Anshul Shivhare MTech AI 17828

Pattern Recognition and Neural Networks

Homework 1

Problem 1

This is a 2-class classification problem with two dimensional feature vector and class conditional densities are normal and we have 3 data set for problem one, and all 3 data set have different class 2 means. and other parameters are same for all.
NOTE: Naive Bayes algorithm is based on the naive assumption which is conditional independence of every pair of feature vector given the value of class variable.

Sub-problem 1

For estimating each class conditional density, we have taken 5, 10, 25, 75 examples by radomly sampling from the given training data. For each case, compare the accuracy of the Bayes classifier with that of nearest neighbour classifier. I assumed that the prior probabilities are equal and then calculated the parameters from different samplings. and then calculated posterior densities and implemented bayes classifier. then from the same samples implemented nearest neighbour classifier.

Table for sub problem 1 below shows the accuracy for both classifier for both cases. and by below table we can say that Bayes classifier outperforms nearest neighbour classifier(which is obvious as it is optimal).

The table described below is the result of training the model on given set of examples and the testing it on the entire test data (These results are averaged on 500 samples)

We can see that as number of training data increases the result of both classifier increases.that is greater number of samples leads to better estimation of parameters.

Number of examples in training data	Accuracy of Bayes Classifier	Accuracy of Neareresst Neighbour Classifier
5	0.5499	0.6121
10	0.6378	0.6301
25	0.71	0.662
75	0.731	0.670
200	0.7421	0.6899

For data in P1a_traindata_2D.txt and P1a_train_data_2D.txt

Estimated parameters converge in probability to true parameters when the number of samples is sufficiently large.
Also we are getting less accuracy because both of the classes mean are not much different.

Number of examples in training data	Bayes Classifier	Neareresst Neighbour Classifier
5	0.6888	0.871
10	0.876	0.911
25	0.952	0.931
75	0.967	0.948
200	0.97	0.95

For data in P1b_traindata_2D.txt and P1b_train_data_2D.txt

Number of examples in training data	Accuracy of Bayes Classifier	Accuracy of Neareresst Neighbour Classifier
5	0.756	0.702
10	0.7872	0.699
25	0.884	0.818
75	0.990	0.984
200	0.995	0.985

As the synethetic data is drawn from normal distribution with mean

$$\mu_1=[0\ \ 0]^T$$

$$\Sigma_1= \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\Sigma_2= \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

μ_2 changes for all 3 data sets, in first case $\mu_2 = [1\ \ 1]^T$, in second case $\mu_2 = [3\ \ 3]^T$ and in third case $\mu_2 = [3\ \ 6]^T$. So the second density is being moved further and further. Hence probability of correctly classifying will increase, So Bayes classifier performs better in the second and third case. Also as we increase number of examples the model also perform better which is intuitively clear cause on increasing the number the hyperparameter of the model become more and more accurate.

In case the nearest neighbour I used *KneighborsCalssifier* from the sklearn library which sees only one nearest neighbour. The distance is taken to be the eucledian distance. As the number of examples increases the accuracy of the nearest neighbour also increases.

Sub-problem 2

Number of examples in training data	For Bayes Classifier
5	0.667
10	0.83
25	0.92
75	0.9599
100	0.972

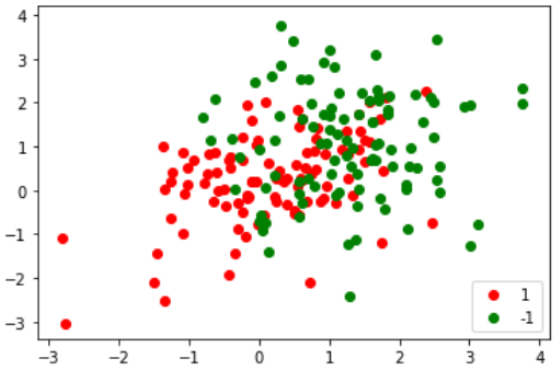
For data in P1b_traindata_2D.txt and P1b_train_data_2D.txt

In second part, Y_train is discarded (removing class label),thus unlabelled data is processed using GuassianMixture model(used sklearn library). it uses EM algorithm to predict the best parameters for the respective densities. after estimating mean and variance from sample train data and using priors, the bayes classifier is implemented. EM algorithm return 3 parameters weights ,mean and variance. weights are used to assign prior probability.

