Using GNUsmail to Compare Data Stream Mining Methods for On-line Email Classification

José M. Carmona-Cejudo\textsuperscript{1}, Manuel Baena-García\textsuperscript{1}, José del Campo-Ávila\textsuperscript{1}, João Gama\textsuperscript{2}, Albert Bifet\textsuperscript{3} and Rafael Morales-Bueno\textsuperscript{1}

\textsuperscript{1}Universidad de Málaga, Spain
\textsuperscript{2}University of Porto, Portugal
\textsuperscript{3}University of Waikato, New Zealand

Manuel Baena-García
Castro Urdiales, October 2011
Email mining

- Spam detection: a two-class problem usually solved with bayesian classifiers.
- Email classification: a multi-class problem to sort email into folders.

Email classification approaches

- Batch learning. The whole dataset is available before the beginning of the learning process.
- Online learning. Data are continually being received and processed over time.

Hypothesis

There is a lack of systems to compare and evaluate different machine learning models for email classification.
**Email mining**

- Spam detection: a two-class problem usually solved with Bayesian classifiers.
- Email classification: a multi-class problem to sort email into folders.

**Email classification approaches**

- Batch learning. The whole dataset is available before the beginning of the learning process.
- Online learning. Data are continually being received and processed over time.

**Hypothesis**

There is a lack of systems to compare and evaluate different machine learning models for email classification.
Introduction

**Context**

**Email mining**
- Spam detection: a two-class problem usually solved with bayesian classifiers.
- Email classification: a multi-class problem to sort email into folders.

**Email classification approaches**
- Batch learning. The whole dataset is available before the beginning of the learning process.
- Online learning. Data are continually being received and processed over time.

**Hypothesis**
There is a lack of systems to compare and evaluate different machine learning models for email classification.
GNUsmail is a framework that allows to compare different email classification algorithms.

We introduce next improvements to GNUsmail:

- to carry out *replicable experimentation*.
- to evaluate data stream mining methods by using:
  - sliding windows.
  - fading factors.
- to use recently proposed statistical tests to compare the performance of online algorithms.
GNUsmail: Architecture and Characteristics

http://code.google.com/p/gnusmail/

**Characteristics**

- Open source framework for online adaptive email classification.
- It contains modules for reading email, preprocessing text and learning.
- The email messages are read as the model is built.

**Architecture**

- **Reading email module** can obtain email messages from local filesystem or remote IMAP server.
- **Text processing module** based on filters that extract attributes from emails.
- **Learning module** into which new algorithms, methods and libraries can be integrated.
Text Preprocessing Module

Structure

- A pipeline of (linguistic) operators which extract relevant features from every mail.
- Some ready-to-use filters are implemented as part of the GNUsmail core, and new ones can be incorporated.

Filters

- Relevant words based on the ranking provided by the tf-idf function.
- Sender, CC.
- Domain of sender.
- Capital letters proportion.
- Language.
- Number of receivers.
Learning Module

Structure

Based on WEKA and MOA frameworks:

- WEKA methods are used with small datasets in environments without time and memory restrictions.
- MOA methods are used in more demanding problems.

WEKA methods

- Multinomial Naïve Bayes
- IBk, k-nearest neighbours
- NN-ge (Nearest Neighbour with Generalised Exemplars)

MOA methods
Learning Module

Structure
Based on WEKA and MOA frameworks:

- WEKA methods are used with small datasets in environments without time and memory restrictions.
- MOA methods are used in more demanding problems.

Weka Methods

MOA Methods
GNUsmail uses MOA by including its tools for evaluation, classification, and drift detection.

- HoeffdingTree
- OzaBag, OzaBoost
- DDM
In data stream contexts, neither cross-validation nor other sampling procedures are suitable for evaluation.

**Prequential measures**
- A prediction is made for each new example.
- Once the real class is known we update a cumulative loss function.

**Forgetting mechanisms**
- Sliding windows.
- Fading factors (preferred method).
**Comparing the performance**

**Adapted McNemar statistic** ($M$)

$$M_i = \text{sign}(a_i - b_i) \times \frac{(a_i - b_i)^2}{a_i + b_i}$$

- $a_i = f_i + \alpha \cdot a_{i-1}$
- $b_i = g_i + \alpha \cdot b_{i-1}$
- $f_i$: 1 iff the example $i$ is misclassified by the first classifier and not by the second one (0 otherwise).
- $g_i$: 1 iff the example $i$ is misclassified by the second classifier and not by the first one (0 otherwise).
**Experimental Setup**

**Initial setup**
- Based on ENRON email dataset.
- We have selected seven specific users.
- We have used only topic folders with more than two messages.
- GNUsmail checks the availability of data, offering to download it.
- The messages are analyzed in chronological order.

