Spam Filters:

Do they work?

Can you prove it?

Gordon V. Cormack

15 February, 2006

University of Waterloo

Why Standardized Evaluation?

- To answer questions!
  - Is spam filtering a viable approach?
  - What are the risks, costs, and benefits of filter use?
  - Which spam filter should I use?
  - How can I make a better spam filter?
- What's the alternative?
  - Testimonials
  - Uncontrolled, unrepeatable, statistically bogus tests
  - Warm, fuzzy feelings

There's no Perfect Test

- But a standardized test should
  - Model real filter usage as closely as possible
  - Evaluate the filter on criteria that reflect its effectiveness for its intended purpose
  - Eliminate uncontrolled differences
  - Be repeatable
  - Yield statistically meaningful results
- Future tests will
  - Challenge assumptions in the current test

TREC – Text Retrieval Conference

- Sponsored by, held at
  - NIST – National Institute for Standards & Technology
  - **http://trec.nist.gov**
- Goals
  - To increase the availability of appropriate evaluation techniques for use by industry and academia, including the deployment of new evaluation techniques more applicable to current systems.
- Format
  - Participants do experiments in one or more *tracks*

What is Spam?

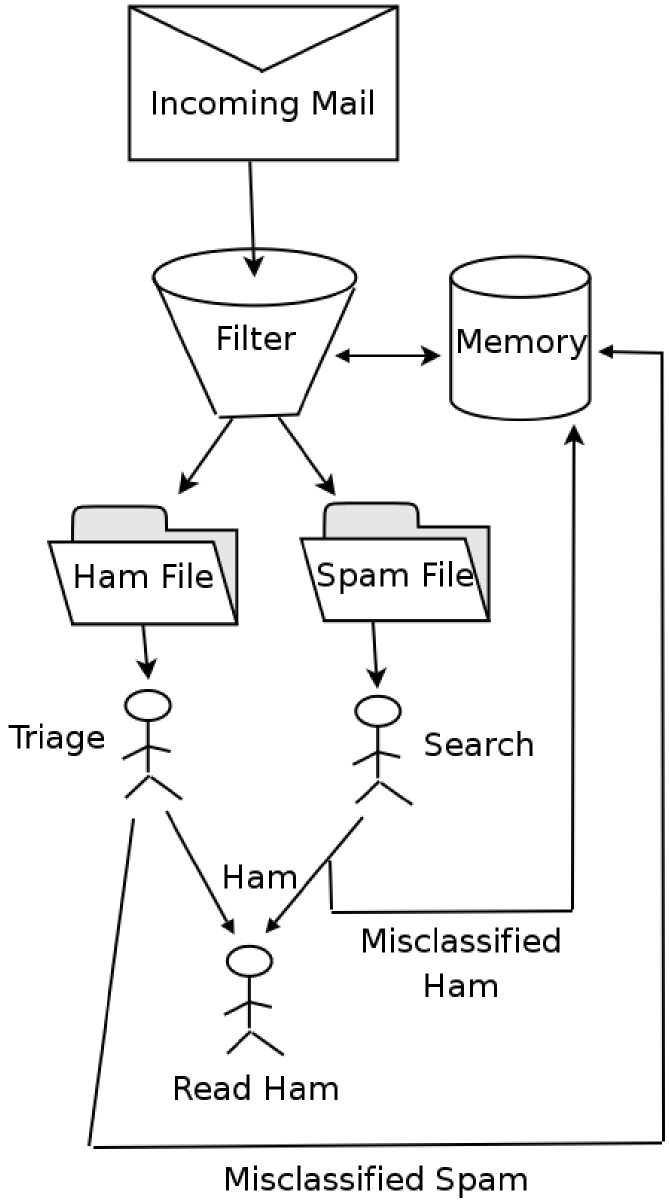
TREC definition

Unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient.

Depends on sender/receiver relationship

Not “whatever the user thinks is spam.”

Spam Filter Usage



- Filter Classifies Email
- Human addressee
  - Triage on ham File
  - Reads ham
  - Occasionally searches for misclassified ham
  - Report misclassified email to filter

Spam Filter Evaluation

- Simulate (replay) incoming email stream
  - *single* stream (for now)
  - chronological order
  - full email message with *original* headers
- Simulate *idealized* user's behaviour
  - reports *all* misclassifications *immediately*
    - spam in ham file (spam misclassification, false negative)
    - ham in spam file (ham misclassification, false positive)
- Capture filter results
- Analyze captured results

Simulating Email Stream

- Identify user
- Secure user's permission (tacit or explicit)
  - this is the hard part
    - User's sensitivities
    - Sender's sensitivities
    - 3rd Parties sensitivities
    - Privacy legislation & ethics
- Capture email exactly as delivered

