nD – 2Class and K- Class Gaussian Discriminant Analysis

1. Problem Statement

The multi feature data set with two classes and K classes should be classified using generative learning model. The data should be trained and tested using the generative learning model. The 10-fold cross validation has to be done to analyze the performance of the model. The accuracy of the model is analyzed with the training and testing error calculated by Mean Square Error.

2. Problem Solution

The solution is to find the parameters of generative learning. The parameters are the prior probability and the likelihood for each class. In order to find the likelihood we compute the mean and covariance of each class. Based on the Baye's rule we find the posterior probability of the given data to each class and determine the class label of the data using discriminant function. The features and the labels are in the form of vectors. The predictor function is used for modeling.

Let X, Y be the feature vector and the label vector,

Baye's Rule:

$P(Y = j| X) = P(Y=j,X)/P(X)$

$= P(X|Y=j)*P(Y=j)/P(X)$

P(X|Y=j) is the likelihood

P(Y=j) is the prior probability of class j

$\mu_j = (\sum I(Y^i=j)X^{(i)})/m_j$ Mean of class j

$\sum j = (\sum I(Y^i=j)(X^{(i)}- \mu_j) (X^{(i)}- \mu_j)^T)) /m_j$

3. Implementation Details

Two multidimensional with two class data and K class have been used. The same steps followed in one dimensional two class date set is followed here.

I have implemented the code in ipython notebook. The filename has to be mentioned and in the tool bar option “Cell” -> “Run All” will implemented the whole file and the results will be printed.

4. Results and Discussion

N Dimensional Two Class Dataset

The data set used is "**breast-cancer-wisconsin.data.txt**"

a) Find the classes and prior probability

```
The Classses are [2, 4]
The Classes Count  {2: 458, 4: 241}
The prior probabiliy {2: 0.6552217453505007, 4: 0.3447782546494993}
```

The classes represent the class labels, the classes count is the number of training data in each class. The prior probability is the probability distribution of each class.

b) The mean for each class,

```
Mean for mulivariate features
2 [ 2.95633188  1.32532751  1.44323144  1.36462882  2.12008734  1.30567686
  2.10043668  1.29039301  1.06331878]

shape of covariane matrix (9, 9)

4 [ 7.19502075  6.57261411  6.56016598  5.54771784  5.29875519  7.56431535
  5.97925311  5.86307054  2.58921162]

shape of covariane matrix (9, 9)
```

The mean for each class is calculated based on the formula stated above.

After find the parameter it’s used in membership function to predict the class label. Below is the sample code for membership.

```
y = np.dot((data-mean[i]).transpose(),np.linalg.inv(sigma[i]));       n= np.dot(y,(data-mean[i]))

        x = math.log(prior[i])-(math.log(det))/float(2)- (n)/float(2)
```

c) Prediction over the entire dataset and below is the sample prediction when the prediction is done over entire training dataset,

Predicted Value	True Value
2	[2]
2	[4]
2	[2]
4	[4]
4	[4]
2	[2]
2	[2]
4	[4]
2	[2]
4	[4]

d) Cross Validation 10 fold

Fold 0 Accuracy 0.971428571429
Fold 1 Accuracy 0.971428571429
Fold 2 Accuracy 0.928571428571
Fold 3 Accuracy 0.928571428571
Fold 4 Accuracy 0.957142857143
Fold 5 Accuracy 0.971428571429
Fold 6 Accuracy 0.971428571429
Fold 7 Accuracy 0.957142857143
Fold 8 Accuracy 0.971428571429
Fold 9 Accuracy 0.898550724638
Average Accuracy 0.952712215321

e) Cross Validation 10 fold MSE

Fold 0 Testing Error [0.22857143]
Fold 1 Testing Error [0.22857143]
Fold 2 Testing Error [0.4]
Fold 3 Testing Error [0.11428571]
Fold 4 Testing Error [0.11428571]
Fold 5 Testing Error [0.05714286]
Fold 6 Testing Error [0.]
Fold 7 Testing Error [0.34285714]
Fold 8 Testing Error [0.17142857]
Fold 9 Testing Error [0.28985507]
Average Mean Square Error
Training Error Testing Error
0.172308779368 0.194699792961

f) Evaluation over the entire data set

Confusion Matrix
[[435 23]
[6 235]]
Accuracy 0.958512160229
Precision
[0.98639455782312924, 0.91085271317829453]
Recall
[0.94978165938864634, 0.975103734439834]
F_score
[0.96774193548387089, 0.94188376753507008]

g) Precision through the cross validation

Precision

Class2	Class4
0.977272727273	0.961538461538
1.0	0.923076923077
0.971428571429	0.885714285714
1.0	0.807692307692
0.976744186047	0.925925925926
0.97619047619	0.964285714286
0.978723404255	0.95652173913
1.0	0.88
1.0	0.916666666667
0.978723404255	0.727272727273

Mean Precision
[0.98590828 0.89486948]

