CS584: Assignment 2

Pooja Mankani
Department of Computer Science
Illinois Institute of Technology
03/20/2016

Abstract

The report illustrates the implementation of generative learning using Gaussian Discriminant Analysis and Naïve Bayes using synthetic and real data sets. The 10 fold cross validation for all the implementations is considered and the training error and testing error is evaluated. A custom model is created and the corresponding python built in model is also created using sklearn package in python. The errors of each of the model is compared and analyzed.

1. **Problem Statement**

   The following problems are considered in the implementation:

   a. **Gaussian Discriminant Analysis:**
      - The GDA can be used for single and multiple feature along with two class and multiple class.
      - The GDA model would not give a good accuracy in case single feature is considered. This is because the data set is a combination of multiple features and in case only a single feature the accuracy of the system will not be as required.
      - The GDA can be used for IRIS and other data sets.

   b. **Naïve Bayes:**
      - The naïve bayes classification can be used for text classification where the words are converted into features.
      - The features that are used can be considered as binary where only the occurrence of word is considered. This is generally used in Bernoulli classification where the features are either 1 or 0 based on the occurrence.
      - The other case of Naïve bayes is when the features is the frequency count of the word in the document. This is used in Binomial classification where the features are the count of the words in each of the document.
      - The feature matrix used in these cases is of the below format:

	Feature1	Feature2	Feature3 …….	Featuren
Doc1				
Doc2				
Doc3				
Doc4				
……				
Docn				

2. **Proposed solution**

   a. **1D 2 class GDA :**

   - The GDA is applied on IRIS dataset for one dimensional feature. The first column is extracted as the data set for one dimensional.
   - Also the only 2 classes are considered rather than 3 classes that is present in the original dataset.
   - The mean, standard deviation, prior probability is calculated.
   - The membership function is calculated using these values. A discriminant function is used to predict the class of the sample based on which class has higher membership function.

```
if membership[0]-membership[1] > 1
        y_predicted = 0
else
        y_predicted = 1
```

   - The accuracy, confusion matrix, precision, recall, F1 measure is calculated

   b. **nD 2 class GDA:**

   - The GDA is applied on IRIS dataset for n dimensional feature. All the columns are used for the classification.
   - In this case only 2 classes are considered and the mean, standard deviation prior probability is calculated. The membership function is calculated and the discriminant function is used to predict the class of the sample.
   - The Precision v/s Recall curve is plotted and the area under the curve is calculated. The GDA gives 100% accuracy when 2 class n dimensional data is considered.
   - The accuracy, confusion matrix, precision, recall, F1 measure is calculated

   c. **nD n class GDA:**

   - The GDA is applied on the IRIS dataset for n dimensional and n classes. The entire dataset is considered for this classification.
   - The membership function is calculated and the discriminant function predicts the classification of the sample.
   - The accuracy, confusion matrix, precision, recall, F1 measure is calculated

   d. **Naïve Bayes with Bernoulli features:**

   - In this case IMDB database (https://www.dropbox.com/s/xk4glpk61q3qrg2/imdb.tgz?dl=1) is used to perform the classification.
   - The dataset has two classes positive and negative and is divided on training and testing data.
   - The alpha is calculated using 10 fold cross validation on the training data and an optimal alpha is retrieved. This is then used on the testing data and values are predicted.
   - The accuracy, confusion matrix, precision, recall, F1 measure is calculated across the folds and the average mean of these are also calculated for the model analysis.
   - Since Bernoulli features is taken into consideration the features are converted into binary values where 1 denotes the word occurrence and 0 denotes the word does not exist. This feature matrix is then used for the classification.

e. Naïve Bayes with Binomial features:

- The same IMDB dataset is used however and the prediction is made based on the alpha class, prior probability. The membership function is calculated and the corresponding class values are predicted based on the discriminant function.
- Since binomial features are considered the frequency of the word is taken into consideration rather than the occurrence of the word.

f. Comparison and Analysis of results:

- The accuracy of the custom and the built in model is compared and the analyzed.
- Also the accuracy across the models is compared with each other and the better model that fits the dataset concluded.

3. Implementation Details:

a. **Design issues:** The following design issues were witnessed:

- Selecting a dataset that would fit correctly in the model was one of the design issues. This is because the accuracy of the model would be hampered in case the most prominent or most important feature is not considered for the classification.
- In case n-class GDA the calculation of standard deviation is difficult in case then number of classes are high. If there are more number of classes the standard deviation becomes n*n hence this would be a huge matrix.
- The prior probability of the samples used is evenly split across the dataset used. This is an ideal scenario. However, in case the real world actual dataset is used the prior probability is may vary and the results will vary as required.
- The conversion of the text documents to feature matrix was one of the design issues faced while performing the analysis.

b. **Problems faced:** The following issues were faced:

- Cross validation: The implementation of cross validation to retrieve the training and testing indices was complex to obtain using python array and matrix manipulation. Hence in this case KFold cross validation of sklearn package was used to evaluate the same.
- Feature matrix: The conversion of text documents to feature matrix was complex using python array and matrix multiplication. Hence in this case the Count_Vectorizer package is used to get the features from the documents.
- The epsilon value used for smoothing in the naïve bayes classification was difficult to calculate. Hence in this case multiple results were considered and the most optimal value for epsilon was picked.

c. **Implementation Details:**

Python Notebook	Analysis	Command
Assignment_2.ipynb	1D 2-class GDA	input_data=read_input('iris.dat') X=create_feature_matrix(input_data) Y=create_y_matrix(input_data) do_cross_validation(X, Y,1, [1,2], n_folds=10, verbose=True))
Assignment_2.ipynb	nD 2-class GDA	input_data=read_input('iris.dat') X=create_feature_matrix(input_data) Y=create_y_matrix(input_data) do_cross_validation(X, Y,4, [1,2], n_folds=10, verbose=True))
Assignment_2.ipynb	nD n-class GDA	input_data=read_input('iris.dat') X=create_feature_matrix(input_data) Y=create_y_matrix(input_data) do_cross_validation(X, Y,4, [1,2,3], n_folds=10, verbose=True))
Assignment_2.ipynb	Naïve Bayes with Bernoulli Features	get_data() path = 'data' pos_train_files = get_files(path + os.sep + 'train' + os.sep + 'pos') neg_train_files = get_files(path + os.sep + 'train' + os.sep + 'neg') all_train_files = pos_train_files + neg_train_files pos_test_files = get_files(path + os.sep + 'test' + os.sep + 'pos') neg_test_files = get_files(path + os.sep + 'test' + os.sep + 'neg') all_test_files = pos_test_files + neg_test_files X_test = vec.transform(all_test_files) y_test = np.array([1] * len(pos_test_files) + [0] * len(neg_test_files)) matrix, vec = do_vectorize(all_train_files, binary=False) print("Bernoulli Feature") accuracy, precision, recall, f_1_score, acc1, c_matrix1, precision1, recall1, f_score1, y_predicted = do_cross_validation_NB(matrix, labels, X_test, y_test,'bernoulli', n_folds=10)
Assignment_1_Gausian_Kernel.ipynb		get_data() path = 'data' pos_train_files = get_files(path + os.sep + 'train' + os.sep + 'pos') neg_train_files = get_files(path + os.sep + 'train' + os.sep + 'neg') all_train_files = pos_train_files + neg_train_files pos_test_files = get_files(path + os.sep + 'test' + os.sep + 'pos') neg_test_files = get_files(path + os.sep + 'test' + os.sep + 'neg') all_test_files = pos_test_files + neg_test_files X_test = vec.transform(all_test_files) y_test = np.array([1] * len(pos_test_files) + [0] * len(neg_test_files)) matrix, vec = do_vectorize(all_train_files, binary=False) print("Binomial Feature") accuracy, precision, recall, f_1_score, acc1, c_matrix1, precision1, recall1, f_score1, y_predicted = do_cross_validation_NB(matrix, labels, X_test, y_test,'binomial', n_folds=10)

4. Results and Discussions:
   a. **1D 2 class GDA:**
      - **Training Parameters calculated**

Parameter	**Class 1**	**Class 2**
Mean	5.006	5.936
Standard Deviation	0.121764	0.261104

      - **Overall Parameters computed:**

Parameter	**Result**
Accuracy	0.86
Precision	0.8103
Recall	0.94
F1 Measure	0.870

        Confusion Matrix: [[47 3]
        [11 39]]
      - **Average Parameters Computed after 10 folds:**

Parameter	**Result**
Accuracy	0.86
Precision	0.8098
Recall	0.9333
F1 Measure	0.856

b. nD 2 class GDA:

- **Training Parameters calculated**

Parameter	Feature	Class 1	Class 2
Mean	Feature 1	5.006	5.936
	Feature 2	3.418	2.77
	Feature 3	1.464	4.26
	Feature 4	0.244	1.326
Standard Deviation	Feature 1	0.121764	0.261104
	Feature 2	0.098292	0.08348
	Feature 3	0.015816	0.17924
	Feature 4	0.010336	0.054664