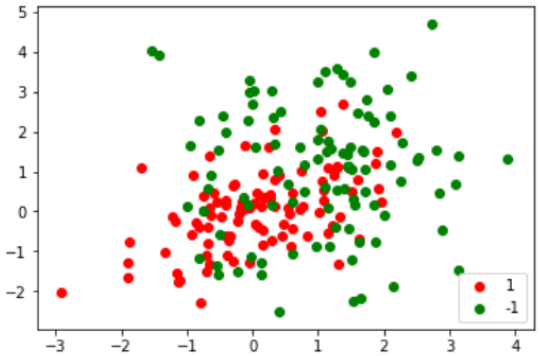
By varying the number of examples the table shows above the accuracy obtained of test cases. Here also as the number of training data increases the accuracy increases.

As here means are far apart(i.e m1=[0,0] and m2=[3,3]), hence the classification accuracy is better. also the mean and covariance matrix return by EM algorithm is approximately equal to actual mean and covariance of data.

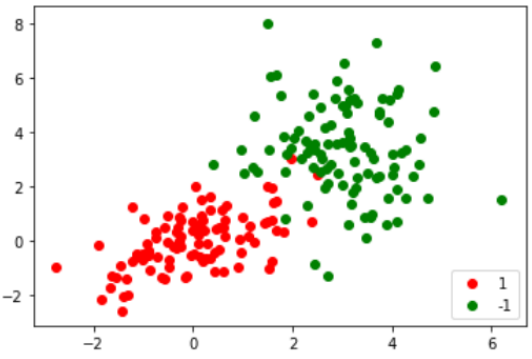
Scatter Plot of train dataset a



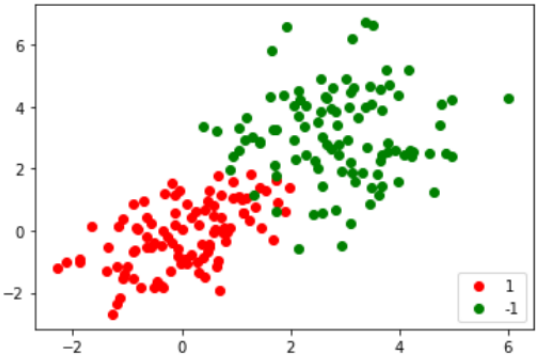
Scatter Plot test dataset a



Scatter Plot of train dataset b



Scatter Plot of test dataset b



Sub-problem 3

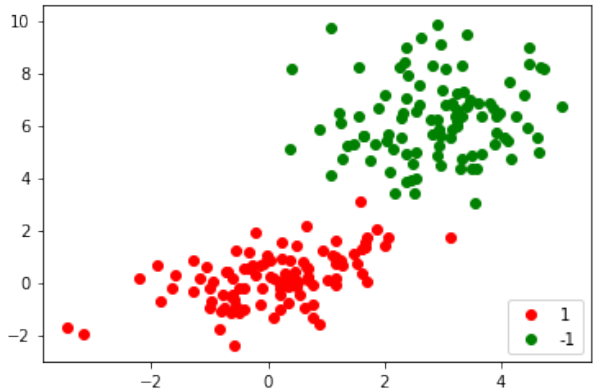
Here implemented 2 bayes classifier, for first bayes classifier taking both class conditional densities as normal and for second bayes classifier taking one as normal and other class conditional density as exponential.
For exponential conditional density the parameter is lamda, which is inverse of class mean.
here prior probabilities for both class in bayes classifier is taken as equal(i.e 0.5).
Hence using all these the posterior is calculated and bayes classifier is implemented.
i.e using the test class posterior is calculated and classifier predicts the class with maximum posterior probability.

The table shows more accuracy for the both densities coming from Guassian distribution rather than one exponential and one Guassian. The was expected as we knew that the examples drawn are from Gaussian.

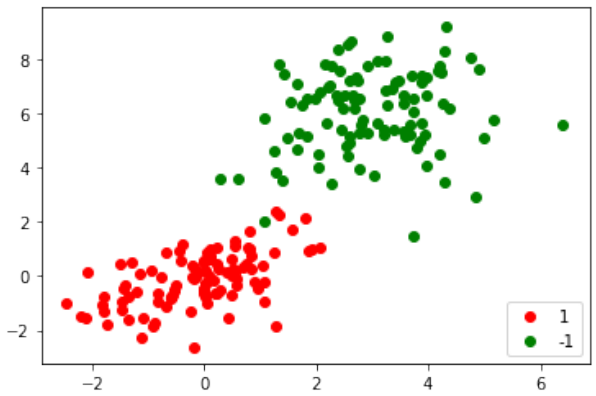
Here also as number of samples increases accuracy in both case increases, also from the result we can say that when both the density are gaussian than it gives better result compared to one exponential and one gaussian. one reason for this behaviour can be that, because means are far apart ([0,0] and [3,6])

Number of examples in training data	When both density are Gaussian	When we assume first as Gaussian and second as Exponential
10	0.871	0.531
25	0.947	0.578
75	0.957	0.621
100	0.982	0.64

Scatter Plot of train dataset c



Scatter Plot of test dataset c



Problem 2

Problem 2 is a 2-class classification problem with twenty-dimensional feature vectors. The class conditional densities are normal. The problem has three subproblems.

