Assignment 3: Naive Bayes Classification

Team
Roychowdhury, Saikat <rychwdh2@illinois.edu>; 3 CREDITS
Abhinav Sharma <abhinavsharma3105@gmail.com> : 3 CREDITS
Shyam Rajendran <srajend2@illinois.edu> : 3 CREDITS

INDEX

Part 1: Digit classification

Part 1.1 Single pixels as features

Estimated priors P(class)

We estimated the priors for each class from the

“class” -> “P(class)

"0" -> "0.0958"
"1" -> "0.1126"
"2" -> "0.0976"
"3" -> "0.0986"
"4" -> "0.107"
"5" -> "0.0868"
"6" -> "0.1002"
"7" -> "0.11"
"8" -> "0.0924"
"9" -> "0.099"

Maximum a Posteriori (MAP) classification

Below is the screenshot of our MAP confusion matrix run with K = 1. The diagonal indicates the classification success for each digit class [0-9]

```
*******************************************
       DIGIT IMAGE CLASSIFICATION : BINARY
*******************************************
TOTAL TEST DOCUMENTS READ   :1000


*** CONFUSION MATRIX ***

    84.444%     0.000%     1.111%     0.000%     1.111%     5.556%     3.333%     0.000%     4.444%     0.000%
     0.000%    96.296%     0.926%     0.000%     0.000%     1.852%     0.926%     0.000%     0.000%     0.000%
     0.971%     2.913%    77.670%     3.883%     0.971%     0.000%     5.825%     0.971%     4.854%     1.942%
     0.000%     2.000%     0.000%    79.000%     0.000%     3.000%     2.000%     6.000%     2.000%     6.000%
     0.000%     0.935%     0.000%     0.000%    76.636%     0.000%     2.804%     0.935%     1.869%    16.822%
     2.174%     2.174%     1.087%    13.043%     3.261%    67.391%     1.087%     1.087%     2.174%     6.522%
     1.099%     6.593%     4.396%     0.000%     4.396%     5.495%    75.824%     0.000%     2.198%     0.000%
     0.000%     5.660%     2.830%     0.000%     2.830%     0.000%     0.000%    72.642%     2.830%    13.208%
     1.942%     0.971%     2.913%    13.592%     1.942%     5.825%     0.000%     0.971%    60.194%    11.650%
     1.000%     1.000%     1.000%     3.000%     9.000%     2.000%     0.000%     2.000%     1.000%    80.000%
-------------------------
OVERALL ACCURACY :77.010%
-------------------------
```

MAP Vs ML observation

We compared the results with and without omitting the prior terms during classification and below are our results for K=1 (binary classification)

```
ML CLASSIFICATION OMITTING PRIOR

*****************************************
        DIGIT IMAGE CLASSIFICATION : BINARY
*****************************************
TOTAL TEST DOCUMENTS READ   :1000


*** CONFUSION MATRIX ***

    84.444%     0.000%     1.111%     0.000%     1.111%     5.556%     3.333%     0.000%     4.444%     0.000%
     0.000%    96.296%     0.926%     0.000%     0.000%     1.852%     0.926%     0.000%     0.000%     0.000%
     0.971%     2.913%    77.670%     3.883%     0.971%     0.000%     5.825%     0.971%     4.854%     1.942%
     0.000%     2.000%     0.000%    79.000%     0.000%     3.000%     2.000%     6.000%     2.000%     6.000%
     0.000%     0.935%     0.000%     0.000%    75.701%     0.000%     2.804%     0.935%     1.869%    17.757%
     2.174%     2.174%     1.087%    13.043%     3.261%    67.391%     1.087%     1.087%     2.174%     6.522%
     1.099%     6.593%     4.396%     0.000%     4.396%     5.495%    75.824%     0.000%     2.198%     0.000%
     0.000%     5.660%     2.830%     0.000%     2.830%     0.000%     0.000%    72.642%     2.830%    13.208%
     1.942%     0.971%     2.913%    13.592%     1.942%     5.825%     0.000%     0.971%    60.194%    11.650%
     1.000%     1.000%     1.000%     3.000%     9.000%     2.000%     0.000%     2.000%     1.000%    80.000%
-------------------------
OVERALL ACCURACY :76.916%
-------------------------
```

