CSL 603 - Machine Learning (Lab 4)

Naive Bayes Classifier and K-Mean Clustering

Rishabh Singh - 2016CSB1054

November 20 , 2018

Abstract

In this lab I implemented a naive bayes classifier to classify mails as Spam or Ham ('Not Spam') . I also considered different values of m to evaluate the hyper parameter.

I also implemented K-mean Clustering to classify the MNIST handwritten digits' dataset.

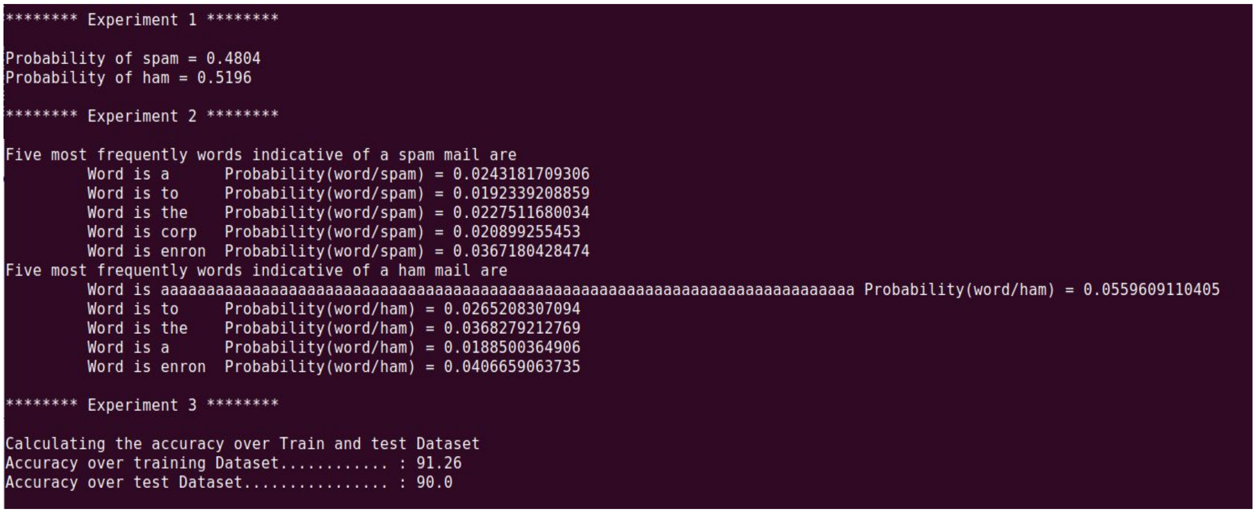
Naive Bayes Classifier

Introduction

Naive Bayes classifier is a probabilistic classifier which is based on Bayes' theorem. IT assumes that all the input features are independent and hence calculate the posterior for a particular class given example. For classification I calculated posterior of that example for class spam and class ham and then the give class label as argmax(probability(class/x)).

The results obtained from the given dataset is as follows….

Observations:



```
******** Experiment 1 ********

Probability of spam = 0.4804
Probability of ham = 0.5196

******** Experiment 2 ********

Five most frequently words indicative of a spam mail are
         Word is a      Probability(word/spam) = 0.0243181709306
         Word is to     Probability(word/spam) = 0.0192339208859
         Word is the    Probability(word/spam) = 0.0227511680034
         Word is corp   Probability(word/spam) = 0.020899255453
         Word is enron  Probability(word/spam) = 0.0367180428474
Five most frequently words indicative of a ham mail are
         Word is aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa Probability(word/ham) = 0.0559609110405
         Word is to     Probability(word/ham) = 0.0265208307094
         Word is the    Probability(word/ham) = 0.0368279212769
         Word is a      Probability(word/ham) = 0.0188500364906
         Word is enron  Probability(word/ham) = 0.0406659063735

******** Experiment 3 ********

Calculating the accuracy over Train and test Dataset
Accuracy over training Dataset............ : 91.26
Accuracy over test Dataset................ : 90.0
```

Fig : 1 Details of the naive bayes classifier on our dataset

The above figure shows the probabilities of each class and the most frequently words of both the classes. The word "aaaaa......aaaaa" has maximum probability in class ham while word "enron" has maximum probability in class spam. The accuracy observed on the training dataset is 91.26% while that on test dataset is 90.0%. I faced a lot of difficulty in computing the prior probabilities as multiplication makes it close to zero, which causes decrease in accuracy due to wrong classification. So i added log of prior and then compared the log posterior probabilities to give class labels.

