Spam Filtering with Naive Bayes – *Which* Naive Bayes?

Vangelis Metsis[1,2], Ion Androutsopoulos[1] and Georgios Paliouras[2]

[1]Department of Informatics,
Athens University of Economics & Business

[2]Institute of Informatics &
Telecommunications, N.C.S.R. "Demokritos"

"We use a Naive Bayes classifier…"

- Naive Bayes is very popular in spam filtering.
  - Almost as accurate in SF as SVMs, AdaBoost, etc.
  - Much simpler, easy to understand and implement.
  - Linear computational and memory complexity.
- But there are many NB versions. ***Which one?***
  - Bayes' theorem + naive independence assumptions.
  - Different event models, instance representations.
  - Differences in performance, some unexpected.

What you are about to *hear*...

- A short discussion of 5 NB versions.
  - Multivariate Bernoulli NB (Boolean attributes)
  - Multinomial NB (frequency-valued attributes)
  - Multinomial NB with Boolean attributes (***strange!***)
  - Multivariate Gauss NB (real-valued attributes)
  - Flexible Bayes (John & Langley, kernels)
  - Better understanding may lead to improvements.
- Experiments on 6 ***new*** non-encoded datasets.
  - Approximations of 6 user mailboxes, preserving order of arrival, emulating ham:spam fluctuation, ...

What you are *not* going to hear...

- "Bayesian" methods that do *not* correspond to what is known as Naive Bayes, nor "Bayesian".
  - Though it would be interesting to compare!
- Filters that use information *other* than the bodies and subjects of the messages.
  - Operational filters include additional attributes or components for headers, attachments, etc.
- Filters trained on data from *many* users.
  - We only consider personal filters, each trained incrementally on messages from a single user.

Message representation

Get rich fast ! ! ! Visit now our online...

$$\vec{x} = \langle x_1, x_2, \ldots x_m \rangle$$

- Each message is represented by a vector of *m* attribute values (features).
- Each attribute corresponds to a token.

alternatives

  - Boolean attributes (token in message or not)
  - TF attributes (occurrences of token in message)
  - normalized TF (TF / message length in tokens)
- Attribute selection: token must occur in >4 training messages + Information Gain.

Message classification

Get rich fast ! ! ! Visit now our online...

$$\vec{x} = \langle x_1, x_2, \ldots x_m \rangle$$

From Bayes' theorem:

$$P(spam|\vec{x}) = \frac{P(spam) \cdot P(\vec{x}|spam)}{P(\vec{x})} \qquad P(ham|\vec{x}) = \frac{P(ham) \cdot P(\vec{x}|ham)}{P(\vec{x})}$$

- Classify as spam iff $P(spam|\vec{x}) \geq T$.
  - Varying $T \in [0,1]$: tradeoff between **wrongly** blocked hams (FPs) vs. **wrongly** blocked spams (FNs).

The multivariate Bernoulli NB

Get rich fast ! ! ! Visit now our online...

$$\vec{x} = \langle x_1, x_2, x_3, \ldots x_m \rangle = \langle 0, 1, 1, \ldots, 0 \rangle$$

"money" "rich" "!" "unsubscribe"

- Each Boolean attribute x_i shows if the corresponding token t_i occurs in the message.
- Event model: *m* ***independent*** Bernoulli trials.
  - Select independently the value of each attribute.

$$p(\vec{x}|spam) = \prod_{i}^{m} p(x_i|spam) = \prod_{i}^{m} p(t_i|spam)^{x_i} \cdot \left(1 - p(t_i|spam)\right)^{1-x_i}$$

$$p(t_i|spam) = \frac{1 + M_{t_i, spam}}{2 + M_{spam}}$$

$M_{t_i, spam}$: training spams with t_i

M_{spam}: training spams

$$p(\vec{x}|ham) = \ldots$$

The multinomial NB

Get rich fast ! ! ! Visit now our online...

$$\vec{x} = \langle x_1, x_2, x_3, \ldots x_m \rangle = \langle 0, 1, 3, \ldots, 0 \rangle$$

"money" "rich" "!" "unsubscribe"

- Each attribute x_i shows how many times the corresponding token t_i occurs in the message.
- Event model: pick *independently* with replacement tokens up to the length of the message, counted in tokens.

