CS534 — Implementation Assignment 2

Naive Bayes (total 70pts)

In this assignment you will implement the Naive Bayes classifier for document classification with both the Bernoulli model and the Multinomial model. For Bernoulli model, a document is described by a set of binary variables, and each variable corresponds to a word in the vocabulary V and represents its presence/absence. The probability of observing a document $\mathbf{x}$ given its class label y is then defined as:

$$p(\mathbf{x}|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}(1 - p_{i|y})^{(1-x_i)}$$

where $p_{i|y}$ denotes the probability that the word i will be present for a document of class y. If $x_i = 1$, the contribution of this word to the product will be $p_{i|y}$, otherwise it will be $1 - p_{i|y}$.

For the Multinomial model, a document is represented by a set of integer-valued variables, and each variable x_i also corresponds to the i-th word in the vocabulary and represents the number of times it appears in the document. The probability of observing a document $\mathbf{x}$ given its class label y is defined as:

$$p(\mathbf{x}|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}$$

Here we assume that each word in the document follows a multinomial distribution of $|V|$ outcomes and $p_{i|y}$ is the probability that a randomly selected word is word i for a document of class y. Note that $\sum_{i=1}^{|V|} p_{i|y} = 1$ for $y = 0$ and $y = 1$.

Your implementation need to estimate $p(y)$, and $p_{i|y}$ for $i = 1, \cdots, |V|$, and $y = 1, 0$ for both models. For $p(y)$, you can use MLE estimation. For $p_{i|y}$, you MUST use Laplace smoothing for both types of models.

Apply your Naive Bayes classifiers to the provided 20newsgroup data. By that I mean learn your models using the training data and apply the learned model to perform prediction for the test data and report the performance.

One useful thing to note is that when calculating the probability of observing a document given its class label, i.e., $p(\mathbf{x}|y)$, it can and will become overly small because it is the product of many probabilities. As a result, you will run into underflow issues. To avoid this problem, you should operate with log of the probabilities.

Basic implementation:

1. (5 pts) Please explain how you use the log of probability to perform classification.

   In order to predict $p(y|x)$, we must use Bayes' Rule:

   $$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

   To maximize the probability of a document x belonging to class y,

   $$\arg\max_{y \in Y} p(y|x) = \arg\max_{y \in Y} \frac{p(x|y)p(y)}{p(x)}$$

   which is equivalent to:

   $$\arg\max_{y \in Y} p(y|x) = \arg\max_{y \in Y} p(x|y)p(y)$$

   where $p(y)$ is learned from training:

   $$p(y) = \frac{\text{\# of documents in class y}}{\text{\# of documents in all classes}}$$

For the Bernoulli Model, we can assign a class as follows:

$$\arg\max_{y \in Y} p(y|x) = \arg\max_{y \in Y} \left\{ p(y) \prod_{i=1}^{|V|} p(w_i|y)^{x_i} (1 - p(w_i|y))^{1-x_i} \right\}$$

Where $x_i = 1$ if word i appears in document x, and $x_i = 0$ if word i is absent from document x. Taking the log of both sides, we get:

$$\arg\max_{y \in Y} \log p(y|x) = \arg\max_{y \in Y} \left\{ \log p(y) + \sum_{i=1}^{|V|} x_i \log p(w_i|y) + \sum_{i=1}^{|V|} (1 - x_i) \log(1 - p(w_i|y)) \right\}$$

For the Bernoulli Model with Laplace smoothing, $p(w_i|y)$ was learned as:

$$\frac{\text{(total \# of documents in class y that contain word i)} + 1}{\text{(total \# of documents in class y)} + 2}$$

For the Multinomial Model, we assign a class as follows:

$$\arg\max_{y \in Y} p(y|x) = \arg\max_{y \in Y} \left\{ p(y) \prod_{i=1}^{|V|} p(w_i|y)^{x_i} \right\}$$

Where x_i is the number of times word i appears in document x. Taking the log of both sides, we get:

$$\arg\max_{y \in Y} \log p(y|x) = \arg\max_{y \in Y} \left\{ \log p(y) + \sum_{i=1}^{|V|} x_i \log p(w_i|y) \right\}$$

For the Multinomial Model with Laplace smoothing, $p(w_i|y)$ was learned as:

$$\frac{\text{(total count of word i in all class y documents)} + 1}{\text{(total word count of all class y documents)} + |V|}$$

A document receives the label y that yields the highest probability $p(y|x)$ for that document x.

