Spam Filtering with Naïve Bayes – Which Naïve Bayes?

Metsis, Androutsopoulus, Paliouras

Presentation by Brett Meyer

Naïve Bayes

- Statistical method employed by text classification systems
- Uses Bayesian probability with the “naïve” assumption of conditional independence
- Surprisingly effective despite this assumption

Naïve Bayes

- Usually takes a “Bag of Words” approach when used with text classification
- Puts all tokens into a vector of size m
- For each instance, a vector ‹x_1, … , x_m› is created where each x_i is one of two types of values:
  - A Term Frequency, which calculates the number of times each token appears in a given instance
  - A Boolean value as to whether the token appears in the instance

Naïve Bayes

- Generally, for a vector x the probability that an instance with vector x belongs in category c is

$$p(c \mid \vec{x}) = \frac{p(c) \cdot p(\vec{x} \mid c)}{p(\vec{x})}$$

- In the case of spam filtering, given c_h as a ham category, c_s as a spam category, and a threshold T = 0.5, a message is classified as spam when

$$\frac{p(c_s) \cdot p(\vec{x} \mid c_s)}{p(c_s) \cdot p(\vec{x} \mid c_s) + p(c_h) \cdot p(\vec{x} \mid c_h)} > T$$

Naïve Bayes Versions

- Multi-variate Bernoulli
- Multinomial w/ Term Frequency attributes
- Multinomial w/ Boolean attributes
- Multi-variate Gaussian
- Flexible Bayes

Multi-variate Bernoulli NB

- Each message is treated as a set of tokens
- Each x_i in each message vector has a Boolean value as to whether the token occurs in the message
- Each message is seen as a result of m Bernoulli trials, where each trial decides whether or not each token will occur in the message
- Uses a Laplacean prior to estimate $p(t \mid c)$ where t is a token

Multinomial NB, Term Frequency attributes

- Each message is treated as a bag of tokens
- Each x_i in each message vector has a numeric value as to the number of times the token occurs in the message
- Each message is seen as picking each token from the vector of attributes with probability $p(t \mid c)$ for each token
- Additional assumption that the number of tokens chosen does not depend on the category
- This is limiting for spam filtering because it assumes that the probability for receiving a long spam message appears to be less than that of receiving an equally long ham message
- Uses a Laplacean prior to estimate $p(t \mid c)$ where t is a token

Multinomial NB, Boolean attributes

- Same as with Term Frequency attributes, except the attributes are Boolean

- Differs from Multi-variate Bernoulli NB because it does not directly take into account the absence of tokens from the message, and uses a different Laplacean prior for estimating $p(t \mid c)$

Multi-variate Gauss NB

- Modifies the Multi-variate Bernoulli NB to use real-valued attributes by assuming that each attribute follows a Gaussian distribution
- Mean and typical deviation estimated from the training data
- Employed *normalized* Term Frequencies for Multi-variate Gauss NB and Flexible Bayes
  - Term Frequencies divided by the total number of occurrences in the message
  - Takes into account the message's length

Flexible Bayes

- Instead of using a single normal distribution for each attribute per category, takes $p(x_i \mid c)$ to be the average of many normal distributions with different mean values but the same typical deviation
- Number of normal distributions is the number of values an attribute can take for each category
- Each of these numbers is used as the mean of a normal distribution of that category
- By averaging several normal distributions, can approximate the true distributions of real-valued attributes more closely than Multi-variate Gauss NB

Dataset

- No uniform benchmark dataset for measurement
- Mostly because of privacy issues
- But wait! Enron's emails just became public record!
- Chose 6 Enron employee email accounts, cleaned up to only include ham messages

Dataset

- Four different spam sources
  - Spam Assassin corpus
  - Honeypot project
  - spam collection of Bruce Guenter
  - spam collected by Georgios Paliouras
- First three collected through the use of traps
- Removed duplicates from first three and merged first and second source
- Duplicates left in for fourth source since it didn’t use traps, so the duplicates were part of a normal traffic stream

Dataset

- Merged each of the six ham message collections with one of the three spam collections
- Varied the ham-spam ratio so that in the first three resultant datasets the ham-spam ratio was 3:1, while in the last three it was 1:3
- Around five to six thousand messages in each benchmark dataset

Table 1: Composition of the six benchmark datasets.

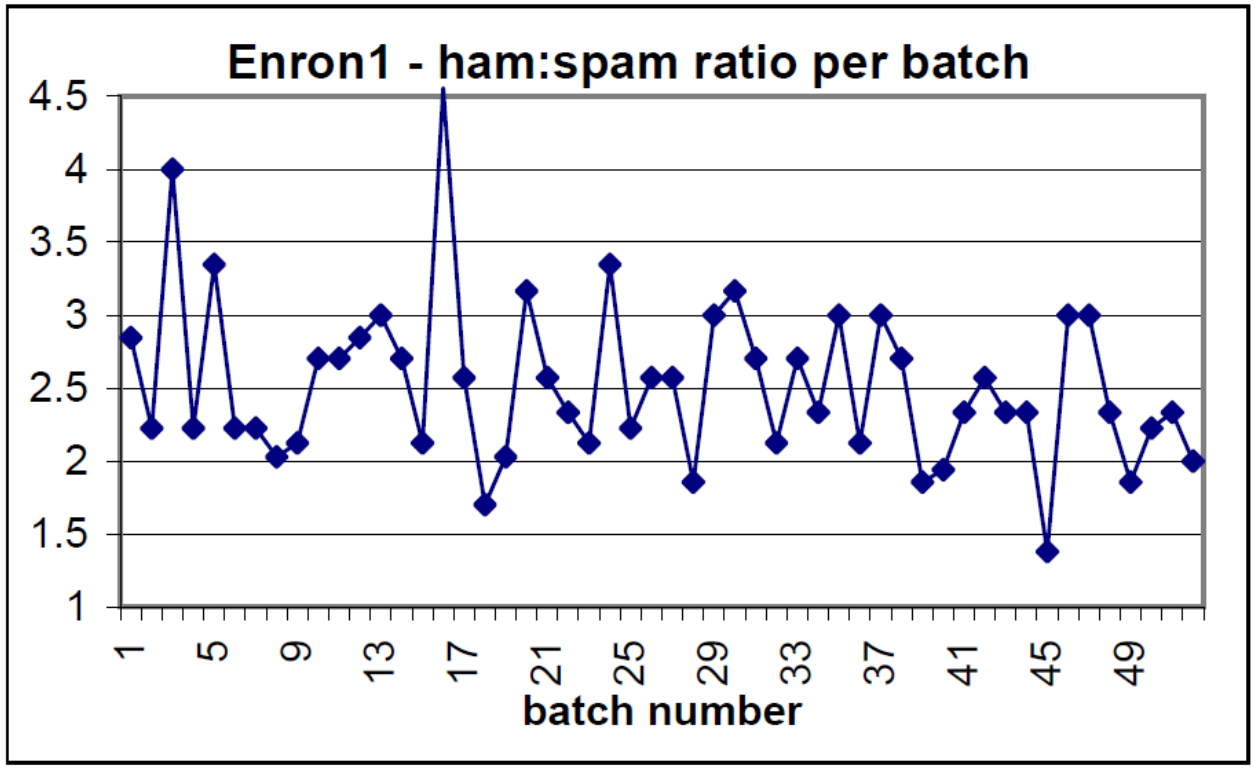
ham + spam	ham:spam	ham, spam periods
farmer-d + GP	3672:1500	[12/99, 1/02], [12/03, 9/05]
kaminski-v + SH	4361:1496	[12/99, 5/01], [5/01, 7/05]
kitchen-l + BG	4012:1500	[2/01, 2/02], [8/04, 7/05]
williams-w3 + GP	1500:4500	[4/01, 2/02], 12/03, 9/05]
beck-s + SH	1500:3675	[1/00, 5/01], [5/01, 7/05]
lokay-m + BG	1500:4500	[6/00, 3/02], [8/04, 7/05]

