td><strong>Gordon Cormack</strong>&lt;br&gt;University of Waterloo, Canada</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.918251</td>
<td><strong>Nikolaos Trogkanis</strong>, National Technical University of Athens, Greece&lt;br&gt;<strong>Georgios Paliouras</strong>, National Center of Scientific Research &quot;Demokritos&quot;, Greece</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.907398</td>
<td><strong>Kushagra Gupta</strong>, Vikrant Chaudhary, Nikhil Marwah, Chirag Taneja&lt;br&gt;Inductis India Pvt Ltd</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.899241</td>
<td><strong>D’yakonov Alexander</strong>&lt;br&gt;Moscow State University, Russia</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.893333</td>
<td><strong>Wenyuan Dai</strong>&lt;br&gt; Apex Data &amp; Knowledge Management Lab, Shanghai Jiao Tong University</td>
</tr>
</tbody>
</table>
DMC colors spam red, non-spam green

Spam
- www.lmg2-dvd.co.uk
- www.f2films.co.uk
- www.gifthunt.co.uk
- www.home-loans-online.co.uk
- www.abfinance.co.uk
- www.insurance-quote.co.uk
- irish-swingers.connect4fun.co.uk

Non-spam (normal)
- lib1.leeds.ac.uk
- www.babyfriendly.org.uk
- www.learningservices.org.uk
- www.preparingforemergencies.gov.uk
- www.hintsandthings.co.uk
- www.guardian.co.uk
- www.psnc.org.uk
DMC applied to what?

Text! (actually, any stream of bits)

hostname
  of host to be classified
  of incoming links
  of outgoing links

html content
  page(s) on host  *(which pages?)*
  text, markup, formatting  *(just a bit stream to DMC)*
  excerpts of pages  *(first or last 2500 bytes)*

http server response

10 filters in total (9 DMC, 1 OSBF-Lua)
Combining the 10 filter runs

Each run yields a *spamminess score* \( s_n \) for each host.

Convert to *log-odds* \( L_n \) using training data:

\[
L_n = \log \left( \frac{\left| \{ i \mid s_i \leq s_n \text{ and ith message is spam} \} \right|}{\left| \{ i \mid s_i \geq s_n \text{ and ith message is ham} \} \right|} + \epsilon \right)
\]

Naïve combination

sum \( L_n \) over all runs

Slightly better combination

*logistic regression* to compute weighted sum
## Results (10-fold cross validation)

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>$F_1$</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>homebig</td>
<td>.939</td>
<td>.634</td>
<td>.064</td>
</tr>
<tr>
<td>homebig.tail</td>
<td>.938</td>
<td>.626</td>
<td>.056</td>
</tr>
<tr>
<td>httponly</td>
<td>.867</td>
<td>.481</td>
<td>.124</td>
</tr>
<tr>
<td>bodyonly</td>
<td>.933</td>
<td>.627</td>
<td>.184</td>
</tr>
<tr>
<td>wget</td>
<td>.942</td>
<td>.622</td>
<td>.121</td>
</tr>
<tr>
<td>wget.tail</td>
<td>.942</td>
<td>.619</td>
<td>.135</td>
</tr>
<tr>
<td>wget.osbf</td>
<td>.929</td>
<td>.635</td>
<td>.200</td>
</tr>
<tr>
<td>hostname</td>
<td>.864</td>
<td>.424</td>
<td>.095</td>
</tr>
<tr>
<td>ingraph</td>
<td>.952</td>
<td>.639</td>
<td>.383</td>
</tr>
<tr>
<td>outgraph</td>
<td>.834</td>
<td>.289</td>
<td>.021</td>
</tr>
<tr>
<td>log-odds</td>
<td>.975</td>
<td>.796</td>
<td>-</td>
</tr>
<tr>
<td>logistic</td>
<td>.980</td>
<td>.803</td>
<td>-</td>
</tr>
</tbody>
</table>
Collaborative Proposal

Combine all Web Spam Challenge Submissions!

Really, really naïve approach

spamminess = # spam votes among participants

Naïve approach

requires training results for log-odds calculation

Logistic regression

requires training results for weight calculation

Let's build the ultimate filter

send me your data (training + test)

gvcormac@uwaterloo.ca
Content-based
Web Spam Detection

Gordon V. Cormack

University of Waterloo
Dynamic Markov model (DMC)

This example implements a 1\textsuperscript{st} order Markov model

\textbf{A} means following 0; \textbf{B} means following 1

Outputs \(f\) on edges are frequencies

\[ \text{Prob}(1 \text{ following } A) = \frac{4}{2 + 4} = 0.667 \]

\(f\) incremented after each transition
DMC State Cloning

State A, input 1, Prob 0.67
B visited 16 times previously
  4 from A; 12 from elsewhere
B should be cloned because it is visited from distinct contexts several times

B cloned to create B'
  $f$ divided in 4:12 ratio in proportion to previous visits
  $f$ incremented as usual