2. (10 pts) Report the overall testing accuracy (number of correctly classified documents over the total number of documents) for both models.

Table 1: Comparing Two Models

Bernoulli Accuracy	Multinomial Accuracy
0.6240	0.7952

The Bernoulli Model yielded better accuracy when the Dirichlet prior for the Multinomial Model was set to 1 (Table 1). One important difference between these two models is that for the Multinomial Model, the count probabilities of a given word decays exponentially with the number of times it appears in a document. However, this does not mean that the Bernoulli Model is better for this particular data set. We can improve the Multinomial test accuracy by trying different Dirichlet prior values and applying heuristics to reduce vocabulary, as we will see below.

3. (5 pts) Are there any news groups that are confused more often than others? Why do you think this is? To answer this question, you might want to produce a K by K confusion matrix, where K is the number of classes, and the i, j-th entry of the matrix shows the number of class i documents being predicted to belong to class j. A perfect prediction will have only diagonal elements in this confusion matrix.

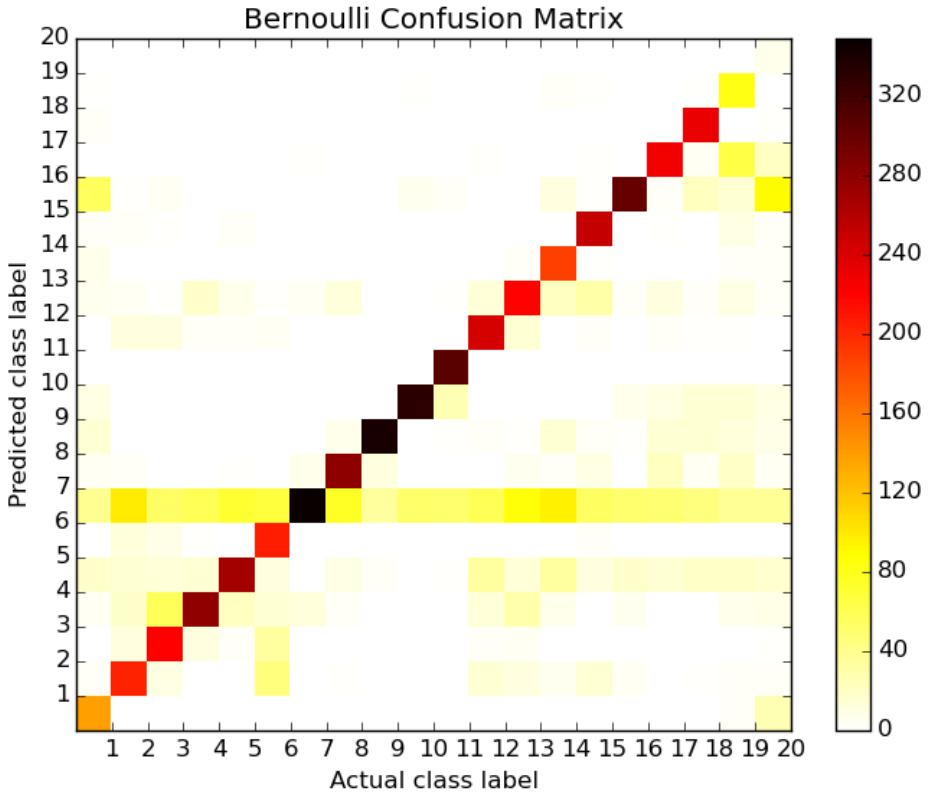

Figure 1

20	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**7**
19	2	0	1	0	0	0	0	0	1	2	0	1	0	4	2	1	0	2	**83**	1
18	4	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	**233**	0	2
17	1	0	0	0	0	0	2	1	1	1	0	2	1	1	2	1	**228**	5	65	21
16	57	2	5	0	0	0	0	0	0	6	4	1	0	11		**301**	4	23	15	91
15	3	4	2	0	3	0	1	0	0	0	0	1	0	1	**251**	0	2	0	10	3
14	7	1	0	0	0	1	0	0	0	0	0	0	3	**189**	2	0	1	1	4	4
13	6	5	2	18	7	2	5	13	0	1	2	14	**220**	22	30	4	12	3	10	3
12	1	12	11	4	3	5	1	0	1	0	1	**244**	15	0	3	0	3	2	2	1
11	0	0	0	0	0	0	0	0	0	0	**308**	1	0	0	0	0	0	0	0	0
10	10	0	1	0	0	0	0	0	1	**331**	27	1	0	1	0	7	10	16	16	10
9	15	0	1	0	0	0	0	7	**341**	0	2	3	2	15	3	2	15	15	13	9
8	5	3	1	1	2	1	8	**280**	11	0	0	1	6	4	10	0	22	5	20	5
7	39	99	54	59	72	68	**349**	76	35	53	51	59	87	96	55	51	50	46	38	37
6	0	13	9	2	0	**206**	1	0	0	0	0	1	0	0	3	0	0	0	0	0
5	18	15	14	16	**269**	12	1	10	4	1	2	34	14	34	11	19	15	20	20	17
4	5	19	58	**279**	22	16	13	4	0	0	1	14	29	7	1	6	1	1	7	9
3	1	11	**221**		4	34	1	1	0	0	0	3	5	1	1	0	1	1	1	2
2	4	**204**	10	1	1	45	1	2	1	1	1	15	11	6	16	5	0	2	3	3
1	**139**	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	3	26
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Bernoulli Confusion Matrix