There are three data sets again and the training and test datafiles are named similarly. For all three cases, μ1 is a vector of all zeros and μ2 is a vector of all ones.

The question is similary to part b of question 1, but instead of 20 dimensional. For each class conditional density, the size of data used for estimation was varied as 10, 20, 50, 200, 300, 500 (random sampling from training data) for each class .

Implemented same as part b of question 1.posterior probabilities of both the classes are calculated, the
maximum posterior class is classified.

Note: In case of Nearest Neighbour the Euclidean distance were compared.

In first 2 parts, the covariance matrices of both the class are equal so the resulting classifier will be linear classifier (Hyperplane) $W^TX + w_0$. In first case variance is(relatively) low, the density would be a narrow peak near the mean. In second case the covariance matrics is larger than the first one, so second density will have more spread around the means. There will be low accuracy.as the overlap of densities would be more and hence the classification may be erroneous.

Number of examples in training data	Bayes Classifier	Nearest neighbouir Classifier
10	0.78	0.813
50	0.895	0.891
100	0.966	0.917
300	0.989	0.964

For data in P2a_traindata_20D.txt and P2a_train_data_20D.txt

Number of examples in training data	Bayes Classifier	Nearest neighbouir Classifier
10	0.672	0.591
50	0.824	0.671
100	0.934	0.815
300	0.947	0.864

For data in P2b_traindata_20D.txt and P2a_train_data_20D.txt

Number of examples in training data	Bayes Classifier	Nearest neighbouir Classifier
10	0.687	0.754
50	0.912	0.820
100	0.969	0.901
300	0.972	0.926

For data in P2c_traindata_20D.txt and P2c_train_data_20D.txt

for 3rd case The two covariance matrices are not equal with non zero values in off diagonal locations.hence it(covariance) can be used for estimating densities.
The result also shows low accuracy for the second case which was intuitively clear. On comparingwith nearest neighbour classifier Bayes perform better. Also on increasing the examples the accuracy of both the classifier increases.

Question 3

This is 2-class problem with one dimensional feature vector. There are two sub problems here. In both cases, class conditonal densities are mixtures of gaussian as specified below.
for both two subproblems we implement 3 classifiers:
1.assuming class conditional density as mixture gaussian(two), and use EM algorithm
2.assuming the same as single gaussian, and use Maximum Likelihood
3.implement nearest neighbour

Note: For every part I am taking whole training dataset for density estimation.

For 1st classifier we will estimate the class conditional densities with the help of EM algorithm. This will return the mean ,variance and weights(priors).
For 2nd classifier we do ML estimation and then bayes classifier.
For 3rd classifier we implement Nearest neighbour classifier.

SubProblem 1

for sample size 50

Classifier	**Bayes testing accuracy**
Classifier_1(using EM)	93.5
Classifier_2(ML estimation)	93
Classifier_3(NN)	86

For different sample size NN performs different

Sample size	**NN accuracy**
5	72.5
20	81
50	86

SubProblem 2

for sample size 50

Classifier	**Bayes testing accuracy**
Classifier_1(using EM)	73.5
Classifier_2(ML estimation)	57.8
Classifier_3(NN)	66.5

For different sample size NN performs different

Sample size	**NN accuracy**
5	62
20	63.7
50	66.5

For a given class the probability as caliculated as:

$$p_i(x)=w_0*N(x,\mu_0,\Sigma_0)+w_1*N(x,\mu_1,\Sigma_1)$$

where w_0 is the weight given to the single density, $N(x,\mu_0,\Sigma_0)$ is the normal density of test case 'x' with mean μ_0 and covariance Σ_0

whichever p_i is maximum we assign the class to the test case

In estimation using MLE ,The mean was taken of training set and same for covariance, and using this means and covariance density is estimated. and then we implement bayes classifier.