Simulating *Idealized* User

- Capture
  - Filter result for each message (ham/spam)
  - User's reports of misclassified ham or spam
- But Real Users are not *Ideal*
  - err and are inconsistent
  - slow and haphazard in reporting misclassification
- Real User involved in pilot evaluation
  - vets disagreements between user and filter
- Gold Standard ham/spam judgement

Standardized Filter Interface

Filter implements (Linux or Windows) commands

initialize

create necessary files & servers (cold start)

classify *filename*

read *filename* which contains exactly 1 email message

write one line of output:

classification score auxiliary_file

train *judgement filename classification*

take note of gold-standard *judgement*

finalize

clean up: kill servers, remove files

Tool Kit for Filter Evaluation

initialize

for each *judgement, filename* in corpus

  classify *filename* > *classification, score*

  train *judgement filename classification*

  record *judgement, filename, classification, score*

finalize

[later]

analyze & summarize recorded judgements

Participant Filters

Group	Filter Prefixes
Beijing University of Posts and Telecommunications	kidSPAM1, kidSPAM2, kidSPAM3, kidSPAM4
Chinese Academy of Sciences (ICT)	ICTSPAM1, ICTSPAM2, ICTSPAM3, ICTSPAM4
Dalhousie University	dalSPAM1, dalSPAM2, dalSPAM3, dalSPAM4
IBM Research (Segal)	621SPAM1, 621SPAM2, 621SPAM3
Indiana University	indSPAM1, indSPAM2, indSPAM3, indSPAM4
Jozef Stefan Institute	ijsSPAM1, ijsSPAM2, ijsSPAM3, ijsSPAM4
Laird Breyer	lbSPAM1, lbSPAM2, lbSPAM3, lbSPAM4
Massey University	tamSPAM1, tamSPAM2, tamSPAM3, tamSPAM4
Mitsubishi Electric Research Labs (CRM-114)	crmSPAM1, crmSPAM2, crmSPAM3, crmSPAM4
Pontificia Universidade Catolica Do Rio Grande Do Sul	pucSPAM1, pucSPAM2, pucSPAM3
Universite Paris-Sud	azeSPAM1, azeSPAM2
York University	yorSPAM1, yorSPAM2, yorSPAM3, yorSPAM4

Non-participant Filters

Filter	*Run Prefix*	*Configuration*
Bogofilter	bogofilter	0.92.2
DSPAM	dspam-tum	3.4.9, train-until-mature
	dspam-toe	3.4.9, train-on-errors
	dspam-teft	3.4.9, train-on-everything
Popfile	popfile	0.22.2
Spamassassin	spamasasb	3.0.2, Bayes component only
	spamasasv	3.0.2, Vanilla (out of the box)
	spamasasx	3.0.2, Mr. X configuration
Spamprobe	spamprobe	1.0a

