

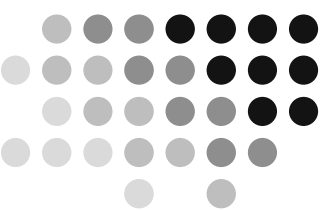
History, Techniques and Evaluation of Bayesian Spam Filters

José María Gómez Hidalgo

Computer Systems

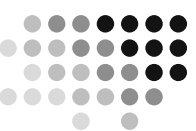
Universidad Europea de Madrid

<http://www.esp.uem.es/~imgomez>

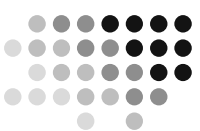


Historic Overview

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



Primitive Heuristic Filters



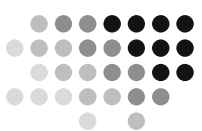
- 1994-97 Primitive Heuristic Filters
 - Hand coding simple IF-THEN rules
 - if “Call Now!!!” occurs in message then it is spam
 - Manual integration in server-side processes (procmal, etc.)
 - Require heavy maintenance
 - Low accuracy, defeated by spammers obfuscation techniques

Advanced Heuristic Filters



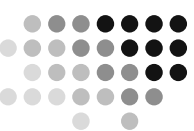
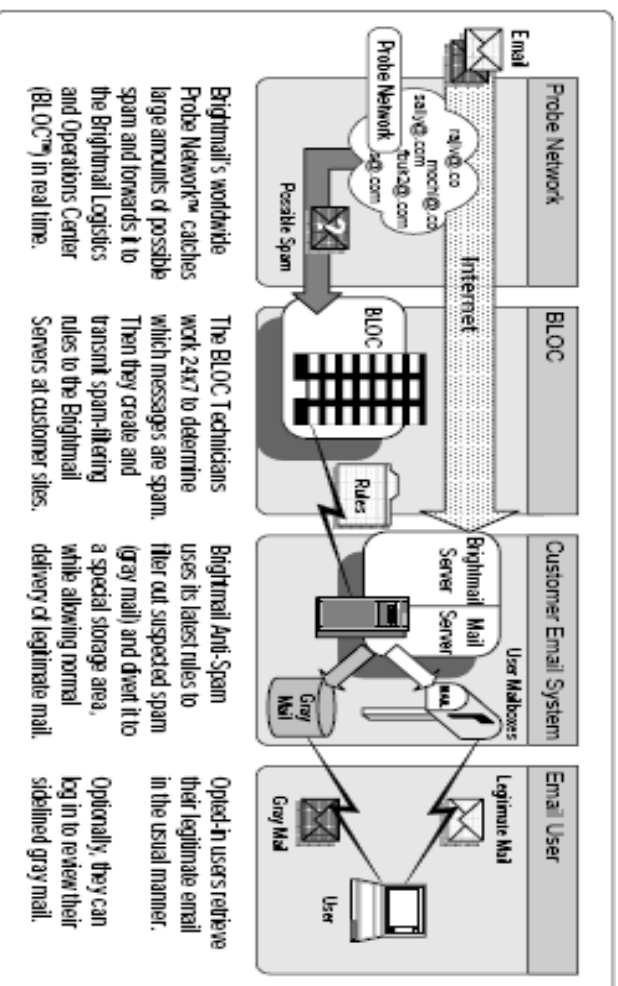
- 1998-2000 Advanced Heuristic Filters
 - Wiser hand-coded spam AND legitimate tests
 - Wiser decision = require several rules to fire
 - Brightmail’s Mailwall (now in Symantec)
 - For many, first commercial spam filtering solution
 - Network of spam traps for collecting spam attacks
 - Team of spam experts for building tests (BLOC)
 - Burdensome user feedback (private email)





Advanced Heuristic Filters

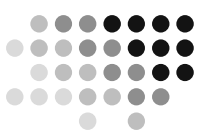
- Mailwall processing flow [Brightmail02]



