CS 584 Machine Learning
Assignment 02 Report
Haripriya Pandya

Part 1: 1-d 2-class Gaussian Discriminant Analysis

Problem Statement:
This is a type of machine learning algorithm involving generative learning where the assumed distribution of data is of type Gaussian. In this particular instance it only involves one feature per example.

Proposed Solution:
In order to conduct the analysis one needs to create a generative model first, for which parameters are computed first. Calculating the mean and sigma for this distribution would provide parameters for creating a model. By inputing those parameters into a function called 'membership function/likelihood function' gives the probability of a certain example for belonging to a certain class. After such calculation we can use discriminating function to predict the class with highest likelihood value.

Implementation details:
For this analysis I used the **Iris data se**t and preprocessed the dataset appropriately to make it 1-d 2class distribution. By following the mathematical formula of this type of distribution I calculated mean and sigma calculating functions. I used the similar technique to compute the likelihood function. The function **"oneD_2class_GDA"** performs the actual function of prediction, which is also called by the **"cross_validation1D"** function which performs the k-fold cross validation at the end. For this entire implementation I have mainly used the inbuilt python function and used the sklearn library to compute accuracy, recall, precision, confusion matrix and f-measure.

Result and discussions:
At the end of this analysis I was able to achieve 100% accuracy. I attribute a fair percent of accuracy to the dataset itself. Using the description available of the dataset, I chose the feature values that were the highest indicators of the class value. In this instance it shows the perfect accuracy for classification purposes.

References:

Part 2: nD 2-class Gaussian Discriminant Analysis

Problem Statement
This is a type of machine learning algorithm involving generative learning where the assumed distribution of data is of type Gaussian. In this particular instance it involves multiple feature values per example.

Proposed Solution
In order to conduct the analysis one needs to create a generative model first, for which parameters are computed first. This problem differs from the first, in that one needs to calculte

the mean vector and covariance instead of mean and sigma, for this distribution, which would provide parameters for creating a model. By inputing those parameters into a function called 'membership function/likelihood function' gives the probability of a certain example for belonging to a certain class. After such calculation we can use discriminating function to predict the class with highest likelihood value.

Implementation details
For this analysis I used the **Iris data se**t and preprocessed the dataset appropriately to make it n-d 2class distribution. By following the mathematical formula of this type of distribution I calculated mean vector and covariance calculating functions. I used the similar technique to compute the likelihood function. The function **"nd_experiment"** performs the actual function of prediction, which is also called by the **"cross_validation_nd_2class"** function which performs the k-fold cross validation at the end. For this entire implementation I have mainly used the inbuilt python function and used the sklearn library to compute accuracy, recall, precision, confusion matrix and f-measure.

Result and discussions
At the end of this analysis I was able to achieve 97.00% accuracy. I attribute a fair percent of accuracy to the dataset itself. Using the description available of the dataset, I chose the feature values that were the highest indicators of the class value.

Part 3: nD k-class Gaussian Discriminant Analysis

Problem Statement
This is a type of machine learning algorithm involving generative learning where the assumed distribution of data is of type Gaussian. In this particular instance it involves multiple feature values per example with multiple class belonging.

Proposed Solution
In order to conduct the analysis one needs to create a generative model first, for which parameters are computed first. This problem differs from the first, in that one needs to calculate the mean vector and covariance instead of mean and sigma, for this distribution, which would provide parameters for creating a model. By inputing those parameters into a function called 'membership function/likelihood function' gives the probability of a certain example for belonging to a certain class. After such calculation we can use discriminating function to predict the class with highest likelihood value.

Implementation details
For this analysis I used the **Iris data se**t and preprocessed the dataset appropriately to make it n-d 2class distribution. By following the mathematical formula of this type of distribution I calculated mean vector and covariance calculating functions. I used the similar technique to compute the likelihood function. The function **"nd_experiment"** performs the actual function of prediction, which is also called by the **"cross_validation_nd_kclass"** function which performs the k-fold cross validation at the end. For this entire implementation I have mainly used the inbuilt python function and used the sklearn library to compute accuracy, recall, precision, confusion matrix and f-measure.

Result and discussions:

At the end of this analysis I was able to achieve 95.33% accuracy. I attribute a fair percent of accuracy to the dataset itself. Using the description available of the dataset, I chose the feature values that were the highest indicators of the class value. However I also observe that with increase in feature values, the model parameters have changed and that has showed impact on the accuracy.

Part 4: Naive Bayes with Bernoulli features

Problem Statement
This is a type of machine learning algorithm involving generative learning where the assumed distribution of data is Naive Bayes, as it includes the assumption in calculation of the likelihood function based on Naive Bayes rule. In this particular instance it involves multiple feature values of Bernoulli type, which are boolean of nature.

Proposed Solution
In order to conduct the analysis one needs to create a generative model first, for which parameters are computed first. By inputing those parameters into a function called 'membership function/likelihood function' gives the probability of a certain example for belonging to a certain class. After such calculation we can use discriminating function to predict the class with highest likelihood value.

Implementation details
For this analysis I used the **Spambase data set** and preprocessed the dataset appropriately to make the features boolean instead of various decimals. By following the mathematical formula of this type of distribution I calculated prior class probability and the relative alpha values calculating functions. I used the similar technique to compute the membership function. The function **"NB_Bern_exp"** performs the actual function of prediction, which is also called by the **"cross_validation_NBBer"** function which performs the k-fold cross validation at the end. For this entire implementation I have mainly used the inbuilt python function and used the sklearn library to compute accuracy, recall, precision, confusion matrix and f-measure.

Result and discussions
At the end of this analysis I was able to achieve 87.04% accuracy. I attribute the lack of percent of accuracy to the dataset itself. In case of Bernoulli features we do not consider the values of feature but we neutralize it to either 0 or 1, hence it devalues the features and model is trained such way. However I also observe that with increase in feature values, the model parameters have changed and that has showed impact on the accuracy.

Part 5: Naive Bayes with Binomial features

Problem Statement
This is a type of machine learning algorithm involving generative learning where the assumed distribution of data is Naive Bayes, as it includes the assumption in calculation of the likelihood function based on Naive Bayes rule. In this particular instance it involves multiple feature values of Binomial type, which does not neutralize the values into boolean values.

Proposed Solution

In order to conduct the analysis one needs to create a generative model first, for which parameters are computed first. By inputing those parameters into a function called ‘membership function/likelihood function’ gives the probability of a certain example for belonging to a certain class. After such calculation we can use discriminating function to predict the class with highest likelihood value.

Implementation details

For this analysis I used the **Spambase data set** and preprocessed the dataset appropriately to my analysis and appropriately changed the document length of each document. By following the mathematical formula of this type of distribution I calculated prior class probability and the relative alpha values calculating functions. I used the similar technique to compute the membership function. The function **NB_Bino_exp** performs the actual function of prediction, which is also called by the **“cross_validation_NBBi”** function which performs the k-fold cross validation at the end. For this entire implementation I have mainly used the inbuilt python function and used the sklearn library to compute accuracy, recall, precision, confusion matrix and f-measure.

Result and discussions

At the end of this analysis I was able to achieve 82.54% accuracy. I have observed that in this specific case the length of each document has showed a great effect on the accuracy. I have added a constant value to the final length of a document. With higher values of constant the accuracy seem to have decreased and vice versa. However with very low values have also had slightly decreasing effect on accuracy. That shows the overall effect of term frequency on the accuracy.