Figure 2

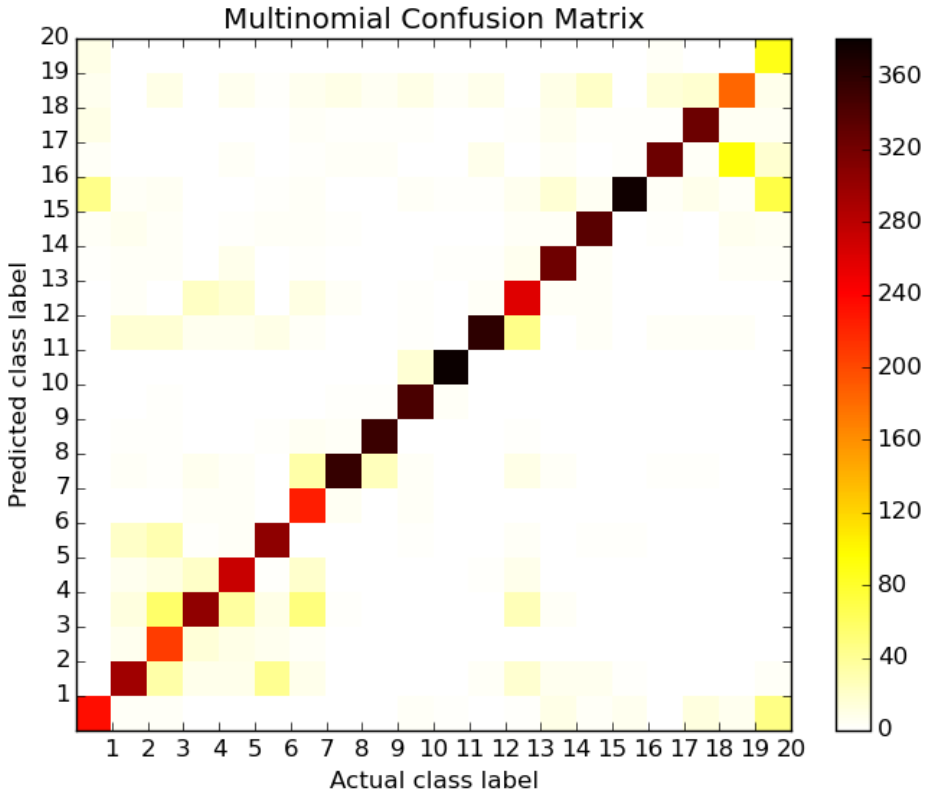

Figure 3

20	9	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	4	0	1	**89**
19	7	1	9	0	6	2	7	9	5	10	3	8	0	10	23	1	16	20	**184**	8
18	10	0	0	0	0	1	4	2	2	2	1	0	2	9	2	2	4	**326**	5	5
17	3	0	0	1	3	1	3	5	5	1	1	8	0	3	1	2	**324**	3	95	19
16	45	5	6	0	2	2	3	1	1	5	2	2	6	18	5	**377**	3	7	3	70
15	3	6	4	1	3	3	3	2	0	0	1	0	4	3	**334**	0	2	0	7	5
14	2	4	4	0	10	0	2	0	1	0	2	2	6	**322**	4	1	1	0	3	3
13	1	4	1	23	15	0	12	3	0	2	0	2	**257**	3	4	0	0	0	0	0
12	1	19	17	7	7	10	3	1	1	2	1	**363**	46	1	4	1	3	4	3	1
11	1	0	1	1	0	0	1	0	0	17	**382**	0	1	0	1	0	1	1	0	0
10	0	0	2	0	1	1	0	2	2	**341**	3	0	0	1	0	0	0	1	0	0
9	0	2	2	0	1	2	6	3	**353**	1	0	0	2	0	1	0	1	1	0	1
8	0	3	1	6	5	0	32	**358**	26	3	0	1	11	2	0	0	2	1	1	0
7	0	1	0	5	2	1	**221**	4	0	2	1	1	1	0	0	0	1	0	0	0
6	1	22	34	4	3	**306**	1	1	0	4	0	2	3	0	2	3	0	0	1	0
5	0	6	10	21	**271**	2	20	0	0	1	0	1	6	0	0	1	1	0	0	0
4	0	12	59	**302**	37	10	51	2	0	1	0	1	27	3	0	0	0	0	0	0
3	0	5	**205**	15	9	7	4	0	0	0	0	1	0	1	0	0	0	0	0	0
2	0	**269**	32	8	8	42	9	1	1	0	0	2	19	7	7	2	0	1	1	3
1	**235**	3	3	0	0	0	0	1	0	4	2	0	2	10	3	7	1	11	6	47
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Multinomial Confusion Matrix

Figure 4

From the confusion matrix it is easy to see that class 7 was highly confused with the Bernoulli Model. This corresponds to windows.x, which presumably is about the graphics package called x-window. However, in the Multinomial model, there was no such confusion. This suggests that certain words in the windows.x file are fairly common words that show up at least one time in all the other classes; however, they are used with a much higher frequency in the windows.x class, which allows the Multinomial Model to properly discriminate. You also can observe some confusion in both models between word 1 and word 20, which corresponds to atheism and religion.christian. It makes sense that these classes would share significant portions of their vocabulary.

Priors and overfitting:
(20 pts) In this part, we will focus on the multinomial model and experiment with different priors. In particular, your last set of experiments use' Laplace smoothing for MAP estimation, which corresponding to using $Dirichlet(1+\alpha, ..., 1+\alpha)$ with $\alpha = 1$ as the prior. In this part, you will retrain your classifier with different values of α between 10^{-5} and 1 and report the accuracy on the test set for different α values. Create a plot with value of α on the x-axis and test set accuracy on the y-axis. Use a logarithmic scale for the x-axis. Comment on how the test set accuracy change as α changes and provide a short explanation for your observation.