As can be seen, the difference between ML and MAP is not very significant. **~ 1%**

Experimenting with different K values

We also ran the classifier with different values of Laplace Smoothing factors in the range 1-50. We observed that maximum accuracy was achieved when K was 1.

```
*****************************************
DIGIT IMAGE CLASSIFICATION : BINARY CLASSIFIER
-----------------------------------------------------------------
K , Accuracy ( in % )
1.0,77.00973220352992
2.0,76.51786976979153
3.0,76.20100217137053
4.0,76.10391479272975
5.0,75.67663337849181
6.0,75.6679377263179
7.0,75.45924207414399
8.0,75.36334915321942
9.0,75.36626177457865
10.0,75.37906450582798
11.0,75.14612271423155
12.0,75.14162792818337
13.0,75.03173781829325
14.0,74.93584489736867
15.0,74.83875751872789
16.0,74.5230843780231
17.0,74.4143887258492
18.0,74.3991510176005
19.0,74.49349064024202
20.0,74.3847949880681
21.0,74.49590609917921
22.0,74.38152120324091
23.0,74.48152120324092
24.0,74.16704252025231
25.0,73.96125948943762
26.0,73.74112042909053
27.0,73.64403305044975
28.0,73.54694567180897
29.0,73.1279722027722
30.0,72.93379744549064
31.0,72.93379744549064
32.0,72.73076217067522
33.0,72.73076217067522
34.0,72.52087206078511
35.0,72.52087206078511
36.0,72.5220665185013
37.0,72.32789176121975
38.0,72.21919610904584
39.0,72.31265405297107
40.0,72.10687102215638
41.0,72.09817536998247
42.0,72.09817536998247
43.0,72.09817536998247
44.0,72.00108799134169
45.0,72.00108799134169
46.0,71.99834023534243
47.0,71.99834023534243
48.0,71.88964458316852
49.0,71.78964458316851
50.0,71.69255720452773
```

PART 1 EXTRA CREDIT

Implement Ternary Features

Instead of considering the pixel values as either "1" or "0" depending on whether it is "background" : space or "foreground" : "+" or "#" , we gave unique values for each type of pixel to have ternary classification.

Below is the screen shot of the classifier with K=1.
We can see a slight improvement over binary classification.

```
*****************************************
        DIGIT IMAGE CLASSIFICATION : TERNARY
*****************************************
TOTAL TEST DOCUMENTS READ   :1000


*** CONFUSION MATRIX ***

    83.333%     0.000%     1.111%     0.000%     0.000%     6.667%     4.444%     0.000%     4.444%     0.000%
     0.000%    95.370%     0.000%     0.000%     0.000%     1.852%     0.926%     0.000%     1.852%     0.000%
     0.971%     2.913%    74.757%     5.825%     0.971%     0.971%     5.825%     1.942%     4.854%     0.971%
     0.000%     2.000%     0.000%    80.000%     0.000%     3.000%     2.000%     5.000%     3.000%     5.000%
     0.000%     0.000%     0.000%     0.000%    77.570%     0.935%     1.869%     0.935%     1.869%    16.822%
     2.174%     1.087%     1.087%    13.043%     3.261%    68.478%     1.087%     1.087%     2.174%     6.522%
     0.000%     4.396%     4.396%     0.000%     6.593%     5.495%    75.824%     0.000%     3.297%     0.000%
     0.000%     6.604%     2.830%     0.000%     2.830%     0.000%     0.000%    73.585%     1.887%    12.264%
     0.971%     1.942%     2.913%    11.650%     1.942%     8.738%     0.000%     0.971%    62.136%     8.738%
     1.000%     1.000%     0.000%     2.000%    10.000%     2.000%     0.000%     2.000%     2.000%    80.000%
------------------------
OVERALL ACCURACY :77.105%
-------------------------
```

Apply your Naive Bayes classifier with various features to face data.