**Attributes**
- Sender username, sender domain.
- Number of recipients, body length, capital letters proportion, size of email, subject length.
- Most relevant words.
**Experimental Setup**

**Algorithms**
- OzaBag over NNge, using DDM for concept drift detection.
- NNge.
- Hoeffding Trees.
- Majority class.

**Output**
- GNUsmail plots...
  - the prequential-based metrics
  - and the adapted McNemar test
  ... to visually analyse the differences in performance.
## Results

Folder-wise prequential accuracies with bagging of DDM and NN-ge

<table>
<thead>
<tr>
<th>Folder</th>
<th>Correct/Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>beck-s (101 folders)</td>
<td>1071/1941</td>
<td>55.18%</td>
</tr>
<tr>
<td>europe</td>
<td>131/162</td>
<td>80.86%</td>
</tr>
<tr>
<td>calendar</td>
<td>104/123</td>
<td>84.55%</td>
</tr>
<tr>
<td>recruiting</td>
<td>89/114</td>
<td>78.07%</td>
</tr>
<tr>
<td>doorstep</td>
<td>49/86</td>
<td>56.97%</td>
</tr>
<tr>
<td>kaminsky-v (41 folders)</td>
<td>1798/2699</td>
<td>66.62%</td>
</tr>
<tr>
<td>universities</td>
<td>298/365</td>
<td>81.64%</td>
</tr>
<tr>
<td>resumes</td>
<td>420/545</td>
<td>77.06%</td>
</tr>
<tr>
<td>personal</td>
<td>154/278</td>
<td>55.4%</td>
</tr>
<tr>
<td>conferences</td>
<td>163/221</td>
<td>73.76%</td>
</tr>
<tr>
<td>lokay-m (11 folders)</td>
<td>1953/2479</td>
<td>78.78%</td>
</tr>
<tr>
<td>tw_commercial_group</td>
<td>1095/1156</td>
<td>94.72%</td>
</tr>
<tr>
<td>corporate</td>
<td>345/400</td>
<td>86.25%</td>
</tr>
<tr>
<td>articles</td>
<td>152/232</td>
<td>65.51%</td>
</tr>
<tr>
<td>enron.t_s</td>
<td>86/176</td>
<td>48.86%</td>
</tr>
<tr>
<td>williams-w3 (18 folders)</td>
<td>2653/2778</td>
<td>95.5%</td>
</tr>
<tr>
<td>schedule_crawler</td>
<td>1397/1398</td>
<td>99.91%</td>
</tr>
<tr>
<td>bill_williams_iii</td>
<td>1000/1021</td>
<td>97.94%</td>
</tr>
<tr>
<td>hr</td>
<td>74/86</td>
<td>86.05%</td>
</tr>
<tr>
<td>symsees</td>
<td>74/81</td>
<td>91.36%</td>
</tr>
</tbody>
</table>
Replicable Experimentation

**RESULTS FOR BECK-S**

![Prequential based results for beck-s](image)

(a) Prequential error

(b) Fading factors preq. ($\alpha = 0.995$)

**FIGURE:** Prequential based results for beck-s
RESULTS FOR KITCHEN-L

Figure: Prequential based results for kitchen-l

(a) Prequential error

(b) Fading factors preq. (\(\alpha = 0.995\))
**Adapted McNemar test**

Figure: OzaBag vs. NN-ge using fading factors with $\alpha = 0.995$
Adapted McNemar test

**Figure**: OzaBag vs. Hoeffding tree using fading factors with $\alpha = 0.995$
Conclusion and Future Work

- We have presented different methods to evaluate data stream algorithms.
- We have incorporated to GNUsmail recently proposed evaluation methods.
- Such evaluation methods improve prequential error measures.
- McNemar test is adequate as a tool to compare the online performance in the domain of email classification.
- Current online learning algorithm implementations needs to known all the attributes before the learning itself.
- Future methods should support online addition of new features.
Using GNUsmail to Compare Data Stream Mining Methods for On-line Email Classification

José M. Carmona-Cejudo\textsuperscript{1}, Manuel Baena-García\textsuperscript{1}, José del Campo-Ávila\textsuperscript{1}, João Gama\textsuperscript{2}, Albert Bifet\textsuperscript{3} and Rafael Morales-Bueno\textsuperscript{1}

\textsuperscript{1}Universidad de Málaga, Spain
\textsuperscript{2}University of Porto, Portugal
\textsuperscript{3}University of Waikato, New Zealand

Manuel Baena-García
Castro Urdiales, October 2011