Pattern Recognition and Neural Networks Assignment – 1

Name – Saankhya Subrata Mondal Sr. No. – 17990

Dept. – CSA Degree – M. Tech in AI

Question 1

1.1 –

For both subproblems a and b, Bayes classifier and nearest neighbor classifier algorithms were implemented. Here, the training and test dataset has equal number of examples from both the classes 1 and -1. The feature vector is of dimension 2 i.e. – there are 2 features for each sample. Firstly, 5, 10, 25, and 75 examples were sampled from training set. It was ensured that the sampled training set is not much skewed towards one of the class. The classifiers were implemented taking into consideration various instances of training data (since training data is generated randomly). Later, I have used the whole training set for training the Bayes classifier and in finding the nearest neighbor. Due to randomness in sampling, I have written the range of test accuracy obtained after training 10 different classifiers of each type wherever possible. Averaging the test accuracy can also be done.

Bayes classifier – From the training set, the prior probabilities were calculated for both the classes. As mentioned in the instructions, the data is synthetic, and the class means and co-variance matrices (assuming normal density) from which the data was generated were provided. However, I have calculated the class means and co-variance matrices for both the classes. I continued to train the classifier till it achieved reasonably good accuracy (more than 70% in a) and 90% in b)) on the training set. After this, the posterior probability of both the classes were computed for each test sample. Each test sample was assigned the class which has a higher posterior probability. Accuracy was computed by calculating the number of misclassifications.

Nearest neighbor – For each test example, the closest training sample from the training set was found out. Euclidean norm was used to find the closest training sample. The test sample was then assigned the class of the closest training sample. The same set of training examples were used to find the nearest neighbors as was used in training the Bayes classifier.

Subproblem a) –

No. of training examples (n)	Class 1 means	Class 1 co-variance matrix	Class 2 means	Class 2 co-variance matrix
5	[-0.1822, 0.79663]	[[1.315, -0.63]; [-0.63, 0.8164]]	[1.13, 1.6043]	[[0.582, -1.00]; [-1.00, 1.72]]
10	[0.2471, 0.8196]	[[0.52,0.189]; [0.189, 0.465]]	[0.151, 1.475]	[[0.91, 0.43]; [0.43, 1.034]]
25	[0.444, 0.286]	[[1.068, 0.119]; [0.119, 1.008]]	[1.256, 1.025]	[[0.52, 0.053]; [0.053, 0.858]]
75	[0.046, 0.273]	[[1.015, 0.552]; [0.552, 0.8326]]	[1.13, 0.937]	[[0.9, -0.108]; [-0.108; 2.059]]
200	[0.158, 0.245]	[[1.016, 0.4535]; [0.4535, 1.014]]	[1.266, 1.04]	[[0.91, 0.04]; [0.04, 1.513]]

No. of training examples (n)	Bayes classifier accuracy on test set (in %)	Nearest neighbor classifier accuracy on test set (in %)
5	50	63
10	75	68
25	76	66-71.5
75	76	66.5-69
200	76.5	69

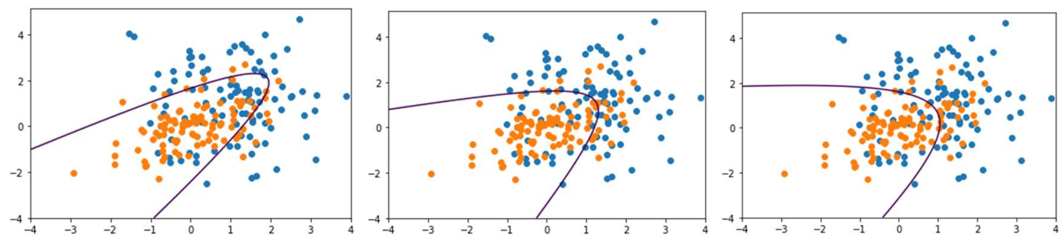

Fig 1 – Bayes classifier boundary for n = 25, 75, 200 respectively

Subproblem b) –

No. of training examples (n)	Class 1 means	Class 1 co-variance matrix	Class 2 means	Class 2 co-variance matrix
5	[1.252, -0.10135]	[[0.2403, 0.5949]; [0.5949, 1.47284]]	[3.18, 3.457]	[[0.2944, 0.724]; [0.724, 1.898]]
10	[0.072, 0.276]	[[0.128,0.034]; [0.034, 0.8798]]	[2.5352, 3.785]	[[0.716, -1.367]; [-1.367, 3.07]]
25	[0.089, -0.0933]	[[0.5903, 0.103]; [0.103, 0.538]]	[3.125, 3.471]	[[1.019, -0.409]; [-0.409, 0.996]]
75	[0.128, -0.072]	[[1.236, 0.5428]; [0.5428, 0.8795]]	[3.10, 3.36]	[[1.091, -0.149]; [-0.149; 3.627]]
200	[-0.023, 0.011]	[[1.072, 0.615]; [0.615, 1.06]]	[3.014, 3.34]	[[0.9811, -0.16]; [-0.16, 2.873]]

