ISCO630E-ASSIGNEMNT-5

Conclusion

By Nairit Banerjee-IIT2016505

Question 1

We need to classify a given mail as Spam or Ham using Naive Bayes Classifier.

After importing the data, the first task is to clean it and tokenize the mail contents.

Cleaning refers to removal of duplicate entries, special characters, punctuations and stop words.

Now we split the cleaned messages and obtain a list of words and corresponding label pairs.

We also perform stemming(converting morphologically similar words to root word).

We split the entire dataset into 70:30 ratio for training and testing purposes respectively.

Now we train the following four probabilities:

1) Probability that a word occurs in spam mails

2) Probability that a word occurs in ham mails

3) Probability that any given mail is spam

4) Probability that any given mail is ham

To predict whether a mail is spam or not, we see which of the following is greater,

$$P(spam|w1 \cap w2 \cap ... \cap wn)\ versus\ P(\sim spam|w1 \cap w2 \cap ... \cap wn)$$

Thus we compute the following,

$$P(spam|w1 \cap w2 \cap ... \cap wn) = \frac{P(w1 \cap w2 \cap ... \cap wn|spam).P(spam)}{P(w1 \cap w2 \cap ... \cap wn)}$$

Considering each word independent to each other, we can simplify above expression as,,

$$\frac{P(w1|spam).\,P(w2|spam)\ldots P(wn|spam).\,P(spam)}{P(w1).\,P(w2)..\,P(wn)}$$

In order to find probability of each word being in a spam/ham mail, we create two dictionaries. One of them stores all words appearing in spam mails of training data and the other one stores all words appearing in ham mails.

We also insert corresponding probability of a mail being spam if that word appears. For those words appearing in test dataset but not in train dataset, we set the probability to 0.5, as we cannot judge the mail being spam or not.

Testing on 1672 unseen mails, we got an accuracy of 95.39.

In order to evaluate the classifier we compute the confusion matrix,

Predicted		**0**	**1**	**All**
Actual				
0		1388	56	1444
1		21	207	228
All	1409	263	1672	

We additionally compute the following metrics,

Ham precision: 0.9850958126330731

Ham recall: 0.961218836565097

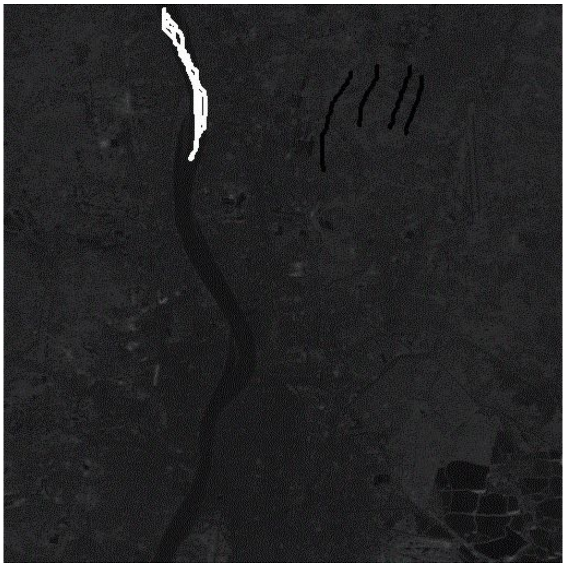
Spam precision: 0.7870722433460076

Spam recall: 0.9078947368421053

Question 2

We are given 4 satellite images of Hoogly river, corresponding to R-band, G-band, B-band, and I-band images. We need to choose 50 sample river points and 100 sample non-river points and using Naive Bayes Classification, predict class of each pixel in am 512*512 image.

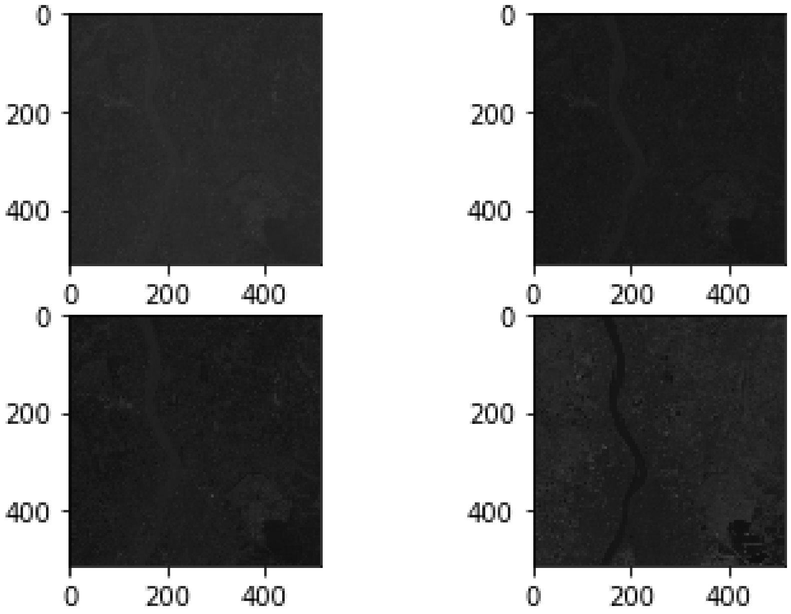
For obtaining the river and non-river sample locations, we manually do some markings in the image as follows.