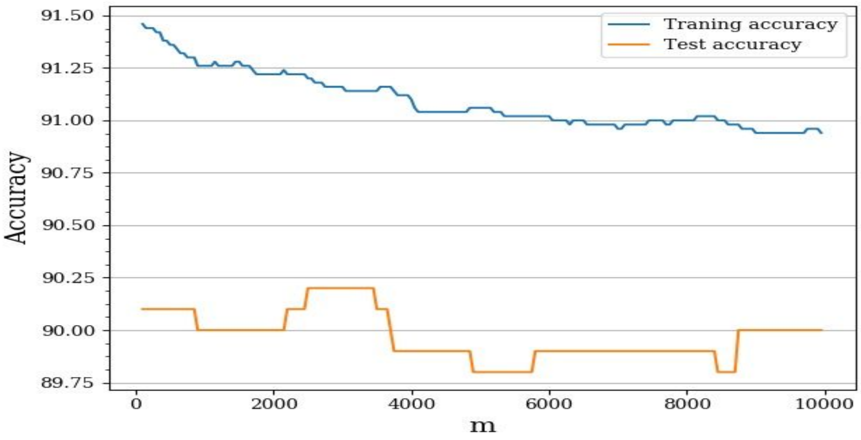


Fig 2: Variation of Accuracy vs m (with constant p)

From the above graph I observer that as we increase m (with constant p) the Training accuracy decreases. This can be explained by the fact that as we increase m the prior terms becomes insignificant because it is inversely proportional to m, and the model tends towards random classification.

The model works on the probabilities computed by the words occurring in a spam mail. If I had to beat this classifier I would add more words that have higher posterior probability of not spam in my actually “spam” mail.

K-Mean Clustering

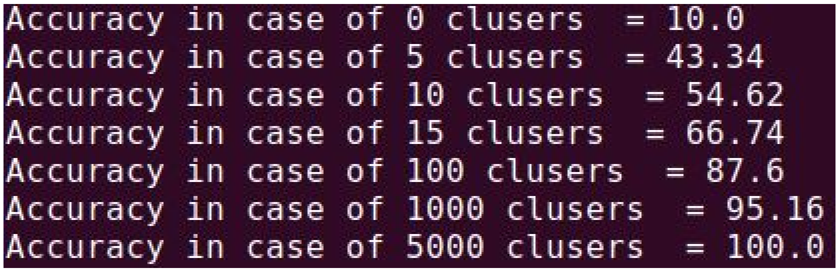
Introduction

In this part of the assignment I used MNIST handwritten digits dataset to perform K-Mean clustering. I used sklearn’s inbuilt ”KMeans” function to perform clustering. I calculated the confusion matrix and the accuracy over various value of m.

The results obtained from the given dataset is as follows….

