Predicting the Future

Finite State Machines

Testing, Probability, Statistics

and other unpleasant things

Gordon V. Cormack

University of Waterloo

University of Waterloo

Data Compression

50'000€ Prize for Compressing Human Knowledge

(widely known as the Hutter Prize)

Compress the 100MB file `enwik8` to less than the current record of about 16MB

- The Task
- Motivation
- **Detailed Rules for Participation**
- Previous Records
- More Information
- **Newsgroup on the contest and prize**
- History
- Committee
- Donations
- **Frequently Asked Questions**
- Contestants
- Links
- Disclaimer

News: *Alexander Ratushnyak is also the **second** Winner! Congratulations!*

... the contest continues ...

Being able to compress well is closely related to intelligence as explained below. While intelligence is a slippery concept, file sizes are hard numbers. Wikipedia is an extensive snapshot of Human Knowledge. If you can compress the first 100MB of Wikipedia better than your predecessors, your (de)compressor likely has to be smart(er). The intention of this prize is to encourage development of intelligent compressors/programs.

The Task

Create a compressed version (self-extracting archive) of the 100MB file `enwik8` of less than about 16MB. More precisely:

Honeywell MK III CMU

DMC

- Information theory
- Automata theory
- Markov processes
- Probability & stats
- Machine learning
- 250 lines of code
- Evaluation & measurement
- Application (avionic telemetry)

theory + practice + evaluation + application

Predict Human Actions

University of Waterloo

SpamOrHam

353777 votes counted! (**High Scores:** #1 456, #2 428, #3 235)

Your longest run of agreement with the filters is 8 (for this session); your current run is 8 (keep playing!).

Scores for all your sessions will be used in determining the winner; just use the same email address each time you visit spamorham.org

---

Thanks gvcormac@uwaterloo.ca! On the last email you said ham and the spam filters said ham

So far you've found 0 possible filter errors and verified 8 emails

---

Click one of these buttons:

This is Spam - I'm not sure - This is Ham

☐ Flag message as funny

As displayed by Microsoft Outlook

Reuter's Contract Negotiation - Message (Plain Text)

From: Corey, Paula [Paula.Corey@ENRON.com] Sent: Tue 5/21/2002 10:36 PM

To: Davidson, Binky; Vanderhorst, Barry; Constantine, Chris; Germany, Chris; Story, S. Craig; Coleman, David; Draper, David; Leboe, David; Sewell, Doug; Piper, Greg; Hewitt, Jess P.; Parks, Joe; Sarnowski, Julie; Kelly, Katherine L.; Kitagawa, Kyle; Shireman, Kristann; Valencia, Laura; Schlueter, Mark; Polsky, Phil; Rieker, Paula; Ryder, Patrick; Barbe, Robin; Barbe, Robin; Cady, Rachel; Plauche, Stephen; Wells, Tori L.; Mathias, Vridhay;

Cc: Dopson, LaMetrice; Harrington, Stephen; Parsons, Andrew; Hughes, Evan

Subject: Reuter's Contract Negotiation

```
You have been identified as a current subscriber to our Reuter's Cobra or
I-Finance Market Data. We are currently in the process of re-negotiating
these licenses. Please respond no later than Tuesday, May 28th with your
Co# and RC. Future ongoing services will be allocated to the appropriate
cost centers.
This information will allow us to negotiate in an informed manner.
Individual costs will be determined at the time we have gathered all
subscriber details.

I appreciate your assistance with this effort.

Paula Corey
```

Predictive models work for

data compression

spam detection

- viruses, phishing, IM, SMS, blog, Web spam
- insensitive to language, alphabet, coding method
- heterogeneous, multimedia, metadata

plagiarism detection, authorship attribution

intrusion detection

game playing

Need **well defined tasks** and **evaluation**!

Data Compression

Given a stream of bits

Represent the stream in fewer bits

Trick:

- predict each bit in turn (as a probability p)
- encode as -log2(p) bits (on average)
  - arithmetic coding
  - optimal given p

Measure success! Compress some data!

What is Spam?

