On-line Spam Filter Fusion

Thomas Lynam & Gordon Cormack

originally presented at SIGIR 2006
On-line vs Batch Classification

- Batch Hard Classifier
  - separate training and test data sets
  - Given ham/spam classification of training set
  - Compute ham/spam class for each message

- On-line Soft Classifier
  - Chronological sequence
  - Compute *spamminess* for each in sequence
    - ham/spam class by comparing to fixed threshold
  - Given ham/spam classification afterwards
    - Immediate, correct feedback (idealized user)
Measures of Success & Failure

- ROC Curve
- ROC Area *above* the curve (as percentage)
- Ham & spam misclassification rates
  - $Sm(\%)$ when threshold set for $Hm(\%) = .1$
- 95% confidence intervals
  - For ROC area (logit transformed)
  - For difference between ROC areas (logit trans)
    - Significant result: difference interval excludes 0
Pilot Test ROC
(Mr X corpus)
Pilot Tests K Subsets
(Mr X corpus)

![Graph showing performance metrics for different K-subsets.](image-url)
TREC 2005 SPAM TRACK

- 4 corpora
  - 1 public, 3 private
- submit runs on public corpus
- submit filter to be run on private corpora
- 53 runs (different filters)
- 17 different organizations represented
## TREC Spam Track Corpora

<table>
<thead>
<tr>
<th></th>
<th>Ham</th>
<th>Spam</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr X</td>
<td>9038</td>
<td>40048</td>
<td>49086</td>
</tr>
<tr>
<td>S B</td>
<td>6231</td>
<td>775</td>
<td>7006</td>
</tr>
<tr>
<td>T M</td>
<td>150685</td>
<td>19516</td>
<td>170201</td>
</tr>
<tr>
<td>Full</td>
<td>39399</td>
<td>52790</td>
<td>92189</td>
</tr>
<tr>
<td>Aggregate</td>
<td>205253</td>
<td>113129</td>
<td>318482</td>
</tr>
</tbody>
</table>
TREC Filter Performance Distribution

(1−ROCA)\% – Aggregate Pseudo–Corpus

Thomas Lynam and Gordon Cormack - University of Waterloo - SIGIR 2006
Fusion Methods

- Best System (Baseline)
- Voting
- SumScore
- Log-odds Averaging
- SVM
- Logistic Regression
Log-odds Averaging

- 53 unknown systems
  - unknown min/max scores.
  - linear/nonlinear scoring
- How to normalize scores?

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

- **SVM**\textsuperscript{light}
  - default kernel and parameters
  - log-odds averaging used as features
- training set sizes of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000
- output used as spamminess score
Logistic Regression

- LR-TRIRLS logistic regression package
- weights predict prior classification
- Negative weights considered over-fitting
- initial weight equal $\frac{1}{\text{number of filters}}$
- training set sizes of 0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 2100, 4100, 9100, 19100, 39100, 69100, 99100, 129100, 159100.
- weighted average uses as spamminess score
ROC
(Full Corpus)
S B Results

1-ROCA% and sm%@hm%=.1
Aggregate Results

1-ROCA%

sm%@hm%=.1
Subset Experiment

- logistic regression subset selection
  - eliminate smallest filter weight
  - recompute logistic-regression weight
  - repeat
- train on Mr X and S B corpora
- subset size of
  2, 3, 4, 8, 16 ..., largest subset with only positive weights
Training on Mr X Corpus
Results on Full Corpus

1-ROCA%

sm%@hm%=.1

Thomas Lynam and Gordon Cormack - University of Waterloo - SIGIR 2006
## MrX-derived subsets on trec05p-1

<table>
<thead>
<tr>
<th>Subset</th>
<th>(1-ROCA)%</th>
<th>sm%@hm% = .1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrx23</td>
<td>.007*** (.006-.009)</td>
<td>.79*** (.62-.99)</td>
</tr>
<tr>
<td>mrx16</td>
<td>.007*** (.006-.009)</td>
<td>.84*** (.69-1.02)</td>
</tr>
<tr>
<td>mrx8</td>
<td>.009*** (.007-.011)</td>
<td>.88*** (.71-1.08)</td>
</tr>
<tr>
<td>mrx4</td>
<td>.012*** (.009-.015)</td>
<td>1.07*** (.82-1.39)</td>
</tr>
<tr>
<td>mrx3</td>
<td>.012*** (.010-.016)</td>
<td>1.15*** (.92-1.44)</td>
</tr>
<tr>
<td>mrx2</td>
<td>.016 (.012-.021)</td>
<td>1.31** (1.01-1.68)</td>
</tr>
<tr>
<td>best</td>
<td>.019 (.015-.023)</td>
<td>1.78 (1.42-2.22)</td>
</tr>
</tbody>
</table>
Base Filter Participation in Subsets (by Separate Performance)
TREC 06 MrX II Corpus

![Graph showing classification performance]
1-ROCA(%) on Mrx II

- Logodds: 0.196 (.007 - .05)
- Vote: 0.224 (.009 - .05)
- Ofl: 0.363 (.02 - .06)

Significance
- Logodds – Ofl $p < .04$ (96% confidence)
- Vote – Ofl $p < .06$ (94% confidence)
All fusion methods substantially outperformed the best system.
On small corpus SVM and Logistic regression are less effective.
Voting seems more stable.
log-odds essential for other methods.
negative LR weights not always overfitting.
Conclusions

- Voting works surprisingly well
- Log-odds averaging works a little better
- Logistic Regression is slightly better
- SVM is the best for large corpus
- 53 filters not feasible
- predicting good small subsets possible
Future Work

- explore meta analysis
- different methods of score normalization
- apply fusion to other areas
Questions?
<table>
<thead>
<tr>
<th>Subset</th>
<th>(1-ROCA)%</th>
<th>sm%@hm%=.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrx23</td>
<td>.007***</td>
<td>.006-.009</td>
</tr>
<tr>
<td>mrx16</td>
<td>.007***</td>
<td>.006-.009</td>
</tr>
<tr>
<td>mrx8</td>
<td>.009***</td>
<td>.007-.011</td>
</tr>
<tr>
<td>mrx4</td>
<td>.012***</td>
<td>.009-.015</td>
</tr>
<tr>
<td>mrx3</td>
<td>.012***</td>
<td>.010-.016</td>
</tr>
<tr>
<td>mrx2</td>
<td>0.02</td>
<td>.012-.021</td>
</tr>
<tr>
<td>best</td>
<td>0.02</td>
<td>.015-.023</td>
</tr>
</tbody>
</table>
SpamAssassin Corpus ROC curves
Mr X Corpus ROC Curves