CS5011: Introduction to Machine Learning **Fall 2014**

Programming Assignment 1 Report

26/09/2014 *Ditty Mathew, CS13D018*

1 Linear Classification

1.1 Creating DS1

1.1.1 Objective

We have to generate two classes with 10 features where each class is given by a multivariate Gaussian distribution, with both the class sharing same co-variance matrix.

1.1.2 Details of Experiment

Mean and Co-variance has chosen as follows,
Mean of Class A(Mean1A) = [1,0,0,0,0,0,0,0,0,0]

Mean of Class B(Mean1B) = [2,0,0,0,0,0,0,0,0,0]

Covariance : Randomly generated 10x10 matrix. Let it be A. Then Covariance = $A^T A$

1.1.3 Observation

Mean of two classes should be close for overlap between the class. So Mean decides the linear separability of data.

1.2 Linear classifier using regression on indicator variables

1.2.1 Objective

To learn a linear classifier by using regression on indicator variable.

1.2.2 Details of Experiment

Experimented for the data generated DS1 and also with different sets of mean and co-variance and some other set of mean also. Other set of mean chose are as follows,
Mean2A =[1,0,0,0,0,0,0,0,0,0]

Mean2B =[1.1,0,0,0,0,0,0,0,0,0]

Mean3A =[1,1,1,1,1,1,1,1,1,1]

Mean3B =[5,5,5,5,5,5,5,5,5,5]

Since co-variance matrix is randomly generated, it has not been mentioned. Classes are labeled as 0 and 1. So if $f(x_i) <= 0.5$, $yi_{predict} = 0$ else 1

1.2.3 Evaluation Details

The details of result for each set of Mean are as follows,

Coefficient for Mean1A,Mean1B = -5.09564817e-02, 3.68222158e-01, -1.65855970e-02, 5.94640604e-02, 9.66634581e-02, -2.06716021e-02, -2.59041519e-01, -9.70990651e-03, -5.47248286e-06, 2.93440784e-02, -2.21585239e-01

Coefficient for Mean2A,Mean2B = 0.35001674, 0.14908351, 0.06557866, -0.006822, -0.03901959, -0.05351156, 0.02764919, -0.21378781, -0.03756652, 0.15370071, 0.0104775

Coefficient for Mean3A,Mean3B = -0.24052163, 0.02200995, 0.02223361, -0.02958265, -0.02865028, 0.02243879, 0.03886829, -0.00534119, -0.01137986, -0.02045257, 0.23706911

Measure	**Mean1A, Mean1B**	**Mean2A, Mean2B**	**Mean3A, Mean3B**
Accuracy	0.8	0.52	1
Precision	0.79	0.51	1
Recall	0.8	0.55	1
F-measure	0.8	0.53	1

1.2.4 Observation

As the mean of two class is close by, accuracy is less since it is not linearly separable. In the case of Mean3A, Mean3B ; data of two class is far away from each other. So there is no overlap and classified correctly.

1.3 KNN Classifier

1.3.1 Objective

To learn Knn classifier and to analyze how it works while varying the value of k.

1.3.2 Details of Experiment

1.3.3 Results

Accuracy, precision, recall and F-measure for different values of k, for data generated with mean - Mean1A,Mean1B has shown in the table below.

Measure	k=1	k=5	k=10	k=50	k=100
Accuracy	0.63	0.65	0.69	0.72	0.71
Precision	0.61	0.63	0.69	0.70	0.69
Recall	0.66	0.68	0.64	0.73	0.73
F-measure	0.63	0.66	0.66	0.71	0.71

.

For mean - Mean2A, Mean2B, evaluation details has shown in the following table.

Measure	k=1	k=5	k=10	k=50	k=100
Accuracy	0.51	0.48	0.49	0.5	0.5
Precision	0.50	0.47	0.48	0.49	0.49
Recall	0.57	0.51	0.41	0.6	0.65
F-measure	0.53	0.49	0.44	0.54	0.56

1.3.4 Observation

Since here we are using one gaussian per class, linear decision boundary is the best one can do. Then also, KNN is working almost close to(not worse) regression on indicator variables. As per the experiments when there was less overlap, higher value of k worked better.

