Historic Overview

- 2003-now: Second Generation Bayesian Filters
- 2001-02: First Generation Bayesian Filters
- 1998-2000: Advanced Heuristic Filters
- 1994-97: Primitive Heuristic Filters

Jose Maria Gomez Hidalgo

http://www.esp.uem.es/~jmgomez
Primitive Heuristic Filters

Hand coding simple IF-THEN rules

if "Call Now!!!" occurs in message
then it is spam

Manual integration in server-side processes (procmail, etc.)

Require heavy maintenance

For many, first commercial spam filtering solution

BrightMail’s MailWall (now in Symantec)

Wiser hand-coded spam AND legitimate tests

Burdensome user feedback (private email)

1994-97 Primitive Heuristic Filters

Advanced Heuristic Filters

1998-2000 Advanced Heuristic Filters

Wiser decision = require several rules to fire

BrightMail’s MailWall (now in Symantec)

For many, first commercial spam filtering solution

Hand coding simple IF-THEN rules

1994-97 Primitive Heuristic Filters

Techniques

Hand coding simple IF-THEN rules

Manual integration in server-side processes

Procmail, etc.

Require heavy maintenance

For many, first commercial spam filtering solution

Burdensome user feedback (private email)
Advanced Heuristic Filters

- Used by the very spammers to test their spam
- Open source and widely used spam filtering solution
- Caveats
  - Used by the very spammers to test their spam
  - Limited adaptation to users' email

SpamAssassin

- Open source and widely used spam filtering solution
- Uses a combination of techniques
- Blacklisting, heuristic filtering, now Bayesian filtering, etc.
- Tests contributed by volunteers
- Tests scores optimized manually or with genetic programming

Caveats

- Used by the very spammers to test their spam
- Limited adaptation to users' email
Advanced Heuristic Filters

SpamAssassin tests samples

Some techniques given up by spammers

HTML obfuscation

Percentage of spam email in a collection

Finding the test(s) long along time

Success

They interpret it as a

Courtesy of Steve Webb

Pu06

Advanced Heuristic Filters
First Generation Bayesian Filters

First Generation Bayesian Filters Overview

- Machine Learning spam-legitimate email characteristics from examples
- (Simple) tokenization of messages into words
- Machine Learning algorithms (Naïve Bayes, C4.5, Support Vector Machines, etc.)
- Batch evaluation
- Fully adaptable to user email – accurate
- Combisable with other techniques

First Generation Bayesian Filters

Spammers still trying to guess how to defeat them

A hit

Popularized by Paul Graham’s “A Plan for Spam”

Early research work by Androstopoulos,

Proposed by [Sahami98] as an application of Text Categorization

2001-02 First Generation Bayesian Filters
First Generation Bayesian Filters

- Tokenization [Graham92]
  - Scan all message = headers, HTML, Javascript
  - Token constituents
    - Alphanumeric characters, dashes, apostrophes, and dollar signs
    - Ignore HTML comments and all number tokens
    - Tokens occurring less than 5 times in training corpus
    - Case

- Breaking messages into pieces
- Defining the most relevant spam and legitimate features
- Feeding learning with appropriate information

[Reference: Baldwin98]
First Generation Bayesian Filters

Bayesian learning [Grahams02]

Message Probability

\[
\frac{\prod_{T \in TM} p(T)}{\sum_{T \in TM} p(T)} = p(S | T)
\]

Token Probability

\[
\frac{S}{ST} + \frac{L}{LT} + \frac{2.1T}{ST} = p(T)
\]

- Neural Networks (e.g., Perceptron)
- Statistical learners (e.g., Support Vector Machines)
- Lazy learners (e.g., K Nearest Neighbors)
- Rule based classifiers (e.g., Ripper)
- Decision trees (e.g., C4.5)
- Probabilistic (Bayesian and Markovian) methods
- Dozens of algorithms and classification functions
- E.g.: Building rules algorithmically instead of by hand
- Inducing a classifier automatically from examples
- Learning
First Generation Bayesian Filters

Batch evaluation – Technical Literature

- Focus on end-user features including accuracy
- Usually accuracy and error, sometimes weighted
- False positives (blocking ham) worse than false negatives
- Not allowed training on errors or test messages
- Undisclosed test collection => Non reproducible tests

