1. Problem Statement

1.1 Gaussian discriminant analysis

In this question:

1.1.1 First it needs to select suitable data from UCI machine Learning database.

1.1.2 Pick data iris and it could be used as 2-class (combine 2 classes into 1 class) or n-class.

1.1.3 Applying python build in method to achieve gaussian function.

1.1.4 Applying python to calculate the model parameter also known as learning procedure.

1.1.5 Applying the learnt model to classify the test data and calculate the performance of the model.

1.2 Naïve Bayes

In this question:

1.2.1 First it needs to select suitable data from UCI machine Learning database.

1.2.2 Pick data SMS spam data and it could be used as 2-class (spam or ham).

1.2.3 Calculate the possibility of every individual text or word its possibility of log likelihood and use this likely hood to estimate every other word's possibility and the word form a message, calculate the possibility of the message to be a spam message.

2 Proposed solution

2.1 If the random variable X obeys a positional parameter of μ and the scale parameter is a normal distribution of σ, it is recorded as:

$$X \sim N(\mu, \sigma^2)$$

Then its probability density function is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The mathematical expectation or expectation of the normal distribution μ is equal to the positional parameter, which determines the position of the distribution; the square root or standard deviation σ of the variance σ^2 is equal to the scale parameter, which determines the magnitude of the distribution. [1]

$$p(y) = \phi^y(1-\phi)^{1-y}$$
$$p(x|y=0) = \frac{1}{2\pi^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right)$$
$$p(x|y=1) = \frac{1}{2\pi^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)$$

Maximum likelihood estimation of four parameters: φ, μ

$$\begin{aligned}\ell &= log\prod_{i=1}^{m} p(x^{(i)}, y^{(i)})\\ &= log\prod_{i=1}^{m} p(x^{(i)}|y^{(i)})p(y^{(i)})\\ &= \sum_{i=1}^{m}\Bigg[y^{(i)}log\phi + (1-y^{(i)})log(1-\phi)\\ &\quad -\frac{1}{2}(x-\mu_{y^{(i)}})^T\Sigma^{-1}(x-\mu_{y^{(i)}}) + log\frac{1}{2\pi^{n/2}|\Sigma|^{1/2}}\Bigg]\end{aligned}$$

$$\nabla_\phi \ell = \sum_{i=1}^{m}\left(y^{(i)}\frac{1}{\phi} - (1-y^{(i)})\frac{1}{1-\phi}\right)$$

$$0 = \sum_{i=0}^{m} y^{(i)} - \sum_{i=0}^{m}\phi$$

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

$$\begin{aligned}\nabla_{\mu_0}\ell &= \nabla_{\mu_0}\sum_{i=1}^{m}\left(\left[-\frac{1}{2}(x^{(i)}-\mu_0)^T\Sigma^{-1}(x^{(i)}-\mu_0)\right]1\{y^{(i)}=0\}\right)\\ &= \sum_{i=1}^{m}\left(\left[\frac{1}{2}\Sigma^{-1}(x^{(i)}-\mu_0)\right]1\{y^{(i)}=0\}\right)\end{aligned}$$

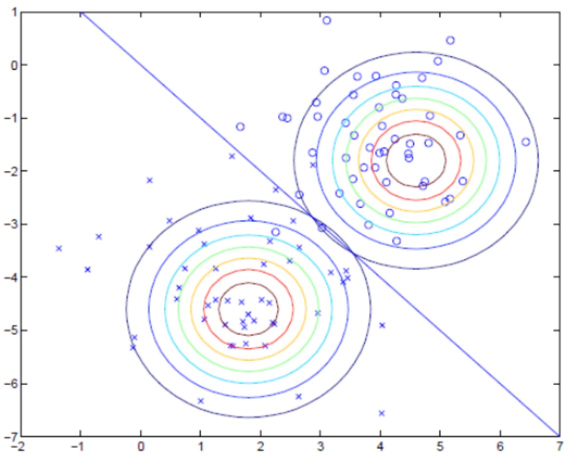
$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}}$$

$$\begin{aligned}\nabla_\Sigma\ell &= \nabla_\Sigma\sum_{i=1}^{m}\left(-\frac{1}{2}(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-1}(x^{(i)}-\mu_{y^{(i)}}) + log\frac{1}{2\pi^{n/2}|\Sigma|^{1/2}}\right)\\ &= \sum_{i=1}^{m}\left((x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T\Sigma^{-2} - \Sigma^{-1}\right)\end{aligned}$$

$$\sum_{i=1}^{m}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T = m\Sigma$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T$$

So far, we have obtained an estimate of the parameters φ, μ0, μ1, Σ. So, we can predict new data by argmax p(x|y)p(y). [2]



2.2 Bayes' theorem is stated mathematically as the following equation:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Where P(A|B) is the probability that event A will occur in the event that event B occurs.

