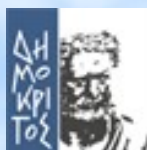


Spam Filtering with Naive Bayes - *Which* Naive Bayes?

Vangelis Metsis^{1,2}, Ion Androutsopoulos¹ and
Georgios Paliouras²



¹Department of Informatics,
Athens University of Economics & Business



²Institute 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...


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

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

alternatives

Message classification

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

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

From Bayes' theorem:

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

- Classify as **spam** iff $P(\text{spam}|\vec{x}) \geq T$.
 - 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...

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

"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.

$$p(\vec{x} | spam) = \prod_i^m p(x_i | spam) = \prod_i^m p(t_i | spam)^{x_i} (1 - p(t_i | spam))^{1-x_i}$$

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

← training spams with t_i
← training spams

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

The multinomial NB

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

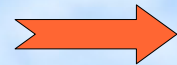
→ $\vec{x} = \langle x_1, x_2, x_3, \dots, x_m \rangle = \langle 0, 1, 3, \dots, 0 \rangle$

"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...


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

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

multinomial distribution:

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

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

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

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

occurrences of t_i in training **spams**

occurrences of all tokens in training **spams**

Multinomial NB, Boolean attributes

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

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

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

- Same as before, but Boolean attributes.

$$p(\vec{x}|spam) = \frac{\prod_{i=1}^m p(t_i|spam)^{x_i}}{x!} \quad p(\vec{x}|ham) = \dots$$

(Note: In the original image, the terms $p(\vec{x}|spam)$, $p(|d|) \cdot |d|!$, and $x!$ are crossed out with blue X's, and $p(\vec{x}|spam)$ is circled in red.)

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

$$p(\vec{x}|spam) = \prod_i p(t_i|spam)^{x_i} \cdot (1 - p(t_i|spam))^{1-x_i}$$

(Note: In the original image, $p(\vec{x}|spam)$ is circled in red, and the second part of the formula is enclosed in a dashed orange oval.)

- 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...

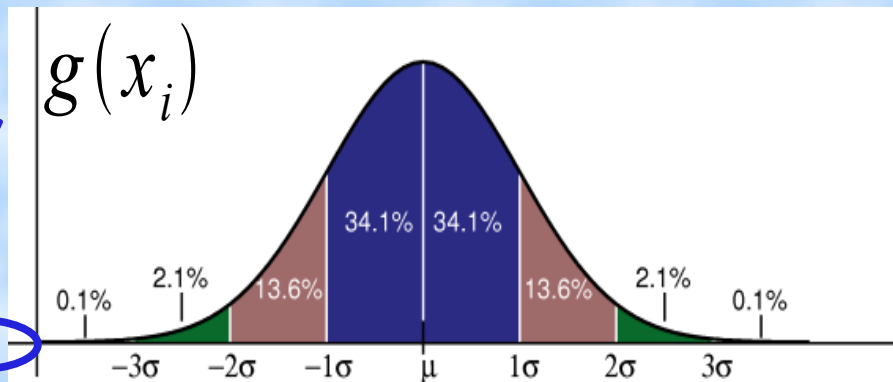
→ $\vec{x} = \langle x_1, x_2, x_3, \dots, x_m \rangle = \langle 0, 0.01, 0.03, \dots, 0 \rangle$

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

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

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

Some
probability
mass is
lost...



estimated from training **spams**

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

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

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

$L_{i,c}$: number of different values x_i has in the training instances of category c.

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

$p(\vec{x}|ham) = \dots$ normal distribution introduced by the l -th value of x_i in the training messages