1.4 Mixture of 3 Gaussians

Chosen means are as follows
mean1A=[1,1,1,1,1,1,1,1,1,1]
mean2A=[5,5,5,5,5,5,5,5,5,5]
mean3A=[10,10,10,10,10,10,10,10,10,10]
mean1B=[1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5]
mean2B=[5.5,5.5,5.5,5.5,5.5,5.5,5.5,5.5,5.5,5.5]
mean3B=[10.5,10.5,10.5,10.5,10.5,10.5,10.5,10.5,10.5,10.5]

Co-variance matrix generated randomly.

1.4.1 Linear classifier by using regression on indicator variables

Coefficients are 4.20988588e-01, 1.16845762e-02, -1.64395453e-02, -2.21802547e-02, 1.23688398e-02, 5.33348832e-02, 1.32639862e-03, -5.28964364e-03, -3.17427559e-02, -1.72779668e-04, 9.01481460e-03

Measure	Values
Accuracy	0.5
Precision	0.49
Recall	0.6
F-measure	0.54

1.4.2 Knn classifier

Measure	k=1	k=5	k=10	k=50	k=100
Accuracy	0.63	0.66	0.66	0.66	0.68
Precision	0.62	0.65	0.65	0.65	0.67
Recall	0.67	0.65	0.65	0.65	0.69
F-measure	0.61	0.65	0.65	0.65	0.68

1.4.3 Observation

In this case Knn classifier is working better than other. But in the case of DS1, regression on indicator variable is better than Knn classifier. If the data is linearly separable regression is better. In other case Knn is better

2 Linear Regression

2.1 Preprocessing Dataset

The given data is of order 1994x128. The first five columns are described to be non-predictive variables. Hence they can be ignored. Now in the remaining 1994x123 matrix, the last column forms the target. Hence, excluding that, the remaining 1994x122 is the data which has to be split into training and test data sets. To make the rest data usable, filled the missing data in two ways.

1. Mean of the missing attribute
2. Interpolation of missing attribute

In order to determine which of the above two techniques is better, we use both the data(filled by sample means and the one filled by interpolation). By observing the RSS error obtained in the next question, we can conclude that both are working similar.

2.1.1 Fit using linear regression

By Mean of missing attribute
The coefficients are stored in file CandC-train2-Mean-regressionCoefficients.txt.
RSS of Best fit data = 6.99
Best fit obtained for data CandC-train2.csv ,Cand-test2.csv

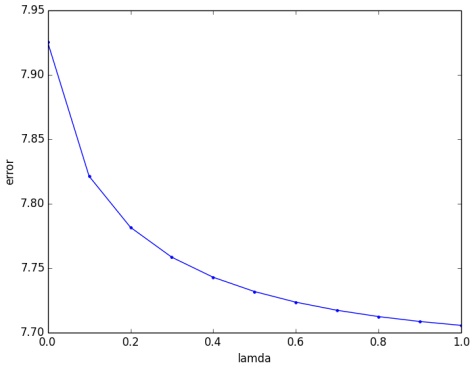
Averaged RSS error over 5 different 80-20 splits = 7.88

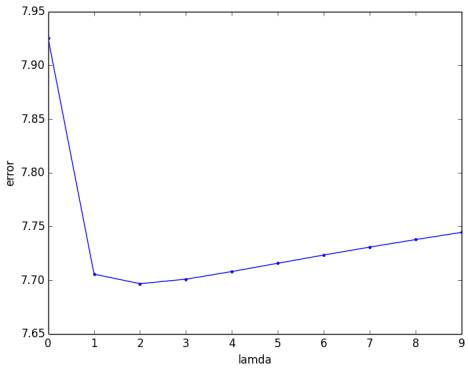
By Interpolation of missing attribute
The coefficients are stored in file CandC-train2-Interpolation-regressionCoefficients.txt.
RSS of Best fit data = 6.95
Best fit obtained for data CandC-train2.csv ,Cand-test2.csv

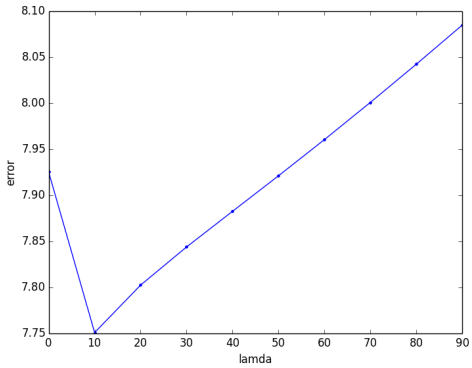
Averaged RSS error over 5 different 80-20 splits = 7.92

2.1.2 Ridge Regression

Experimented for different values of Λ and the following graphs shows the rss error obtained for different lamda values(x axis : lamda, y axis : rss error).