Advanced Heuristic Filters

- 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

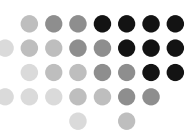




Advanced Heuristic Filters

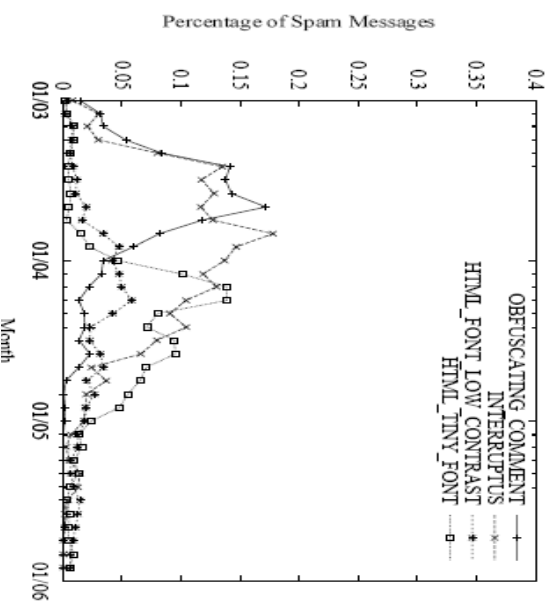
- SpamAssassin tests samples

AREA TESTED	LOCALE	DESCRIPTION OF TEST	TEST NAME	DEFAULT SCORES (local, net, with bayes, with bayes+net)
body		Generic Test for Unsolicited Bulk Email	GTUBE	1000,000
body		Incorporates a tracking ID number	TRACKER_ID	2,000 1,295 2,292 1,032
body		Weird repeated double-quotation marks	WEIRD_QUOTING	1,120 1,200 1,295 1,341
rawbody		Extra blank lines in base64 encoding	MIME_BASE64_BLANKS	0 0 0,184 0,224
rawbody		base64 attachment does not have a file name	MIME_BASE64_NO_NAME	0 0 0,224
rawbody		Message text disguised using base64 encoding	MIME_BASE64_TEXT	2,048 1,522 2,749 1,885
rawbody		MIME section missing boundary	MIME_MISSING_BOUNDARY	1
body		Missing blank line between MIME header and body	MISSING_MIME_HDR_SEP	1
body		Multipart message mostly text/html MIME	MIME_HTML_MOSTLY	1,703 0,699 2,309 1,102
body		Message only has text/html MIME parts	MIME_HTML_ONLY	0,414 0,001 0,389 0,001
rawbody		Quoted-printable line longer than 76 chars	MIME_QP_LONG_LINE	0,159 0 0,234 0
body		HTML and text parts are different	MPART_ALT_DIFF	0,425 0,137 1,142 0
body		HTML and text parts are different	MPART_ALT_DIFF_COUNT	1,649 0 1,607 0,708
body		MIME character set is an unknown ISO charset	MIME_BAD_ISO_CHARSET	3,360 3,360 3,885 4,185
body		Character set indicates a foreign language	CHARSET_FARAWAY	3,200
body		Body contains a ROT13-encoded email address	EMAIL_ROT13	1,600 1,680 1,850 2,000
body		Message body has 70-80% blank lines	BLANK_LINES_70_80	1,499 1,236 1,757 1,805
body		Message body has 80-90% blank lines	BLANK_LINES_80_90	0,272 0,107 0,810 0

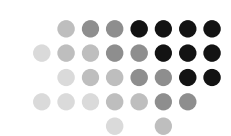


Advanced Heuristic Filters

- SpamAssassin tests along time
 - HTML obfuscation
 - Percentage of spam email in a collection firing the test(s) along time
- Some techniques given up by spammers
 - They interpret it as a success
- Courtesy of Steve Webb [Pu06]

