Hot Topics in Machine Learning (HWS17)
Assignment 2: Naive Bayes

Chen Zongqi

1. Training

Train the dataset mnist in few seconds. During test my function, when I set alpha is 1, I meet divided by zero problem. It might wasted few seconds compared with alpha is 2. Which I need to mention is that I assume alpha is a constant number, so I just calculate MAP with symmetric Dirichlet using K * (alpha -1).

2. Prediction

When I compute labels log probabilities, if I normalize their probabilities and then compute its log, I can get right number and pass the test part. However, I use this function calculating mnist dataset, it will occur divided by zero problem. It will seriously influence prediction accuracy. Without calculating log probability, I use directly logjoint, then I can get good accuracy.

3. Experiments on MNIST digits data

a) The accuracy of model is 83.63%

b) See Figure 1, label 0 contains an error 6 and label 1 contains error 3 and two 7. Figure 2 shows that randomly pick 100 error predictions, we can see label 6 and 7 have obviously less error predictions.

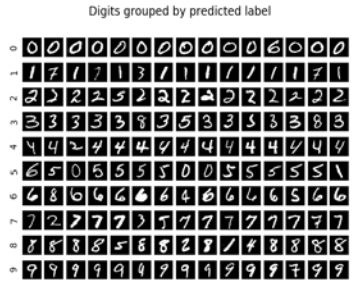

Figure 1

Figure 2

See the confusion matrix below, it can confirm that label 6 and 7 have less prediction errors, particularly label 6 predicts 7 and 9 has no error ,and label 1 as well, it predicts 0, 4, 7 and 9 has no error.

```
[[ 872    0    3    5    3   63   18    1   14    1]
 [   0 1102    8    3    0    3    4    0   15    0]
 [  15   28  816   37   26    8   31   18   49    4]
 [   4   22   28  835    1   29   10   14   45   22]
 [   2    8    6    1  808    2   15    1   20  119]
 [  22   22    5  128   29  602   20   16   20   28]
 [  15   20   16    1   20   30  852    0    4    0]
 [   1   41   15    3   17    0    2  862   18   69]
 [  15   23   11   68   12   31   10    7  759   38]
 [  12   14    5    9   64    8    1   26   15  855]]
```

The reason why it occurs wrong prediction might be that one pixel in our digits has high value in few labels. For example, label 4 and label 9, label 4 wrong predicts label 9 about hundred errors. Label 4 and label 9 have same unique pixels then the probability of both may be higher than others.

4. Model selection (optional)

In my experiment, I set alpha [1, 1.5, 2, 2.5, 5], and results as follow: 0.30675, 0.82775, 0.829416666667, 0.819583333333 and 0.78525. So when alpha is 1, the accuracy is worse than 0.5. And alpha from 1 to 2, increasing to 0.8294 which alpha 2 has the highest accuracy. Later, when alpha is greater than 2, the accuracy is decreased, when alpha equals 5 has lowest value 0.78. It seems add-one smoothing has best performance. The reason might be, when alpha is greater than 2, alpha would influence the value more than the value itself.

5. Generating data

a) Figure 3 is I generate digits by randomly choosing from Dirichlet distribution. It looks quite in a mess and is hard to figure out what digit is.

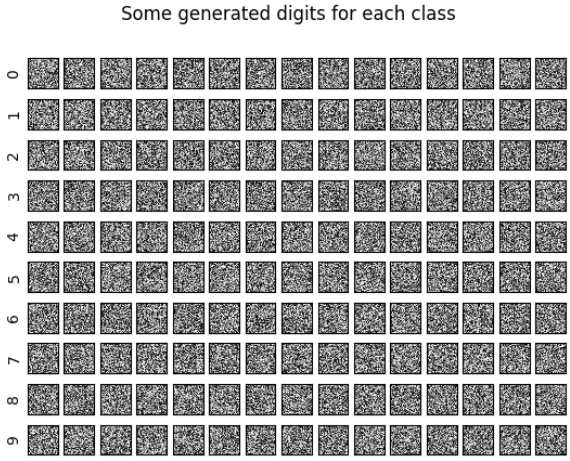

Figure 3

b) In this case, I generate two more digits by alpha = 1 (see Figure 4) and alpha = 5 (see Figure 5). Compared with those digits, we can find that when alpha is smaller than 2, it can figure out what digit is at least. When alpha is greater than 2, the digits show quite in a mess. The reason might be without smoothing, if one pixel value is 0 then it always 0 when generate digits. By contrast, after smoothing one pixel value should be zero but it has value. That's why when alpha is greater than 2, those digits are hard to tell.

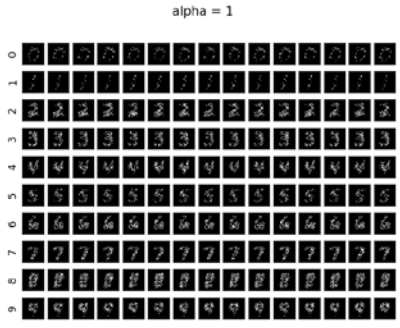

Figure 4

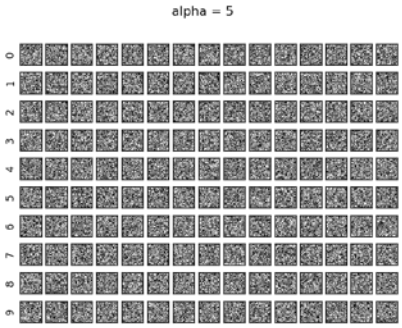

Figure 5