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 and non-spam examples

Spam

```
Hi,
=20
M e R / D / A
V / a G R A
P R O z & C
A m o x / c i I l / n
C i A L / S
V A L / u M
T r & m a d o I
A m B / E N
X & n a x
L e V / T R A
S O m &
=20
http://www.prosebutis.com <http://www.prosebutis.com>=20
```

Non-spam

```
Dear Gord:

Your C program has solved Ok the problem 11102 (Moonshine)
in 0.514 seconds using as much as 420 kbytes of virtual memory.
Congratulations!

--

PS: Check the board at http://acm.uva.es/board/

The Online Judge (Linux acm.uva.es 2.4.18-27.7.x i686)
Judge software version 2.8 [http://acm.uva.es/problemset/]
Wed May 24 23:19:30 UTC 2006
```

Objective: color spam red, non-spam green

Spam

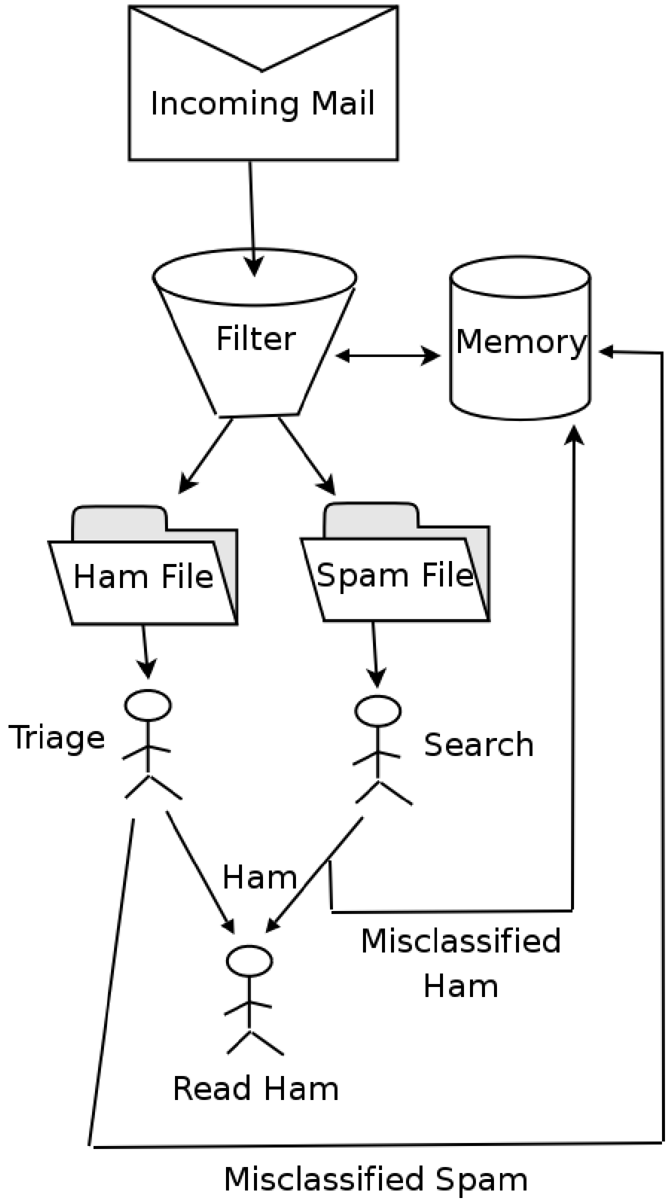
```
Hi,
=20
M e R / D / A
V / a G R A
P R O z & C
A m o x / c i I l / n
C i A L / S
V A L / u M
T r & m a d o I
A m B / E N
X & n a x
L e V / T R A
S O m &
=20
http://www.prosebutis.com <http://www.prosebutis.com>=20
```

Non-spam

(ham)

```
Dear Gord:

Your C program has solved Ok the problem 11102 (Moonshine)
in 0.514 seconds using as much as 420 kbytes of virtual memory.
Congratulations!

--

PS: Check the board at http://acm.uva.es/board/

The Online Judge (Linux acm.uva.es 2.4.18-27.7.x i686)
Judge software version 2.8 [http://acm.uva.es/problemset/]
Wed May 24 23:19:30 UTC 2006
```

How is the coloring used?



Filter Classifies Email

Human addressee

- Triage on ham File
- Reads ham
- Occasionally searches for misclassified ham
- Report misclassified email to filter

Questions to answer

Method to color spam & non-spam (ham)?

How well does the method color?

How well is the overall purpose met?