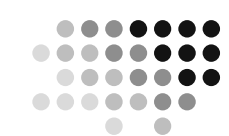


First Generation Bayesian Filters



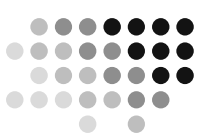
- 2001-02 First Generation Bayesian Filters
- Proposed by [Sahami98] as an application of Text Categorization
- Early research work by Androtsoupoulos, Drucker, Pantel, me :-)
- Popularized by Paul Graham's "A Plan for Spam"
- A hit
- Spammers still trying to guess how to defeat them

First Generation Bayesian Filters

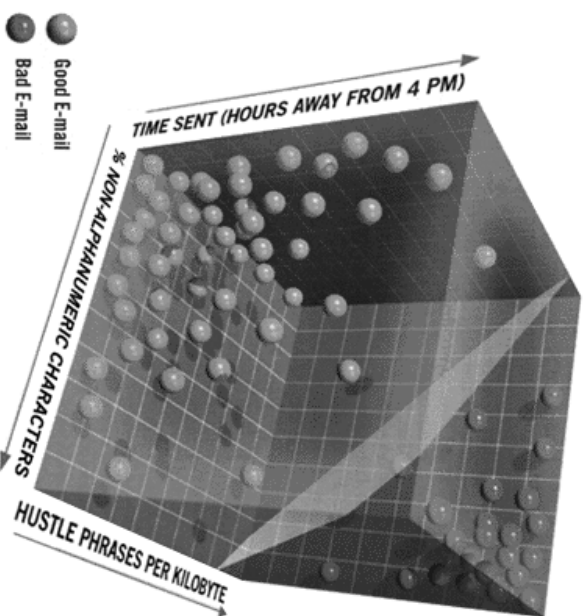


- First Generation Bayesian Filters Overview
- Machine Learning spam-legitimate email characteristics from examples
 - (Simple) tokenization of messages into words
 - Machine Learning algorithms (Naïve Bayes, C4.5, Support Vector Machines, etc.)
 - Batch evaluation
- Fully adaptable to user email – accurate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

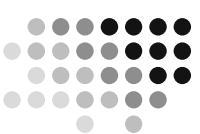
First Generation Bayesian Filters



- Tokenization
- Breaking messages into pieces
 - Defining the most relevant spam and legitimate features
- Probably the most important process
 - Feeding learning with appropriate information
- [Baldwin98]



First Generation Bayesian Filters



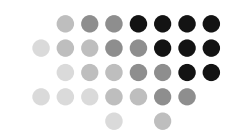
- Tokenization [Graham02]
- Scan all message = headers, HTML, Javascript
- Token constituents
 - Alphanumeric characters, dashes, apostrophes, and dollar signs
- Ignore
 - HTML comments and all number tokens
 - Tokens occurring less than 5 times in training corpus
- Case

First Generation Bayesian Filters



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

First Generation Bayesian Filters



- Bayesian learning [Graham02]

TOKEN PROBABILITY

ST = # times T occurs in spam
S = # spam messages

$$P(T) = \frac{S}{L} \frac{ST}{S + ST}$$

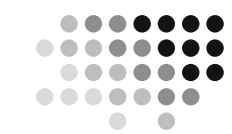
LT = # times T occurs in legitimate email
L = # legitimate messages

MESSAGE PROBABILITY

$$P(S) = \frac{\prod_{T \in TM} P(T)}{\prod_{T \in TM} P(T) + \prod_{T \in TM} (1 - P(T))}$$

TM = set of 15 most extreme tokens (far from .5)

First Generation Bayesian Filters



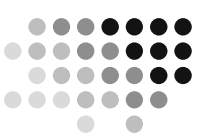
- Batch evaluation
- Required for filtering quality assessment
- Usually focused on accuracy
 - Early training / test collections
 - Accuracy metrics
 - Accuracy = hits / trials
 - Operation regime: train and test
- Other features
 - Prize, ease of installation, efficiency, etc.