In Bayes' theorem, each noun has a customary name:

P(A|B) is the conditional probability that A is known to occur after B, and is also called the posterior probability of A due to the value obtained from B.

P(A) is the prior probability (or edge probability) of A. It is called "a priori" because it does not consider any B factor.

P(B|A) is the conditional probability that B is known to occur after A, and is also called the posterior probability of B due to the value obtained from A.

P(B) is the prior probability or edge probability of B.

According to these terms, Bayes' theorem can be expressed as: posterior probability = (likelihood * prior probability) / normalized constant, that is, the posterior probability is proportional to the product of prior probability and similarity. In addition, the ratio P(B|A)/P(B) is sometimes referred to as standardized likelihood, and the Bayesian theorem can be expressed as: posterior probability = standard likelihood * prior probability

That is, we have m samples, each sample has n features, and the feature output has K categories, defined as C1, C2, ..., CK. From the sample we can learn the a priori distribution P of naive Bayes (Y=Ck) (k=1, 2, ... K), and then learn the conditional probability distribution P(X=x|Y=Ck)=P(X1=x1, X2=x2,...Xn=xn |Y=Ck), then we can use the Bayesian formula to get the joint distribution P(X,Y) of X and Y. The joint distribution P(X, Y) is defined as:

P(X, Y=Ck)=P(Y=Ck)P(X=x|Y=Ck)=P(Y=Ck)P(X1=x1, X2=x2,...Xn=xn|Y =Ck)(1)(2)

It can be seen from the above equation that P(Y=Ck) is relatively easy to find by the maximum likelihood method, and the obtained P(Y=Ck) is the frequency of the category Ck appearing in the training set. But P(X1=x1, X2=x2,...Xn=xn|Y=Ck) is difficult to find, which is a super-complex conditional distribution with n dimensions. The naive Bayesian model makes a bold assumption here that the n dimensions of X are independent of each other, so that:

P(X1=x1,X2=x2,...Xn=xn|Y=Ck)=P(X1=x1|Y=Ck)P(X2=x2|Y=Ck)...P(Xn=xn |Y=Ck)

3 Implementation details

3.1 Gaussian Discriminate analysis

Select a dataset form UCI machine learning database, which data should be a classification dataset, and it's better to have more than one feature and more than one class. Turn the class label to the number which also represent the dataset. And build the dataset which could be used in training and testing. Using mathematical method to calculate the Gaussian parameter Φ, μ0, μ1, Σ. Then using the gaussian function with the parameter to apply on the test data which would give the 'possibility' of the test data belong to the 2 or more class and then build a function g(x) to decide which of the class the test set should belong. Calculating the confusion matrix and get the precision, recall, accuracy and F-measure.

The difference between n dimension to 1 dimension is the change in the coefficient index and the increase in the dimension of the sigma matrix, which cause the discriminant function to become complex.

Problem:

Mathematical theory is relatively simple to implement, but the dimensions of processing data when using Python will be different.

The main problem is that python is not skilled.

3.2 Naïve Bayes

Select data from dataset SMS spam dataset.

Load the dataset and create a dataset of the term segmentation set and the class tag. Create a thesaurus based on the sample. The occurrence of each sample in the thesaurus is counted, and a single sample is mapped into the thesaurus. The occurrence of a single sample in the thesaurus is counted, with 1 indicating occurrence and 0 indicating no occurrence. Calculate conditional probability and class label probability. The Bayesian classification algorithm is trained to calculate the four probability distributions required. Use test data to achieve classification.

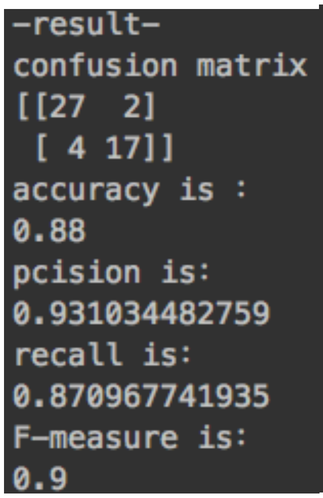
Problem:

Data is difficult to use because data literal data needs to be split into individual words, and words are combined to form an overall frequency of occurrence. However, the data contains many special symbols and other languages, making it difficult to segment the data. However, if you build your own data (dividing the entire sentence into a single vocabulary), you can achieve good discrimination.

