A Selective Learning Model for Spam Filtering

Didier Colin, Catherine Roucairol, Ider Tseveendorj
Prism Laboratory
University of Versailles Saint-Quentin en Yvelines
France

March 26, 2009
Motivations of this work

The selective learning model

Experiments and results

Online application

Conclusion
The spam filtering problem

- Two approaches for spam filtering:
  - Knowledge engineering
  - Machine learning: text classification

- Spam filtering is not a typical text classification problem:
  - Adversarial classification: Classifying against an opponent who will try to delude/break the filter
  - Need for autonomy: Maintaining accuracy over time with minimal human intervention
  - False-positive issue: No acceptable false positive rate
Idea

- Learning all messages is generally a bad idea
- Assumption: existence of a harmful knowledge
- Basic idea: identify these messages and do not learn them
- *Formulate the learning process as an optimization problem, and introduce a decision variable*
- Purposes:
  - Protect the filter against deluding strategies
  - Provide better behaviour over time by preventing natural degeneration of the filter
  - Give the filter better generalization capability
Why a selective approach?

- Human communications are inherently redundant
- Human languages often contain misleading informations
- Especially true in the case of spam (repetitive commercial strategies, deceptive messages)
- These characteristics may be difficult to capture in a feature selection scheme
Problem formulation

- Problem formulation: finding a training subcorpus such that training on it maximizes the resulting filter’s accuracy on the evaluation corpus
- A typical corpus: $10^3$ to $10^6$ learning messages
- A typical classifier learns in polynomial time
Problem formulation

- Problem formulation: finding a training subcorpus such that training on it maximizes the resulting filter’s accuracy on the evaluation corpus
- A typical corpus: $10^3$ to $10^6$ learning messages
- A typical classifier learns in polynomial time

→ we opt for a meta-heuristic implementation
Implementation

- Genetic implementation
- Data: a set of messages $C$, a classifier $f$
- Representations
  - Solution: boolean vector $X$ of dimension $|C|$, $X_i = 1$ if message $i$ is selected
  - Fitness: $A(f_{C(X)}, C)$, weighted accuracy of resulting filter on the set $C$, $C(X) = \{c_i \in C | X_i = 1\}$
- Operations
  - Selection: elitist
  - Cross-over: one point
  - Mutation: random bit inversion
Genetic operations
Genetic operations

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & \ldots & 0 \end{bmatrix} \]
Genetic operations

\[
X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}
\]

\[
Y = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}
\]
Genetic operations

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]
\[ Y = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \]
\[ X' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \]
\[ Y' = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \]

One-point cross-over
Genetic operations

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]
\[ Y = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \]
\[ X' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \]
\[ Y' = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \]

One-point cross-over

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]
Genetic operations

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]

\[ Y = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \]

\[ X' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \]

\[ Y' = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \]

One-point cross-over

\[ X = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]

\[ X'' = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]

Mutation
Experiments protocol

- Data sets: lingspam corpus¹ (481 spams, 2412 legitimate messages), SpamAssassin (1897 spams, 4150 legitimate messages)

- Classifier: Bernoulli naive bayesian, 60 words vocabulary

- Parameters:
  - population size: 10 to 100
  - mutation rate: 5 to 75
  - initial solutions: random selection of 10% legitimate message and 50% spam

- Metric: Total Cost Ratio = \( \frac{A(f_C(x), C)}{A(f_{\emptyset}, C)} \)

Results: TCR evolution for various population size
Results: TCR evolution for a population of 25 individuals
Results: Overview

Table: Comparison of spam precision and spam recall for exhaustive and selective learning algorithm

<table>
<thead>
<tr>
<th></th>
<th>Exhaustive learning</th>
<th>Selective learning (initial)</th>
<th>Selective learning (best)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>96.82 %</td>
<td>96.85 %</td>
<td>98.72 %</td>
</tr>
<tr>
<td>Recall</td>
<td>88.33 %</td>
<td>89.60 %</td>
<td>96.47 %</td>
</tr>
</tbody>
</table>

- Better solutions found at the first iteration
- TCR improved by a factor 4
- Best solutions contain only 1/3 of the lingspam corpus
Results on SpamAssassin

Bernoulli naive bayesian performs bad (TRC < 1)
Initial solutions must be almost exhaustive
Selective learning do not bring much improvement
Online selective learning

- Initial learning is only half of the job
- Is online selective learning possible?
- Assuming no-user feedback
- Corpus → flow
- For each incoming message, a decision problem: shall we learn it?
- Idea: for each incoming message, test if learning this message improves the filter’s precision over the N previous messages (learning window)
Online selective learning algorithm

**Input:** $W_i$, the i-th message on the mail flow, $f$, a classifier, $N$, an integer

```
begin
  $f' \leftarrow \text{copy}(f)$
  if $f(W) \geq \lambda$
    then learn($f'$, $W$, spam)
  else learn($f'$, $W$, ham)
  $C \leftarrow \{W_j, i - N \leq j \leq i\}$
  if $A(f, C) \geq A(f', C)$
    then return false
  else return true
end
```

**Algorithm 1:** Online selective learning

Didier Colin, Catherine Roucairol, Ider Tseveendorj
Prism Laboratory
University of Versailles Saint-Quentin en Yvelines
France

A Selective Learning Model for Spam Filtering
TCR evolution, regular lingspam

- Little to no improvements
- Slight loss for window = 50, 25
- Slight gain for window = 500
- But global evolution is even
- Easy mail flow $\rightarrow$ conservative learning strategies
TCR evolution, noisy lingspam (5%)
Conclusions

- A learning model specifically designed to address the issues of spam filtering
- Easy to implement...
- Good synergy with existing techniques
- Not tied to a specific classification model
Perspectives and future works

- Efficient heuristics for initial solutions?
- Make use of non learned data
- Dynamic variations of online selective window
Thank you!