(a) Ward's minimum variance criterion minimizes the total within-cluster variance. To implement this method, at each step find the pair of clusters that leads to minimum increase in total within-cluster variance after merging.(From Wikipedia)

(b) Ward-Linkage gives correct clusters as both classes are uniformly distributed separately. So the variance of euclidean distance is minimised if datapoints are clustered according to classes.

2.5 D31 dataset

None of the clustering algorithms converge/take a long time to converge.

3 NAIVE BAYES

3.1 MAXIMUM LIKELIHOOD ESTIMATION ASSUMING MULTINOMIAL LIKELIHOOD

3.1.1 ALGORITHM FOLLOWED

TRAINMULTINOMIALNB($\mathbb{C}$, $\mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}$, c)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}$, c)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c$, t)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** V, $prior$, $condprob$

APPLYMULTINOMIALNB($\mathbb{C}$, V, $prior$, $condprob$, d)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] \mathrel{+}= \log condprob[t][c]$
6 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

Figure 13.2 Naive Bayes algorithm (multinomial model): Training and testing.

-Credits – Textbook by Manning
TRAINING

1. Each document has been converted to a frequency count of the words in the Vocabulary and appended as a row with it's corresponding label(Spam or Ham) in the Training/Cross-Validation Matrix.

2. The No of Docs has been counted as the no of rows in the Training Set.

3. Vocabulary has been computed as the maximum integer used to encode a word or after preprocessing, the no. of columns in the training matrix.

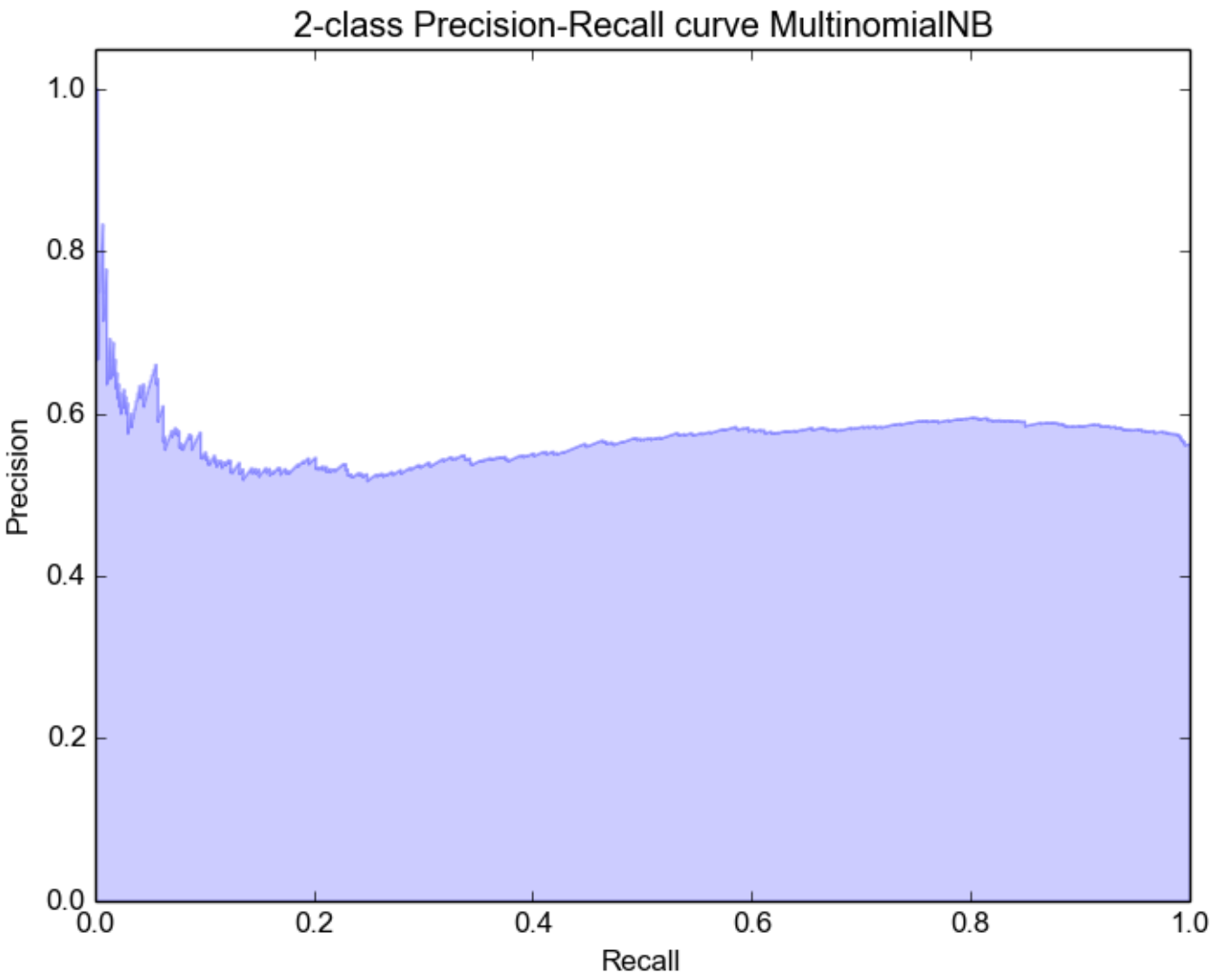
4. Class priors have been found as described in image.
5. Text of all docs is concatenated.
6. Conditional Probability for each word is found.
   TESTING

- Score for each word for each class is found and is classwise compared to predict class.