The multinomial NB – continued

Get rich
fast ! ! !
Visit
now our
online...

$$\vec{x} = \langle x_1, x_2, x_3, \ldots x_m \rangle = \langle 0, 1, 3, \ldots, 0 \rangle$$

"money" "rich" "!" "unsubscribe"

multinomial distribution:

$$p(\vec{x} \mid spam) = p(|d|) \cdot |d|! \prod_{i=1}^{m} \frac{p(t_i \mid spam)^{x_i}}{x_i!}$$

$$p(\vec{x} \mid ham) = \ldots$$

$|d|$: message length in tokens; we ***assume*** it does not depend on the category.

In effect a unigram language model per category; see refs for ***n-gram*** NB versions...

$$p(t_i \mid spam) = \frac{1 + N_{t_i, spam}}{m + N_{spam}}$$

$N_{t_i, spam}$: occurrences of t_i in training spams

N_{spam}: occurrences of all tokens in training spams

Multinomial NB, Boolean attributes

Get rich
fast ! ! !
Visit
now our
online...

$$\vec{x} = \langle x_1, x_2, x_3, \ldots x_m \rangle = \langle 0, 1, 1, \ldots, 0 \rangle$$

"money" "rich" "!" "unsubscribe"

- Same as before, but Boolean attributes.

$$p(\vec{x}|spam) = \cancel{p(|d|)} \cdot \cancel{|d|!} \frac{\prod_{i=1}^{m} p(t_i|spam)^{x_i}}{\cancel{x_i!}} \qquad p(\vec{x}|ham) = \ldots$$

- The multivariate Bernoulli NB (Boolean) considers more directly missing tokens

$$p(\vec{x}|spam) = \prod_{i}^{m} p(t_i|spam)^{x_i} \cdot \left(1 - p(t_i|spam)\right)^{1-x_i}$$

- and uses different estimates of $p(t_i|category)$.

Hold on, isn't this weird?

- An advantage of the multinomial NB is supposed to be that it accommodates TFs.
  - Previous work [McCallum & Nigam, Schneider, Hovold] shows it outperforms the (Boolean) multivariate Bernoulli NB.
- ***Why*** replace TFs with Boolean attributes?
  - It performs even better on Ling-Spam [Schneider].
  - With TF attributes, the multinomial NB in effect assumes that attributes follow Poisson distributions in each category [Eyheramendy et al.], which may not be true.

The multivariate Gauss NB

Get rich fast ! ! ! Visit now our online...

$$\vec{x} = \langle x_1, x_2, x_3, \ldots x_m \rangle = \langle 0,\ 0.01,\ 0.03,\ \ldots,\ 0 \rangle$$

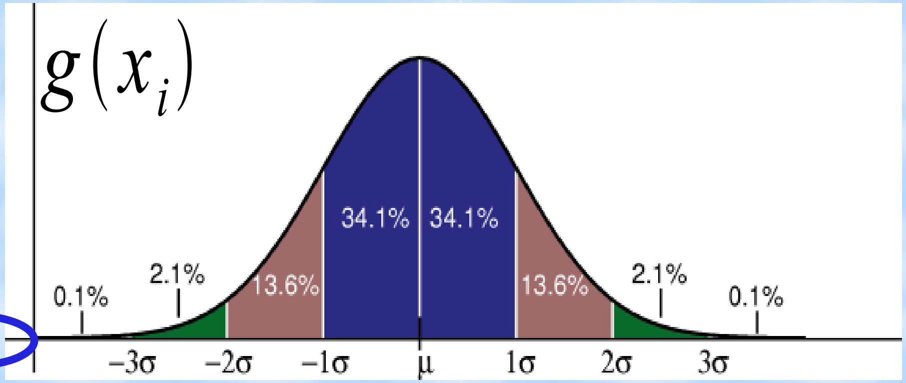
"money" "rich" "!" "unsubscribe"