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

The Enron-Spam datasets

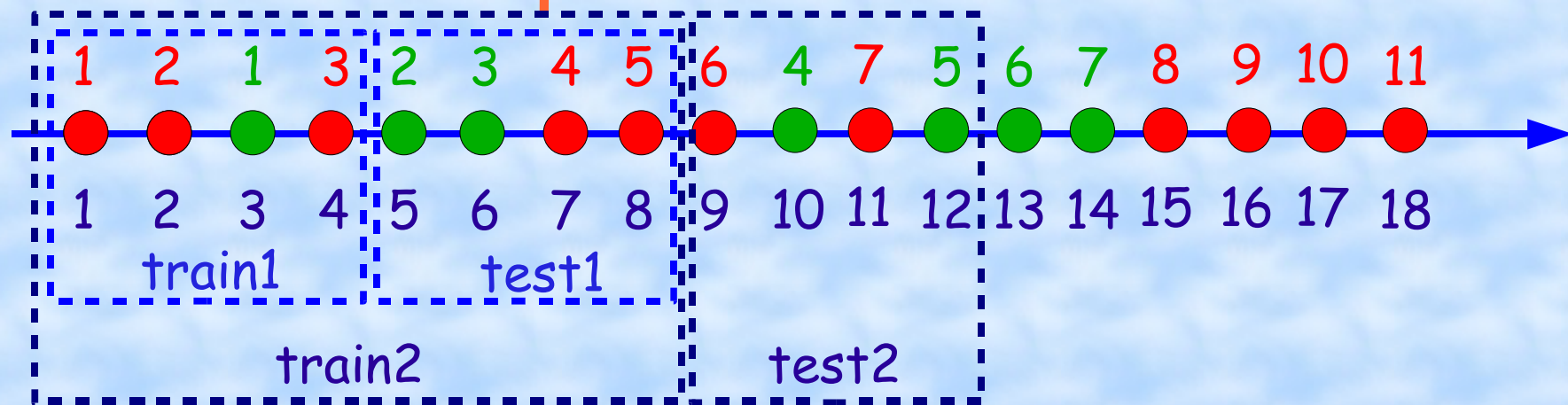
- 6 datasets, each emulating a user mailbox.

- 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

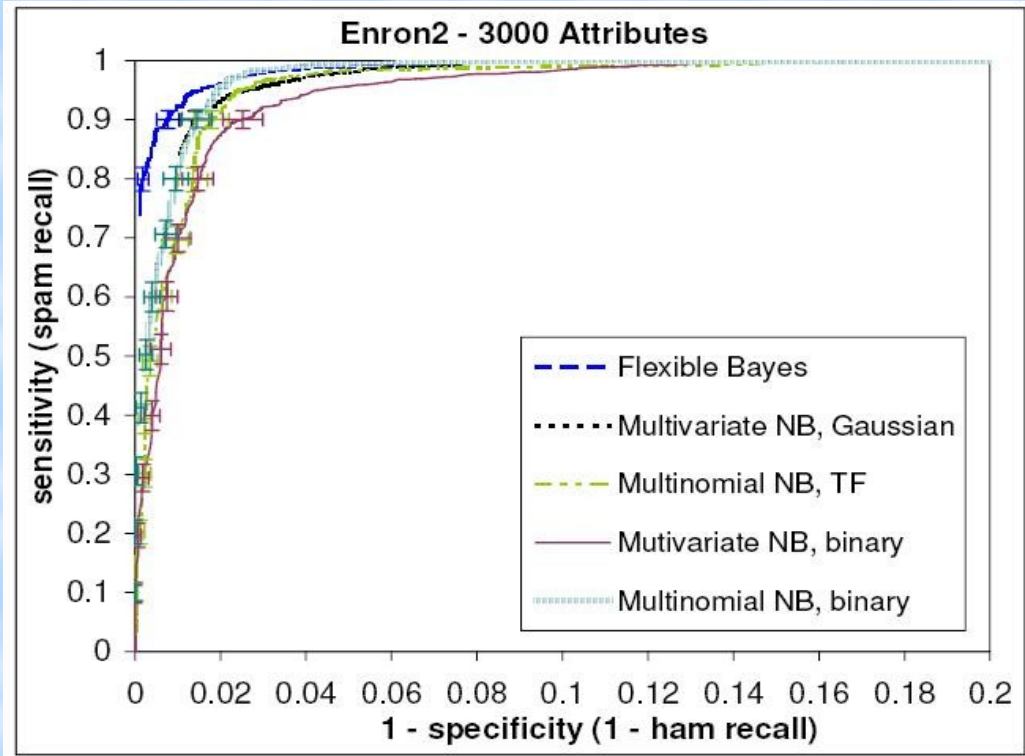