First Generation Bayesian Filters



- Batch evaluation – Technical literature
- Focus on end-user features including accuracy
- 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

First Generation Bayesian Filters

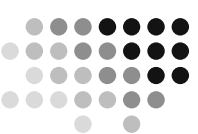


- Batch evaluation – Technical [Anderson04]

REAL-WORLD LABS		Spam Filters			
REPORT CARD		Barracuda Spam Firewall 500	Vircom modusgate	BorderWare MXtreme Mail Firewall Appliance MX-400 3i	Sophos PureMessage 4.5
ANTISPAM ACCURACY (30%)		4.4	4	4.8	4.4
ADDITIONAL FEATURES					
Antivirus (5%)		4	4	5	4.5
Attachment filtering (5%)		4	4.5	5	2
Outlook/Exchange Notes/ Domino Integration (5%)		2	3	3	1
Quarantine (5%)		5	4.5	2	4.5
PRICE (PER USER, PER YEAR)					
1,000 users antispam, antivirus (10%)		5	4	2	3.5
10,000 users antispam, antivirus (10%)		5	4.5	3	3
ARCHITECTURE (15%)		3.5	4	4.5	4.5
MANAGEMENT/CONFIGURATION					
Distributed administration (5%)		1	4	4	5
End-user controls (5%)		4.5	4	2	4
Reporting (5%)		3	2	5	3.5
TOTAL SCORE (100%)		4.02	3.95	3.92	3.87
		B+	B	B	B

A=4.3 B=3.5 C=2.5 D=1.5 F=1.5 AC GRADES INCLUDE * or - IN THEIR RANGES. TOTAL SCORES AND WEIGHTED SCORES ARE BASED ON A SCALE OF 0-5.

First Generation Bayesian Filters

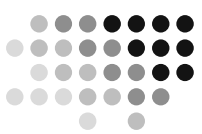


- Batch evaluation – Technical [Anderson04]

Accuracy Test Results

	Inbox	Spam	Total	Inbox common	False negatives	False positives	Nonweighted accuracy	Weighted accuracy
Control	210	1338	1548	210	0	0	100.0%	100.0%
Greenlew Data	263	1285	1548	202	61	8	95.5%	93.5%
IronPort	291	1257	1548	204	87	6	94.0%	92.4%
Brightmail	291	1257	1548	204	87	6	94.0%	92.4%
BorderWare	292	1256	1548	204	88	6	93.9%	92.4%
Barracuda	260	1288	1548	193	67	17	94.6%	90.2%
Sophos	298	1250	1548	199	99	11	92.9%	90.1%
Espion	261	1287	1548	192	69	18	94.4%	89.7%
Katharion	207	1341	1548	182	25	28	96.6%	89.3%
Vircom	221	1327	1548	184	37	26	95.9%	89.2%
Proofpoint	275	1273	1548	193	82	17	93.6%	89.2%

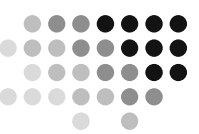
First Generation Bayesian Filters



- 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
- Standardized test collections
- PU1, Lingspam, Spammassassin Public Corpus
- Operation regime
 - Train and test, cross validation (Machine Learning)

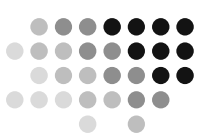
First Generation Bayesian

Filters



- Batch evaluation – Research [Gomez02]
- Comparing several learning algorithms under unknown costs, simple tokenization, Lingspam
- ROC Convex Hull analysis
 - $X = \text{False Positive Rate}$, $Y = \text{True Positive Rate}$
=> Spam captured under few False Positives
 - Plots for an algorithm over a number of cost conditions or thresholds ($P(\text{spam}) > T$)
- Data points obtained by 10 fold cross validation
- Slope ranges and convex hull