- Facilitating delivery of good email
- “filtering spam” is just a means to the end

Models and prediction

Given a sequence of bits, predict the next one (***x***)

`1011011011011011011`***x***

x is *probably* **0**

`01011011101111011111`***x***

x is *probably* **1**

How *'probably?'*

Prob(***x*** = **0** following `1011011011011011011`)

Prob(***x*** = **1** following `01011011101111011111`)

Model

abstracts the string of bits; used to predict *behavior*

0^{th} order Markov model

Count the number of *zeros* & the number of *ones*:

`1011011011011011011`$\boldsymbol{x}$

zeros: **6** *ones:* **13**

Use the proportion of *ones* to estimate

Prob($\boldsymbol{x} = \mathbf{1}$) = 13/19 = 0.68

Doesn't seem like such a good estimate

how can we validate it?

- intuition
- testimonial
- faith
- experiment

1^{st} order Markov model

Count the number of *ones* and *zeros* following a `0`, and the number following a `1`

`1011011011011011011`***x***

following 0: *zeros*: **0** *ones*: **6**

following 1: *zeros*: **6** *ones*: **6**

Use the proportion of *ones* following `1` to estimate

Prob(***x*** = **1** *following* **1**) = 6/12 = 0.5

Still doesn't seem like such a good estimate

but better than 0^{th} order

2^{nd} order Markov model

- Count the number of *ones* and *zeros* following `00`, and `01`, and `10`, and `11`.

  `1011011011011011011`x

  following 00: *zeros*: **0** *ones*: **0**

  following 01: *zeros*: **0** *ones*: **6**

  following 10: *zeros*: **0** *ones*: **6**

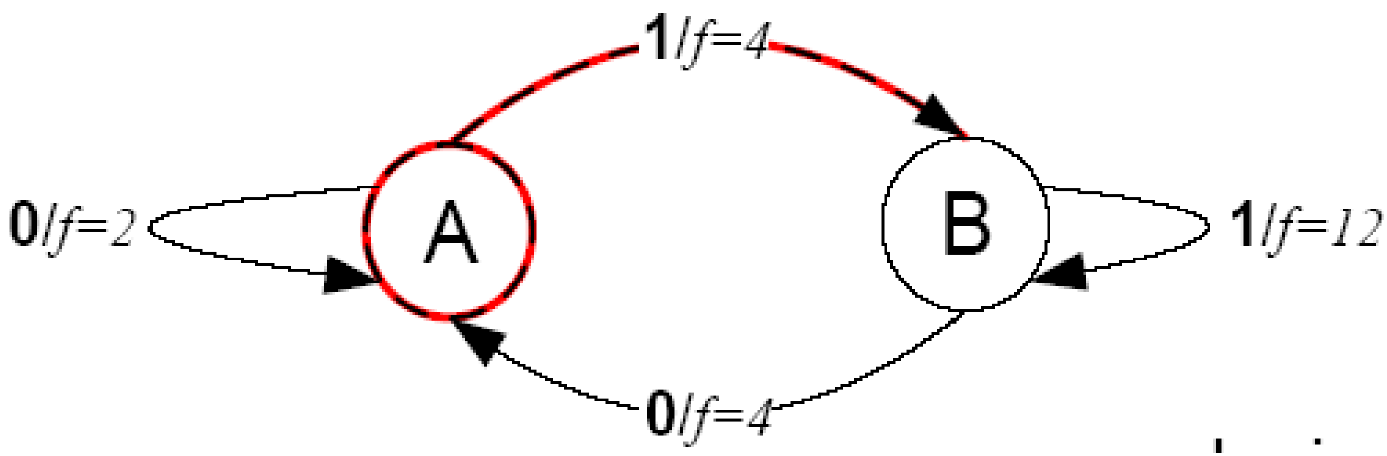
  following 11: *zeros*: **5** *ones*: **0**

- Use the proportion of *ones* following `11` to estimate

  Prob($\boldsymbol{x}$ **= 1** *following* **11**) = 0/5 = 0

- Overconfident! (Overfitted model)

Dynamic Markov model (DMC)