The average RSS obtained for some lamda values are as follows:

Lambda	Avg RSS
0	7.92
0.5	7.73
1	7.70
10	7.75
50	7.92
100	8.12

For lambda =1, we are getting best fit.
It is possible to use this method for feature selection. Ridge regression

shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares. Therefore, the best fit we can achieve at minimum RSS and it is 7.7.

3 Feature Extraction

3.1 PCA

The given data is of order 2000x3. Extracted the feature along first component axis and used the training data in the projected space to train linear regression with indicator random variables. Then the learned model is used to classify the test instances. The results are given below:
Confusion matrix with actual labels along the rows and predicted labels along the columns:

	1	2
1	121	79
2	79	121

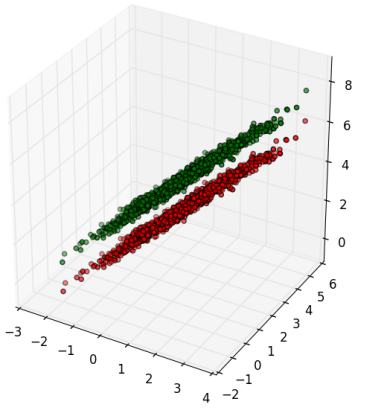
.

The per-class precision, recall and f-measures are shown below:

	Precision	Recall	F-measure
class1	0.61	0.61	0.61
class2	0.61	0.61	0.61

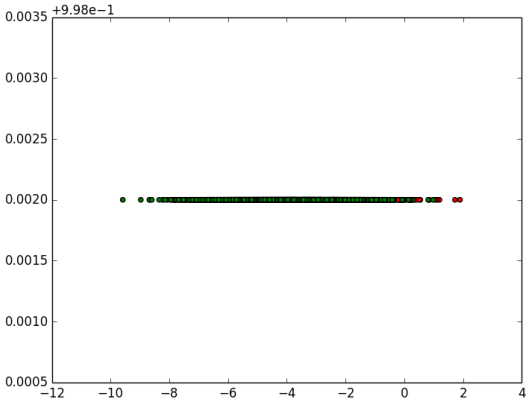
.

The plot indicating the original data is shown below,



.

The plot indicating the projected data is shown below.



3.1.1 LDA

The same dataset is used here. Extracted the projected dataset in the derived feature space. The results are given below:
Confusion matrix with actual labels along the rows and predicted labels along the columns:

	1	2
1	200	0
2	0	200

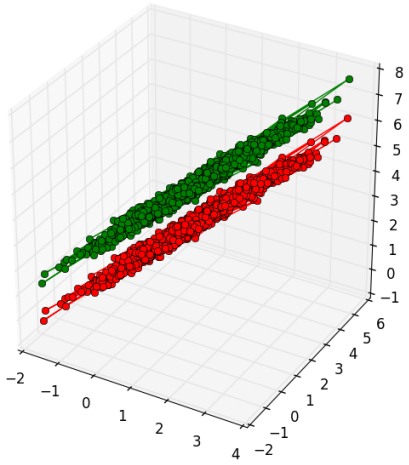
The per-class precision, recall and f-measures are shown below:

	Precision	Recall	F-measure
class1	1	1	1
class2	1	1	1

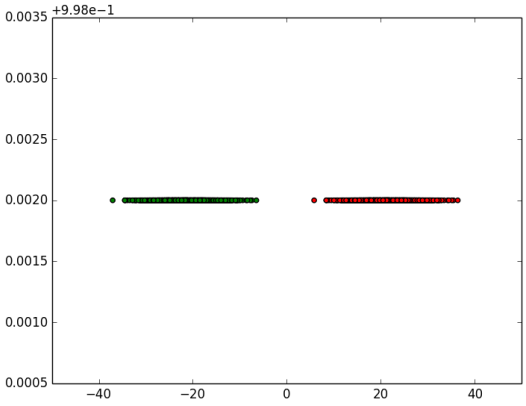
The plot indicating the original data is shown below,

.

.



The plot indicating the projected data is shown below.



3.2 Observation

LDA is doing well than PCA. In PCA we are classifying the projected data. So the structure of input data is loosing after projection. That is why we are not able to classify correctly even though there is linear decision boundary in the input data. But in LDA the classification is considering the structure of input

data and it is classifying correctly.