Dataset

- The benchmark datasets were then modified as follows:
  - Messages sent by the owner of the mailbox were removed
  - All HTML tags and headers of the messages were removed
  - All spam messages written in non-Latin character sets were removed

Dataset

- One main objective was to emulate real-world spam conditions, e.g. incremental retraining and evaluation
- The original ordering of the ham messages was preserved, and spam injected at random intervals with a varying distribution rate



Training the spam filter

- For each ordered dataset, the incremental retraining and evaluation procedure was implemented as follows:
  - Split the sequence of messages into batches $b_1, \ldots, b_i$ of k adjacent messages each, preserving the order of arrival
  - For $i = 1$ to $l - 1$, train the filter (including attribute selection) on the messages of batches 1, … , i, and test it on the messages of b_{i+1}
- k set to 100

Evaluation

- Spam recall $\left(\frac{TP}{TP+FN}\right)$
- Ham recall $\left(\frac{TN}{TN+FP}\right)$
- ROC curves (spam recall vs. 1 – ham recall)
- Learning curves of incremental retraining and evaluation

Experiment

- Did not assign attributes to tokens that are too rare
  - Discarded tokens that did not occur in at least 5 messages of the training data
- Ranked remaining attributes by information gain, and used only the m best (remember that each message transforms to a vector $\langle x_1, \ldots, x_m \rangle$ with m attributes)
- Experimented with m = 500, 1000, 3000

Results

- Best results achieved with 3000 attributes
- Differences in effectiveness for fewer attributes very small
- Differences are insignificant across all five versions of NB and for all threshold values
- Thus, the increased number of attributes for greater effectiveness may not justify the increased computational cost of the filter, even though the increase is linear

NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6
FB	7.87	3.46	1.43	1.31	0.11	0.34
MV Gauss	5.56	4.75	1.97	12.7	3.36	5.27
MN TF	0.88	0.95	0.20	0.50	0.75	0.18
MV Bernoulli	2.10	0.95	1.09	0.45	1.14	0.88
MN Boolean	2.31	1.97	2.04	0.43	0.39	0.20

Table 2: Maximum difference ($\times 100$) in spam recall across 500, 1000, 3000 attributes for $T = 0.5$.

NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6
FB	0.61	0.23	1.72	0.54	0.48	0.34
MV Gauss	1.17	0.75	5.94	1.77	5.91	4.88
MN TF	2.17	1.38	1.02	0.61	1.70	1.22
MV Bernoulli	1.47	0.63	6.37	2.04	2.11	1.22
MN Boolean	0.53	0.68	0.10	0.48	1.36	2.17

Table 3: Maximum difference ($\times 100$) in ham recall across 500, 1000, 3000 attributes for $T = 0.5$.

Results

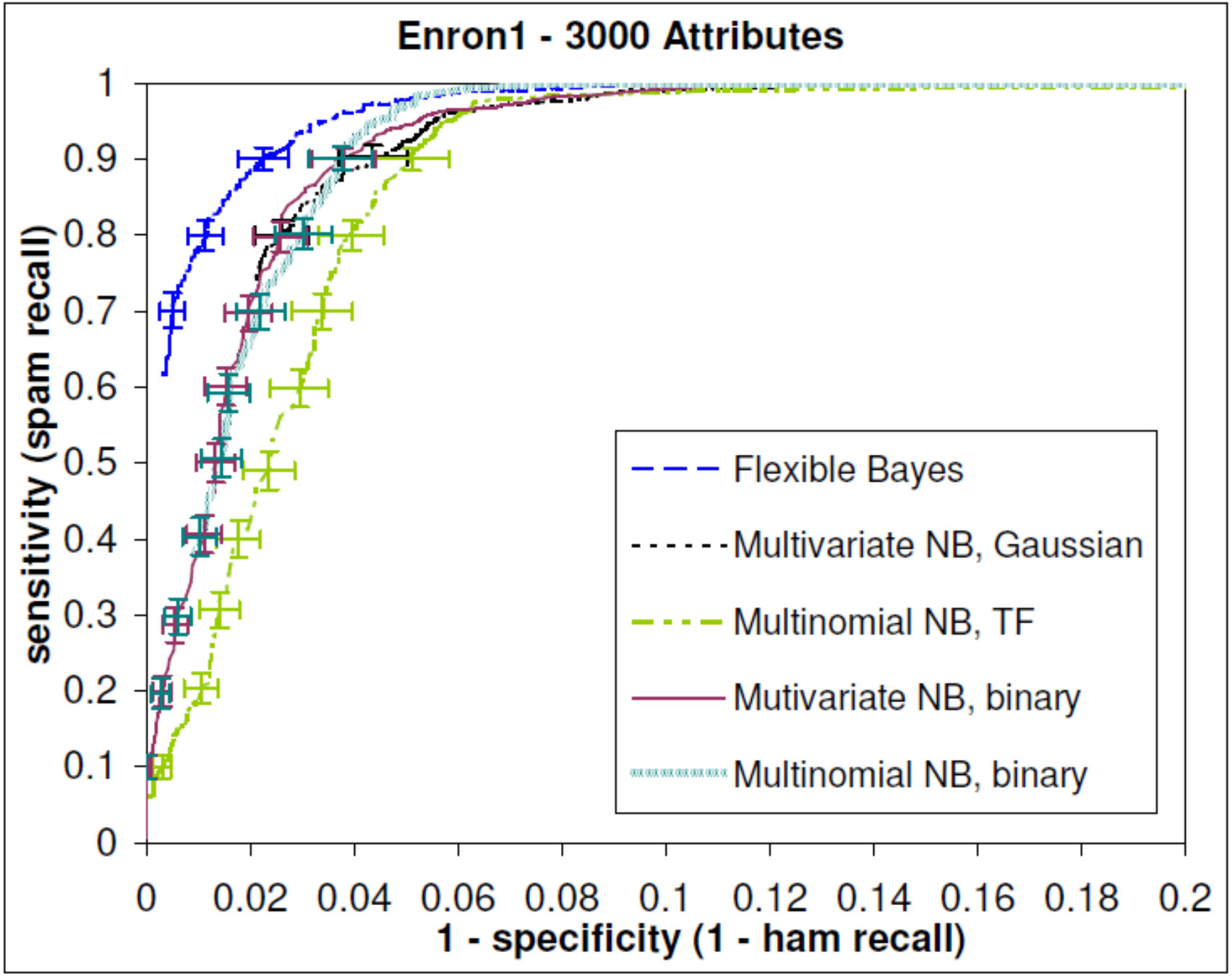
- Multinomial NB w/ Boolean attributes performs best in 4 out of 6 datasets
- Flexible Bayes performs best in the other 2
- Differences in performance between versions of NB typically very small