We applied the Naïve Bayes classifier to the face data. Below is the confusion matrix with K=1.

```
*************************************************
        FACE IMAGE CLASSIFICATION
*************************************************
TOTAL TEST DOCUMENTS READ   :150


*** CONFUSION MATRIX ***

    88.312%    11.688%
     6.849%    93.151%
--------------------------
OVERALL ACCURACY :90.731%
--------------------------
```

Part 2 (for everybody): Text Document Classification

Spam detection

Spam email classification : Naive Bayes classifier with K=1

```
*********************************************
        EMAIL CLASSIFICATION
*********************************************
TOTAL TEST DOCUMENTS READ   260


*** CONFUSION MATRIX ***

    96.923%     3.077%
     1.538%    98.462%
-------------------------
OVERALL ACCURACY :97.692%
-------------------------
```

NonSpam Class classification Rate : 96.923%
Spam Class classification Rate : 98.423%

Experimenting with different K values

We also ran the classifier for K=1-50 and below are the results

****
SPAM CLASSIFICATION
****

K, Accuracy
1 ,97.692
2 ,97.308
3 ,97.308
4 ,97.308
5 ,96.538
6 ,96.538
7 ,96.538
8 ,96.538
9 ,96.538
10 ,96.538
11 ,96.538
12 ,96.538
13 ,96.538
14 ,95.769
15 ,95.769
16 ,95.769
17 ,95.769
18 ,95.769
19 ,95.769

20 ,95.769
21 ,95.769
22 ,95.769
23 ,95.769
24 ,95.769
25 ,95.769
26 ,95.769
27 ,95.769
28 ,95.769
29 ,95.769
30 ,95.769
31 ,95.769
32 ,95.769
33 ,95.769
34 ,95.769
35 ,95.769
36 ,95.769
37 ,95.769
38 ,95.769
39 ,95.769
40 ,95.769
41 ,95.769
42 ,95.769
43 ,95.769
44 ,95.769
45 ,95.769
46 ,95.769
47 ,95.769
48 ,95.769
49 ,95.769
50 ,95.769

Eight newsgroups

Results with K=1 as below

```
*******************************************
        NEWS CLASSIFICATION
*******************************************
TOTAL TEST DOCUMENTS READ   263


*** CONFUSION MATRIX ***

    97.059%     0.000%     0.000%     0.000%     2.941%     0.000%     0.000%     0.000%
     0.000%    84.848%     0.000%    12.121%     3.030%     0.000%     0.000%     0.000%
     0.000%     0.000%    97.222%     0.000%     0.000%     0.000%     0.000%     2.778%
     0.000%     0.000%     0.000%    89.286%     3.571%     0.000%     0.000%     7.143%
     2.128%     0.000%     0.000%     0.000%    97.872%     0.000%     0.000%     0.000%
     0.000%    40.000%     0.000%     0.000%    10.000%    40.000%     0.000%    10.000%
     0.000%     0.000%     0.000%     0.000%     0.000%     0.000%   100.000%     0.000%
     3.448%     3.448%     0.000%     6.897%     0.000%     0.000%     0.000%    86.207%
-------------------------
OVERALL ACCURACY :86.562%
-------------------------
```

Classification rate per class:

CLASS	CLASSIFICATION RATE
sci.space	97.059%
comp.sys.ibm.pc.hardware	84.848%
rec.sport.baseball	97.22%
comp.windows.x	89.286%
talk.politics.misc	97.872%
misc.forsale	40%
rec.sport.hockey	100%
comp.graphics	86.207%

To complete
dataset, and over 80% on the newsgroup dataset. Additionally, for each class, report the top 20 words with the highest likelihood. Finally, as in Part 1.1, take the pair of classes from the email dataset and four highest-confusion pairs from the newsgroup dataset, and display the top 20 words with the highest log-odds ratio for that pair of classes.

Extra Credit for Part 2

bag-of-words representations of the documents using word cloud maps

We created a visualization of bag of words and have hosted it online @ <URL PASTE>

Screenshot of the visualization as below < PASTE SCREEN SHOT BELOW>

Statement of individual contribution

SHYAM RAJENDRAN	**Implemented** Part1 + Single pixels as features classification + Ternary feature based classification + Face data classification Part2 + Spam Detection + Newsgroup classification
ABHINAV SHARMA	
SAIKAT ROYCHOWDHURY	