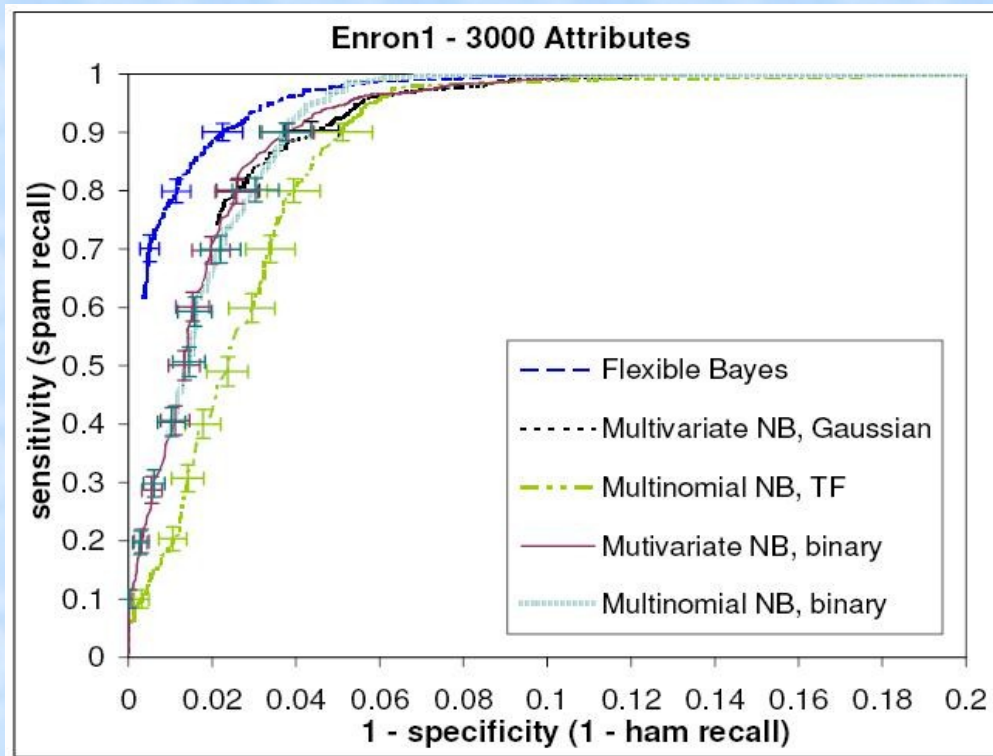
- Removed self-addressed messages, duplicates from spam traps, HTML, attachments, headers.
- Varying ham:spam ratios (approx. 3:1, 1:3).
- Available in both raw and preprocessed form.