- Attribute values: TFs / msg. length (in tokens).
- Independence assumption + assume attributes follow normal distributions per category.

$$p(\vec{x}|spam) = \prod_i^m p(x_i|spam) = \prod_i^m g(x_i; \mu_{i,spam}, \sigma_{i,spam})$$

estimated from training spams

Some probability mass is lost...



$$p(\vec{x}|ham) = \ldots$$

Flexible Bayes [John & Langley]

- Same as multivariate Gauss NB, but for each x_i we have as many normal distributions as the number of values x_i has in the training data.

l-th value of x_i in the training messages

$$p(\vec{x}|spam) = \prod_i^m p(x_i|spam) = \prod_i^m \frac{1}{L_i} \cdot \sum_{l=1}^{L_i} g(x_i; \mu_{i,l}, \sigma_{spam})$$

L_i: number of different values of x_i in spam training messages

$1/\sqrt{M_{spam}}$

normal distribution introduced by the l-th value of x_i in the spam training messages

$$p(\vec{x}|ham) = \ldots$$

- Multiple normal distributions allow us to approximate better the real distributions.

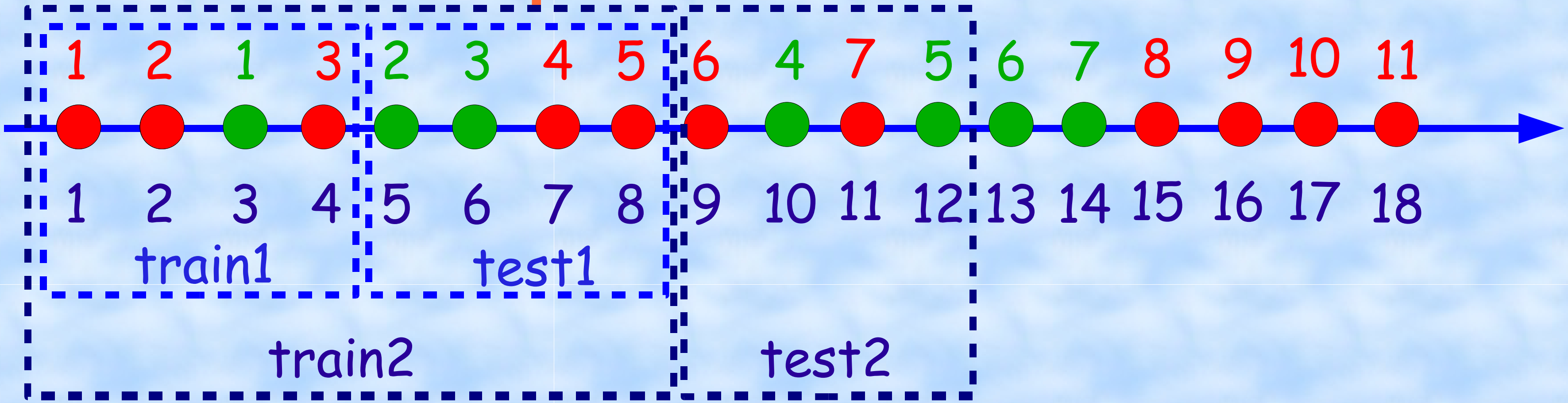
The Enron-Spam datasets

- 6 datasets, each emulating a user mailbox.
  - Hams from 6 Enron users.
  - Spams from 3 sources (G. Paliouras, B. Guenter, SpamAssassin+HoneyPot)

ham + spam	ham : spam
farmer-d + GP	3672 : 1500
kaminski-v + SH	4361 : 1496
kitchen-l + BG	4012 : 1500
williams-w3 + GP	1500 : 4500
beck-s + SH	1500 : 3675
lokay-m + BG	1500 : 4500