Public Corpus & Subsets

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

University of Waterloo

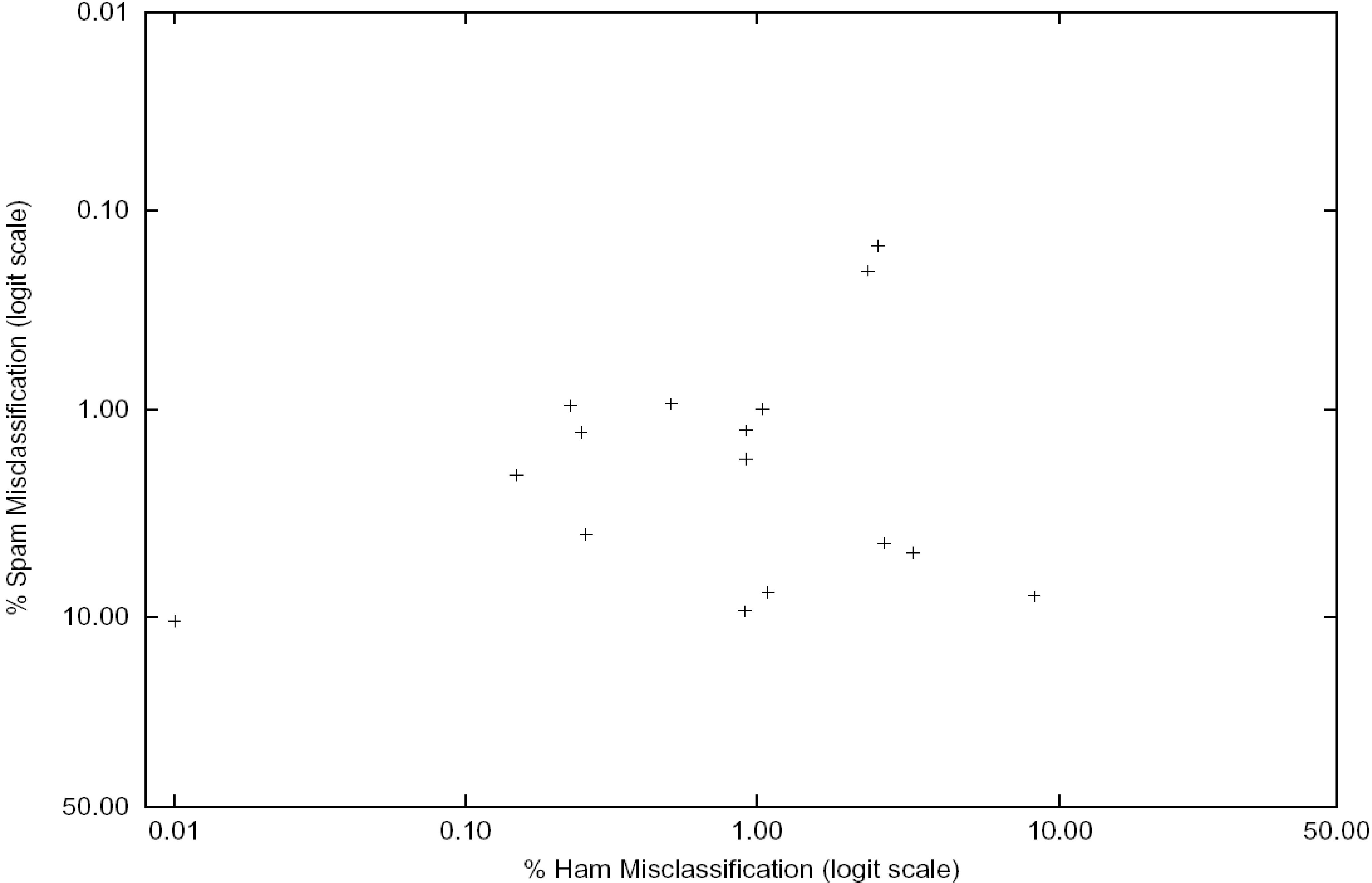
Analysis – Binary Classification

		Gold Standard Judgement	
		ham	spam
Filter	ham	a	b
Classification	spam	c	d

a: ham (correctly classified) [true negative]
b: spam misclassification [false negative]
c: ham misclassification [false positive]
d: spam (correctly classified) [true positive]

c/(a+c): ham misclassification rate (hm%)
b/(b+d): spam misclassification rate (sm%)

Hm% vs Sm% - Public Corpus



Logistic Average Misc%

logit transforms probability to log odds

- odds $x = x\ /\ (100\% - x)$
- logit $x = \log(\text{odds } x)$

range -∞ .. ∞ with symmetric algebraic properties

- 0.1% - 0.01% equals 99.9% - 99.99%
  - nearly equals 1% - 0.1%, 99.99% - 99.999% etc.
  - i.e. each represents a *tenfold* performance difference

logistic average misclassification

- $lam\% = \text{logit}^{-1}\,(\text{logit } hm\% + \text{logit } sm\%)/2$
  - improvements in *lm%, hm%* rewarded equally
    - (similar to geometric mean in Robust Track)