3.1.2 CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.95	0.97	0.96
Legit	0.98	0.96	0.97
Average	0.97	0.97	0.97

3.1.3 PRECISION-RECALL CURVE:



3.2 BERNOULLI NAIVE BAYES:

3.2.1 ALGORITHM FOLLOWED:

TRAINBERNOULLINB($\mathbb{C}$, $\mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 **for each** $t \in V$
7 **do** $N_{ct} \leftarrow$ COUNTDOCSINCLASSCONTAININGTERM($\mathbb{D}, c, t$)
8 $condprob[t][c] \leftarrow (N_{ct} + 1)/(N_c + 2)$
9 **return** $V, prior, condprob$

APPLYBERNOULLINB($\mathbb{C}, V, prior, condprob, d$)
1 $V_d \leftarrow$ EXTRACTTERMSFROMDOC(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in V$
5 **do if** $t \in V_d$
6 **then** $score[c] \mathrel{+}= \log condprob[t][c]$
7 **else** $score[c] \mathrel{+}= \log(1 - condprob[t][c])$
8 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

Figure 13.3 NB algorithm (Bernoulli model): Training and testing. The add-one smoothing in Line 8 (top) is in analogy to Equation (13.7) with $B = 2$.

Algorithm has been followed exactly as in textbook.

3.2.2 CLASSIFICATION REPORT

3.2.3 CLASSIFICATION SCORES:

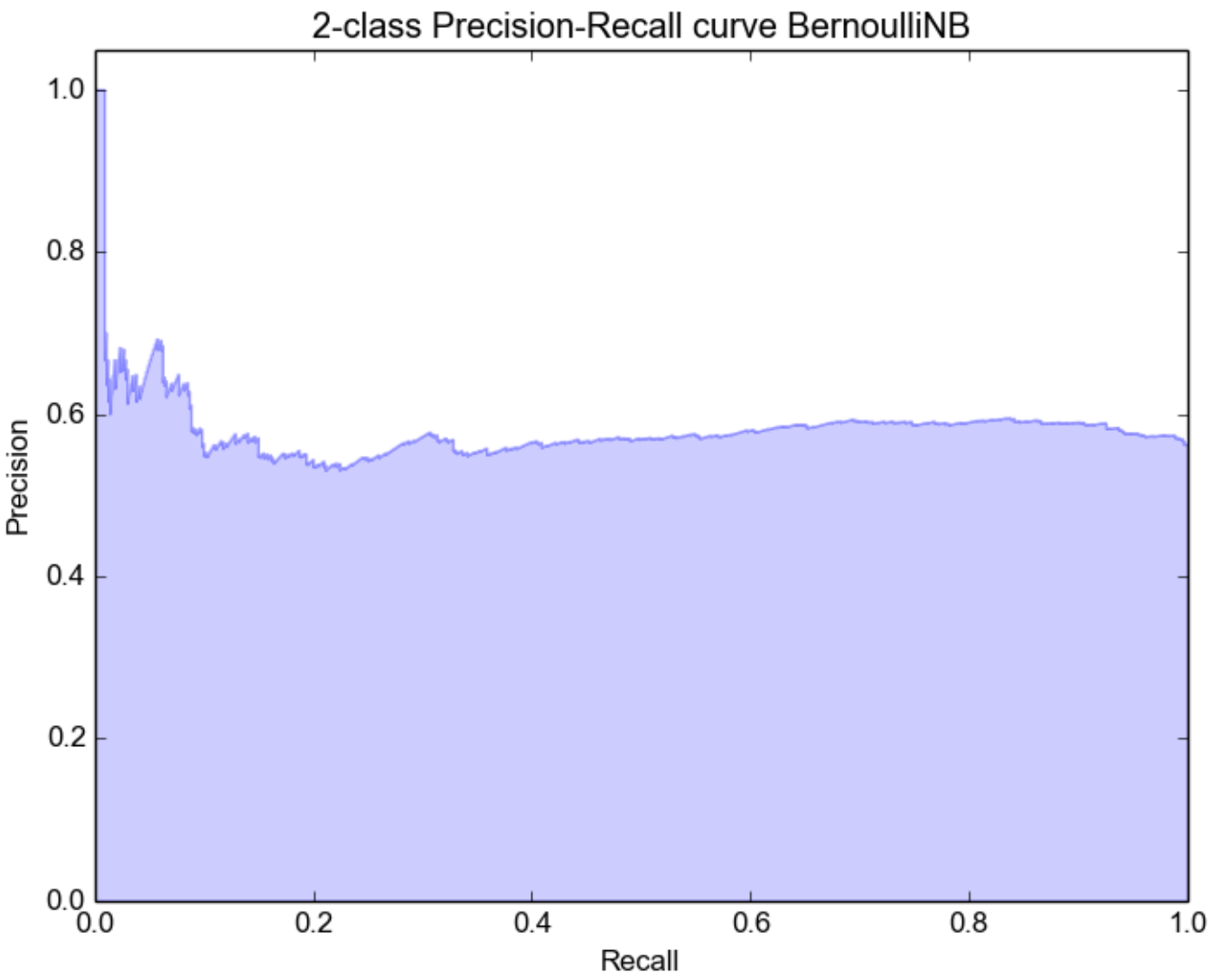
	Precision	Recall	F-score
Spam	0.99	0.83	0.90
Legit	0.88	0.99	0.93
Average	0.93	0.92	0.92