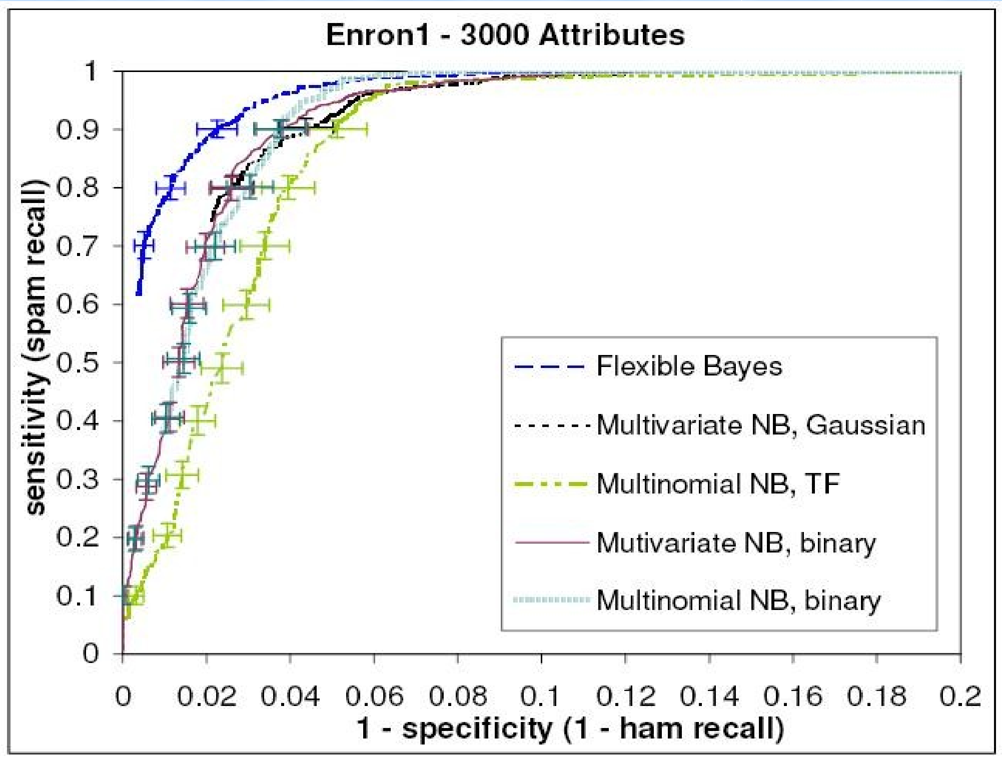
- Removed self-addressed messages, duplicates from spam traps, HTML, attachments, headers.
- Varying ham:spam ratios (approx. 3:1, 1:3).
- Available in both raw and preprocessed form.

