Matthew Letter

Project 2

CS529 Machine Learning

March 19, 2015

Introduction:

A high-level description on how your code works. :

The overall goal of the code was to use Naïve Bayes for classifying documents. The first part of which was training a Bayesian system. The data.txt and label.txt file were first parsed and turned into document objects and stored in a list. This list was used to train the Bayesian classifier. The document list was passed on and the probability of each word for each topic was calculated. Beta was set to 1/|vocabulary|. The reverted value for any word not found in a category was set to beta/|words in category|. $P(x_i|y_j)$ was set to |word in category|+beta/|words in category| and $P(y)$ was set to |documents in category/|number of documents|, and $P(y|x)$ was obtained by taking the max value of the log of $P(y_j)$ and adding the sum of all log $P(x_i)$. Which ever had the highest value was used as the classifier for the category. After the classifier was trained the test documents were loaded and parsed using the same approach as the training data. The test values were then classified using the trained Bayesian classifier. The number of correctly and incorrectly classified documents was recorded down and used to determine the % accuracy. The accuracy was the number of correct over |documents|.

The accuracies you obtain under various settings:

See figure 2

Explain which options work well and why:

When beta is set to between 10^{-2} and 10^{0} the accuracy was maximized. It was also noted that if the top 100 most common words are removed the accuracy is greatly increased. When beta is too small outliers will not get smoothed out. Conversely if beta is too large important information starts to become background noise.

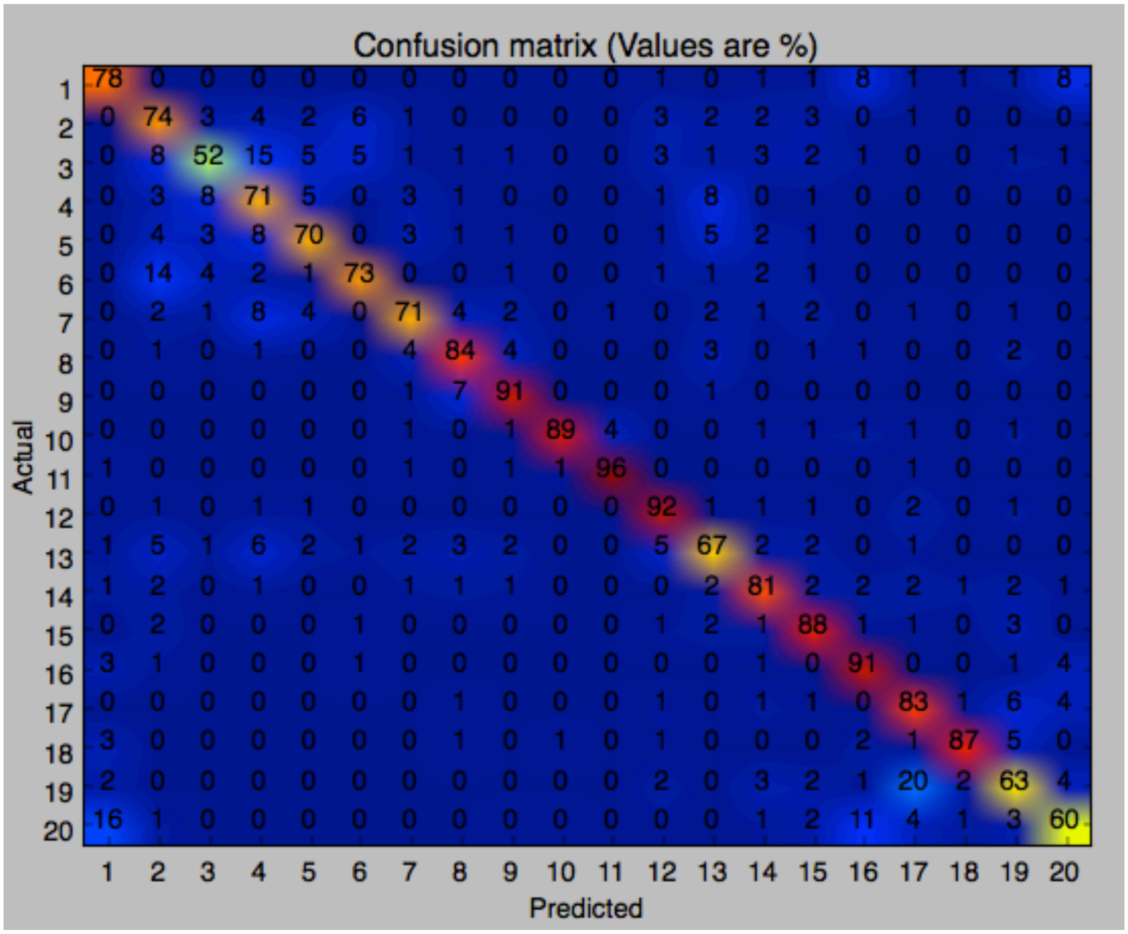
Questions:

Question 1:

It would be difficult to accurately estimate parameters because each document would be considered a unique index yet there are more than the 1000 words in each document (50,000). This makes it hard to get an accurate probability for each word at a unique index.