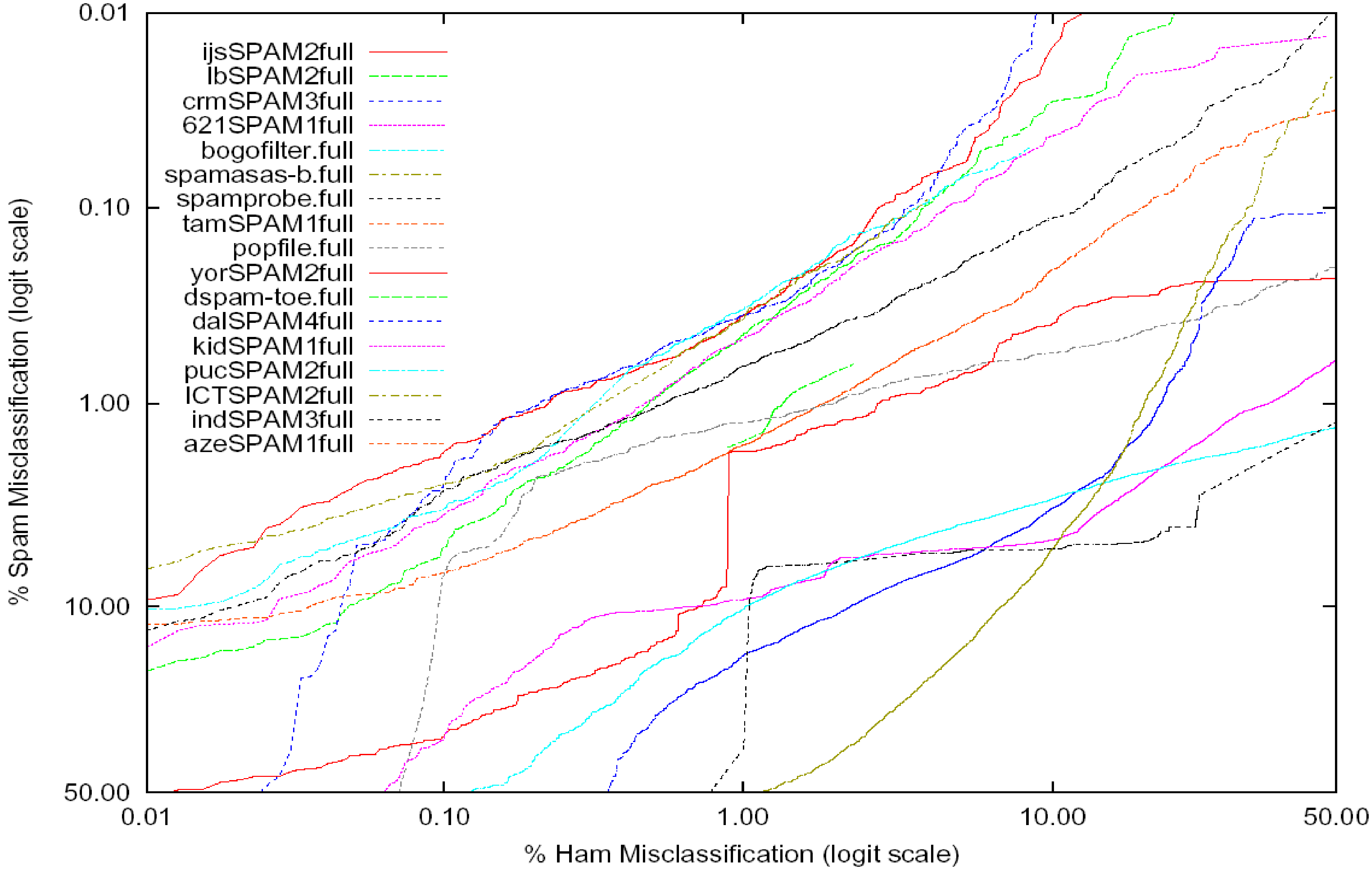
Classification – Public Corpus

Run	Hm%	Sm%	Lam%
bogofilter	0.01	10.47	0.30
ijsSPAM2	0.23	0.95	0.47
spamprobe	0.15	2.11	0.57
spamasas-b	0.25	1.29	0.57
crmSPAM3	2.56	0.15	0.63
621SPAM1	2.38	0.20	0.69
lbSPAM2	0.51	0.93	0.69
popfile	0.92	1.26	0.94
dspam-toe	1.04	0.99	1.01
tamSPAM1	0.26	4.10	1.05
yorSPAM2	0.92	1.74	1.27
indSPAM3	1.09	7.66	2.93
kidSPAM1	0.91	9.40	2.99
dalSPAM4	2.69	4.50	3.49
pucSPAM2	3.35	5.00	4.10
ICTSPAM2	8.33	8.03	8.18
azeSPAM1	64.84	4.57	22.92

Analysis – Ham/Spam Tradeoff

- Most filters compute *spamminess*
  - if *spamminess* > *threshold* then classify as spam
  - else classify as ham
- *threshold* value is arbitrary
  - higher threshold =
    - fewer ham misclassifications
    - more spam misclassifications
- ROC (Receiver Operating Characteristic) Curve
  - vary *threshold,* plot ham misc. vs. spam misc.
  - Area under curve approaches 100% (perfect filter)
  - We report (1-ROCA)% [degree of imperfection]

ROC Curves – Public Corpus



University of Waterloo

Measures – Public Corpus

Run	(1-ROCA)%	*Rank*	Sm% @ Hm%=0.1	*Rank*	Lam%	*Rank*
ijsSPAM2	0.02	1	1.8	1	0.5	2
lbSPAM2	0.04	2	5.2	7	0.7	7
crmSPAM3	0.04	3	2.6	3	0.6	5
621SPAM1	0.04	4	3.6	6	0.7	6
bogofilter	0.05	5	3.4	5	0.3	1
spamasas-b	0.06	6	2.6	2	0.6	3
spamprobe	0.06	7	2.8	4	0.6	4
tamSPAM1	0.16	8	6.9	8	1.1	10
popfile	0.33	9	7.4	9	0.9	8
yorSPAM2	0.46	10	34.2	10	1.3	11
dspam-toe	0.77	11	88.8	15	1.0	9
dalSPAM4	1.37	12	76.6	13	3.5	14
kidSPAM1	1.46	13	34.9	11	3.0	13
pucSPAM2	1.97	14	51.3	12	4.1	15
ICTSPAM2	2.64	15	79.5	14	8.2	16
indSPAM3	2.82	16	97.4	16	2.9	12
azeSPAM1	28.89	17	99.5	17	22.9	17