- **Overall Parameters computed:**

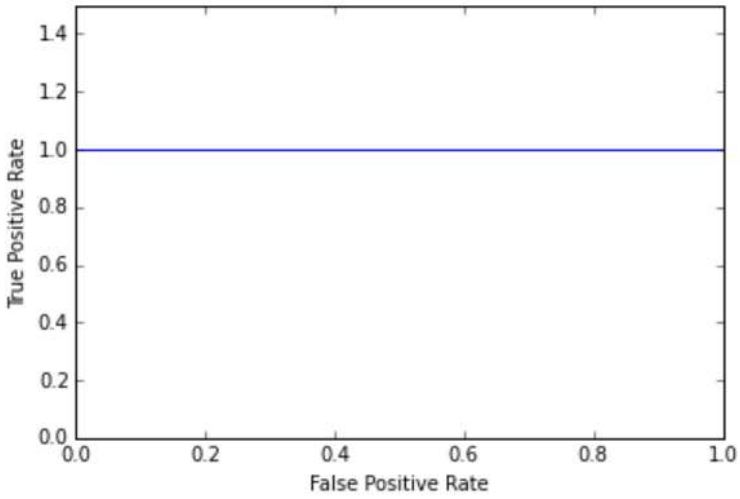
Parameter	Result
Accuracy	1
Precision	1
Recall	1
F1 Measure	1

Confusion Matrix: [[50 0]
[0 50]]

- **Average Parameters Computed after 10 folds:**

Parameter	Result
Accuracy	1
Precision	1
Recall	1
F1 Measure	1

- **Precision v/s Recall Curve**



- **Area under the curve: 1**
- **Accuracy obtained 100%**

c. nD n class GDA:

- **Training parameters calculated:**

Parameter	Feature	Class 1	Class 2	Class 3
	Feature 1	5.006	5.936	6.588
	Feature 2	3.418	2.77	2.974
	Feature 3	1.464	4.26	5.552
Mean	Feature 4	0.244	1.326	2.026

- **Sigma:**

Class 1: [[0.121764 0.098292 0.015816 0.010336]
[0.098292 0.142276 0.011448 0.011208]
[0.015816 0.011448 0.029504 0.005584]
[0.010336 0.011208 0.005584 0.011264]]

Class 2: [[0.261104 0.08348 0.17924 0.054664]
[0.08348 0.0965 0.081 0.04038]
[0.17924 0.081 0.2164 0.07164]
[0.054664 0.04038 0.07164 0.038324]]

Class 3: [[0.396256 0.091888 0.297224 0.048112]
[0.091888 0.101924 0.069952 0.046676]
[0.297224 0.069952 0.298496 0.047848]
[0.048112 0.046676 0.047848 0.073924]]

- **Overall Parameters computed:**

Parameter	Result
Accuracy	0.9866
Precision	0.9866
Recall	0.9866
F1 Measure	0.9866

Confusion Matrix: [[50 0 1]
[0 49 1]
[0 1 49]]

- **Average Parameters Computed after 10 folds:**

Parameter	Result
Accuracy	0.9733
Precision	0.9771
Recall	0.9733
F1 Measure	0.9722

Conclusion:

- **For IRIS dataset 1D 2class gives bad performance since the feature taken into consideration is not a prominent or dependent one.**
- **However, for nD 2 class we get 100% accuracy which is the best performance since all the features are taken into consideration. The area under the precision and recall curve is 1 since the accuracy is 100%.**
- **For nD n class we get an accuracy of 97% with a few errors in the prediction.**

d. Naïve Bayes with Bernoulli features:

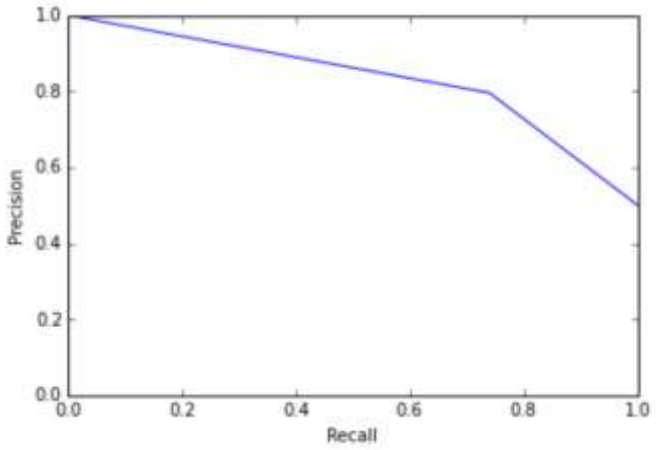
- **Average Parameters computed with 10 folds:**

Parameter	Result
Accuracy	0.745
Precision	0.8551
Recall	0.6657
F1 Measure	0.6894

- **Overall Parameters computed:**

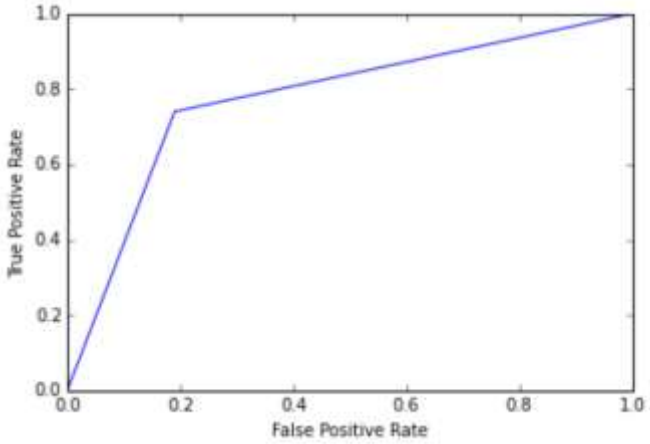
Parameter	Result
Accuracy	0.775
Precision	0.7957
Recall	0.74
F1 Measure	0.7668