Question 2:

Overall testing accuracy was 78.521% when beta was set to 1/|Vocabulary|.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	78	0	0	0	0	0	0	0	0	0	0	1	0	1	1	8	1	1	1	8
2	0	74	3	4	2	6	1	0	0	0	0	3	2	2	3	0	1	0	0	0
3	0	8	52	15	5	5	1	1	1	0	0	3	1	3	2	1	0	0	1	1
4	0	3	8	71	5	0	3	1	0	0	0	1	8	0	1	0	0	0	0	0
5	0	4	3	8	70	0	3	1	1	0	0	1	5	2	1	0	0	0	0	0
6	0	14	4	2	1	73	0	0	1	0	0	1	1	2	1	0	0	0	0	0
7	0	2	1	8	4	0	71	4	2	0	1	0	2	1	2	0	1	0	1	0
8	0	1	0	1	0	0	4	84	4	0	0	0	3	0	1	1	0	0	2	0
9	0	0	0	0	0	0	1	7	91	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	1	0	1	89	4	0	0	1	1	1	1	0	1	0
11	1	0	0	0	0	0	1	0	1	1	96	0	0	0	0	0	1	0	0	0
12	0	1	0	1	1	0	0	0	0	0	0	92	1	1	1	0	2	0	1	0
13	1	5	1	6	2	1	2	3	2	0	0	5	67	2	2	0	1	0	0	0
14	1	2	0	1	0	0	1	1	1	0	0	0	2	81	2	2	2	1	2	1
15	0	2	0	0	0	1	0	0	0	0	0	1	2	1	88	1	1	0	3	0
16	3	1	0	0	0	1	0	0	0	0	0	0	0	1	0	91	0	0	1	4
17	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	83	1	6	4
18	3	0	0	0	0	0	0	1	0	1	0	1	0	0	0	2	1	87	5	0
19	2	0	0	0	0	0	0	0	0	0	0	2	0	3	2	1	20	2	63	4
20	16	1	0	0	0	0	0	0	0	0	0	0	0	1	2	11	4	1	3	60

Figure 1: Confusion matrix for accuracy of 78.521% and beta = 1/|vocabulary|

Question 3:

Newsgroups 3, 19, 20 are newsgroups that the algorithm appears confused with more than others. This could be because the words in these groups have a high frequency in multiple groups. For example in news group 20 the words used for this newsgroup could also be used frequently in newsgroups 1 and 16 and a little bit with 17 and 19 (based off the confusion matrix)

Question 4:

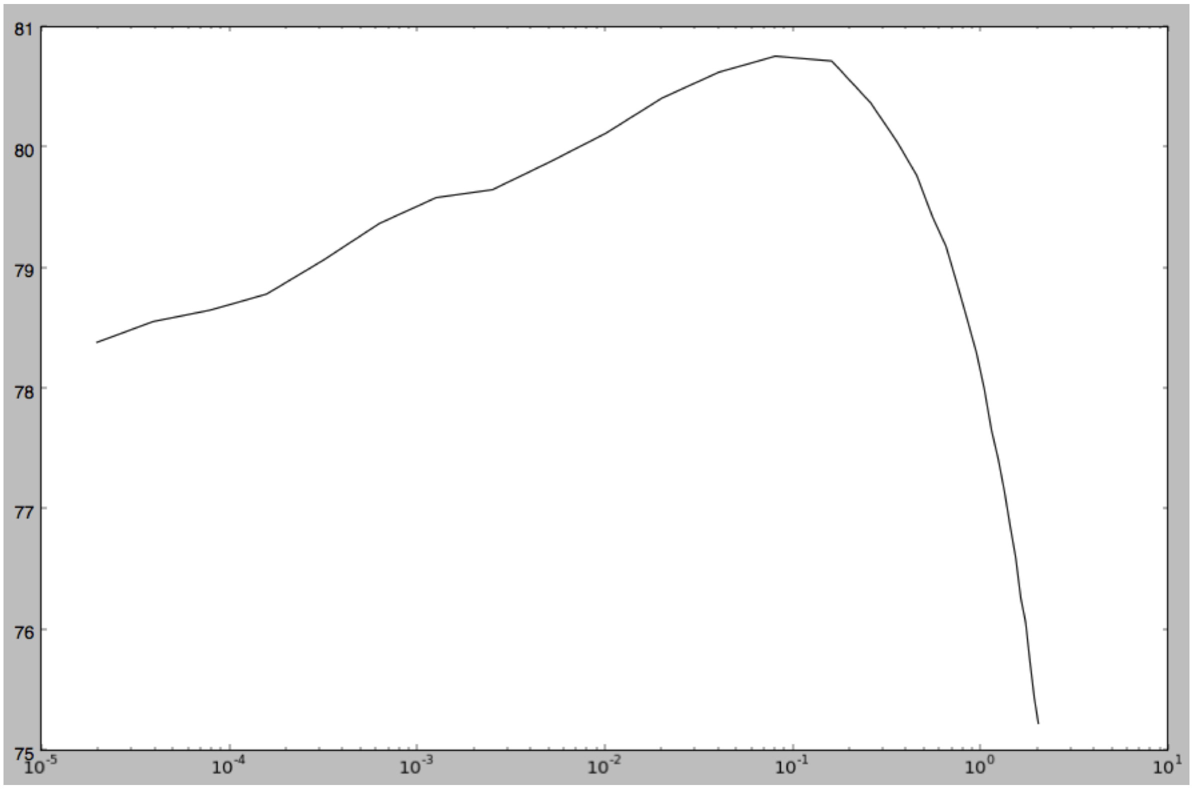


Figure 2: Values of beta set from 0.00001 to 2

The accuracy drops for low beta because inaccuracies are no longer being reduced to the point that they begin to affect the actual classification accuracy. When beta is large actual useful information begins to get degraded.

Question 5:

To choose the top 100 words I ranked the words by finding the words with the highest information gained by choosing the words with the highest probability standard deviations. As the words with the most information gained would be those that had the highest standard deviation. Furthermore I removed the top 100 most frequently used words in the English language from the list of possible words to use for the top 100.

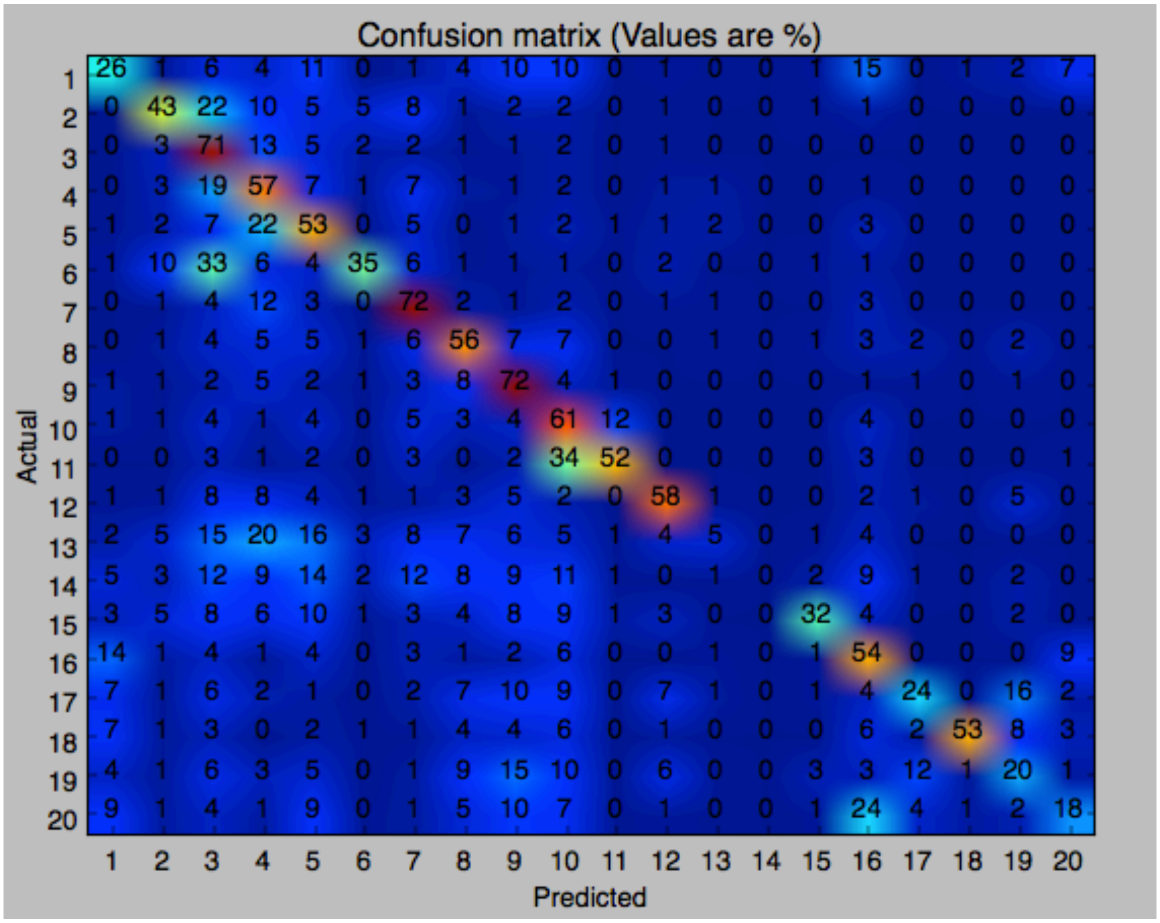
Question 6:

After implementing the proposed solution for question 5 here is the list of top 100 words

libraries , library , remained , english , realizing , ny , wallpapers , brad , for , attributed , knows , nationally , shift , corel , tapscott , cdt , cynical , kevin , blanks , package , printing , jk , iwll , kent , born , grabbing , science , anthology , teenage , keeps , unfortunately , ba , vanheyningen , materials , dominant , ames , information , games , outta , davidr , pattern , attest , server , signature , pretend , opossum , combining , but , correct , light , shoulder , votes , enlighten , hamish , throws , lynn , evaluation , per , tired , noyes , damaging , mantis , deal , people , accept , announced , nfs , guy , leon , stay , socket , nestorius , broken , the , cap , lists , project , questions , virginia , modified , kept , users , theocracy , vogle , mmwang , gif , produce , pretty , discriminate , axes , shoudl , cetera , justin , start , someday , murderers , write , mormons , atheist , nm

Question 7:

After review of the top 100 words that the classifier is relying on I made a confusion matrix using only the top 100 words for classifying the documents. I ended up getting a classification rate of 44.144% accuracy. Based on the below confusion matrix we see that some documents were actually really well classified based on the top 100 words leading me to believe that there is a biased in the data.



Confusion matrix (Values are %)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	26	1	6	4	11	0	1	4	10	10	0	1	0	0	1	15	0	1	2	7
2	0	43	22	10	5	5	8	1	2	2	0	1	0	0	1	1	0	0	0	0
3	0	3	71	13	5	2	2	1	1	2	0	1	0	0	0	0	0	0	0	0
4	0	3	19	57	7	1	7	1	1	2	0	1	1	0	0	1	0	0	0	0
5	1	2	7	22	53	0	5	0	1	2	1	1	2	0	0	3	0	0	0	0
6	1	10	33	6	4	35	6	1	1	1	0	2	0	0	1	1	0	0	0	0
7	0	1	4	12	3	0	72	2	1	2	0	1	1	0	0	3	0	0	0	0
8	0	1	4	5	5	1	6	56	7	7	0	0	1	0	1	3	2	0	2	0
9	1	1	2	5	2	1	3	8	72	4	1	0	0	0	0	1	1	0	1	0
10	1	1	4	1	4	0	5	3	4	61	12	0	0	0	0	4	0	0	0	0
11	0	0	3	1	2	0	3	0	2	34	52	0	0	0	0	3	0	0	0	1
12	1	1	8	8	4	1	1	3	5	2	0	58	1	0	0	2	1	0	5	0
13	2	5	15	20	16	3	8	7	6	5	1	4	5	0	1	4	0	0	0	0
14	5	3	12	9	14	2	12	8	9	11	1	0	1	0	2	9	1	0	2	0
15	3	5	8	6	10	1	3	4	8	9	1	3	0	0	32	4	0	0	2	0
16	14	1	4	1	4	0	3	1	2	6	0	0	1	0	1	54	0	0	0	9
17	7	1	6	2	1	0	2	7	10	9	0	7	1	0	1	4	24	0	16	2
18	7	1	3	0	2	1	1	4	4	6	0	1	0	0	0	6	2	53	8	3
19	4	1	6	3	5	0	1	9	15	10	0	6	0	0	3	3	12	1	20	1
20	9	1	4	1	9	0	1	5	10	7	0	1	0	0	1	24	4	1	2	18

Figure 3: classification with the top 100 words with accuracy of 44.144%