Rank by Statistic & Corpus

Filters	Aggregate ROCA	Aggregate h=.1	Aggregate lam%	trec05p-1/full ROCA	trec05p-1/full h=.1	trec05p-1/full lam%	Mr. X ROCA	Mr. X h=.1	Mr. X lam%	S. B. ROCA	S. B. h=.1	S. B. lam%	T. M. ROCA	T. M. h=.1	T. M. lam%
ijsSPAM2	1	3	3	1	1	2	7	12	11	2	3	5	1	6	6
ijsSPAM1	2	2	3	2	2	4	7	14	13	3	6	17	2	5	5
ijsSPAM4	3	6	6	4	5	8	5	10	16	5	7	15	5	8	7
ijsSPAM3	4	7	12	3	2	5	2	2	8	6	10	22	6	10	18
crmSPAM2	5	1	1	14	11	16	3	11	5	17	13	19	4	2	1
crmSPAM3	6	15	13	7	7	10	16	18	18	1	2	10	7	9	4
crmSPAM4	7	8	1	10	4	2	17	31	14	4	4	11	8	4	2
lbSPAM2	8	11	15	5	13	11	9	13	7	9	14	4	11	17	23
lbSPAM1	9	9	11	6	12	9	13	16	2	8	18	9	13	13	19
tamSPAM1	10	13	17	16	14	22	14	9	15	18	20	20	9	12	14
spamprobe	11	5	5	11	8	6	11	15	4	21	15	12	14	7	7
tamSPAM2	12	18	18	18	22	23	21	29	26	11	27	24	12	14	13
bogofilter	13	14	14	9	9	1	1	3	12	14	17	3	21	16	16
spamasas-b	14	10	7	11	6	6	11	8	10	16	9	7	19	11	12
lbSPAM3	15	21	20	14	20	18	24	37	25	26	44	34	15	18	20
crmSPAM1	16	17	24	17	18	26	19	30	23	24	11	21	20	19	28
lbSPAM4	17	19	23	20	21	28	22	23	30	20	23	32	17	15	22
yorSPAM2	18	20	19	23	25	25	3	7	5	10	16	15	18	23	24
spamasas-x	19	16	8	22	19	15	6	1	3	7	1	1	23	20	17
kidSPAM1	20	30	27	31	26	32	29	32	49	32	46	37	16	29	21
dspam-toe	21	35	16	26	40	20	28	40	21	47	18	6	25	30	15
621SPAM1	22	4	22	8	10	11	40	6	31	23	5	23	3	1	9
621SPAM3	23	12	30	13	15	16	42	4	29	25	8	17	10	3	3
yorSPAM4	24	34	26	25	38	29	30	39	24	52	43	50	22	32	29
dspam-tum	25	22	10	27	29	11	27	36	20	48	21	8	36	22	11

Confidence Intervals

95% Confidence Limits – *see notebook appendix*

Exact binomial probabilities

- *hm%, sm%*

Logistic Regression, parametric model

- Standard error (S.E.) for logit *hm%*, logit *sm%*
  - 95% confidence interval ± 1.96 S.E.
  - agrees well with binomial estimate
- *lam%* S.E. = root-mean-square *hm% S.E, sm% S.E.*
- S.E. for learning-curve slope and intercept

Bootstrap (100 resampled pseudo-corpora)