Here, the classification scores are lower because:

1. In Multinomial distribution, we take into account the number of times a word appears in a document.

2. Whereas in Bernoulli, we estimate class based on the fact if a word occurs in a document or not

3.2.4 PRECISION-RECALL CURVE:



3.3 BAYESIAN PARAMETER ESTIMATION WITH DIRICHLET PRIOR AND MULTNOMIAL LIKELIHOOD

1. The algorithm is exactly the same as the Multinomial case, except for the estmation of conditional probabilities:
2. $p(w|\alpha_1, \alpha_2..\alpha_k) = \frac{T_{ct}+\alpha_t+1}{\sum_{t'} T_{ct'}+\alpha_{t'}+1}$
3. The Dirichlet priors are separate for each class. Each class has a separate set of parameters $(\alpha_1, \alpha_1...\alpha_k)$.

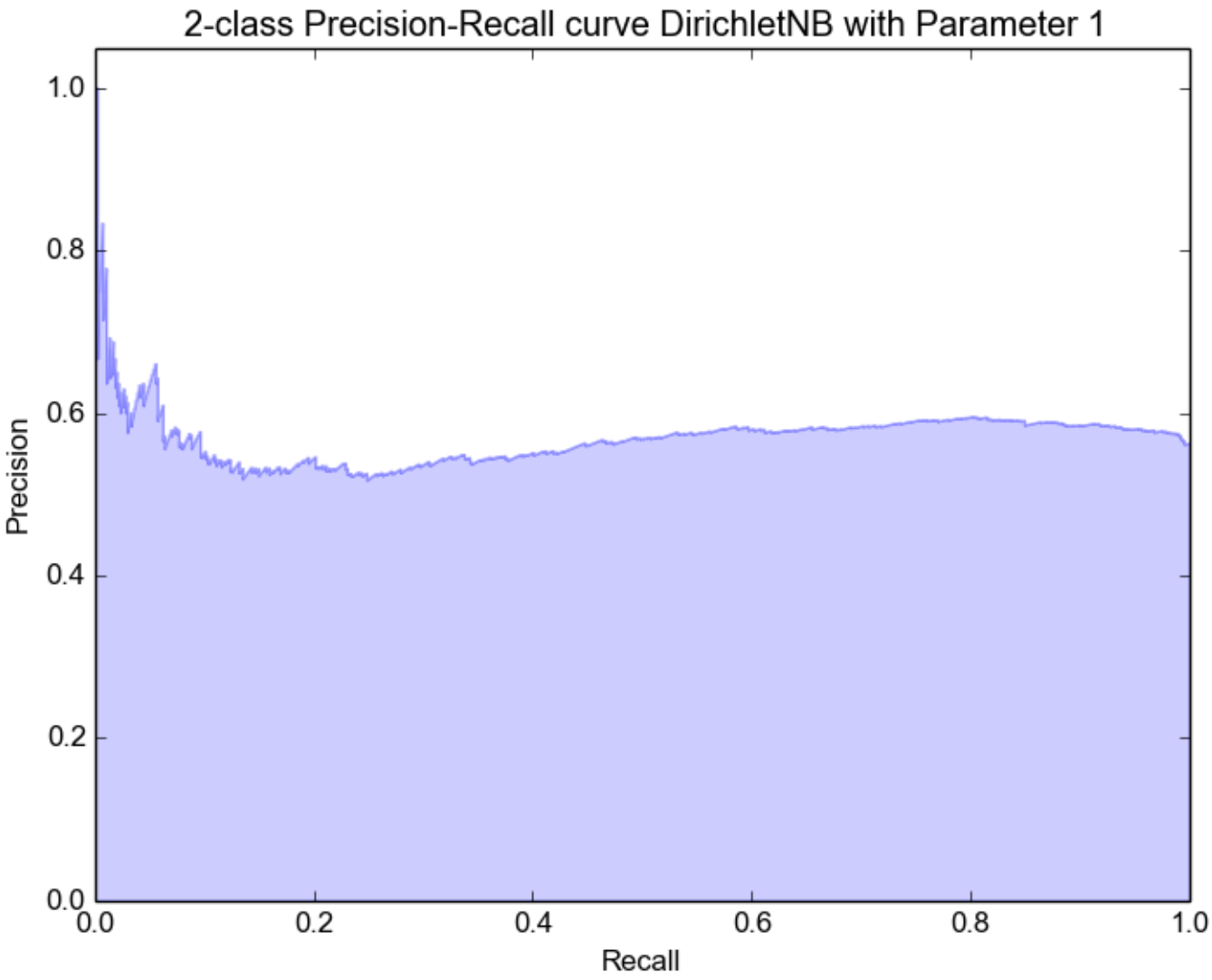
3.3.1 DIRICHLET WHICH ASSUMES ALL WORDS ARE MORE SPAMMY THAN HAMMY:

1. $(\alpha_1, \alpha_1...\alpha_k)$ = (10,10,...10) for Spam Class
2. $(\alpha_1, \alpha_1...\alpha_k)$ = (0,0,...0) for Ham Class

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.95	0.97	0.96
Legit	0.98	0.96	0.97
Average	0.97	0.97	0.97

PRECISION-RECALL CURVE:



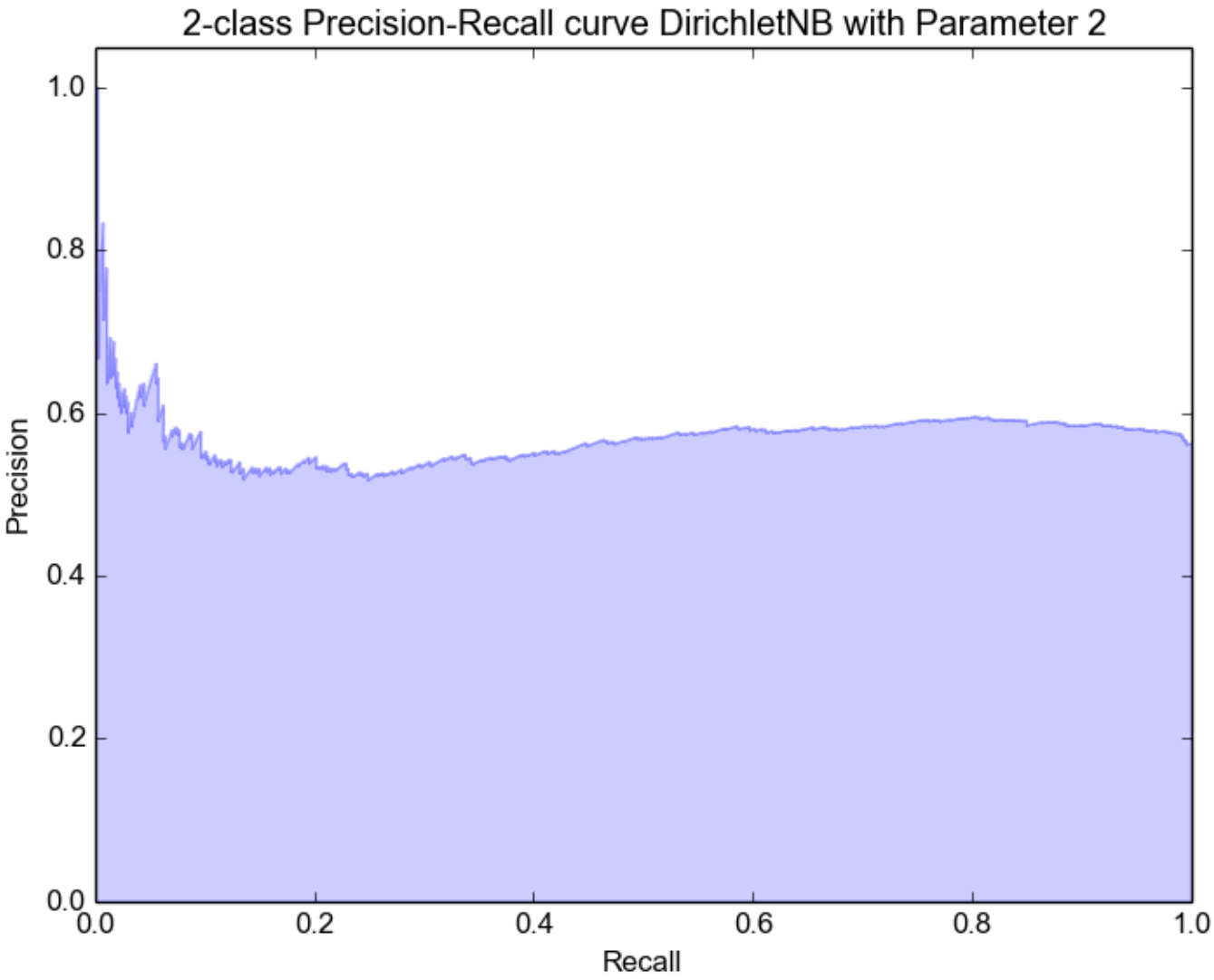
3.3.2 DIRICHLET WITH ESTIMATION OF SPAMMY/HAMMY-NESS:

1. Spamminess of word(the higher likelihood of the class being spam if the document contains the word):
   $\alpha_k = \frac{No.\ of\ spam\ docs\ the\ word\ has\ occured\ in}{Total\ no\ of\ docs\ containing\ word}$
2. Ham Class:
   $\alpha_k = 1 - \frac{No.\ of\ spam\ docs\ the\ word\ has\ occured\ in}{Total\ no\ of\ docs\ containing\ word}$

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.95	0.97	0.96
Legit	0.98	0.96	0.97
Average	0.97	0.97	0.97

PRECISION-RECALL CURVE:



This is supposed to give better score, but the classification scores with multinomial is already high, so not affected much.

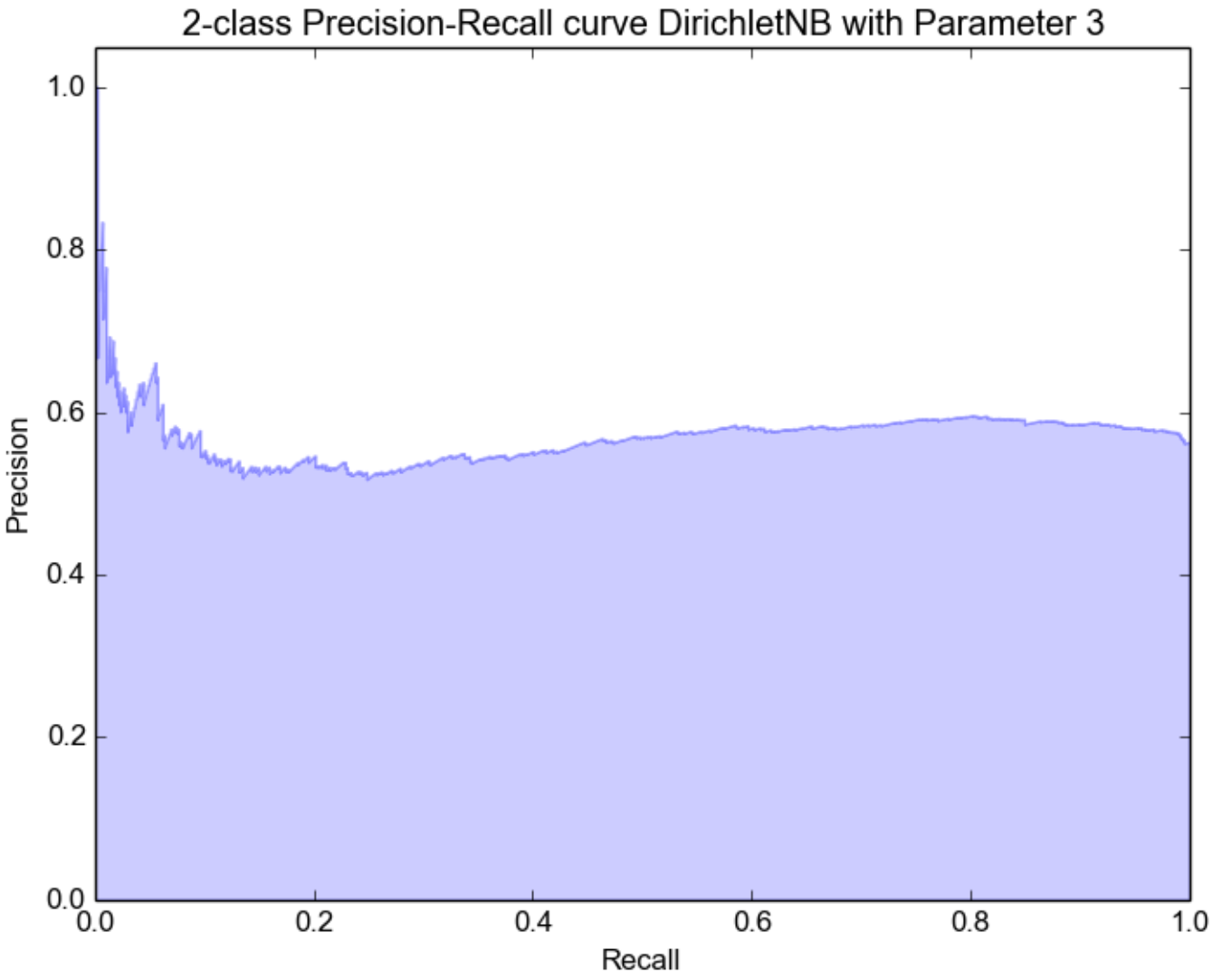
3.3.3 DIRICHLET WHICH ASSUMES ALL WORDS ARE MORE HAMMY THAN SPAMMY:

1. $(\alpha_1, \alpha_1 ... \alpha_k)$ = (10,10,...10) for Ham Class
2. $(\alpha_1, \alpha_1 ... \alpha_k)$ = (0,0,...0) for Spam Class

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.95	0.97	0.96
Legit	0.98	0.96	0.97
Average	0.97	0.97	0.97

PRECISION-RECALL CURVE:



3.4 BAYESIAN PARAMETER ESTIMATION WITH BETA PRIOR AND BERNOULLI LIKELIHOOD

1. Here, we have assumed that Bernoulli Likelihood probability 'p' for each word, for each class has a beta prior, B(α, β)
2. The algorithm is exactly the same as the Bernoulli case, except for the estmation of conditional probabilities:
3. $p(w|\alpha_1, \alpha_2..\alpha_k) = \frac{N_{ct}+\alpha_t+1}{N_c+\alpha_{t'}+\beta_t+2}$
4. Thus the parameters here are $(\alpha_1, \alpha_1...\alpha_k)$ and $(\beta_1, \beta_2...\beta_k)$.

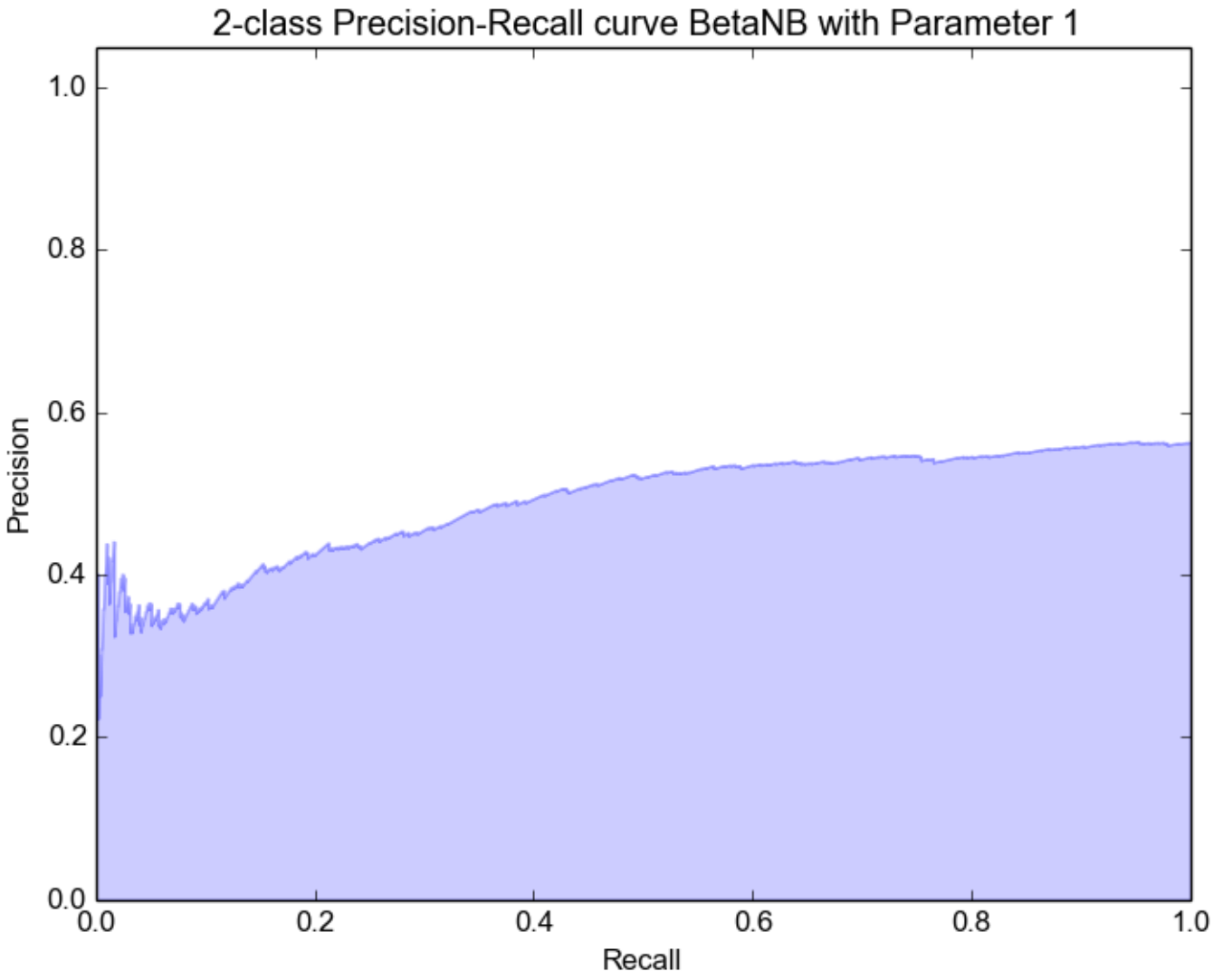
3.4.1 BETA WHICH ASSUMES ALL WORDS ARE MORE SPAMMY THAN HAMMY:

1. $(\alpha_1, \alpha_1...\alpha_k)$ = (10,10,...10)
2. $(\beta_1, \beta_2...\beta_k)$= (0,0,...0)

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.31	0.03	0.03
Legit	0.56	0.95	0.70
Average	0.45	0.55	0.42

PRECISION-RECALL CURVE:



Since we get such bad scores, this is surely a wrong assumption.