h) Recall from Cross Validation

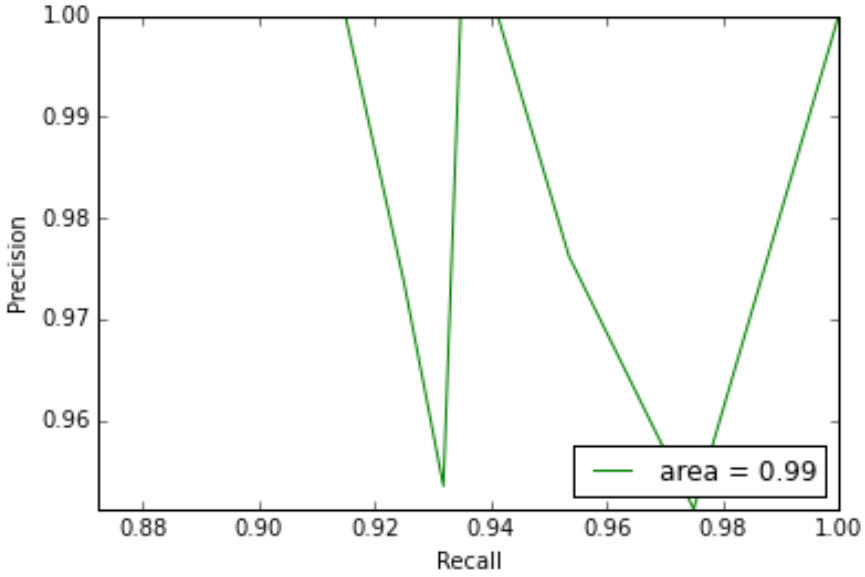
Recall

Class2	Class4
0.977272727273	0.961538461538
0.95652173913	1.0
0.894736842105	0.96875
0.897959183673	1.0
0.954545454545	0.961538461538
0.97619047619	0.964285714286
0.978723404255	0.95652173913
0.9375	1.0
0.958333333333	1.0
0.884615384615	0.941176470588

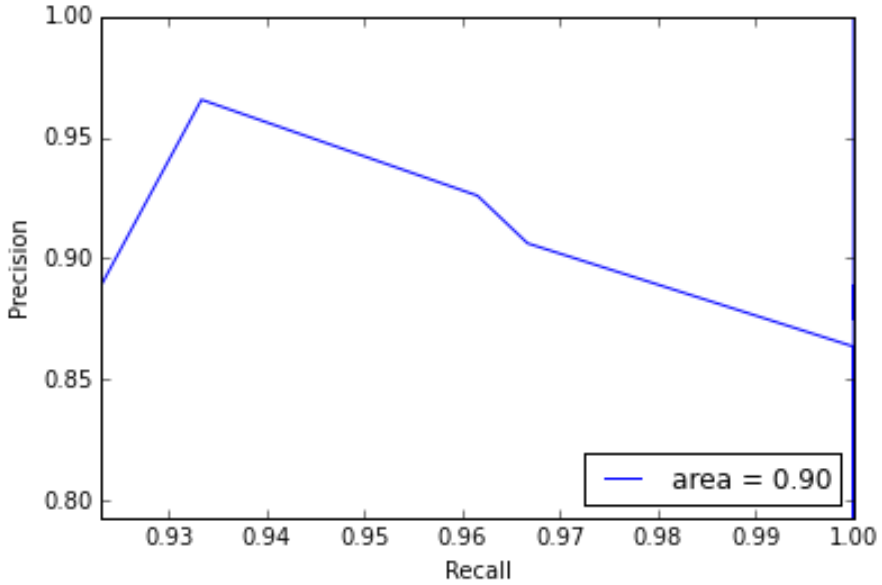
Mean Recall
[0.94163985 0.97538108]

i) Precision Recall curve

Class 2



Class 4



N Dimensional K class Dataset

The data set used is **iris.data.txt**

The classes haven modified as below,

Iris-setosa → 1

Iris-versicolor →2

Iris-virginica →3

a) Find the classes and prior probability

The Classses are [1, 2, 3]
The Classes Count {1: 50, 2: 50, 3: 50}
The prior probabiliy {1: 0.3333333333333333, 2: 0.3333333333333333, 3: 0.3333333333333333}

The classes represent the class labels, the classes count is the number of training data in each class. The prior probability is the probability distribution of each class.

b) The mean for each class,

Mean for mulivariate features

1 [5.006 3.418 1.464 0.244]

shape of covariane matrix (4, 4)

2 [5.936 2.77 4.26 1.326]

shape of covariane matrix (4, 4)

3 [6.588 2.974 5.552 2.026]

shape of covariane matrix (4, 4)

The mean for each class is calculated based on the formula stated above.

c) Prediction over the entire dataset and below is the sample prediction when the prediction is done over entire training dataset,

Predicted Value	True Value
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]
1	[1]

d) Cross Validation 10 fold

Fold 0 Accuracy 0.933333333333
Fold 1 Accuracy 1.0
Fold 2 Accuracy 1.0
Fold 3 Accuracy 0.933333333333
Fold 4 Accuracy 1.0
Fold 5 Accuracy 0.933333333333
Fold 6 Accuracy 1.0
Fold 7 Accuracy 0.933333333333
Fold 8 Accuracy 1.0
Fold 9 Accuracy 1.0
Average Accuracy 0.973333333333

e) Cross Validation 10 fold MSE

Fold 0 Testing Error [0.13333333]
Fold 1 Testing Error [0.]
Fold 2 Testing Error [0.06666667]
Fold 3 Testing Error [0.]
Fold 4 Testing Error [0.]
Fold 5 Testing Error [0.]
Fold 6 Testing Error [0.]
Fold 7 Testing Error [0.]
Fold 8 Testing Error [0.]
Fold 9 Testing Error [0.06666667]
Average Mean Square Error
Training Error Testing Error
0.02 0.0266666666667

f) Evaluation over the entire data set

Confusion Matrix
[[435 23]
[6 235]]
Accuracy 0.958512160229
Precision
[0.98639455782312924, 0.91085271317829453]
Recall
[0.94978165938864634, 0.975103734439834]
F_score
[0.96774193548387089, 0.94188376753507008]

g) Precision through the cross validation

Precision
1.0, 1.0, 0.75
1.0, 1.0, 1.0
1.0, 0.875, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 0.80000000000000004

h) Recall from Cross Validation

Recall
1.0, 0.59999999999999998, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 0.66666666666666663
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 1.0, 1.0
1.0, 0.75, 1.0

5. References

https://en.wikipedia.org/wiki/Generative_model