We now loop through the image and store the locations of 50 river where we encounter a white marking 100 non-river points where we encounter a black marking.

Next we store intensity values of all the four bands into a dataset at the chosen locations.

The four bands looks as follows,



We also store intensity values of all 512*512 pixels in a dataset for all the four-bands. This will serve as the testing data, on which we predict the classes.

Next we calculate mean of river(T1), non-river(T2) classes and obtain the Covariance matrices of the deviation of the sample points from its mean using following formula,

$$\mathrm{Cov}\,(X, Y) = \frac{\sum(X_i - \overline{X})(Y_j - \overline{Y})}{n}$$

Now we loop through all 512*512 pixels we stored and compute,

River_class = (Test_data – T1) ‘ * Inverse (Cov_matrix_River) *(Test_data – T1)

Nonriver class = (Test_data – T2) ‘ * Inverse (Cov_matrix_NonRiver) *(Test_data –T2)

Then we apply multivariate Normal Distribution and obtain the density functions,

p1 = (-0.5) * 1/sqrt(Determinant of Covariance_matrix_Riverclass) * exp(River_class)

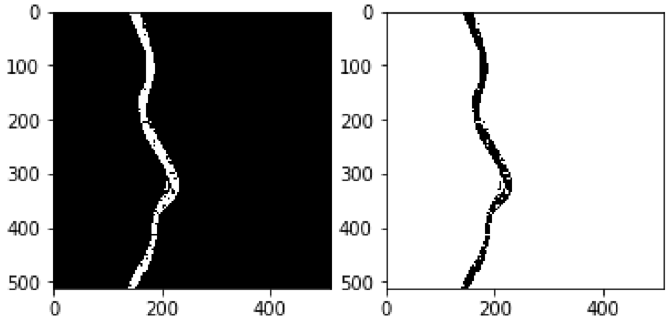
p2 = (-0.5) * 1/sqrt(Determinant of Covariance_matrix_nonRiverclass) * exp(NonRiver_class)

For each pixel location of test image apply Bayes’ rule (P1 * p1) >= (P2 * p2) then

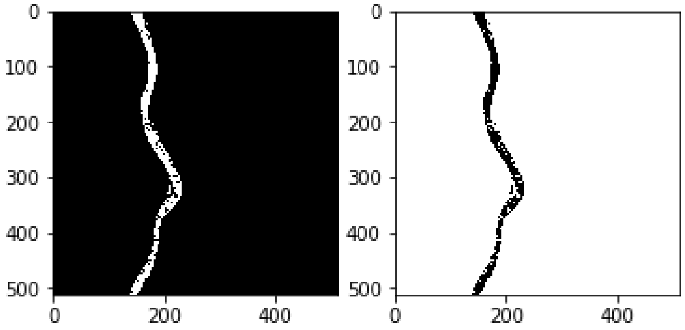
Output_image(i) = 255 (River class) Else

Output_image(i) = 0; (Non-river class)

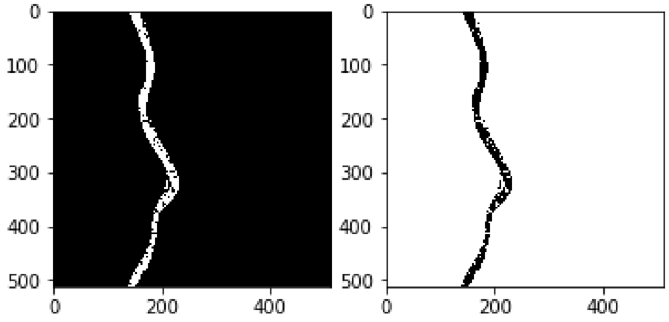
Taking P1 = 0.7 and P2 = 0.3 we get,



Taking P1 = 0.3 and P2 = 0.7 we get,



Taking P1 = 0.5 and P2 = 0.5 we get,


0
100
200
300
400
500
0
200
400
0
100
200
300
400
500
0
200
400