- S.E. for logit *(1-ROCA)%*

Learning Curves

Cumulative

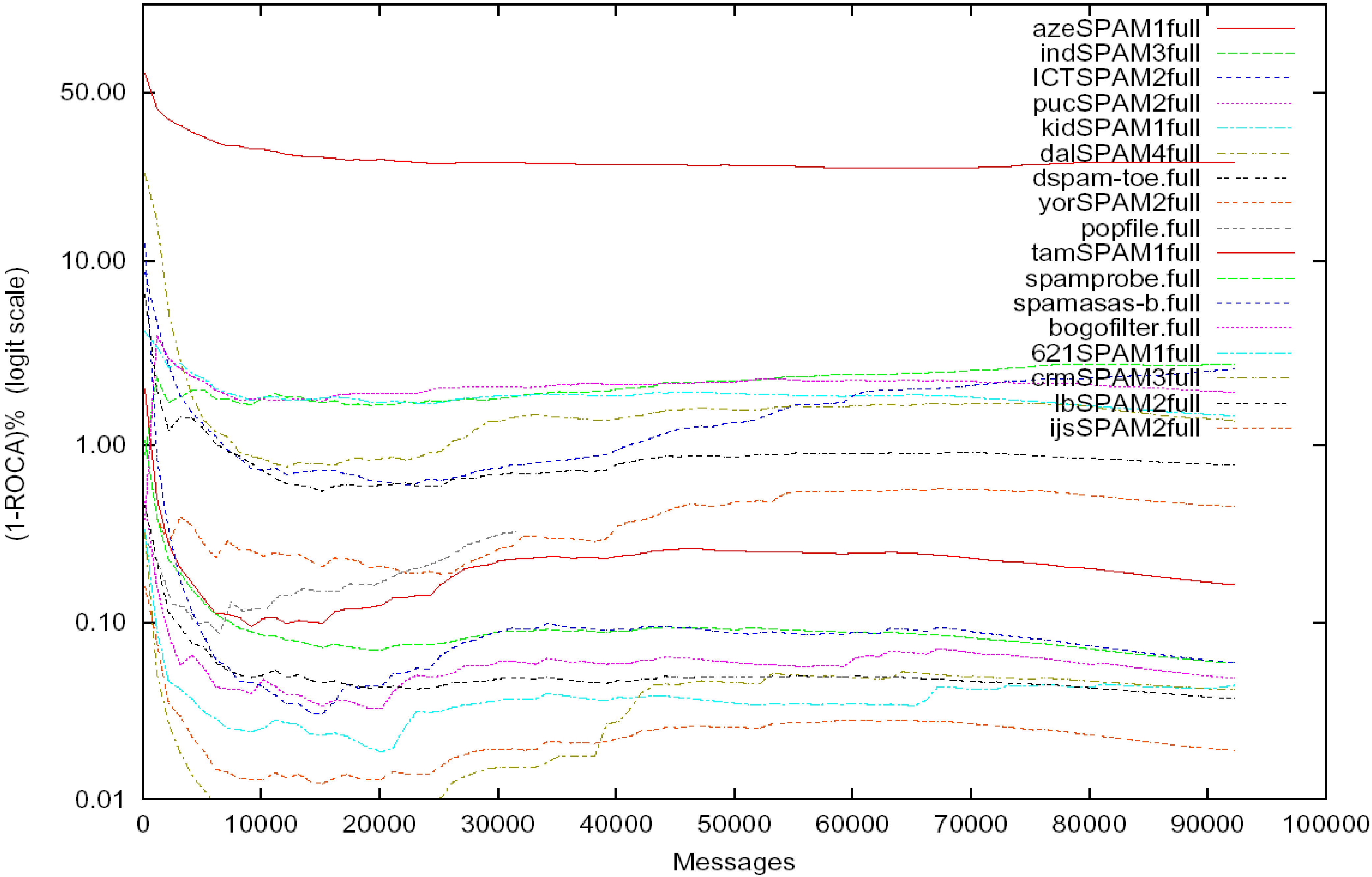
- Report summary statistic e.g. (1-ROCA)%
  - for all prefixes of the corpus
- Reaches asymptote if filter performance constant
- Smooths variations in filter performance (long decay)