No. of training examples (n)	Bayes classifier accuracy on test set (in %)	Nearest neighbor classifier accuracy on test set (in %)
5	50	85-90
10	83.3-87.5	94-97
25	93-97.5	93.5-97
75	97.5-98	96-97
200	98	96

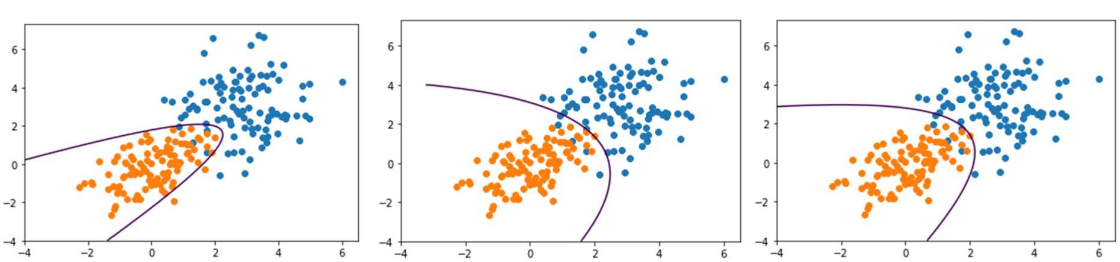

Fig 2 - Bayes classifier boundary for n = 25, 75, 200 respectively

Observations –

- When the number of training examples are 5, NN performs better than Bayes classifier. In fact, Bayes classifier does not learn at all and classifies all the test samples in the same class.
- Bayes classifier accuracy increases as the number of training examples are increased.
- Also, it performs better than or like NN classifier in most cases.
- As can be seen from the table, we achieve higher accuracies in sub-problem b). This is because the class means differ more in this case than in case of sub-problem a). Thus, the data is more separable and hence both the classifiers perform better in the latter case. This can be observed in the scatter plot as well.
- The means and co-variance matrices of both class conditional densities were observed to be approximately equal to the true class means and co-variance matrices.

1.2 –

Expectation-Maximization (EM) algorithm is applied to estimate the two components of the Gaussian Mixture Model. It is an iterative algorithm which is used to estimate the mean, covariances and the weights of the two Gaussian density components present in the GMM. Here, this algorithm has been run with 'k-means' initialization for 10 iterations after which it converges. The following are the estimates obtained for prior probabilities (weights), class means, and class covariance matrices (full covariance was used) for each component –

$$p1 = 0.47658667$$
$$p2 = 0.52341333$$
$$\mu 1 = [-0.09560031, -0.04895932]\ T,$$
$$\mu 2 = [2.96465286, 3.24850602]\ T,$$
$$\Sigma 1 = [0.92316508, 0.49406501; 0.49406501, 0.96721445],$$
$$\Sigma 2 = [0.98585216, -0.02931426; -0.02931426, 2.95639524]$$

It can be observed that around 52% of the training data is estimated to be obtained from class-2 whereas only 50% of the training data comes from class-2. The class means are approximately equal to the original class means. The class co-variance matrices are also approximately equal except the last entry of Σ2 which is 2.956 instead of 2.

Implementing the Bayes classifier using the above estimates, it obtains an accuracy of 97% on the test set which is approximately same as the accuracy obtained (98%) while using the class labels for Bayes classification.

1.3 –

For this question, Bayes classifier assuming both class conditional densities are normal is first implemented. Next, I assumed class conditional density for class-2 to be exponential. The exponential density parameter is the inverse of the mean of the distribution. The classifiers were implemented taking into consideration various instances of training data (since training data is generated randomly).

No. of training examples (n)	Bayes classifier (both normal) accuracy on test set (in %)	Bayes classifier (normal and exponential) accuracy on test set (in %)
5	50	50
10	63-83	57-81
25	93-99	65-88
75	97-99	78-91
200	97.5	86.5

Observations –

- For 5 training examples case, both the classifiers do not learn anything and assign same class to all the test samples (that's why accuracy is 50%).
- As we increase the training samples from 5 to 75, the Bayes classifier (with both normal) gives better and better accuracy. However, in case of the classifier (with one normal and one exponential), the accuracies obtained are not satisfactory.
- Assuming both the densities to be normal gives better results than assuming one of them to be exponential.

Question 2

For all the subproblems, Bayes classifier and nearest neighbor classifier algorithms were implemented. Here, the training and test dataset has equal number of examples from both the classes 1 and -1. The feature vector is of dimension 20 i.e. – there are 20 features for each sample. Firstly, 10, 50, 100, and 300 examples were sampled from training set. It was ensured that the sampled training set is not much skewed towards one of the class. The classifiers were implemented taking into consideration various instances of training data (since training data is generated randomly). Later, I have used the whole training set for training the Bayes classifier and in finding the nearest neighbor. Due to randomness in sampling, I have written the range of accuracy obtained after training 10 different classifiers of each type wherever possible. Averaging the accuracy can also be done.

