1D – 2Class Gaussian Discriminant Analysis

1. Problem Statement

The single feature data set should be classified using generative learning model. The data should be trained and tested using the generative learning model. Generative models are used in machine learning for either modeling data directly (i.e., modeling observations drawn from a probability density function), or as an intermediate step to forming a conditional probability density function. 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:

$P(Y = j \mid X) = P(Y=j,X)/P(X)$

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

$P(X|Y=j)$ 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

For one dimension two class data I have used the subset of multidimensional 2 class dataset. The first step is to select the model. The second step is to find the parameters of the mode. The third step is defining membership function and the final step is discriminant function. The discriminant function is to find the class of the given data using the output of the membership function.

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

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,

The Mean for each class	
2	1.44323144105
4	6.5601659751

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

c) The sigma for each class,

The sigma for each class	
2	0.993502221544
4	6.53683648698

The sigma for each class is found from the stated formula.

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

y = ((data-mean[i])**2)*1.0/2*sigma[i]**2 x = math.log(prior[i])-math.log(sigma[i])- y

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

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

e) Cross Validation 10 fold

Fold	**0**	**Accuracy**	**0.842857142857**
Fold	**1**	**Accuracy**	**0.928571428571**
Fold	**2**	**Accuracy**	**0.971428571429**
Fold	**3**	**Accuracy**	**0.914285714286**
Fold	**4**	**Accuracy**	**0.842857142857**
Fold	**5**	**Accuracy**	**0.914285714286**
Fold	**6**	**Accuracy**	**0.928571428571**
Fold	**7**	**Accuracy**	**0.985714285714**
Fold	**8**	**Accuracy**	**0.957142857143**
Fold	**9**	**Accuracy**	**0.942028985507**
Average Accuracy			**0.922774327122**

f) Cross Validation 10 fold MSE

Fold	**0**	**Testing Error**	**[0.62857143]**
Fold	**1**	**Testing Error**	**[0.28571429]**
Fold	**2**	**Testing Error**	**[0.11428571]**
Fold	**3**	**Testing Error**	**[0.34285714]**
Fold	**4**	**Testing Error**	**[0.62857143]**
Fold	**5**	**Testing Error**	**[0.34285714]**
Fold	**6**	**Testing Error**	**[0.28571429]**
Fold	**7**	**Testing Error**	**[0.05714286]**
Fold	**8**	**Testing Error**	**[0.17142857]**
Fold	**9**	**Testing Error**	**[0.23188406]**
Average Mean Square Error			
Training Error		**Testing Error**	
0.309011532541		**0.308902691511**	

g) Evaluation over the entire data set

```
Confusion Matrix
[[436  22]
 [ 32 209]]
Accuracy 0.922746781116
```

	Class 2	Class 4
Precision	0.93162393162393164	0.90476190476190477]
Recall	0.95196506550218341	0.86721991701244816]
F_score	0.94168466522678185	0.88559322033898302]

h) Precision through the cross validation

	Precision	
	Class2	Class 4
Fold 0	0.8	0.9
Fold 1	0.913043478	0.958333333
Fold 2	0.977272727	0.961538462
Fold 3	0.933333333	0.9
Fold 4	0.80952381	0.892857143
Fold 5	0.944444444	0.8125
Fold 6	0.941176471	0.894736842
Fold 7	0.983333333	1
Fold 8	0.959183673	0.952380952
Fold 9	1	0.764705882
Average	0.926131127	0.903705261

i) Recall from Cross Validation

	Recall	
	Class2	Class 4
Fold 0	0.914285714	0.771428571
Fold 1	0.976744186	0.851851852
Fold 2	0.977272727	0.961538462
Fold 3	0.875	0.947368421
Fold 4	0.918918919	0.757575758
Fold 5	0.944444444	0.8125
Fold 6	0.96	0.85
Fold 7	1	0.909090909
Fold 8	0.979166667	0.909090909
Fold 9	0.928571429	1
Average	0.947440409	0.877044488

5. References

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