This example implements a 1st order Markov model

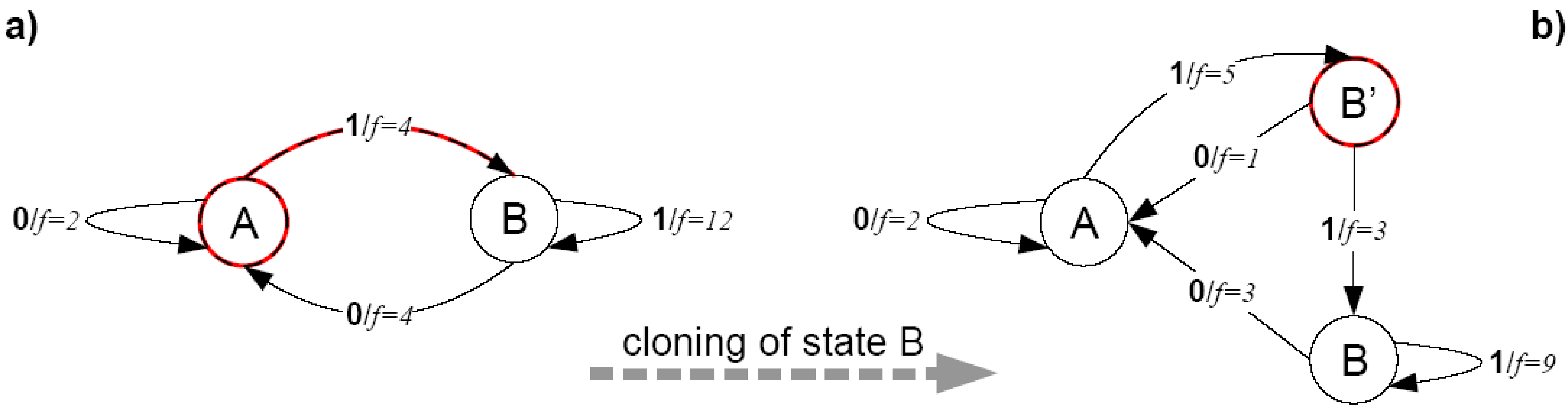
A means *following 0*; B means *following 1*

Outputs ***f*** on edges are frequencies

Prob(**1** *following* A) = 4 / (2 + 4) = 0.667

f incremented after each transition

DMC State Cloning



State A, input 1, Prob 0.67

B visited 16 times previously

- 4 from A; 12 from elsewhere

B should be cloned because it is visited from distinct contexts several times

B cloned to create B'

- *f* divided in 4:12 ratio in proportion to previous visits

f incremented as usual

Data Compression

- Predict each bit in turn
  - DMC
- Construct optimal code
  - arithmetic coding
  - the more probable the shorter the representation
    - $-\log_2$ prob
  - but how do you do a fraction of a bit?
    - many bits at a time
- Google for dmc.c

Likelihood Ratio

Likelihood of a bit (say 0) in spam

`1011011011011011011`**0** $\quad$ Prob($\boldsymbol{x} = \mathbf{0}$)

Likelihood of same bit in non-spam

`0101101110111101111`**0** $\quad$ Prob($\boldsymbol{x} = \mathbf{0}$)

log-likelihood ratio

spamminess = log(Prob($\boldsymbol{x} = \mathbf{0}$) / Prob($\boldsymbol{x} = \mathbf{0}$))

coloring method

spam if *spamminess* > 0; otherwise non-spam

more generally

spam if *spamminess* > $\boldsymbol{t}$; otherwise non-spam

Combining Likelihoods

$$spamminess(x_1 x_2 x_3 \ldots x_n)$$

$$= \log(\text{Prob}(x_1 x_2 x_3 \ldots x_n) \;/\; \text{Prob}(x_1 x_2 x_3 \ldots x_n))$$