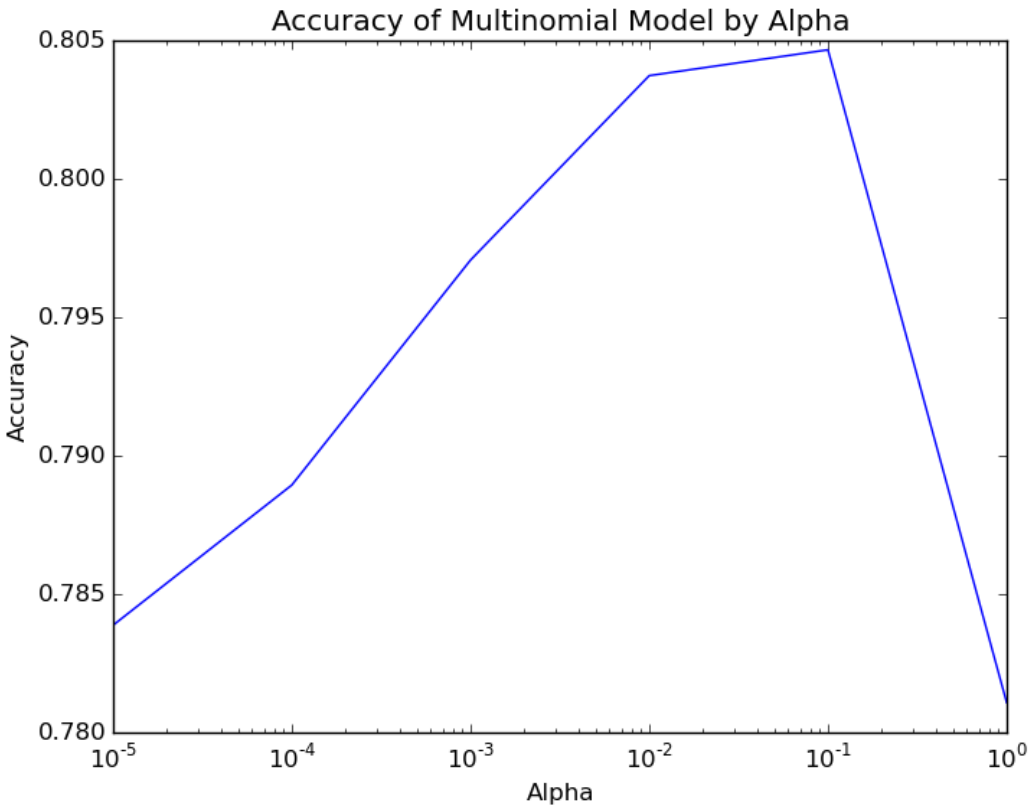

Figure 3

We found that the best α value was 0.1. With this prior, the Multinomial test accuracy exceeds that of the Bernoulli Model (Table 1). It makes sense that the test accuracy drops off at larger values of α because these put more weight on the smoothing terms and hurt the prediction accuracy by reducing the model's discriminative power. Conversely, very small α values reduce the weight on the smoothing terms, which help the model cope with previously unobserved words. As a result, these small α values lead to over-fitting.

Identifying important features:

(20 pts) For this part, design and test a heuristic to reduce the vocabulary size and improve the classification performance. This is intended to be open-ended exploration. Please describe clearly what is your strategy for reducing the vocabulary size and the results of your exploration. A basic pointer to seed your exploration is that we would like to remove words of no discriminative power. How can we measure the discriminative power of a word?

Common words that occur with little or no class preference have the least discriminative power. Thus, a word with poor discriminative power theoretically has a high median multinomial $p(w_i|y)$ value. The median is a better metric than the mean in this case because we do not want to penalize words that are very common in only one specific class. We excluded from our vocabulary different numbers of words with the largest median multinomial $p(w_i|y)$ values and calculated the test accuracy each time for the Multinomial Model (with $\alpha = 0.1$) as well as the Bernoulli Model (Table 2).