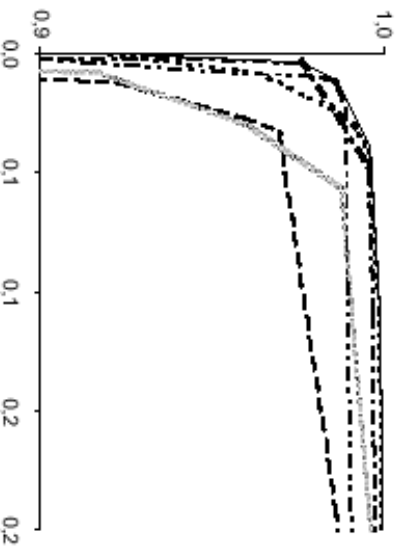
First Generation Bayesian Filters



- Batch evaluation – Research [Gomez02]

Roc curves

Slope ranges



Slope Range	(FP, TP) point	Classifier
[0.000,0.010]	(0.206,1.000)	PAMC <i>i</i> 040
[0.010,0.044]	(0.108,0.999)	SVWE <i>i</i> 005
[0.044,0.357]	(0.040,0.996)	SVTH001
[0.357,1.250]	(0.012,0.986)	ROTH <i>i</i> 020
[1.250,14.750]	(0.004,0.976)	NBWE600
[14.750, ∞]	(0.000,0.917)	SVWE200

FPR between 0 and 0.004 =>

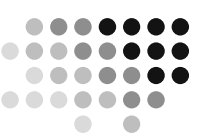
Support Vector Machines lead

FPR between 0.004 and 0.012 =>

Naive Bayes leads

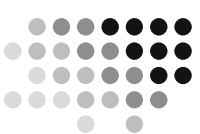
...

Second Generation Bayesian Filters

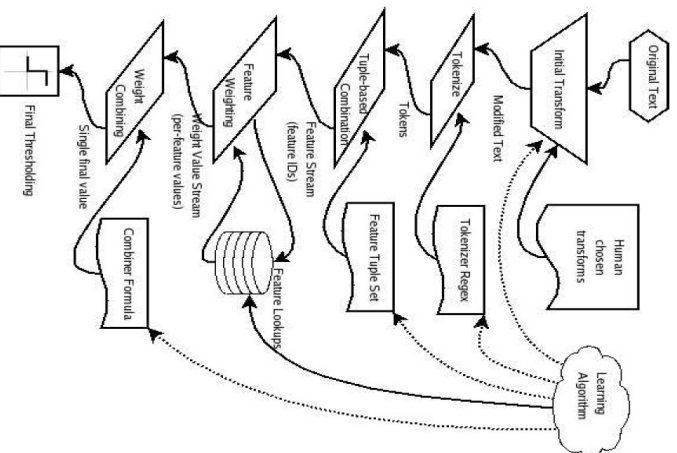


- 2003-now Second Generation Bayesian Filters
 - Significant improvements on
 - Data processing
 - Tokenization and token combination
 - Filter evaluation
 - Filters reaching 99.987% accuracy (one error in 7,000)
- “We have got the winning hand now”
[Zdziarski05]

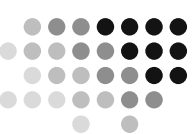
Second Generation Bayesian Filters



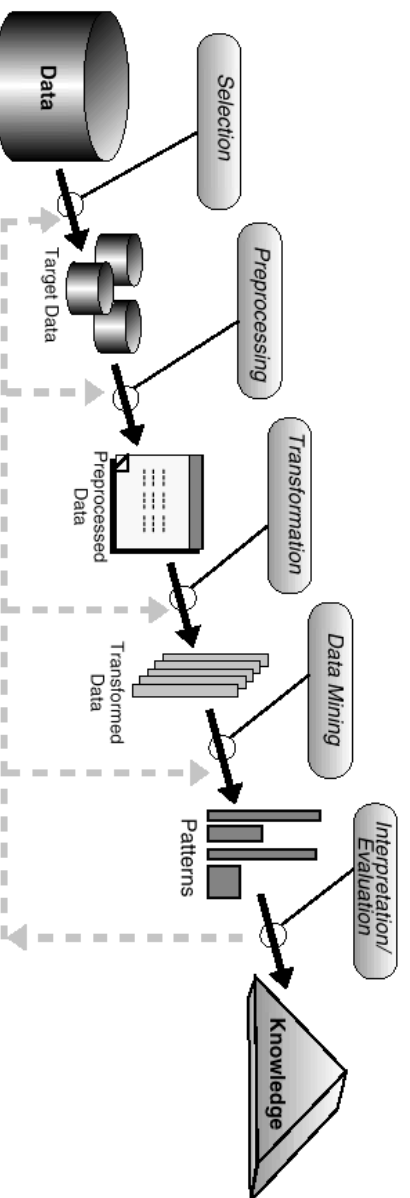
- Unified chain processing [Yerzunis05]
- Pipeline defines steps to take decision
- Most Bayesian filters fit this process
- Allows to focus on differences and opportunities of improvement



