Batch and On-line Spam Filter Comparison

Gordon V. Cormack       Andrej Bratko

University of Waterloo

Jozef Stefan Institute
On-line vs Off-line Evaluation

TREC – Text Retrieval Conference (On-line)

- chronological order, immediate feedback
- real email messages (and filters!)
- soft classification: *spamminess score*

Receiver Operating Characteristic (ROC)

Classical Evaluation (Batch)

- $k$-fold cross validation
- contrived email messages (and filters!)
- hard classification: *spam* or *ham*

accuracy, weighted accuracy, Total Cost Ratio (TCR)
Test Methods and Corpora

TREC 2005 Public Corpus

- on-line test (TREC methodology)
- 10-fold cross validation (random splits)
- 9:1 chronological split
  - on-line test sequence
  - batch test set
- tokenized, obfuscated versions of same corpora

Ling Spam & PU1 Corpora

- 10-fold cross validation
- splits, tokenization, obfuscation defined by corpora
Subject Filtering Methods

$X^2$ (Graham/Robinson)

Bogofilter (Relson, Louis et al.)

Support Vector Machine (Vapnik)

$\text{SVM}^{\text{light}}$ (Joachims)

Logistic Regression (Fisher)

$\text{LR-TRIRLS}$ (Komarek)

Prediction by Partial Matching (Cleary & Witten)

Adaptive PPM-D Classifier (Bratko)

Dynamic Markov Modeling (Cormack & Horspool)

Adaptive DMC Classifier (Cormack)
Prediction by Partial Matching

For each class:

- left context occurrences
- left context + prediction
- log-likelihood estimate
- compressed length

Smoothing/backoff:
- zero occurrence problem

Adaptation:
- increment counts
  assuming in-class

Context (509 spam, 1 ham)

Prediction (0 spam, 1 ham)

Prediction (509 spam, 0 ham)
DMC State Cloning

cloning of state B

Cormack & Bratko Batch and On-line Spam Filter Comparison CEAS 2006
TREC Corpus, On-line

![Graph showing performance of different spam filters](image)

- DMC
- PPM
- Bogofilter
- Logistic
- SVM

Cormack & Bratko: Batch and On-line Spam Filter Comparison CEAS 2006
9:1 Chronological, Batch

[Graph showing misclassified spams vs misclassified hams for different methods such as DMC, PPM, Bogofilter, Logistic, and SVM.]
9:1 Chronological, On-line
<table>
<thead>
<tr>
<th>Method</th>
<th>Full Corpus</th>
<th>9:1 Chronological</th>
<th>10-fold C.V.</th>
<th>9:1 Chronological</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMC</td>
<td>.013 (.010-.018)</td>
<td>.0003 (.0000-.003)</td>
<td>.015 (.012-.018)</td>
<td>.003 (.001-.006)</td>
</tr>
<tr>
<td>PPM</td>
<td>.017 (.014-.021)</td>
<td>.0007 (.0001-.005)</td>
<td>.006 (.004-.009)</td>
<td>.003 (.001-.008)</td>
</tr>
<tr>
<td>Bogofilter</td>
<td>.048 (.038-.062)</td>
<td>.002 (.0001-.041)</td>
<td>.020 (.012-.033)</td>
<td>.009 (.003-.029)</td>
</tr>
<tr>
<td>LR</td>
<td>.068 (.058-.079)</td>
<td>.020 (.003-.135)</td>
<td>.016 (.012-.021)</td>
<td>.12 (.001-.10.1)</td>
</tr>
<tr>
<td>SVM</td>
<td>.075 (.064-.088)</td>
<td>.007 (.0015-.033)</td>
<td>.021 (.015-.029)</td>
<td>.13 (.003-5.6)</td>
</tr>
</tbody>
</table>
### Effect of Order/Adaptation

<table>
<thead>
<tr>
<th>Filter</th>
<th>Training Regimen</th>
<th>On-line Random Order</th>
<th>On-line Corpus Order</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMC</td>
<td>Random Order</td>
<td>.01 (.006-.017)</td>
<td>.007 (.004-.011)</td>
<td>.009 (.006-.015)</td>
</tr>
<tr>
<td>DMC</td>
<td>Corpus Order</td>
<td>.035 (.026-.047)</td>
<td>.037 (.024-.057)</td>
<td>.31 (.25-.37)</td>
</tr>
<tr>
<td>PPM</td>
<td>Batch</td>
<td>.0052 (.003-.01)</td>
<td>.0053 (.003-.009)</td>
<td>.0055 (.003-.01)</td>
</tr>
</tbody>
</table>
Tokenization, On-line
<table>
<thead>
<tr>
<th>Method</th>
<th>On-line</th>
<th>Batch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full Corpus</td>
<td>9:1 Chronological</td>
</tr>
<tr>
<td>tokenized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMC</td>
<td>.013 (.010-.018)</td>
<td>.0003 (.0000-.003)</td>
</tr>
<tr>
<td>obfuscated</td>
<td>.025 (.020-.032)</td>
<td>.0006 (.0001-.006)</td>
</tr>
<tr>
<td>tokenized</td>
<td>.037 (.030-.045)</td>
<td>.0004 (.0000-.0042)</td>
</tr>
<tr>
<td>PPM</td>
<td>.017 (.014-.021)</td>
<td>.0007 (.0001-.005)</td>
</tr>
<tr>
<td>tokenized</td>
<td>.038 (.033-.045)</td>
<td>.0016 (.0003-.009)</td>
</tr>
<tr>
<td>obfuscated</td>
<td>.075 (.066-.084)</td>
<td>.0046 (.0016-.013)</td>
</tr>
<tr>
<td>Bogofilter</td>
<td>.048 (.038-.062)</td>
<td>.002 (.0001-.041)</td>
</tr>
<tr>
<td>obfuscated</td>
<td>.13 (.11-.15)</td>
<td>.024 (.004-.14)</td>
</tr>
</tbody>
</table>
Other Methods

Gradient Descent Logistic Regression  \textit{(Goodman)}

On-line Filter Fusion  \textit{(Lynam \& Cormack)}

Classical machine learning  \textit{(Cast of thousands)}

Naïve Bayes (which naïve Bayes?)

kNN

Perceptron

Winnow

Decision trees

Boosting

Stacking
Ling Spam Corpus

The graph shows the comparison of various spam filters for the Ling Spam Corpus. The x-axis represents the number of misclassified hams (of 2412), while the y-axis indicates the number of misclassified spams (of 481). Different markers and colors are used to represent various filter methods, such as DMC, PPM, Bogofilter, Logistic, SVM, a-Bayes, a-kNN, a-Stack, b-Stack, gh-Eval, p-Suffix, and s-Event.
Conclusions

Batch and on-line are different

    good filters can be adapted to do both well

Feature engineering is important

    email is not just a bag or sequence of tokens

Real filters beat contrived ones

    even on contrived corpora

PPM and DMC effectively filter spam

    fast (100s of messages/sec)
    voracious appetite for RAM (0.5 – 2.0 GB)