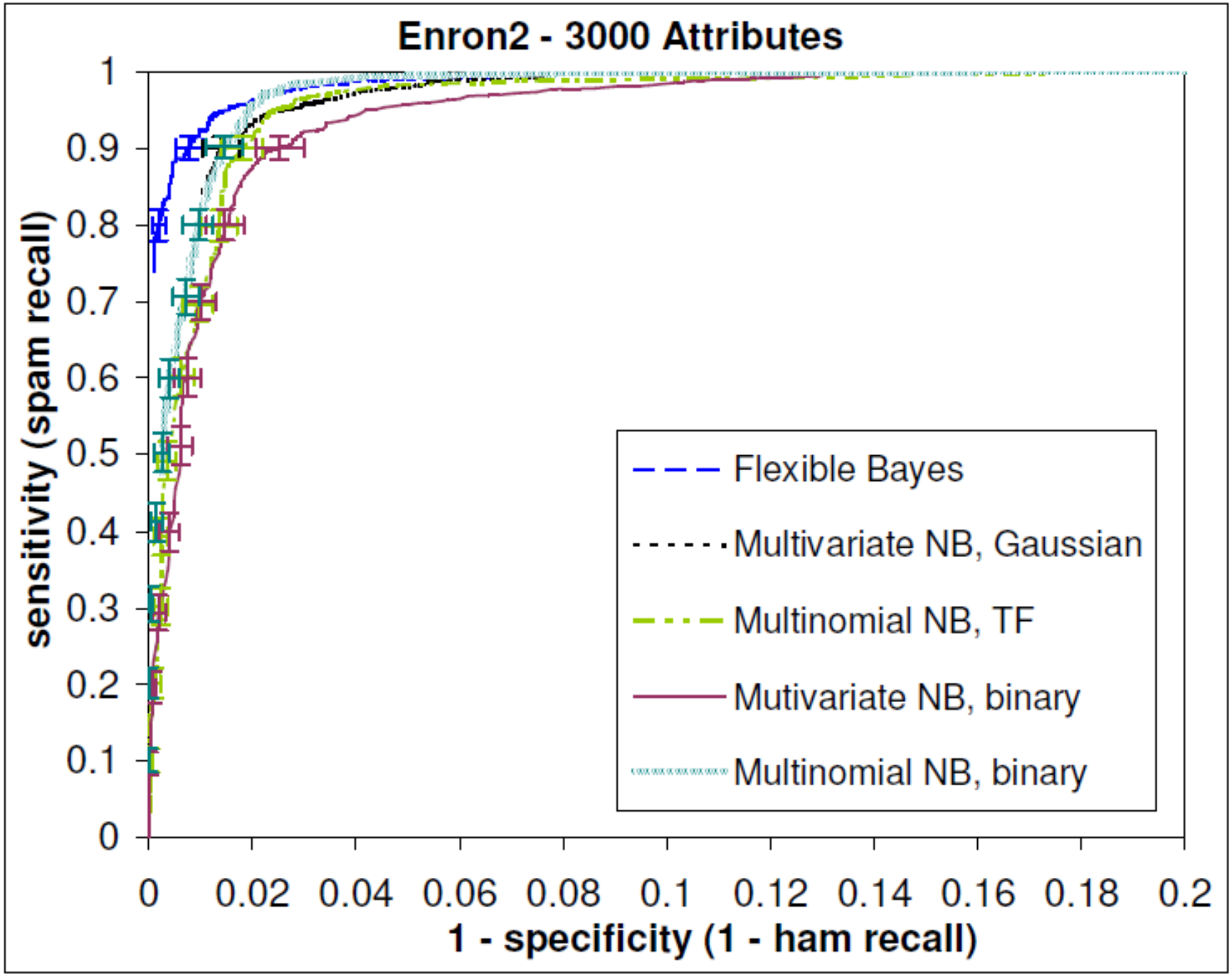
NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6	Avg.
FB	90.50	93.63	96.94	95.78	99.56	99.55	95.99
MV Gauss	93.08	95.80	97.55	80.14	95.42	91.95	92.32
MN TF	95.66	96.81	95.04	97.79	99.42	98.08	97.13
MV Bern.	97.08	91.05	97.42	97.70	97.95	97.92	96.52
MN Bool.	96.00	96.68	96.94	97.79	99.69	98.10	**97.53**

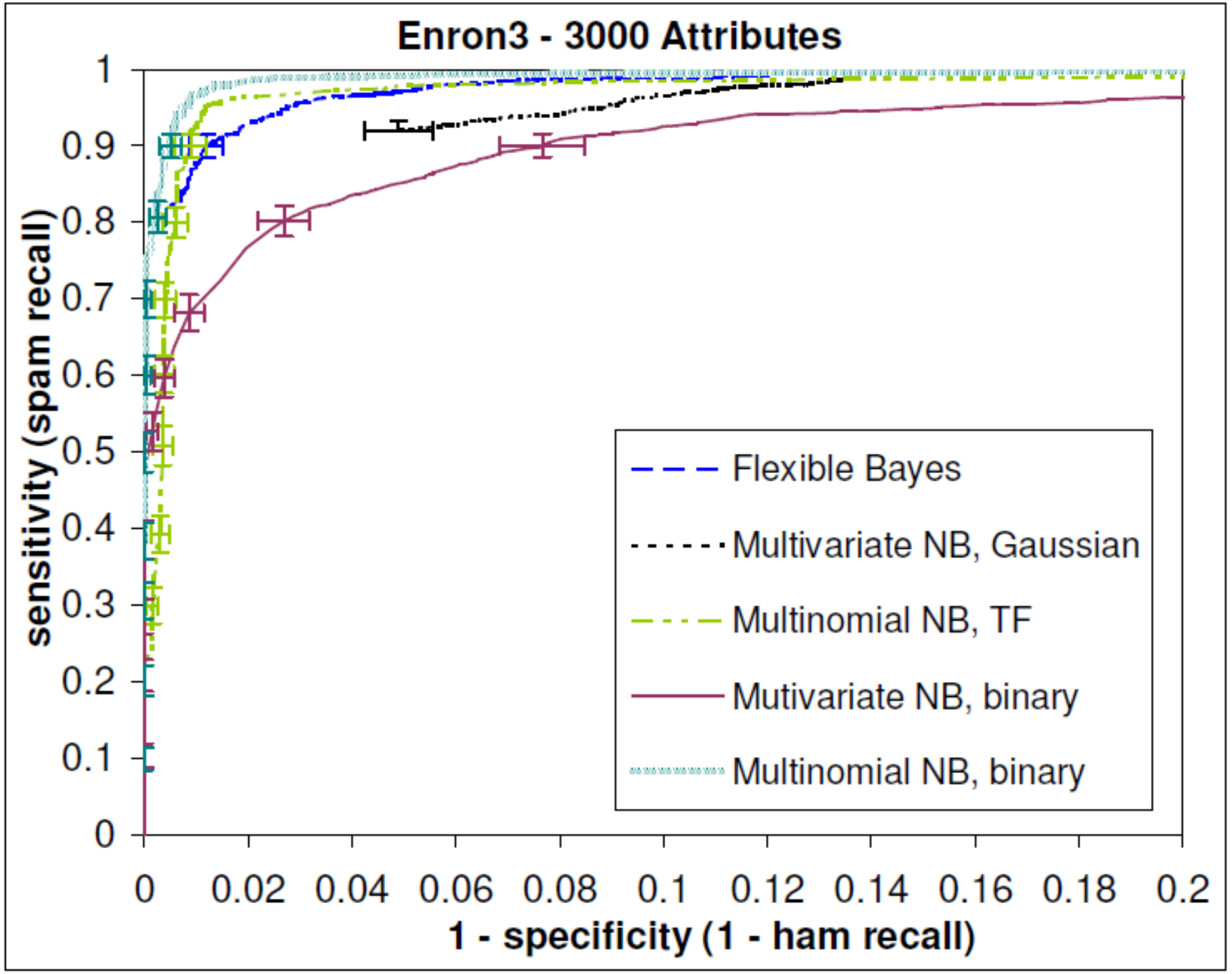
Table 4: Spam recall (%) for 3000 attributes, $T = 0.5$.