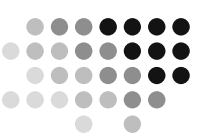
Second Generation Bayesian Filters



- Unified chain processing
- Note remarkable similarity with KDD process [Fayyad96]

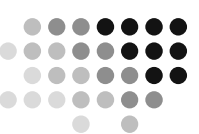


Second Generation Bayesian Filters



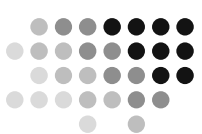
- Preprocessing (1)
- Character set folding
 - Forcing the character set used in the message to the character set deemed “most meaningful” to the end user: Latin-1, etc.
- Case folding
 - Removing case changes
- MIME normalization
 - Unpacking MIME encodings to a reasonable representation (specially BASE64)

Second Generation Bayesian Filters



- Preprocessing (2)
- HTML de bfuscation
 - 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', '1' or | instead of 'l' or 'I', and '\$' instead of 'S'

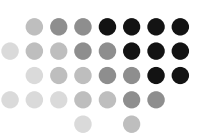
Second Generation Bayesian Filters



- Tokenization
- Token = string matching a Regular Expression
- Examples (CRM111) [Siefkes04]
- Simple tokens = a sequence of one or more printable character
- HTML aware REGEXes = the previous one + typical XML/HTML mark up
 - Start/end/empty tags: <tag> </tag>

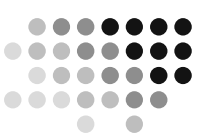
 - Doctype declarations: <!DOCTYPE
 - ETC
- Improvement up to 25%

Second Generation Bayesian Filters



- Tuple based combination
- 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, retaining the last one, e.g. N = 5
 - w4 w5
 - w3 <skip> w5
 - w2 <skip> <skip> w5
 - w1 <skip> <skip> <skip> w5

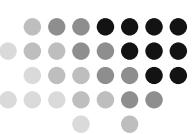
Second Generation Bayesian Filters



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

Tokens:	Viagra	is	great	for
Token Values:	0.92	0.64	0.34	0.71
Bands:	0.90	0.65	0.35	0.70
Patterns:	0.90_0.65_0.35			
	0.65_0.35_0.70			

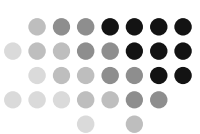
Second Generation Bayesian Filters



- Tuple based combination [Zdziarski05]
- Example: Bayesian Noise Reduction
 - Training
 - Compute sequences values according Grams without bias

GUILTY	INNOCENT		
0.25_1.00_1.00	0.65_0.20_0.00		[0.00900]
0.35_1.00_1.00	1.00_0.60_0.15		[0.21000]
1.00_1.00_0.20	0.00_0.80_0.55		[0.00900]
1.00_0.40_1.00	0.00_0.25_0.90		[0.00900]
1.00_1.00_0.25	0.15_0.05_1.00		[0.00900]
0.55_1.00_1.00	0.60_0.85_0.25		[0.12900]
1.00_1.00_0.35	0.00_0.60_0.90		[0.02000]
0.25_1.00_1.00	0.70_0.05_1.00		[0.17000]