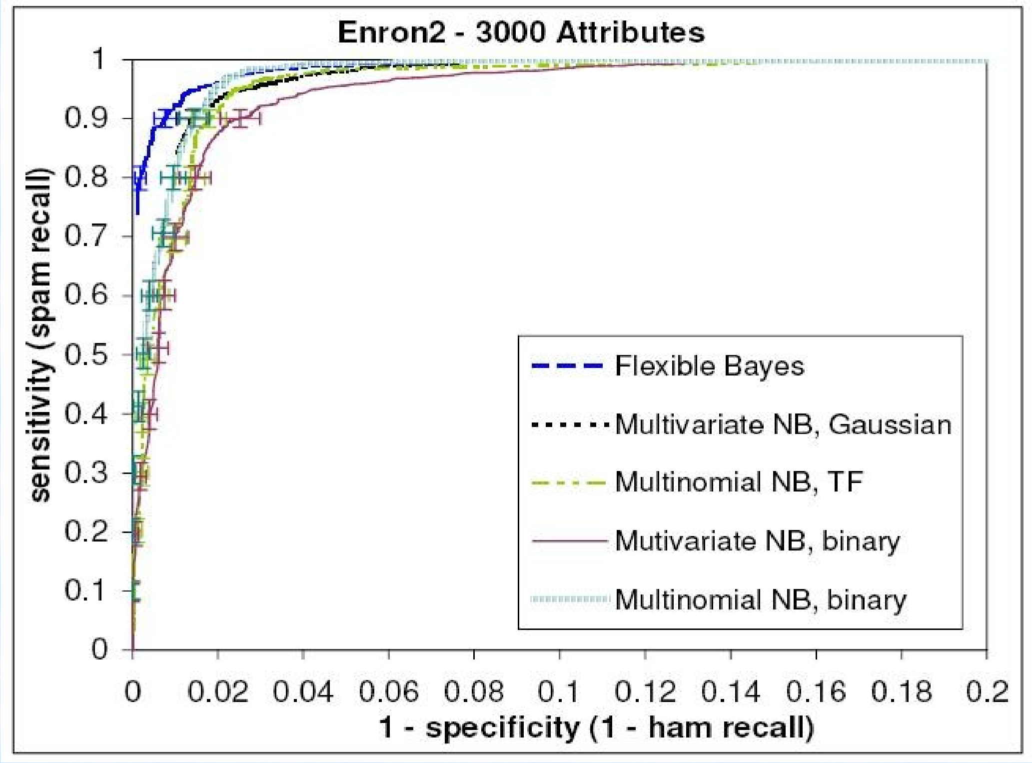
The Enron-Spam datasets – continued



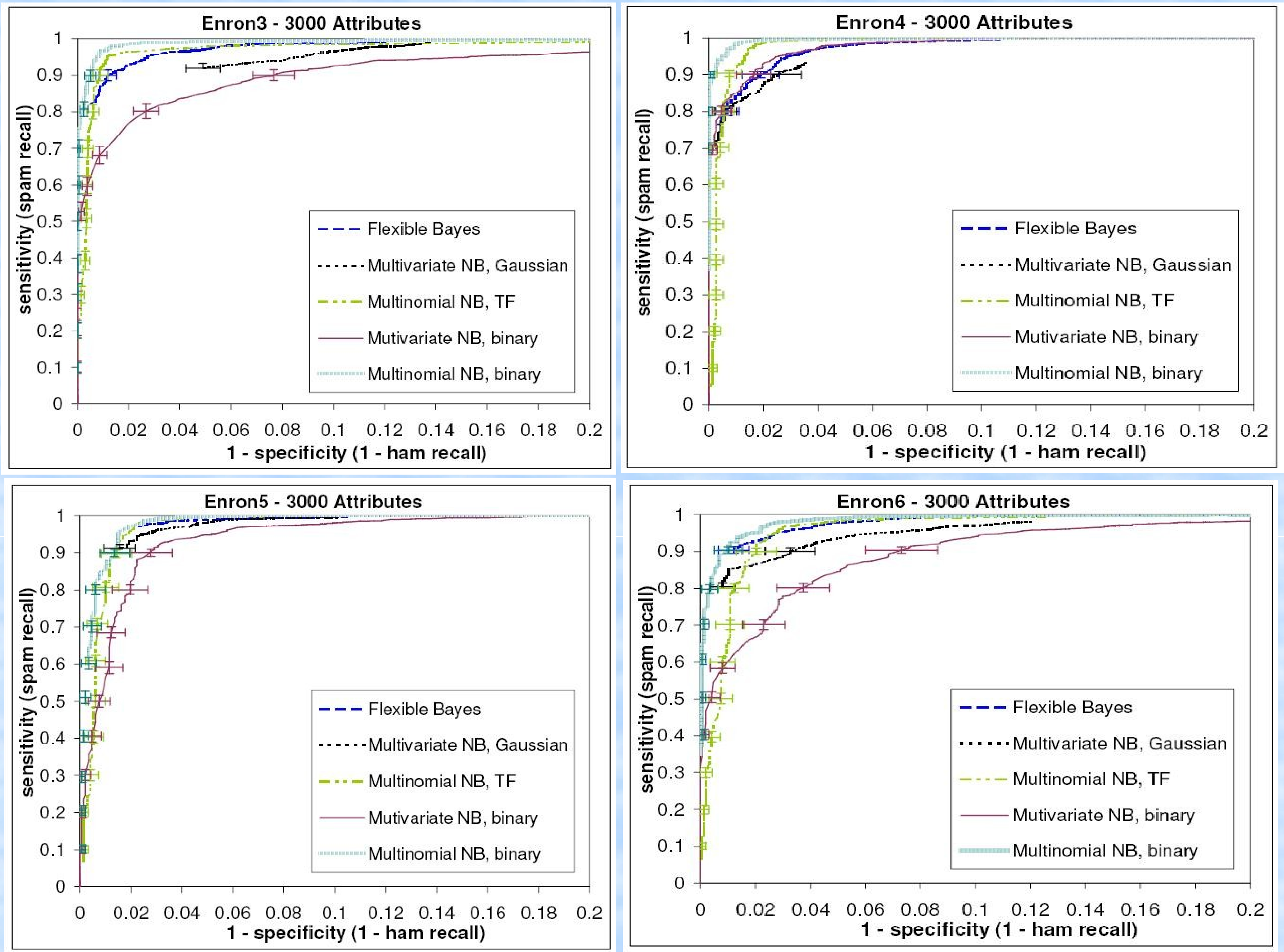
- In each dataset, we maintain the original order of arrival in each category.
- But otherwise, we order randomly, leading to worst-case ham:spam fluctuation.
- Incremental training/testing (batches of 100).
  - The user checks the "spam" folder and retrains every 100 received messages.