- **Confusion matrix:** [[162 38]
  [52 148]]
- **Precision Recall Curve:**



Area under the PR curve: 0.8328

- **ROC Curve**



Area under the ROC curve: 0.7750

e. **Naïve Bayes with Binomial features:**

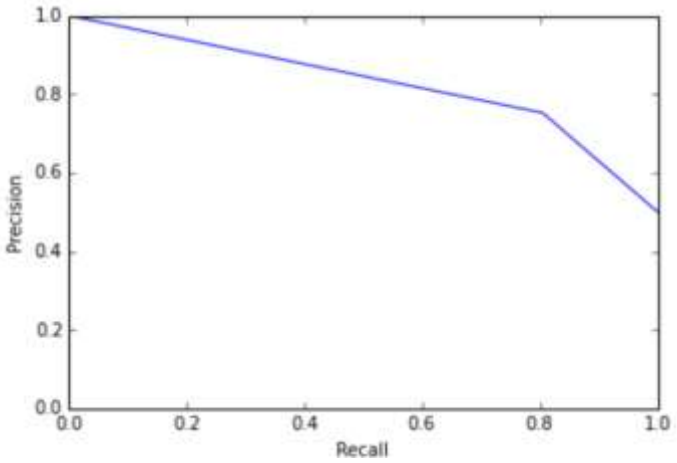
- **Average Parameters computed with 10 folds:**

Parameter	Result
Accuracy	0.7525
Precision	0.7669
Recall	0.7707
F1 Measure	0.7436

- **Overall Parameters computed:**

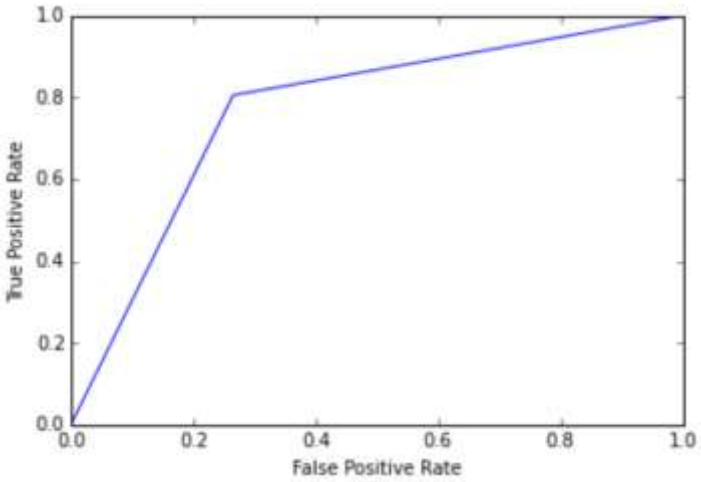
Parameter	Result
Accuracy	0.770
Precision	0.7523
Recall	0.8050
F1 Measure	0.7778

- **Confusion matrix:** [[147 53]
  [39 161]]
- **Precision Recall Curve:**



Area under the PR curve: 0.8274

- **ROC Curve:**



Area under the PR curve: 0.7700

Conclusion:

Custom v/s built in model

Naïve Bayes with Bernoulli Feature			
Custom Model		**Built-in Model**	
Accuracy	0.775	Accuracy	0.775
Naïve Bayes with Binomial Feature			
Custom Model		**Built-in Model**	
Accuracy	0.770	Accuracy	0.775

The accuracy is similar to that of the built in model for this dataset. Moreover, in case of this data set the Bernoulli feature gives better accuracy where the occurrence of the word is taken into consideration and the binomial feature gives slightly lesser accuracy. Hence the occurrence of the word works better than the frequency count with respect to this dataset.

Data Set:

GDA Analysis: IRIS dataset: http://archive.ics.uci.edu/ml/datasets/Iris

Naïve Bayes: IMDB Dataset: https://www.dropbox.com/s/xk4glpk61q3qrg2/imdb.tgz?dl=1

References:

http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#example-model-selection-plot-roc-py

https://github.com/poojamankani/CS579-Online-Social-Network-Analysis/blob/master/asg/a3/a3.ipynb

http://stats.stackexchange.com/questions/7207/roc-vs-precision-and-recall-curves

http://pages.cs.wisc.edu/~jdavis/davisgoadrichcamera2.pdf