Second Generation Bayesian Filters



- Tuple based combination [Zdziarski05]
- Example: Bayesian Noise Reduction
 - Detecting anomalies and dubbing
 - The pattern value must be extreme: [0.00-0.25], [0.75,1.00]
 - The token value must mismatch the pattern value: 0.30 away from the pattern value
 - e.g. less than 0.65 for a 0.95 pattern
 - Ignore the token in classification (but not in training)

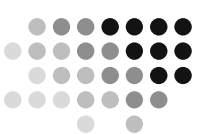
[0.65	0.95	Context value
[is	0.35	Underlying token values
[great	0.70]
[0.15	Context value
[0.65	0.35	Underlying token values
[is	0.35	great
[0.70	for
[]]

Second Generation Bayesian Filters



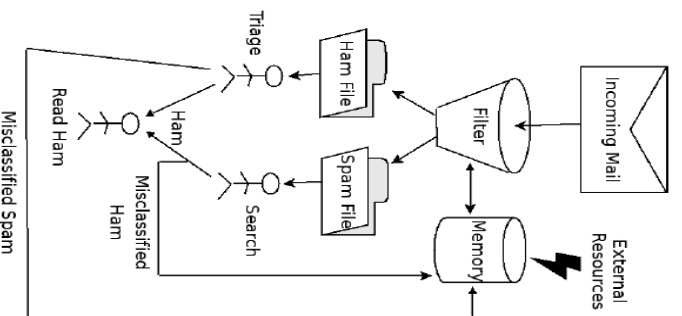
- Learning: weight definition
 - Weight of a token/tuple according to dataset
 - Probably smoothed (added constants)
 - Accounting for # messages = time (confidence)
 - Graham probabilities, increasing Winnow weights, etc.
- Learning: weight combination
 - Combining token weights to single score
 - Bayes rule, Winnow's linear combination
- Learning: final thresholding
 - Applying the threshold learned on training

Second Generation Bayesian Filters



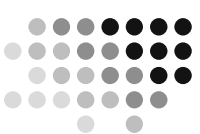
- Accuracy evaluation
 - Online setup
 - Resembles normal end user operation of the filter
 - Sequentially training on errors – time ordering
- As used in TREC Spam Track [Cormack05]
 - Metrics = ROC plotted along time
 - Single metric = the Area Under the ROC curve (AUC)
 - Sensible simulation of message sequence
 - By far, the most reasonable evaluation setting

Second Generation Bayesian Filters



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

Second Generation Bayesian Filters



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

	$S \rightarrow H$	$H \rightarrow S$
$G_0 \rightarrow G_1$	0	278
$G_1 \rightarrow G_2$	4	83
$G_2 \rightarrow G_3$	0	56
$G_3 \rightarrow G_4$	10	15
$G_4 \rightarrow G_5$	0	0
$G_0 \rightarrow G_5$	8	421
G_5	$ H = 9038$	$ S = 40048$

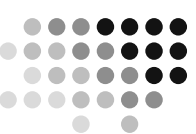
Public Corpora

	Ham	Spam	Total
trec05p-1/full	39399	52790	92189
trec05p-1/ham25	9751	52790	62541
trec05p-1/ham50	19586	52790	72376
trec05p-1/spam25	39399	13179	52578
trec05p-1/spam50	39399	26283	65682

Private Corpora

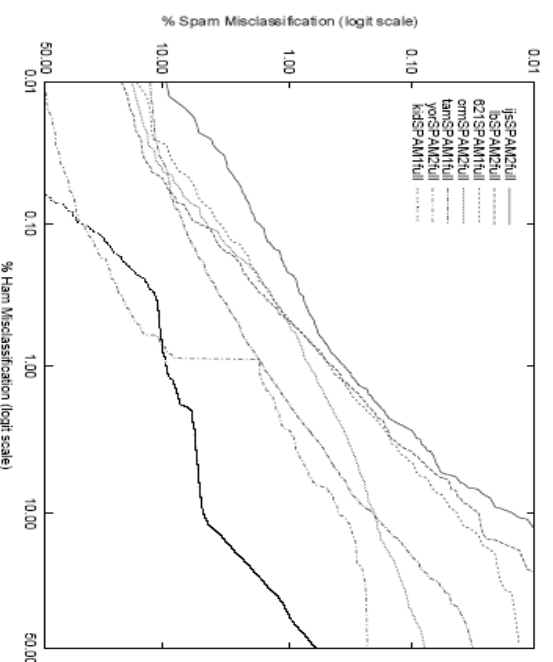
	Ham	Spam	Total
Mr X	9038	40048	49086
S B	6231	775	7006
T M	150685	19516	170201
Total	165954	60339	226293