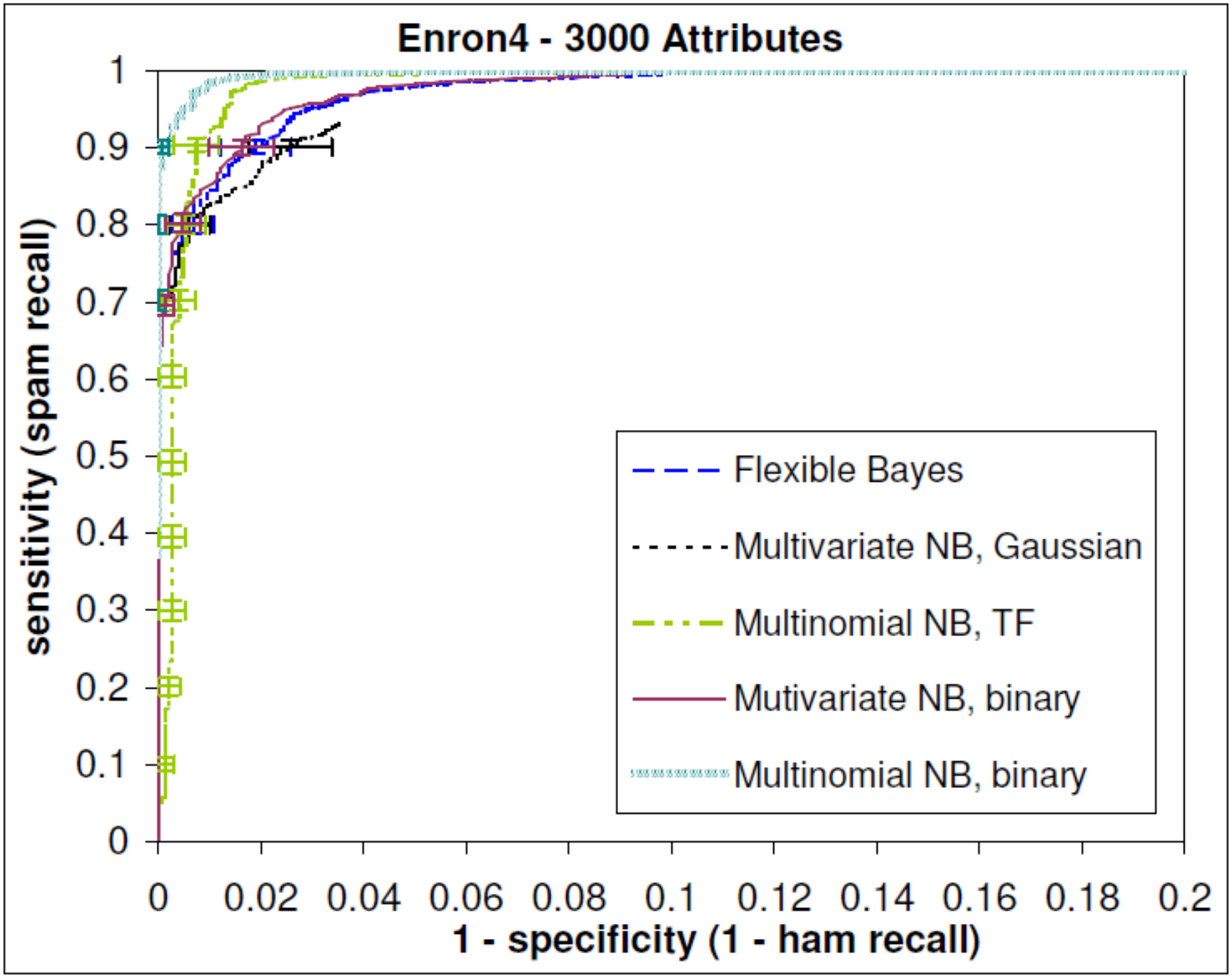
NB version	Enr1	Enr2	Enr3	Enr4	Enr5	Enr6	Avg.
FB	97.64	98.83	95.36	96.61	90.76	89.97	94.86
MV Gauss	94.83	96.97	88.81	99.39	97.28	95.87	95.53
MN TF	94.00	96.78	98.83	98.30	95.65	95.12	96.45
MV Bern.	93.19	97.22	75.41	95.86	90.08	82.52	89.05
MN Bool.	95.25	97.83	98.88	99.05	95.65	96.88	**97.26**

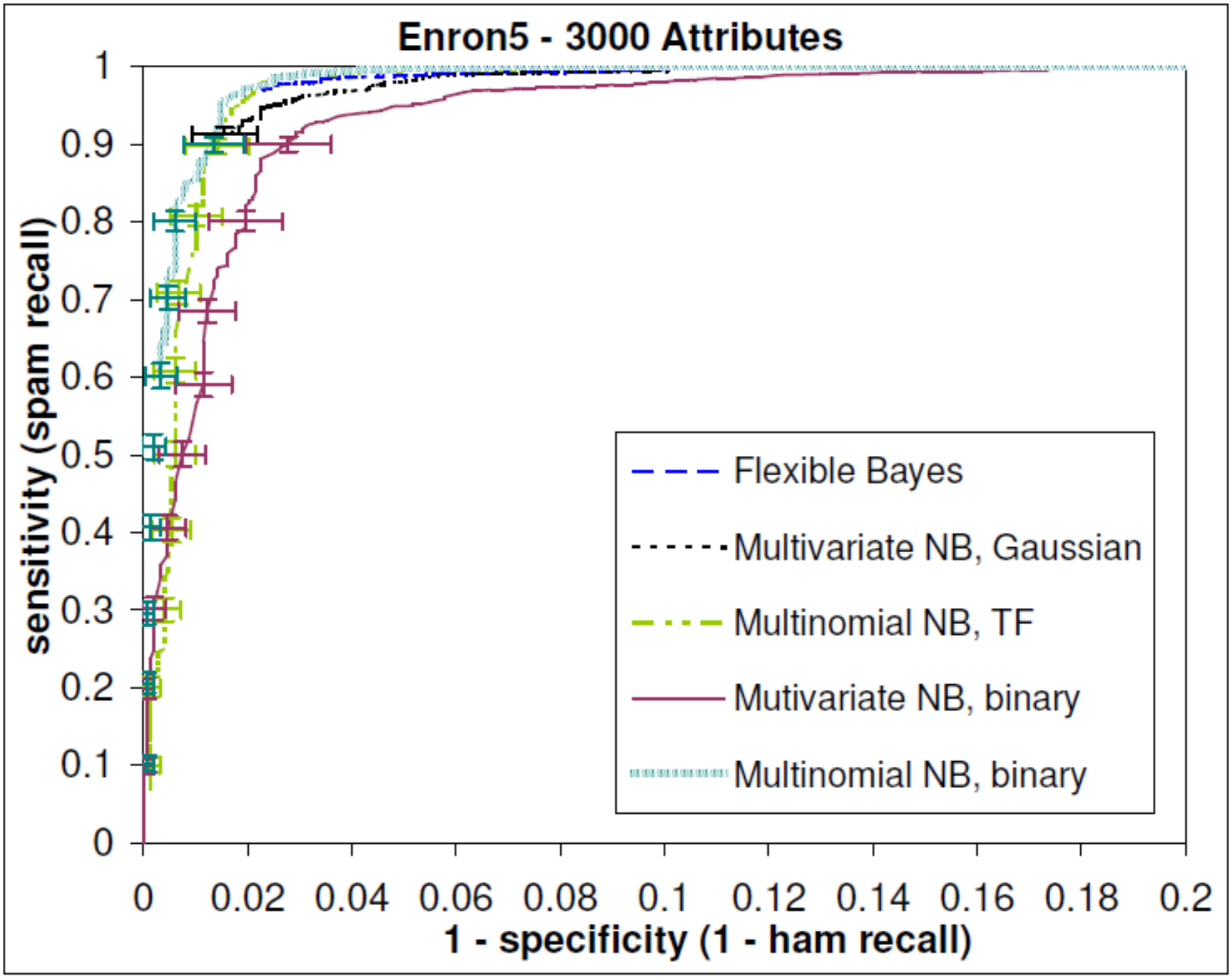
Table 5: Ham recall (%) for 3000 attributes, $T = 0.5$.