Instantaneous

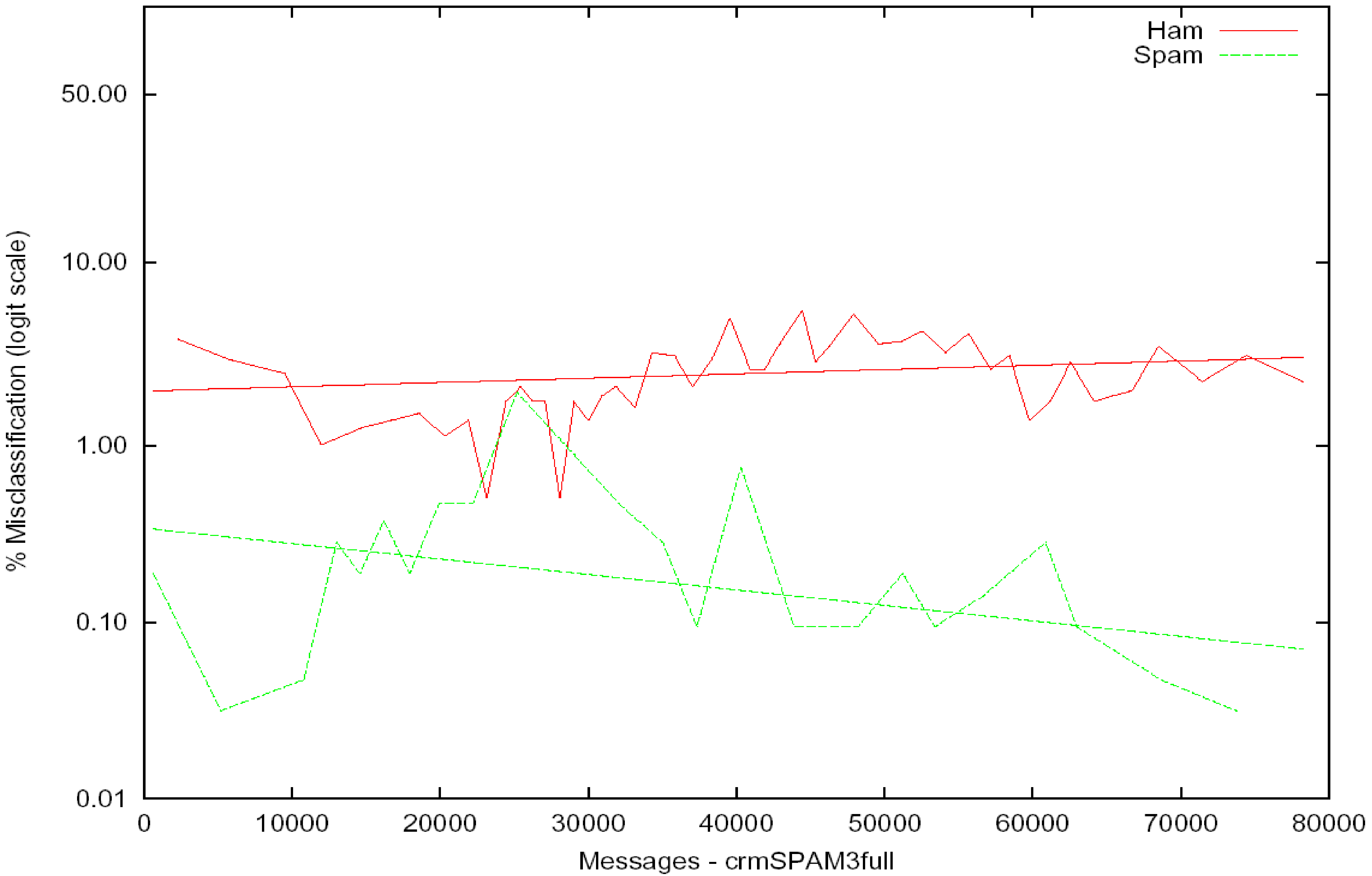
- Estimate hm% and sm% at any given time
  - piecewise approximation
  - logistic regression
    - logit $hm\% = a + bx$
      - best a and b where x is number of messages classified so far
- No suitable estimate (yet) for summary stats