The Enron-Spam datasets - continued

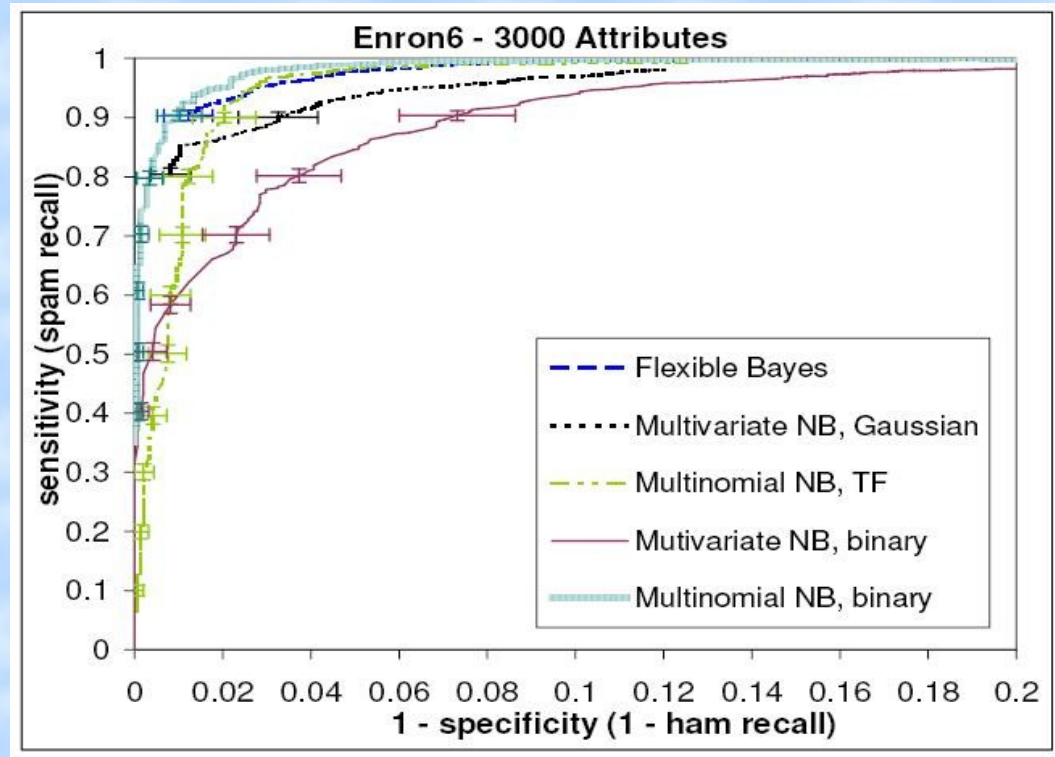
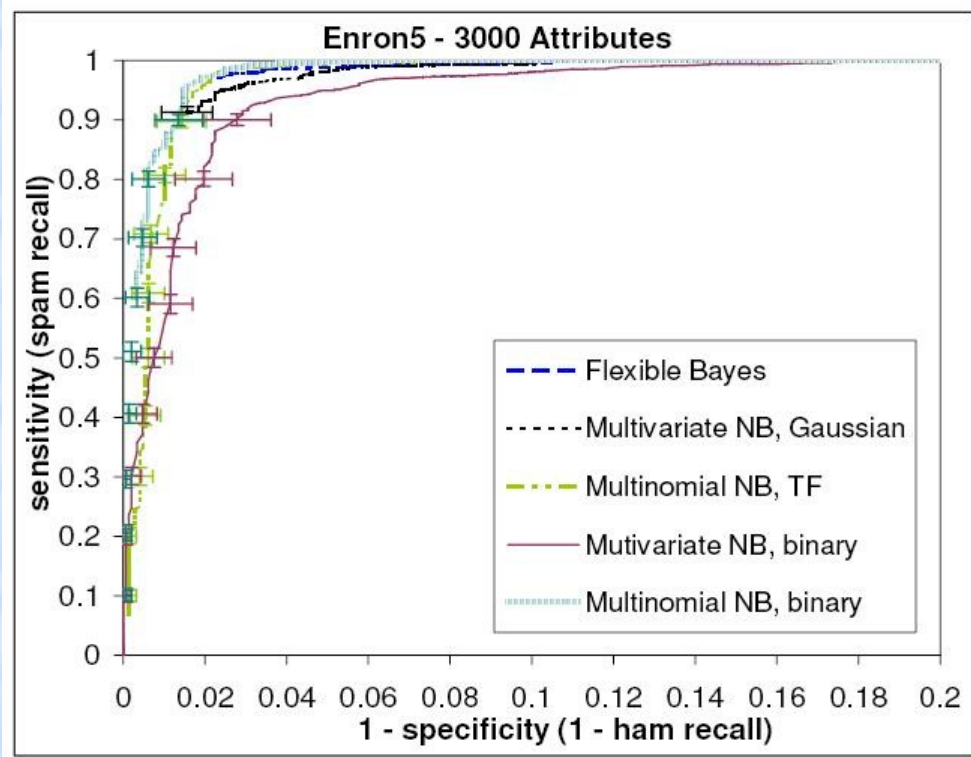
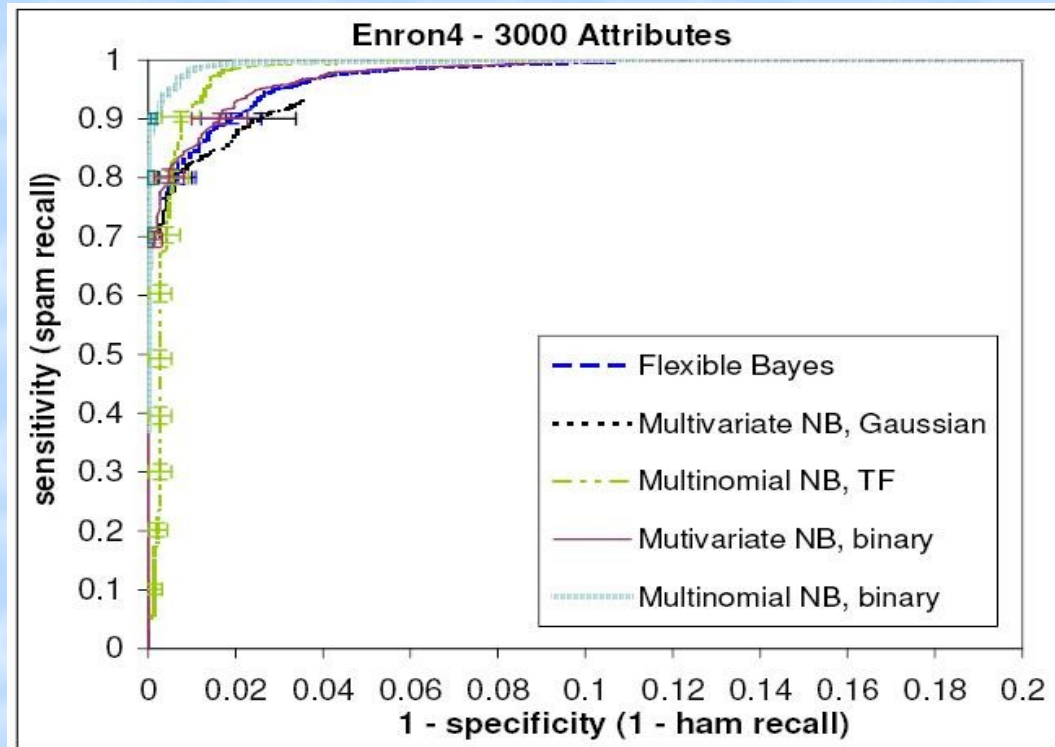
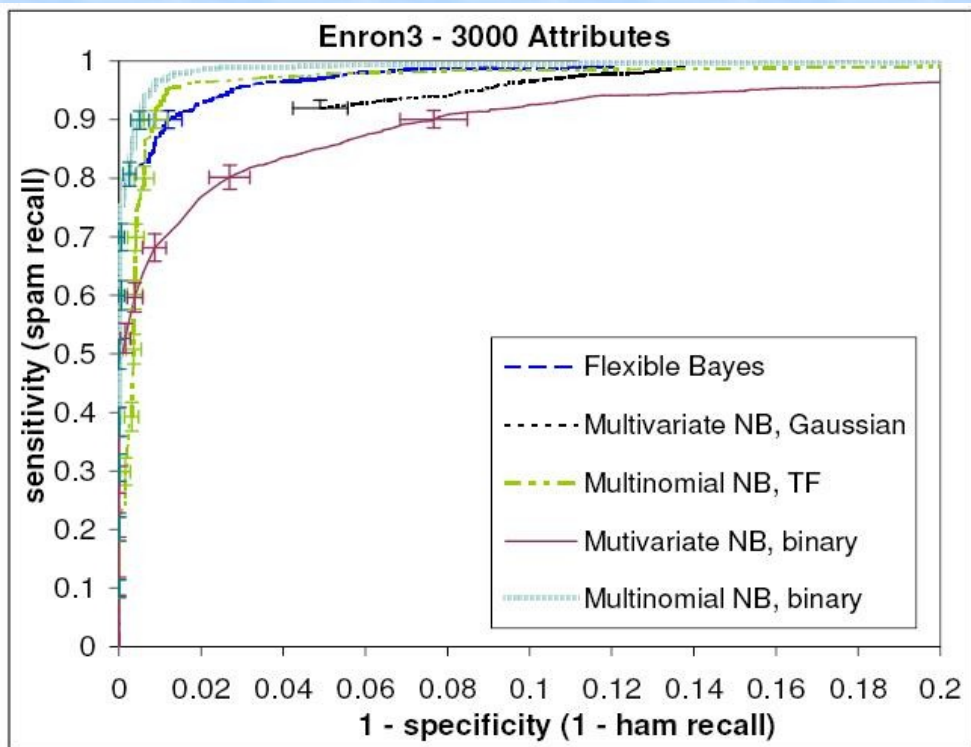


- In each dataset, we maintain the original order of arrival in each category.
- But otherwise, we order randomly, leading to worst-case ham:spam fluctuation.
- Incremental training/testing (batches of 100).
 - The user checks the "spam" folder and retrains every 100 received messages.

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.



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: $O(\text{attributes} \times \text{training_msgs})$.
 - Classification FB: $O(\text{attributes} \times \text{training_msgs})$.
 - Classification others: $O(\text{attributes})$.

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.