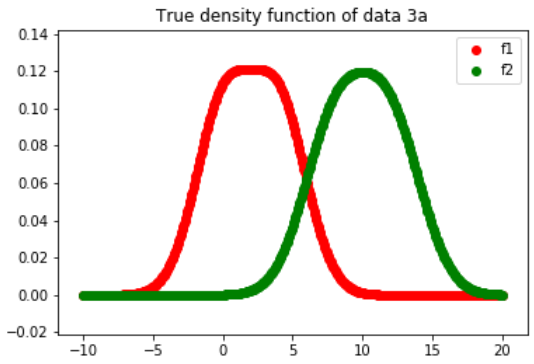
In NN, Euclidian distance was taken for comparing the distance. and with the help of a single neighbour the comparision is done.

On observing the different results(with different sample sizes for all classifiers), I can deduce that if we have sufficient amount of data samples Then EM estimation implementation is best. but if we don't have sufficient data then NN can be used, because estimating the density may result in more error.

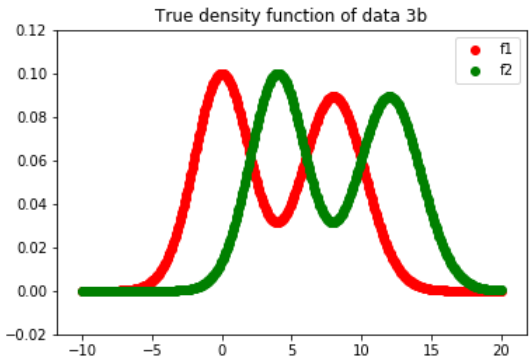
Also in first subproblem, NN performs worst compare to the other two. The other 2 perform good because there is no overlap between the class densities. hence The classification accuracy increases.From data we can see that means for class 1 is 0,4 and for class 2 is 8,12. so intutively we can see that there is no overlap between the 2 means, results in less error.Hence in both case the parameter estimation is approximately correct.Also there is not much difference between MlE estimation and EM one because the densities within class are close enough that even if we use single guassian, there is not much change.

In Second subproblem MLE estimation is worst and EM still performs better than the other 2.
Here we have class 1 means as 0 and 8 , and class 2 means as 4 and 12.
so class densities overlap in this case as the means are overlapping, hence this result in more classification error and less accuracy.

MLE performs the worst because we have only single Gaussian, so the missclassification rate increases (because of density Overlap). also replacing the 2 class conditional densities with a Single Gaussian results in loss of information. This is the reason MLE performs worst than NN in this case.



Data plot 3a



Data plot of 3b

Question 4

This problem is a classification problem(document classification). the dataset consists of 2000 2000 movie reviews and it is a 2-class problem. Using Naive Bayes with 2 different feature vectors 'bag of words' and ' TF-IDF ' compare the performance of the two.

As the samples taken should be randomly distributed, hence i shuffled the data. and then divided it in 70:30 ratio (i.e 70% training data and 30% testing data).

Data Preprocessing is done on whole data, like removing stopwords, Tokenization using NLTK library.Positive Review are Represented as 0 and negative reviews as 1.

For feature extraction from the training data*Countvectorizer* and *Tfidvectorizer* was used from sklearn library for Bag of words model and TF-IDF model.

First we see for Bag of words model

Bag of Words

For bag of words feature representation,a vocabulary is made of all the words that are present in the data. and then we mark the frequency for every word in table. Hence this is done using CountVectorizer() in Scikit-learn library.
Bag of words model is computationally Expensive as the size of matrix will be large.
To train the model i used MultinomialNB()

TF-IDF

Here in TF-IDF, rather than focusing on regular words, the words with more importance are taken into consideration. In TF-IDF model TF stands for term frequency and IDF stands for inverse document. In TF-IDF the words which occurs most in every document is given less Weightage and the words which occurs more in a single document (not in other documents),then it is given more weightage.We make a vocabulary which measures the informativeness of word(i.e log (N/ni)). N is total number of documents, ni is the word appearing in ith document.
IDF reduces the weight given to common words, and highlights the uncommon words in a document.This way we extract the features from the documennt.
To train the model I used MultinomialNB(). and TF-IDF is done using the inbuilt library function in Scikit-learn library TFIDVectorizer().

Comparing both models
Accuracy of Bag of words model was 0.807 and acurracy of TF-IDF was 0.774. Though both have somewhat same accuracy but We can Prefer TF-IDF feature as it do not take into consideration the useless words. it only deals with Relevant words. and also Compared to Bag of words model it uses less space and les computational power.

Train-Test split	Test accuracy(BOW)	Test accuracy(TF-IDF)
60:40	0.74	0.76
70:30	0.807	0.774
80:20	0.864	0.891
90:10	0.842	0.85

Data wordcloud