$$= \log(\text{Prob}(x_1) \;/\; \text{Prob}(x_1)) +$$

$$\log(\text{Prob}(x_2) \;/\; \text{Prob}(x_2)) +$$

$$\log(\text{Prob}(x_3) \;/\; \text{Prob}(x_3)) + \ldots$$

$$\ldots + \log(\text{Prob}(x_n) \;/\; \text{Prob}(x_n))$$

Email spamminess

Let

S be a string consisting of all known spam

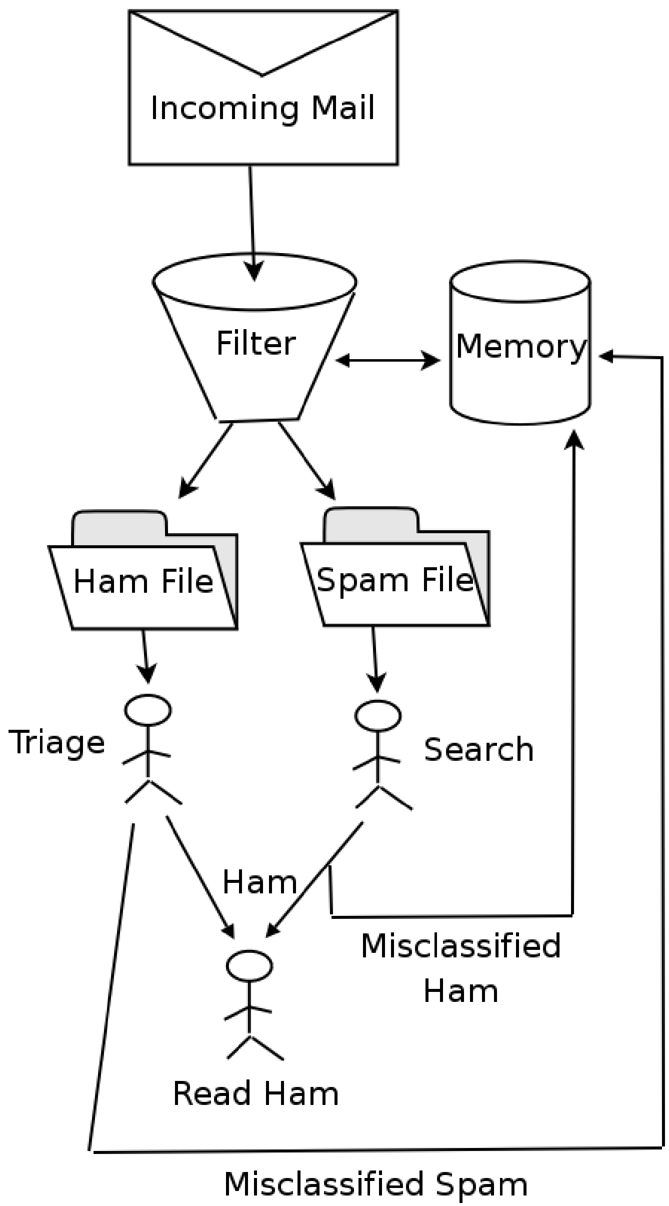
N be a string consisting of all known non-spam

E be an email message

Define spamminess

log (Prob(E following S) / Prob(E following N))