Second Generation Bayesian Filters

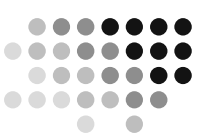


- TREC example results = ROC curve
- Gold
- Jozef Stefan Institute
- Silver
- CRM111
- Bronze
- Laird Breyer

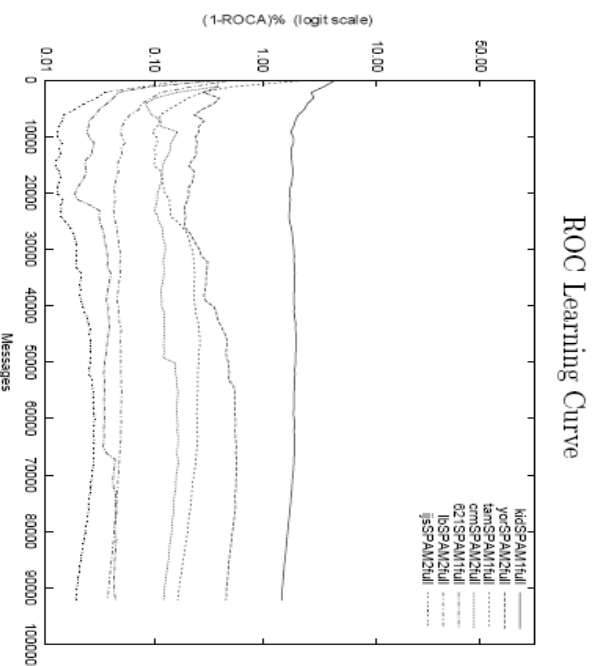
ROC



Second Generation Bayesian Filters



- TREC example results = AUC evolution
- Gold
 - Jozef Stefan Institute
 - Silver
 - CRM111
 - Bronze
 - Laird Breyer

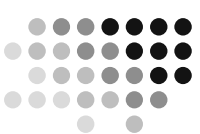


Second Generation Bayesian Filters



- Attacks to Bayesian filters [Zdziarski05]
 - All phases attacked by the spammers
 - See The Spammers Compendium [GraCum06]
 - Preprocessing and tokenization
 - Encoding guilty text in Base64
 - HTML comments (“Hipertextus Interruptus”), small fonts, etc. dividing spammy words
 - Abusing URL encodings

Second Generation Bayesian Filters



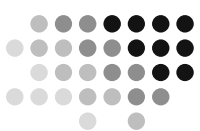
- Attacks to Bayesian filters [Zdziarski05]
- Dataset
 - Mailing list – learning Bayesian ham words and sending spam – effective once, filters learn
 - Bayesian poisoning – more clever, injecting invented words in invented header, making filters learn new hammy words – effective once, filters learn
- Weight combination (decision matrix)
 - Image spam
 - Random words, word salad, directed word attacks
 - Fail in cost effectiveness – effective for 1 user!!!

Conclusion and reflection



- Current Bayesian filters highly effective
 - Strongly dependent on actual user corpus
- Statistically resistant to most attacks
 - They can defeat one user, one filter, once; but not all users, all filters, all the time
- Widespread and effectively combined

Why spam still increasing?



Advising and questions

- Do not miss upcoming events
- CEAS 2006 – <http://www.ceas.cc>
- TREC Spam Track 2006 – <http://trec.nist.gov>

Questions?