Other features

- Operation regime: train and test
- Accuracy = hits / trials
- Accuracy metrics
- Early training / test collections
- Usually focused on accuracy
- Required for filtering quality assessment
- Prize, ease of installation, efficiency, etc.
Batch evaluation – Technical [Anderson04]
First Generation Bayesian Filters

Batch evaluation – Research literature

Focus 99% on accuracy

Accuracy metrics

Increasingly account for unknown costs distribution

Private email user may tolerate some false positives

A corporation will not allow false positives on e.g. orders

Slope ranges and convex hull

Data points obtained by 10-fold cross validation

For thresholds (P(spam) > T)

Plots for an algorithm over a number of cost conditions

Spam captured under few False Positives

X = False Positive Rate, Y = True Positive Rate

ROC Convex Hull analysis

Comparing several learning algorithms under unknown costs, simple tokenization, Lingspam

Standardized test collections

PU1, Lingspam, SpamAssassin Public Corpus

Operation regime

Train and test, cross validation (Machine Learning)
First Generation Bayesian Filters

Batch evaluation – Research [Gomez02]

ROC curves Slope ranges
FPR between 0 and 0.004 => Support Vector Machines lead
FPR between 0.004 and 0.012 => Naive Bayes leads

2003-now Second Generation Bayesian Filters

Second Generation Bayesian Filters

Significant improvements on Data processing
Tokenization and token combination
Filter evaluation
Filters reaching 99.987% accuracy (one error in 7,000)

"We have got the winning hand now"

[Zdziarski05]

"7,000"

Filters
Second Generation Bayesian Filters

Unified chain processing

Note remarkable similarity with KDD process

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96

Fayyad96
Second Generation Bayesian Filters

Preprocessing (1)

- Character set folding
  - Forcing the character set used in the message to the character set deemed "most meaningful" to the end user: Latin-1, etc.

- Case folding
  - Removing case changes

- MIME normalization
  - Unpacking MIME encodings to a reasonable representation (specially BASE64)

Preprocessing (2)

- HTML defacement
  - Dealing with "hypertextus interruptus" and use font and foreground colors to hide hopefully dis-incriminating keywords

- Lookalike transformations
  - Dealing with substitute characters like using '@' instead of 'a', 'i' or 'l', and '$' instead of 'S'
Second Generation Bayesian Filters

Tokenization

Token = string matching a Regular Expression

Examples (CRM111) [Siefkes04]

Simple tokens = a sequence of one or more printable characters

HTML/XML/Etc. = the previous one + typical HTML/XML mark

Improvement up to 25%

Example: Orthogonal Sparse Bigrams

Building tuples from isolated tokens, seeking precision, concept identification, etc.

Example: Orthogonal Sparse Bigrams

Pairs of items in a window of size N over the text.

Building tuples from isolated tokens, seeking precision, concept identification, etc.

Tuple based combination

Filters

Second Generation Bayesian Filters

HTML/Etc.

Doctype declarations: <!DOCTYPE

Start/end/empty tags: <tag> </tag> <br/>

Character

Simple tokens = a sequence of one or more printable characters

Examples (CRM111) [Siefkes04]

Token = string matching a Regular Expression

Tokenization
Second Generation Bayesian Filters

- Tuple based combination [Zdziarski05]
  - Example: Bayesian Noise Reduction
  - Training
    - Compute sequences values according Graham's
  - Instantiation
    - Compute token values according Graham’s formulae and round them to the nearest 0.05
  - Build patterns = probabilities sequences
  - Provide new tokens (probability patterns) and filters
  - Out noisy ones

Provide new tokens (probability patterns) and filters
Out noisy ones

Training
Compute sequences values according Graham's

Instantiation
Compute token values according Graham’s formulae and round them to the nearest 0.05
Build patterns = probabilities sequences

Example: Bayesian Noise Reduction
Provide new tokens (probability patterns) and filters
Out noisy ones

Second Generation Bayesian Filters

Tuple based combination [Zdziarski05]

Example: Bayesian Noise Reduction
Provide new tokens (probability patterns) and filters
Out noisy ones

Second Generation Bayesian Filters

Tuple based combination [Zdziarski05]

Example: Bayesian Noise Reduction
Provide new tokens (probability patterns) and filters
Out noisy ones
Applying the threshold learned on training