Measuring success



Collect email stream

- adjudicate as spam or ham
  - gold standard

Filter email to

- spam file if *spamminess* > ***t***
- ham file otherwise

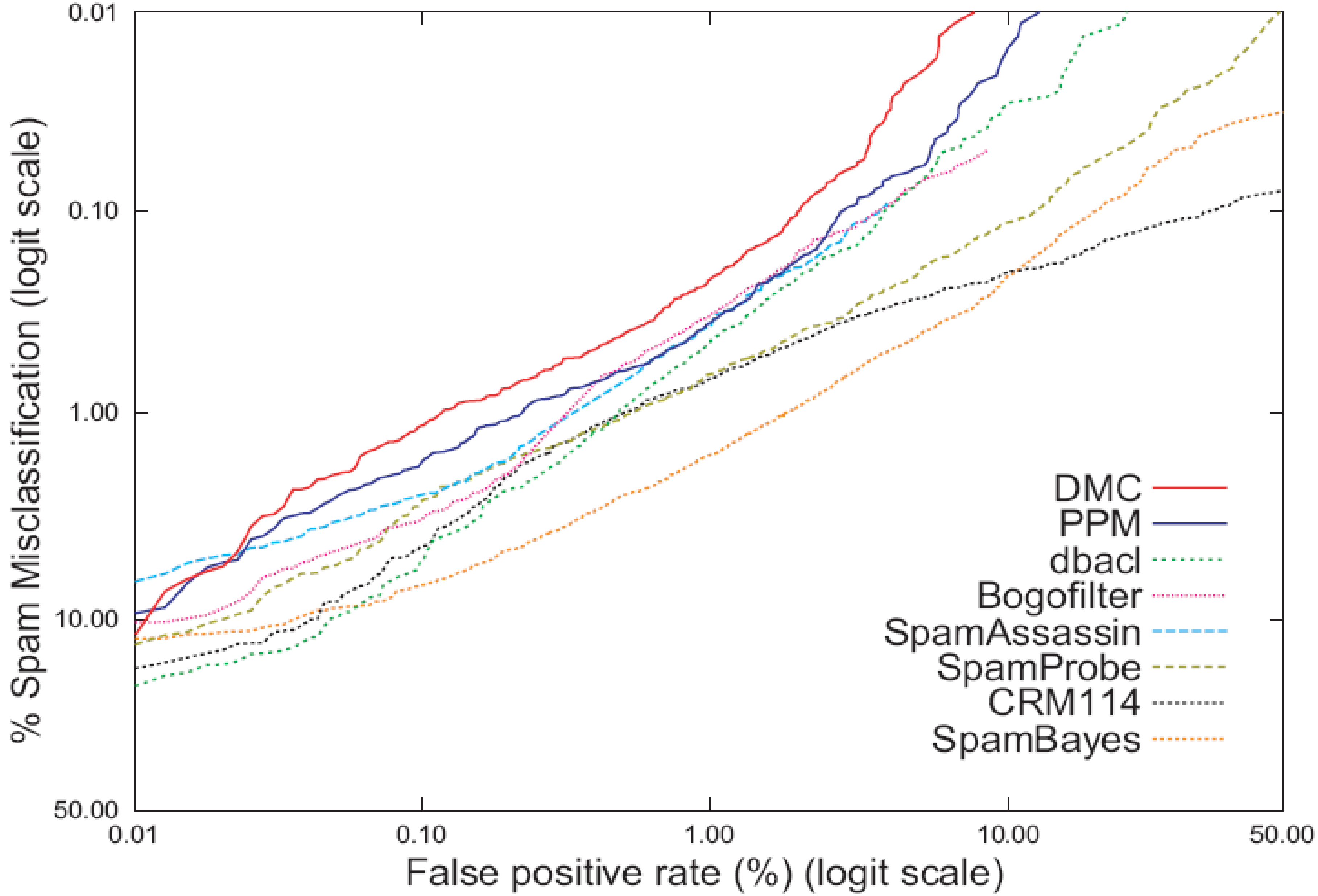
Idealized user

- reports errors immediately

Measure

- false positive rate
- spam misclassification rate

Receiver Operating Characteristic Curve



Summary statistics

TREC public corpus

Filter	1-AUC (%)	SMR at 1% FP	SMR at 0.1% FP	SMR at 0.01% FP
DMC	†**0.013** (0.010 – 0.018)	0.22%	1.17%	14.47%
PPM	†**0.019** (0.015 – 0.023)	0.36%	1.78%	9.89%
dbacl[b]	**0.037** (0.031 – 0.045)	0.45%	5.19%	19.77%
Bogofilter[b]	**0.048** (0.038 – 0.062)	0.33%	3.41%	10.39%
SpamAssassin[b]	**0.059** (0.044 – 0.081)	0.37%	2.56%	7.81%
SpamProbe	**0.059** (0.049 – 0.071)	0.65%	2.77%	15.30%
CRM114[b]	**0.122** (0.102 – 0.145)	0.68%	4.52%	17.17%
SpamBayes[b]	**0.164** (0.142 – 0.189)	1.63%	6.92%	12.55%

† improves on best TREC result ($p < .05$)

b TREC 2005 result

TREC – Text Retrieval Conference

Sponsored by, held at

- NIST – National Institute for Standards & Technology
- **http://trec.nist.gov**