Observation

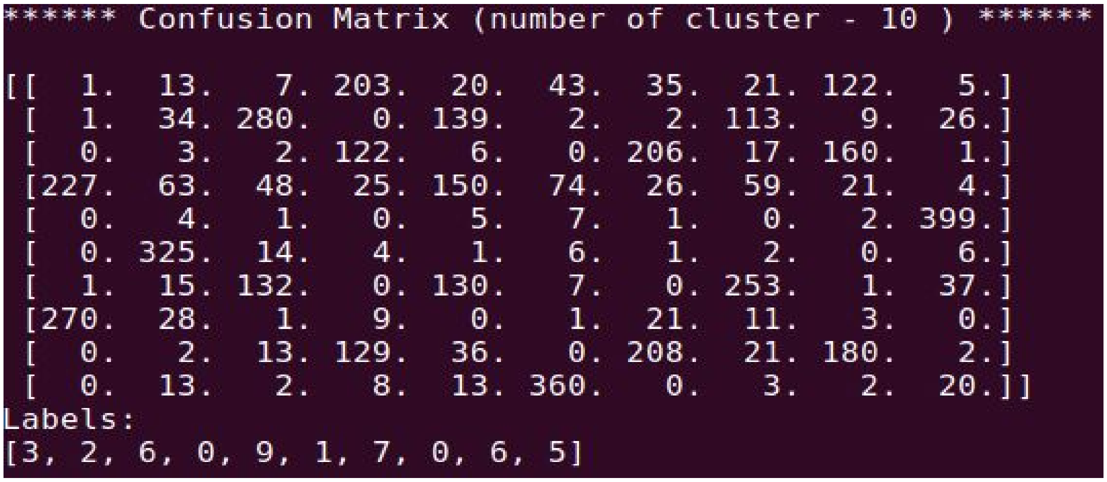
There is a continuous increase upto 100% with increasing number of clusters. This is due to the fact with increasing number of clusters the number of samples in each cluster reduce and hence the model tends to overfitting. I increased the number of cluster to 5000 and observed 100% accuracy because each element has its own cluster.



```
Accuracy in case of 0 clusers  = 10.0
Accuracy in case of 5 clusers  = 43.34
Accuracy in case of 10 clusers  = 54.62
Accuracy in case of 15 clusers  = 66.74
Accuracy in case of 100 clusers  = 87.6
Accuracy in case of 1000 clusers  = 95.16
Accuracy in case of 5000 clusers  = 100.0
```

Fig 3: Accuracy in case of different values of number of clusters.

The confusion matrix and the labels for clusters are shown below.

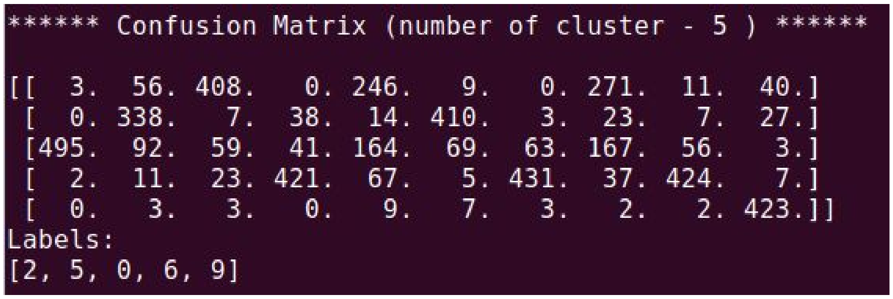


```
****** Confusion Matrix (number of cluster - 10 ) ******

[[  1.  13.   7. 203.  20.  43.  35.  21. 122.   5.]
 [  1.  34. 280.   0. 139.   2.   2. 113.   9.  26.]
 [  0.   3.   2. 122.   6.   0. 206.  17. 160.   1.]
 [227.  63.  48.  25. 150.  74.  26.  59.  21.   4.]
 [  0.   4.   1.   0.   5.   7.   1.   0.   2. 399.]
 [  0. 325.  14.   4.   1.   6.   1.   2.   0.   6.]
 [  1.  15. 132.   0. 130.   7.   0. 253.   1.  37.]
 [270.  28.   1.   9.   0.   1.  21.  11.   3.   0.]
 [  0.   2.  13. 129.  36.   0. 208.  21. 180.   2.]
 [  0.  13.   2.   8.  13. 360.   0.   3.   2.  20.]]
Labels:
[3, 2, 6, 0, 9, 1, 7, 0, 6, 5]
```