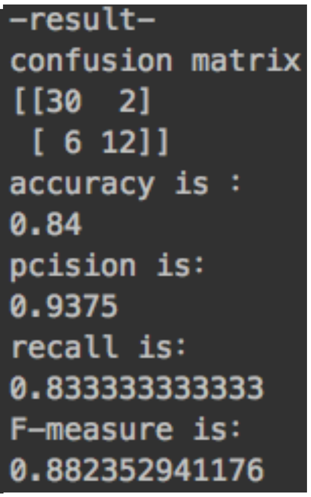
The main problem is that python is not skilled

4 Result and discussion

4.1 1D-2class Gaussian discriminant analysis



```
-result-
confusion matrix
[[27  2]
 [ 4 17]]
accuracy is :
0.88
pcision is:
0.931034482759
recall is:
0.870967741935
F-measure is:
0.9
```

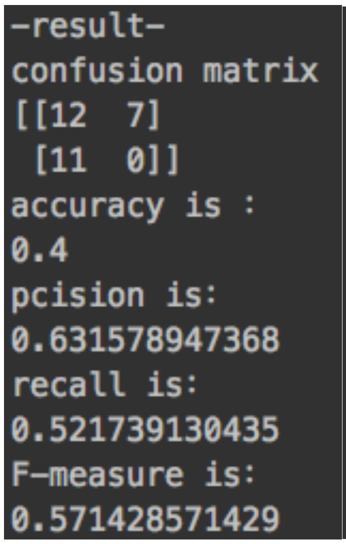


```
-result-
confusion matrix
[[30  2]
 [ 6 12]]
accuracy is :
0.84
pcision is:
0.9375
recall is:
0.833333333333
F-measure is:
0.882352941176
```

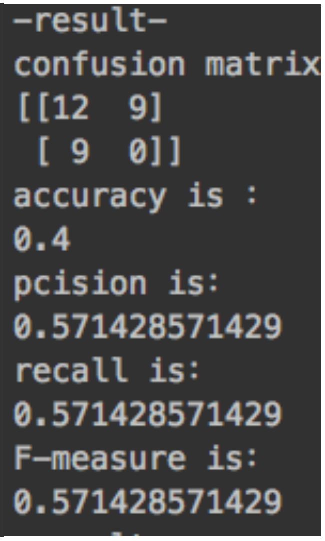
4.2 nD-2class Gaussian discriminant analysis

something wrong with my matrix but it could be implemented in c++.

4.3 nD-kclass Gaussian discriminant analysis



```
-result-
confusion matrix
[[12  7]
 [11  0]]
accuracy is :
0.4
pcision is:
0.631578947368
recall is:
0.521739130435
F-measure is:
0.571428571429
```

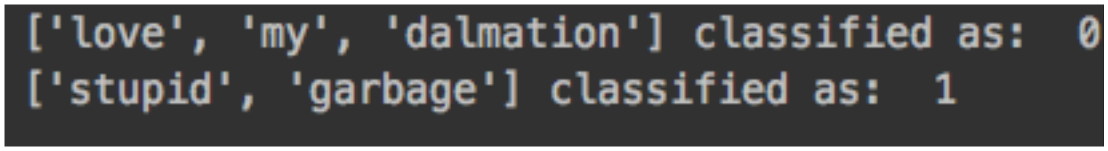


```
-result-
confusion matrix
[[12  9]
 [ 9  0]]
accuracy is :
0.4
pcision is:
0.571428571429
recall is:
0.571428571429
F-measure is:
0.571428571429
```

The classification result of the classifier is not very good.

4.4 Naïve Bayes with Bernoulli features

It's hard to split data to fit the model but it works if I build a smaller dataset.



```
['love', 'my', 'dalmation'] classified as:  0
['stupid', 'garbage'] classified as:  1
```

4.5 Naïve Bayes with Binomial features

5 Reference

[1] Wikipedia. Normal distribution [EB/OL]. https://en.wikipedia.org/wiki/Normal_distribution

[2] CSDN. Gaussian Discriminant Analysis

[EB/OL]. https://blog.csdn.net/sz464759898/article/details/44342923