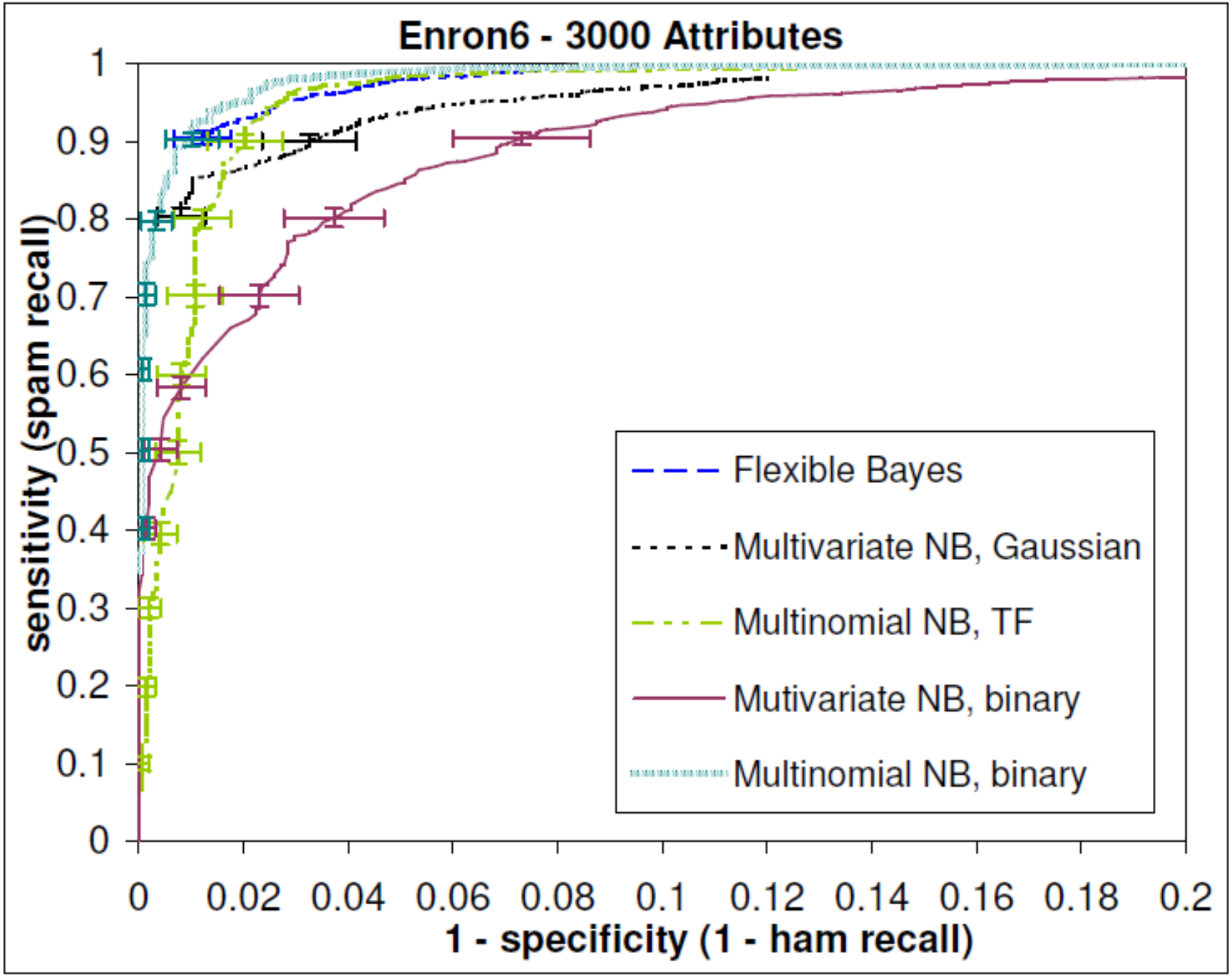

Enron1 - 3000 Attributes
sensitivity (spam recall)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
1 - specificity (1 - ham recall)
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary


Enron2 - 3000 Attributes
sensitivity (spam recall)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
1 - specificity (1 - ham recall)
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary


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

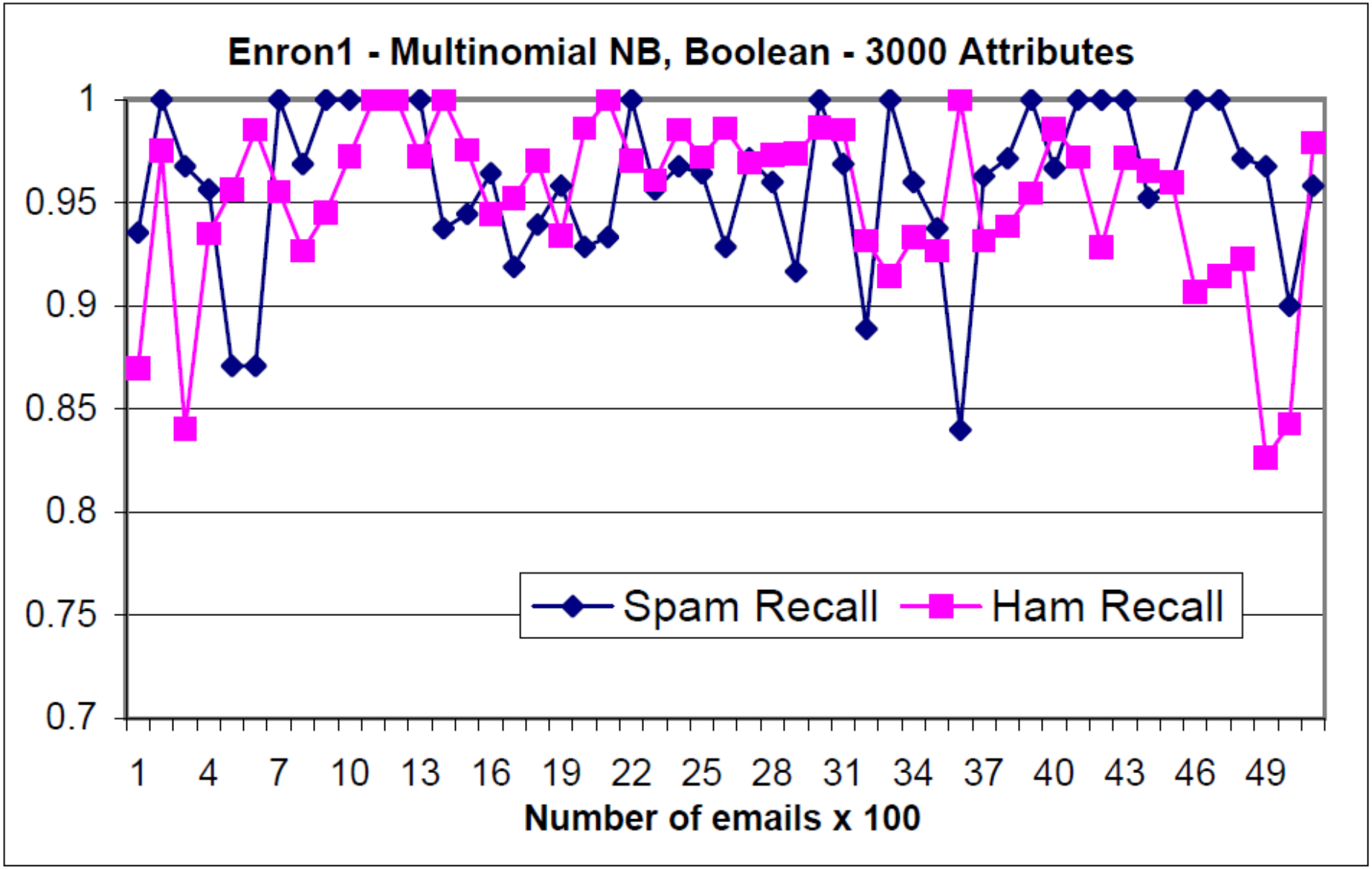

Enron4 - 3000 Attributes
sensitivity (spam recall)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
1 - specificity (1 - ham recall)
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary

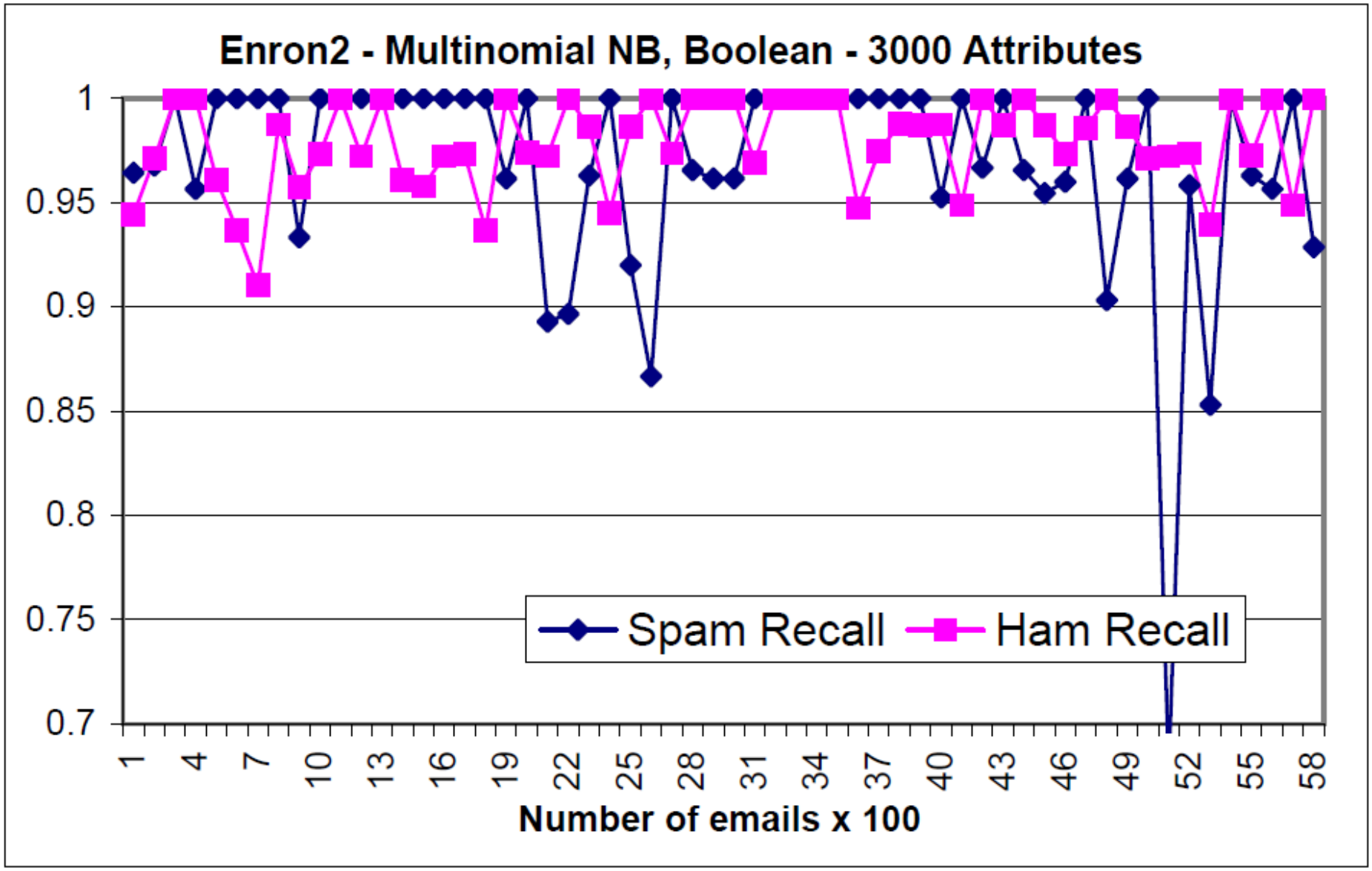

Enron5 - 3000 Attributes
sensitivity (spam recall)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
1 - specificity (1 - ham recall)
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary

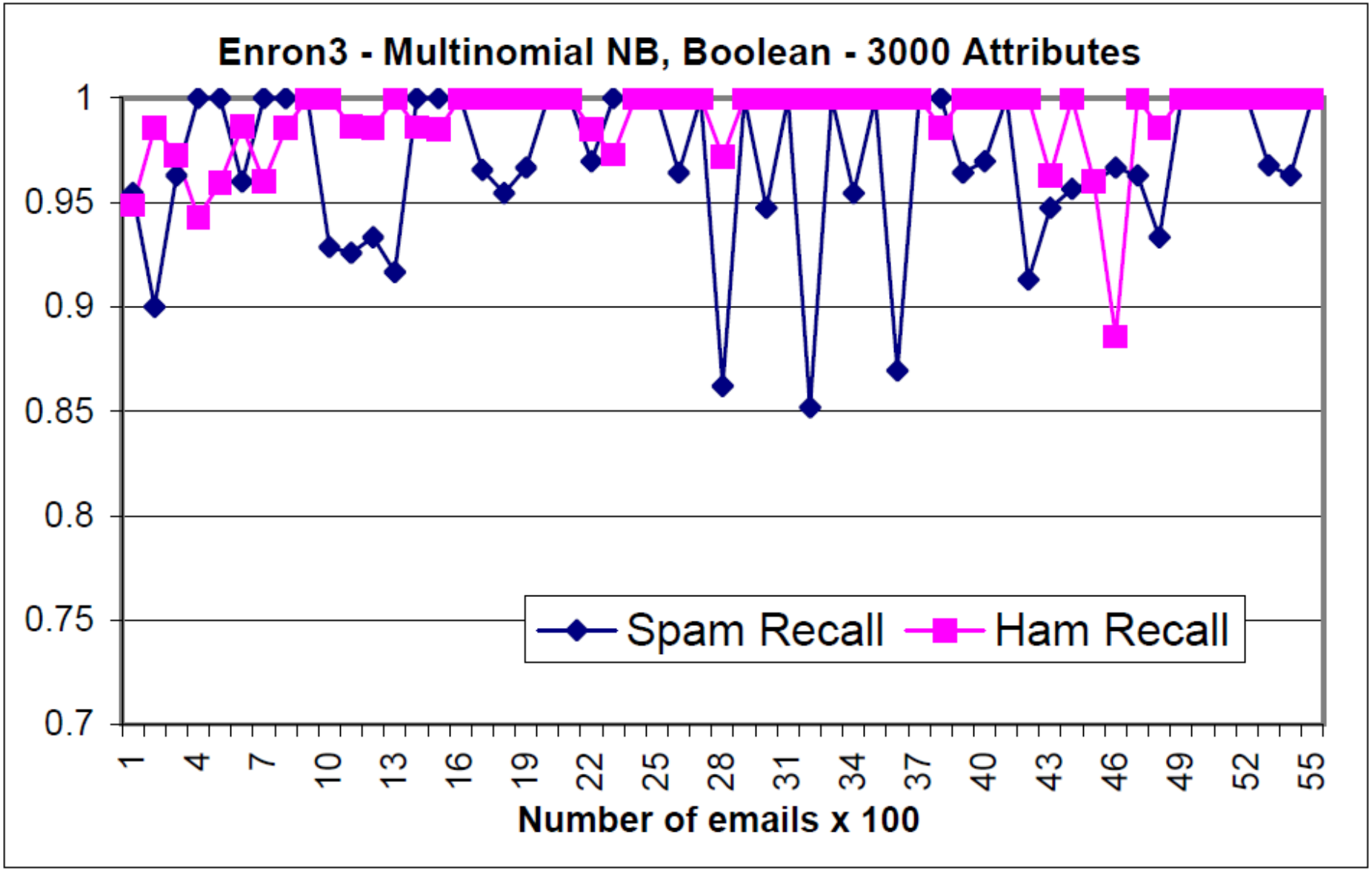

Enron6 - 3000 Attributes
sensitivity (spam recall)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
1 - specificity (1 - ham recall)
Flexible Bayes
Multivariate NB, Gaussian
Multinomial NB, TF
Mutivariate NB, binary
Multinomial NB, binary