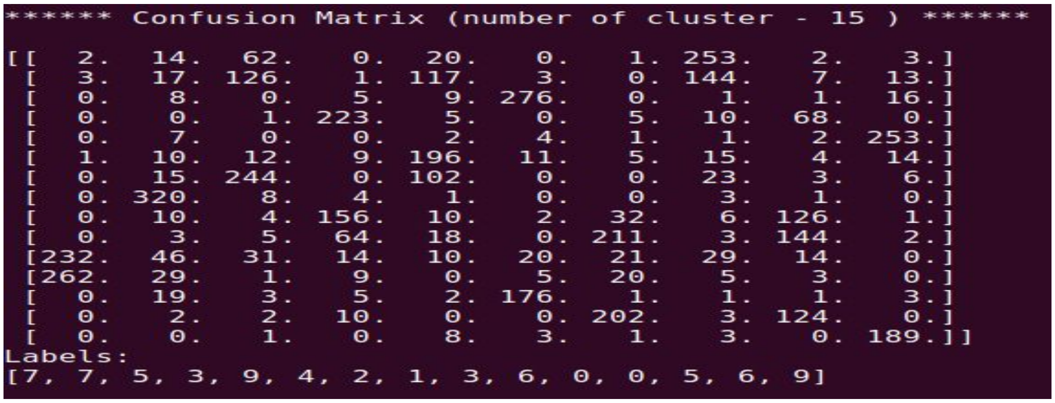
Fig 4 : Confusion matrix for 10 clusters.

From the above figure I observed that Class label 4, 8 is not assigned to any of the clusters which is because the column 4, 8 is never maximum in any of the rows. From that i can infer that 4 and 8 class merges to some other class.



```
****** Confusion Matrix (number of cluster - 5 ) ******

[[  3.  56. 408.   0. 246.   9.   0. 271.  11.  40.]
 [  0. 338.   7.  38.  14. 410.   3.  23.   7.  27.]
 [495.  92.  59.  41. 164.  69.  63. 167.  56.   3.]
 [  2.  11.  23. 421.  67.   5. 431.  37. 424.   7.]
 [  0.   3.   3.   0.   9.   7.   3.   2.   2. 423.]]
Labels:
[2, 5, 0, 6, 9]
```



```
****** Confusion Matrix (number of cluster - 15 ) ******

[[  2.  14.  62.   0.  20.   0.   1. 253.   2.   3.]
 [  3.  17. 126.   1. 117.   3.   0. 144.   7.  13.]
 [  0.   8.   0.   5.   9. 276.   0.   1.   1.  16.]
 [  0.   0.   1. 223.   5.   0.   5.  10.  68.   0.]
 [  0.   7.   0.   0.   2.   4.   1.   1.   2. 253.]
 [  1.  10.  12.   9. 196.  11.   5.  15.   4.  14.]
 [  0.  15. 244.   0. 102.   0.   0.  23.   3.   6.]
 [  0. 320.   8.   4.   1.   0.   0.   3.   1.   0.]
 [  0.  10.   4. 156.  10.   2.  32.   6. 126.   1.]
 [  0.   3.   5.  64.  18.   0. 211.   3. 144.   2.]
 [232.  46.  31.  14.  10.  20.  21.  29.  14.   0.]
 [262.  29.   1.   9.   0.   5.  20.   5.   3.   0.]
 [  0.  19.   3.   5.   2. 176.   1.   1.   1.   3.]
 [  0.   2.   2.  10.   0.   0. 202.   3. 124.   0.]
 [  0.   0.   1.   0.   8.   3.   1.   3.   0. 189.]]
Labels:
[7, 7, 5, 3, 9, 4, 2, 1, 3, 6, 0, 0, 5, 6, 9]
```

Fig 5 : confusion matrix for 5, 15 clusters.

When we change number of cluster to 5 then the clusters classes like 1,3,7 also merges and now we have only 5 different cluster with no splitted cluster.

From k = 15 (i.e. Number of clusters equal to 15), 0,3,5,6,9 splitted to two clusters while 8 stills merges and we do not have any cluster with label equal to 8. This may be due to similar curvy nature of 0,3,6,9 and somewhat 5. We can infer that increasing the number of cluster splits the clusters and if we keep increasing k then 8 will also split to any one of the cluster.