University of Waterloo

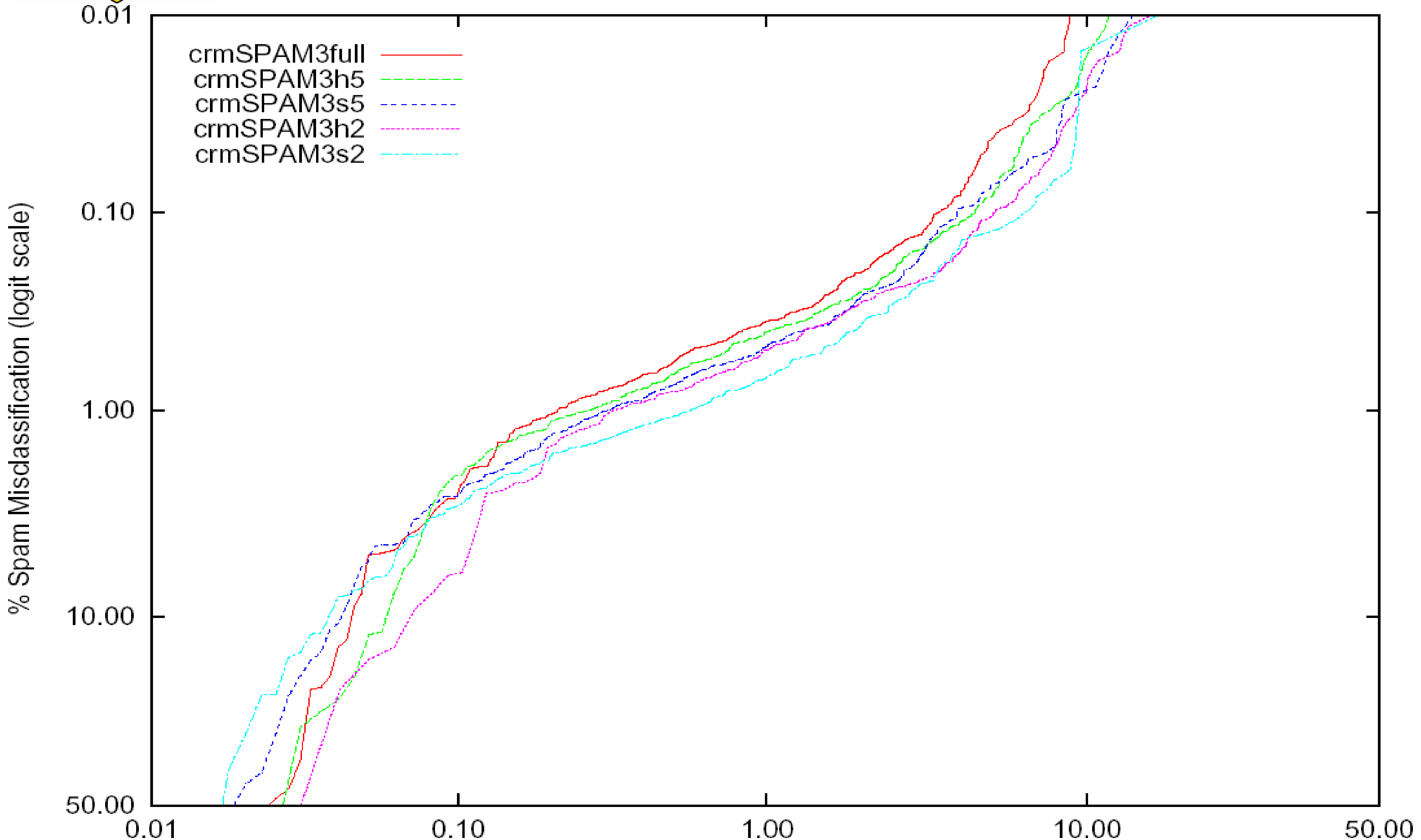
Cumulative ROC Learning



Instantaneous Learning Curves



Ham/spam subsets



Genre Classification

- Not all types of ham are equal!
  - Some more likely misclassified
    - higher likelihood of ending up in spam filter
  - Some more likely missed if filtered
    - can be retrieved from spam file
  - Some more valuable
    - consequences of non-receipt vary dramatically
- Overall downside risk depends on all these factors
- Spam can similarly be classified

Genre (S.B. Corpus)

	Misclassified Spam (of 775 spams)							Misclassified Ham (of 6231 hams)							
	Automated	List	Newsletter	Phishing	Sex	Virus	**Total**	Automated	Commercial	Encrypted	Frequent	List	Newsletter	Personal	**Total**
ijsSPAM2	3	10	4	3	69	2	**91**	4	3	0	0	2	1	0	**10**
lbSPAM2	3	47	12	6	178	11	**257**	1	0	0	0	1	0	0	**2**
crmSPAM3	2	7	10	1	37	2	**59**	4	6	0	1	5	2	3	**21**
621SPAM1	1	6	7	0	10	17	**41**	15	20	0	13	14	8	28	**98**
tamSPAM1	3	40	14	3	147	6	**213**	4	1	0	0	3	0	1	**9**
yorSPAM2	9	11	26	3	114	19	**182**	1	3	0	0	2	3	0	**9**
dalSPAM4	11	23	8	8	249	18	**317**	4	11	0	22	53	10	18	**118**
kidSPAM1	3	8	12	4	74	4	**105**	5	14	1	121	20	2	47	**210**
pucSPAM2	5	28	15	2	264	3	**317**	4	3	9	100	15	2	21	**154**
ICTSPAM2	8	12	17	7	68	10	**122**	4	3	2	8	30	6	14	**67**
indSPAM3	3	22	17	7	220	18	**287**	3	7	0	11	27	60	6	**114**
azeSPAM1	0	16	6	6	43	0	**71**	70	51	126	808	1938	255	360	**3608**

Conclusions

Spam filters work

- still room for improvement

Public corpora work

- finding sources a continuing challenge

Private corpora work

- but we need more rigorous specifications and limits
- burden on volunteers

Spam Filter Test Kit & Methodology

- generally applicable beyond TREC
  - collaborative filtering, different (or no) user feedback, ...

CEAS 2006

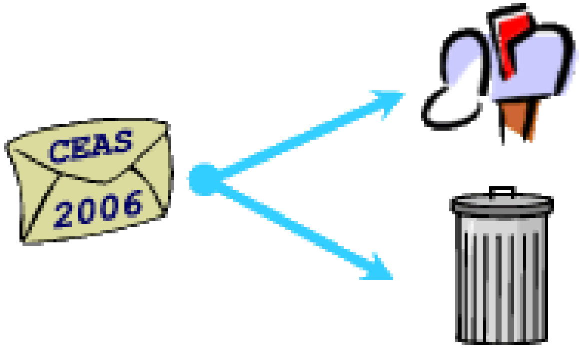
Third Conference on Email and Anti-Spam

27-28 July, 2006

Mountain View, California

http://www.ceas.cc/

submissions: ***23 March, 2006***



Further Resources

TREC - **trec.nist.gov**

- Call for participation (TREC 2006)
- Description of tracks
- Past proceedings

Spam Track – **plg.uwaterloo.ca/~gvcormac/spam**

- Guidelines
- Test jig, analysis tools, sample filters
  - Linux, Unix, or Windows (with Cygwin tools)

Methodology - **plg.uwaterloo.ca/~gvcormac/spamcormack**