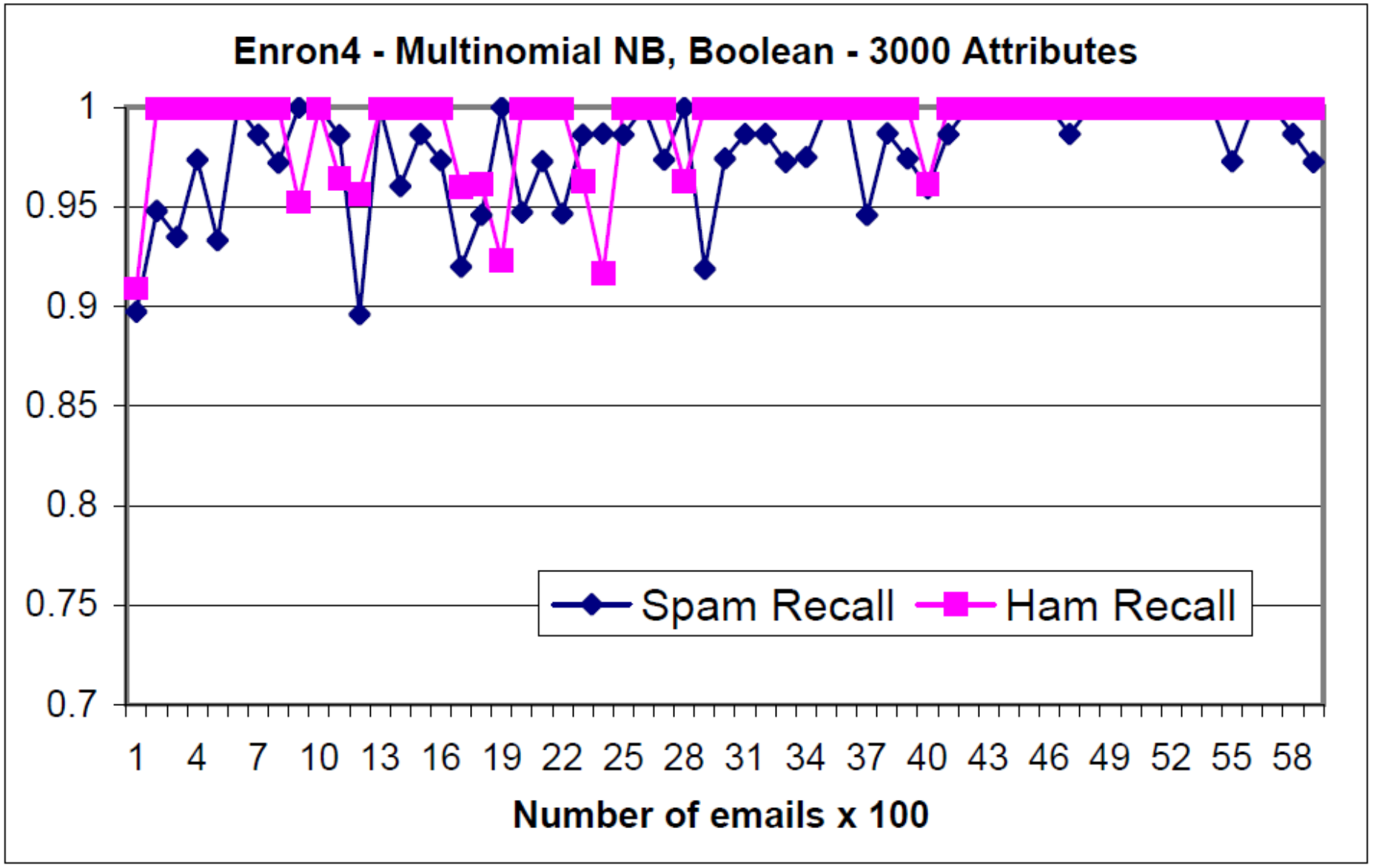
Learning Curves

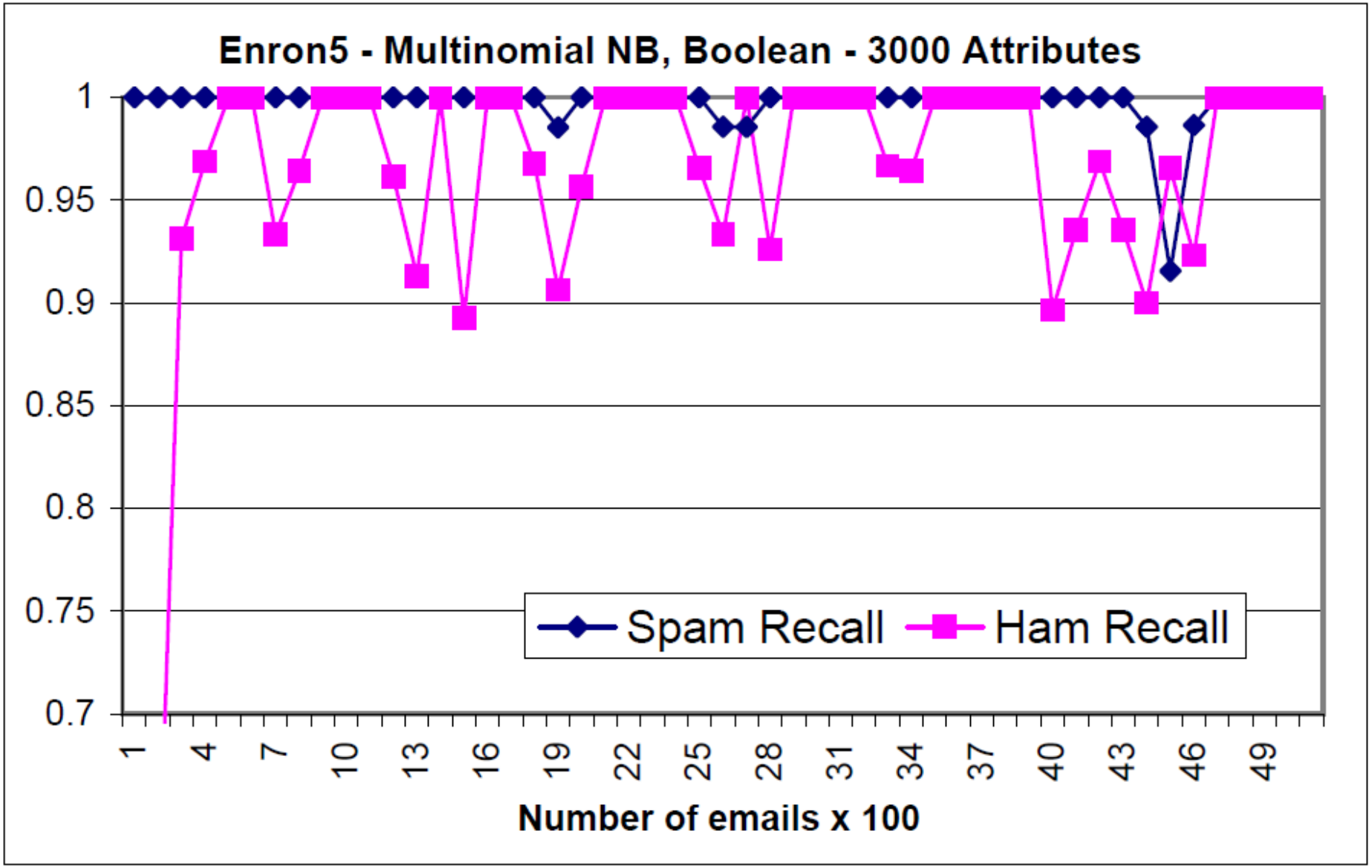
- Provide additional context for the progress made by each classifier over the duration of the experimental runs
- Do not increase monotonically as in other text classification domains
- Most likely due to the unpredictable fluctuation of the ham-spam ratio, changing topics of spam, and adversarial nature of anti-spam filtering

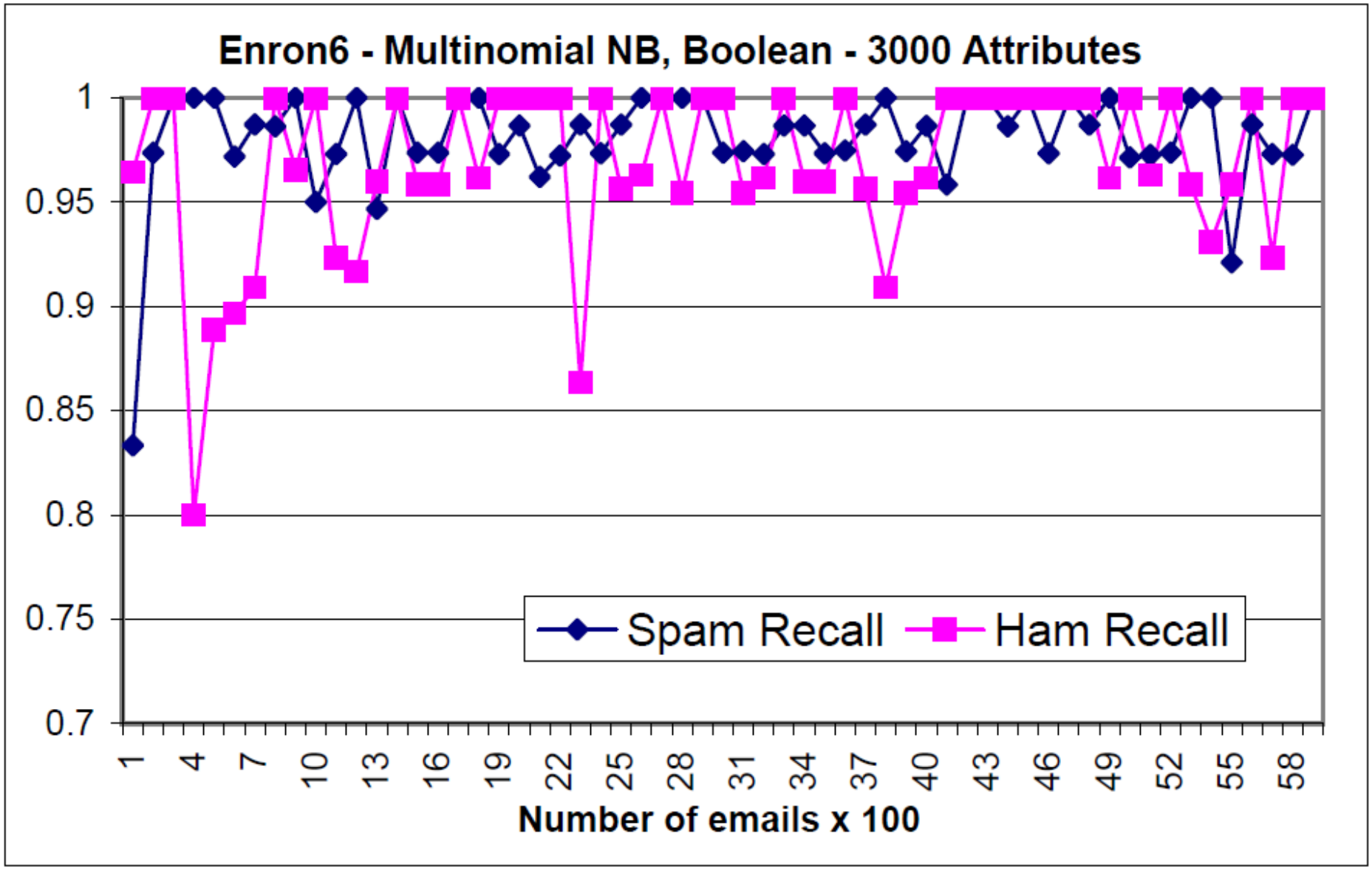

Enron1 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49
Number of emails x 100


Enron2 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
58
Number of emails x 100


Enron3 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
Number of emails x 100


Enron4 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58
Number of emails x 100


Enron5 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
Number of emails x 100


Enron6 - Multinomial NB, Boolean - 3000 Attributes
1
0.95
0.9
0.85
0.8
0.75
0.7
Spam Recall
Ham Recall
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
58
Number of emails x 100

Conclusions

- The two versions of NB used least in in spam filtering, i.e. Flexible Bayes and Multinomial NB w/ Boolean attributes, performed the best in the experiments
- Due to lower computational complexity and smoother trade-off between ham and spam recall, the authors tend to prefer Multinomial NB w/ Boolean attributes
- Best results were achieved with the largest attribute set, but the gain was rather insignificant

Questions?