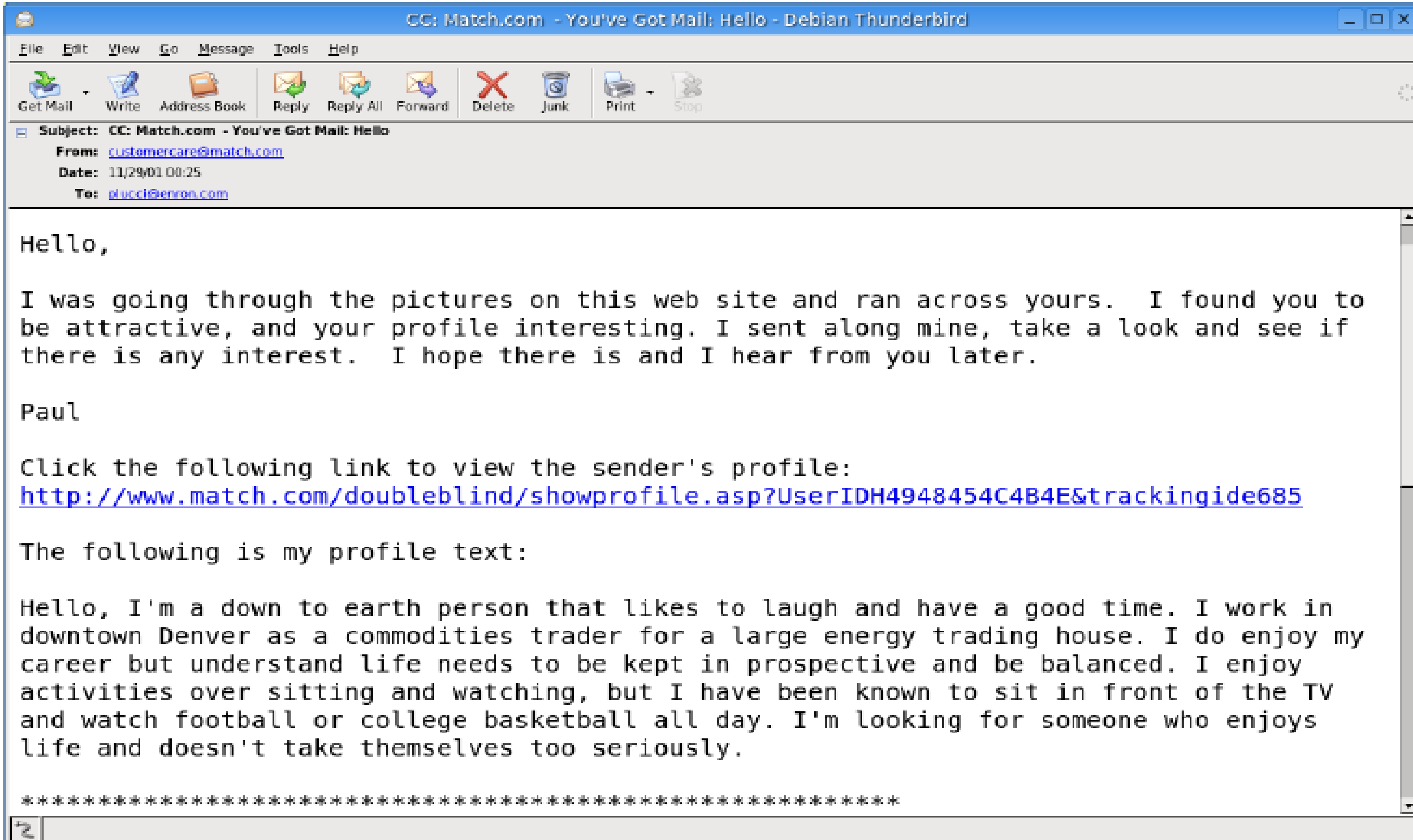
Goals

- 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*
- Standardized evaluation of well-defined tasks

Spam or Ham?



Subject: CC: Match.com - You've Got Mail: Hello
From: customercare@match.com
Date: 11/29/01 00:25
To: plucci@enron.com

Hello,

I was going through the pictures on this web site and ran across yours. I found you to be attractive, and your profile interesting. I sent along mine, take a look and see if there is any interest. I hope there is and I hear from you later.

Paul

Click the following link to view the sender's profile:
http://www.match.com/doubleblind/showprofile.asp?UserIDH4948454C4B4E&trackingide685

The following is my profile text:

Hello, I'm a down to earth person that likes to laugh and have a good time. I work in downtown Denver as a commodities trader for a large energy trading house. I do enjoy my career but understand life needs to be kept in prospective and be balanced. I enjoy activities over sitting and watching, but I have been known to sit in front of the TV and watch football or college basketball all day. I'm looking for someone who enjoys life and doesn't take themselves too seriously.

****

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

More information? *Google!*

cormack spam

TREC spam

DMC spam

DMC compression

ECML challenge

ROC curve

Markov model

PPM spam

OSBF Lua

Bogofilter

spamorham.org

spam conference

email anti-spam

likelihood ratio

machine learning

text classifier

Prediction by Partial Matching (PPM)

For each class:

- left context occurrences
- left context+prediction
- *log-likelihood* estimate
  - compressed length

Smoothing/backoff:

- zero occurrence problem

Adaptation:

- increment counts assuming in-class

ai.stanford.?

Context (509 spam, 1 ham)

ai.stanford.e

Prediction (0 spam, 1 ham)

ai.stanford.*E*

Prediction (509 spam, 0 ham)