Which NB is best? – ROC curves





- The differences are not always statistically significant (95% confidence intervals).
- The rankings differ across the datasets.
- But some consistent top/worst performers.



Enron3 - 3000 Attributes
Enron4 - 3000 Attributes
Enron5 - 3000 Attributes
Enron6 - 3000 Attributes
sensitivity (spam recall)
1 - specificity (1 - ham recall)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.02
0.04
0.06
0.08
0.12
0.14
0.16
0.18
0.2
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary

Which NB is best? - summary

- On all datasets, the multinomial NB did better with Boolean attributes than with TF ones.
  - We confirmed Scheider's observations.
  - But stat. significant difference in only 2 datasets.
- The Boolean multinomial NB was also the top performer in 4/6 datasets, and was clearly outperformed only by Flexible Bayes (in 2/6).
  - But again not always stat. significant differences.
- The multivariate Bernoulli is clearly the worst.

Which NB is best? - continued

- Flexible Bayes impressively superior in 2/6 datasets, and among top-performers in 4/6.
  - But skewed "probabilities", not allowing to reach ham recall > 99.90%, unlike other NB versions.
  - The same applies to the multivariate Gauss NB.
- Flexible Bayes clearly outperforms the multivariate Gauss NB (norm. TF), but not always the multinomial NB with TF attributes.
- Overall the Boolean multinomial NB seems to be the best, but more experiments needed.

How many attributes should I use?

- We tried 500, 1000, 3000 (token) attributes.
- Best results for 3000 attributes, but *very* small differences; see paper.
- May not be worth using very large attribute sets in operational filters.
  - Though linear computational complexity.
  - Training: *O(attributes x training_msgs)*.
  - Classification FB: *O(attributes x training_msgs)*.
  - Classification others: *O(attributes)*.

Anything to remember then?

- Don't just say *"we use Naive Bayes"*...
- Don't use the multivariate Bernoulli NB.
- If you use the multinomial NB, try Boolean.
  - You may also want to consider n-gram models and other improvements; see references.
- Worth investigating further Flexible Bayes.
- Very large attribute sets may be unnecessary.
- 6 new non-encoded emulations of mailboxes.
  - Six real mailboxes coming soon, but PU encoding.