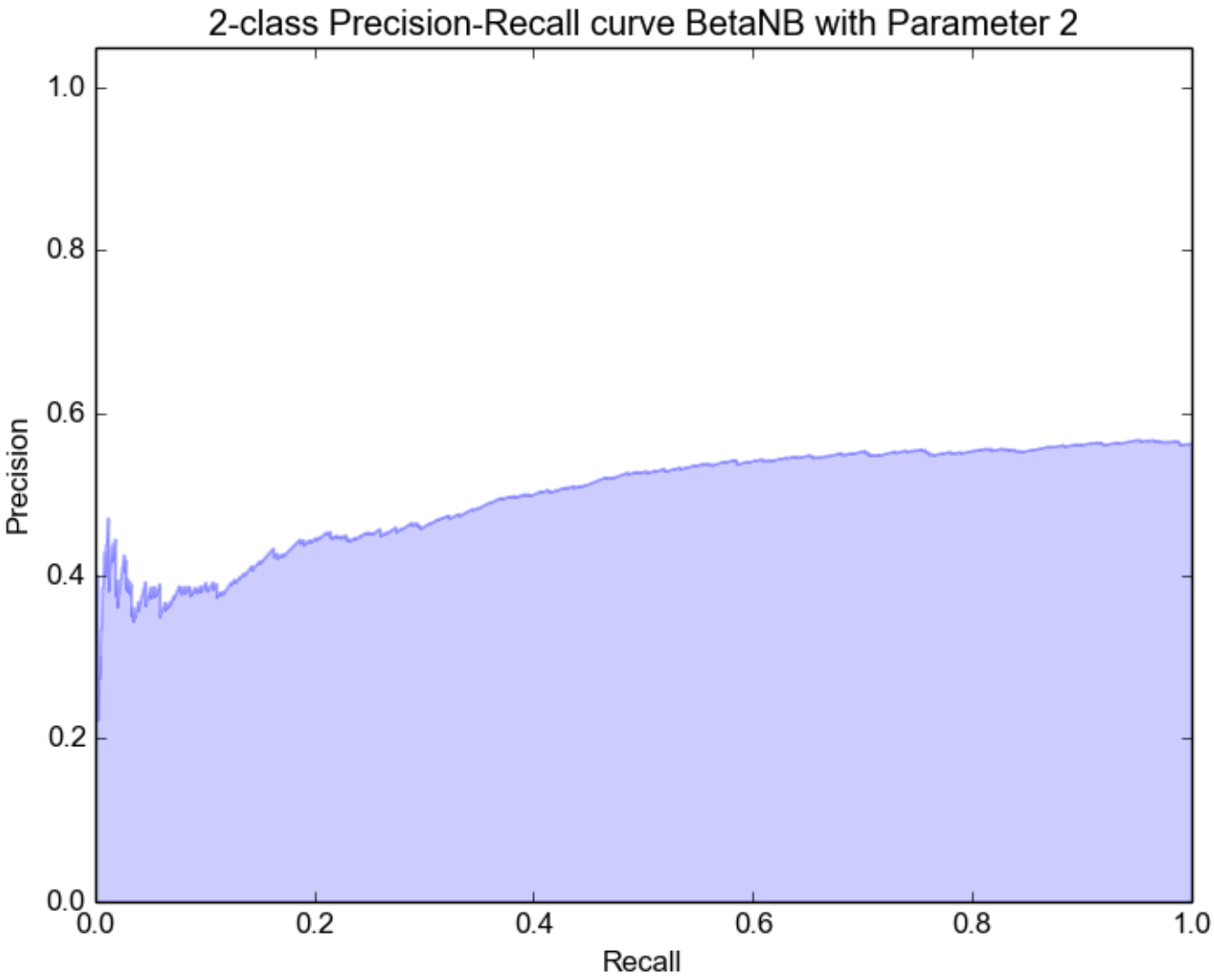
3.4.2 BETA WHICH ASSUMES ALL WORDS EQUALLY SPAMMY OR HAMMY:

1. $(\alpha_1, \alpha_1 ... \alpha_k)$ = (5,5,...5)
2. $(\beta_1, \beta_2 ... \beta_k)$= (5,5,...5)

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.64	0.09	0.16
Legit	0.58	0.96	0.72
Average	0.61	0.58	0.48

PRECISION-RECALL CURVE:



This seems like a better assumption than the previous case, but still not a good assumption.