The experiment was run for the 100-1000 most common words as defined above. We found that removing frequent words provided a slight boost in accuracy, but were surprised that there wasn't a more dramatic change. The majority of this increase in accuracy occurred after removing only 100 words. The improvement for the Bernoulli Model was slightly greater than that for the Multinomial Model, perhaps because these common words do appear in all the different classes and thus add extra uncertainty into the Bernoulli classification, whereas with the Multinomial Model they may by chance occur with slightly different frequencies in the different document classes. One possible explanation for the fact that the improvement wasn't more dramatic is that the number of common words in these classes actually comprises a very small percentage of the vocabulary. Thus, excluding these words wouldn't be expected to drastically improve either model's accuracy.

Table 2: Reducing Vocabulary Size

Excluded Words	Bernoulli Accuracy	Multinomial Accuracy
0	0.624	0.7811
100	0.6394	0.8001
200	0.648	0.8086
300	0.6511	0.8094
400	0.6524	0.8094
500	0.656	0.808
600	0.6557	0.8093
700	0.6567	0.8092
800	0.6555	0.8097
900	0.6519	0.8075
1000	0.6536	0.8068

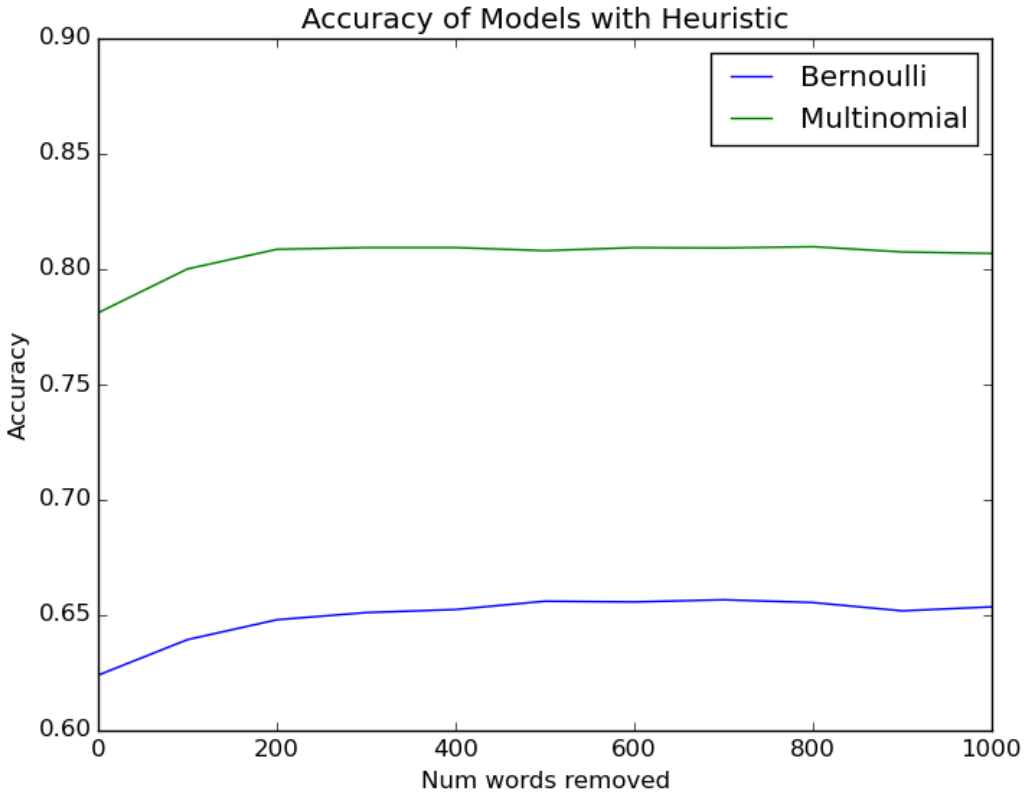
Accuracy of Models with Heuristic
Bernoulli
Multinomial
Accuracy
Num words removed
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
200
400
600
800
1000

Figure 4