Learning: final thresholding
Bayes rule, Winnow's linear combination
Combining token weights to single scores

Learning: weight combination
Graham probabilities, increasing Winnow weights, etc.
Accounting for # messages = time (confidence)
Probabilistic smoothing (added constants)
Weight of a token/tuple according to dataset

Filters
Second Generation Bayesian

<table>
<thead>
<tr>
<th>Context value</th>
<th>0.05</th>
<th>0.35</th>
<th>0.65</th>
<th>0.70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context value</td>
<td>0.05</td>
<td>0.35</td>
<td>0.65</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Example: Bayesian Noise Reduction
Tuple based combination [Zdzieszowski]
Second Generation Bayesian Filters

**Accuracy evaluation**
- Online setup
- Resembles normal end-user operation of the filter
- Sequentially training on errors – time ordering

As used in TREC Spam Track [Cormack05]

**Functions allowed**
- initialize
- train spam message
- train ham message
- classify message
- finalize
- Output by the TREC Spam Filter Evaluation Toolkit

**Sensible simulation of message sequence**
- Single metric = the Area Under the ROC curve (AUC)
- Metrics = ROC plotted along time
- By far, the most reasonable evaluation setting

**Analysis**
- Resembles normal end-user operation of the filter
- Online setup
- Accuracy evaluation

---

**TREC evaluation operation**
- Output by the TREC Spam Filter Evaluation Toolkit
- Functions allowed
- initialize
- classify message
- train ham message
- train spam message
- finalize
Second Generation Bayesian Filters

TREC corpora design and statistics
- ENRON messages labeled by bootstrapping
- Second Generation Bayesian Filters
- TREC example results = ROC curve
- Gold
- Silver
- Bronze
- CRM111
- Jozef Stefan Institute
- Laird Breyer
- CRM111
- Silver
- Jozef Stefan
- Gold

General statistics
- Using several filters labeled by bootstrapping
- ENRON messages
- TREC corpora design and statistics

Filters

Second Generation Bayesian

```
<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>169694</th>
<th>606939</th>
<th>226993</th>
</tr>
</thead>
<tbody>
<tr>
<td>100102</td>
<td>119616</td>
<td>606939</td>
<td>226993</td>
<td></td>
</tr>
<tr>
<td>7006</td>
<td>6234</td>
<td>2717</td>
<td>4004</td>
<td>140066</td>
</tr>
<tr>
<td>1996</td>
<td>9082</td>
<td>6921</td>
<td>1144</td>
<td>140066</td>
</tr>
<tr>
<td></td>
<td>4008</td>
<td>9082</td>
<td>1144</td>
<td>140066</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9082</td>
<td>1144</td>
<td>140066</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Second Generation Bayesian Filters

Attacks to Bayesian filters [Zdziarski05]

Preprocessing and tokenization

Encoding guilty text in Base64

Abusing URL encodings

Tortes, etc. dividing spamish words

HTML comments ("Hipertexts Interruptus", small)

Jozef Stefan Institute

Gold

CRM111

Silver

Laird Breyer

Bronze

All phases attacked by the spammers

See The Spammers Compendium [GraCum06]

Filtering Example Results = AUC Evolution

Second Generation Bayesian Filters
Second Generation Bayesian Filters

Attacks to Bayesian filters [Zdziarski05]

Mailing list – learning Bayesian ham words and sending spam – effective once, filters learn

Bayesian poisoning – more clever, injecting invented words in invented header, making filters learn new

Statistically resistant to most attacks

Strongly dependent on actual user corpus

Current Bayesian filters highly effective

Weight combination (decision matrix)

Random words, word salad, directed word attacks

Image spam

Fall in cost effectiveness – effective for 1 user

Why spam still increasing?

Widespread and effectively combined

all users, all filters, all the time

They can defeat one user, one filter, once; but not

Conclusion and reflection

Current Bayesian filters highly effective

Strongly dependent on actual user corpus

Statistically resistant to most attacks

Widespread and effectively combined

all users, all filters, all the time

Why spam still increasing?

Filter to Bayesian filters [Zdziarski05]

Second Generation Bayesian
Questions?

• CEAS 2006 – http://www.cea5.cc
• Do not miss upcoming events