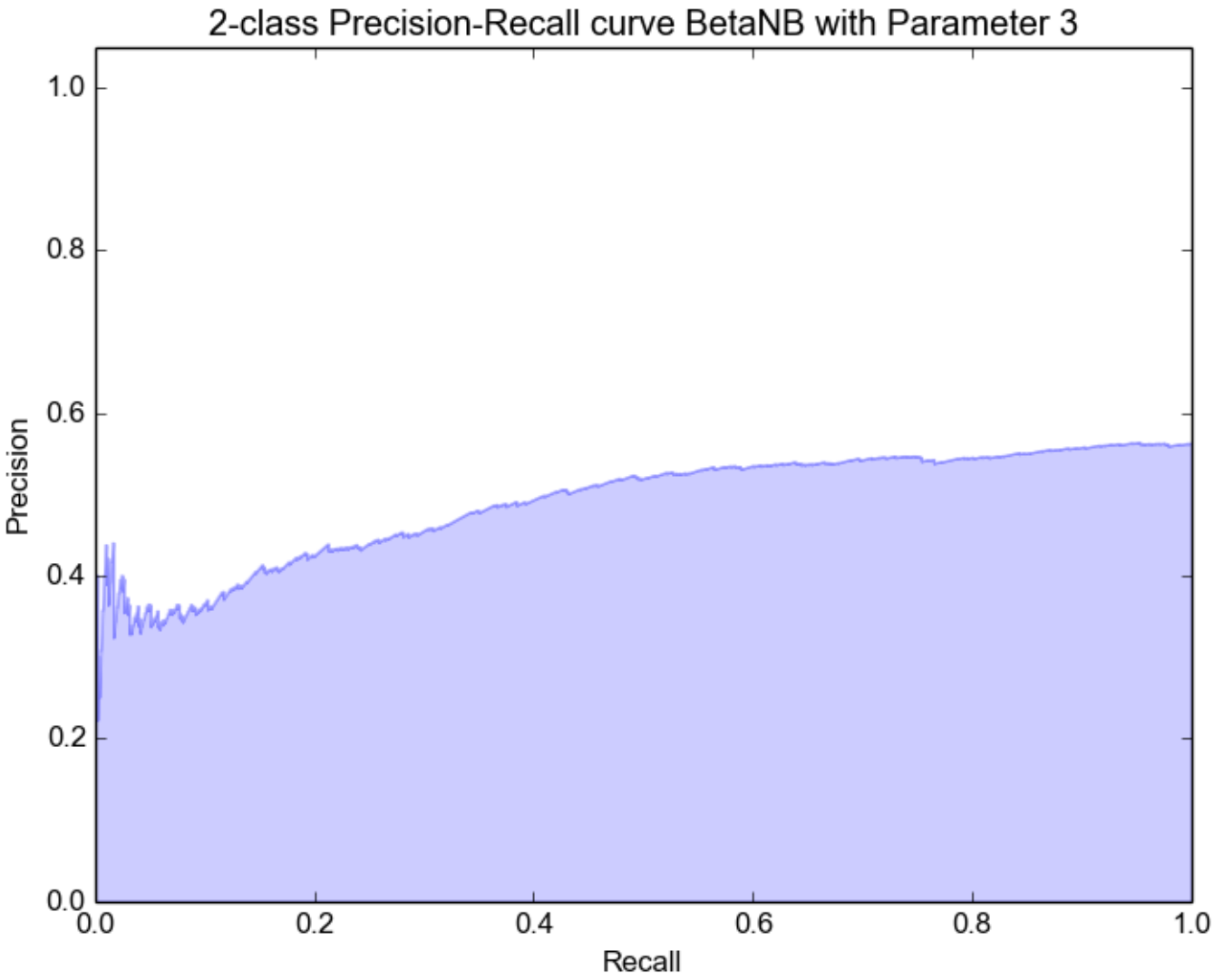
3.4.3 BETA WHICH ASSUMES ALL WORDS ARE MORE SPAMMY THAN HAMMY:

1. $(\alpha_1, \alpha_1 ... \alpha_k)$ = (0,0,...0)
2. $(\beta_1, \beta_2 ... \beta_k)$= (10,10,...10)

CLASSIFICATION SCORES:

	Precision	Recall	F-score
Spam	0.31	0.03	0.05
Legit	0.56	0.95	0.70
Average	0.45	0.55	0.42

PRECISION-RECALL CURVE:



Since we get such bad scores, this is surely a wrong assumption.

1. Basically by enforcing priors, we are skewing the probabilities in a certain direction.
2. Unless the skewed direction is correct, we are actually messing up the probabilities and getting worse results.
3. So, we need to come up with good heuristic to estimate the parameters, thereby the spamminess/hamminess of each word.
4. Expectation-Maximisation for estimating these parameters is probably a technique that could be used.