Subproblem a) –

No. of training examples	Bayes classifier accuracy on test set (in %)	Nearest neighbor classifier accuracy on test set (in %)	Observations
10	45-53	85-93	Bayes classifier performs poorer than NN classifier. In fact, Bayes classifier does not learn anything and classifies all test example as class 2.
50	64-82	93-96	Accuracies increase as the no. of training examples increased. NN classifier performs better than Bayes classifier.
100	87.5-93	93.5-97	NN classifier gives higher accuracy than

			Bayes for various instances.
300	97-98	94-96.5	NN classifier gives higher accuracy than Bayes for various instances.
1000	98.9	95.9	Bayes classifier gives higher accuracy than NN. It gives highest accuracy when all 1000 examples were used for training.

Subproblem b) –

No. of training examples	Bayes classifier accuracy on test set (in %)	Nearest neighbor classifier accuracy on test set (in %)	Observations
10	45-52	70-73	Bayes classifier performs poorer than NN classifier. In fact, Bayes classifier does not learn anything and classifies all test example as class 2.
50	52-70	73-83	Accuracies increase as the no. of training examples increased. NN classifier performs better than Bayes classifier.
100	74-78	72-79	Bayes classifier accuracies are in the same range as that of NN for various instances.
300	84-85	77-81	Bayes classifier gives higher accuracy than NN for various instances.
1000	88.3	79.9	Bayes classifier gives higher accuracy than NN. It gives highest accuracy when all 1000 examples were used for training.

Subproblem c) –

No. of training examples	Bayes classifier accuracy on test set (in %)	Nearest neighbor classifier accuracy on test set (in %)	Observations
10	45-53	92-94	Bayes classifier performs poorer than NN classifier. In fact, Bayes classifier does not learn anything

			and classifies all test example as class 2.
50	75-85	93-97	Accuracies increase as the no. of training examples increased. NN classifier performs better than Bayes classifier.
100	97-99	94-95.5	Bayes classifier gives higher accuracy than NN for various instances.
300	99-99.6	97-98.5	Bayes classifier gives slightly higher accuracy than NN for various instances.
1000	99.7	97.6	Bayes classifier gives higher accuracy than NN. It gives highest accuracy when all 1000 examples were used for training.

Further observations –

- As we increase the training samples from 10 to 300, the sample mean and covariances inch closer to their respective true values.
- NN classifier is more suitable for classification if the sample size is less. As the sample size increase, Bayes classifier performs better than NN classifier.

Question 3

Here, the data is one dimensional and is obtained from mixture of Gaussians. Firstly, the class conditional densities are assumed to be coming from a mixture of two Gaussians. Hence, the weights, means, and covariances were estimated using EM algorithm (with k-means initialization). Then, Bayes classifier was implementing by calculating the posterior probability for both classes calculated using the convex combination of the mixture. Next, Bayes classifier was implemented by assuming class conditional densities to be single Gaussian. Finally, both the classifiers were compared with NN classifier.

Subproblem a)

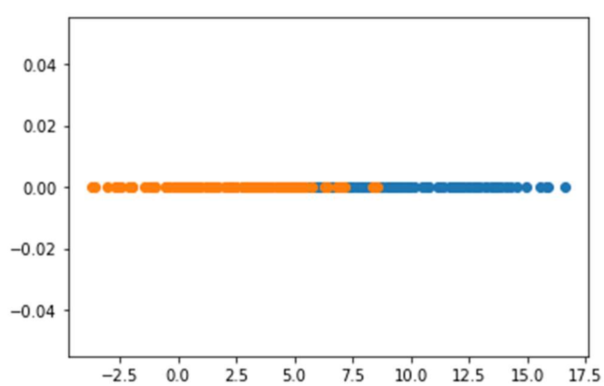

Fig 3 – Scatter plot of test data

The following are the parameters in case of 2 GMM –

Parameters	Class 1 Mixture 1	Class 1 Mixture 2	Class 2 Mixture 1	Class 2 Mixture 2
Weights (prior probabilities)	0.44053	0.559466	0.47269	0.52370
Means	0.35767	3.82633	7.5673	11.5186
Covariances	5.77765	5.09926	3.2917	2.92639

The following are the parameters in case of a single Gaussian –

Parameters	Class 1	Class 2
Weights (prior probabilities)	0.5	0.5
Means	2.2982	9.633
Covariances	8.448	7.052

Type of classifier	2 GMM Bayes Classifier	Single Gaussian Bayes Classifier	Nearest Neighbour Classifier
Test set Accuracy	93.5%	91%	89.5%

Observations –

- The estimated weights, means, and covariances are approximately equal to the true values of the weights, means, and covariances.
- 2 GMM Bayes classifier performs best followed by single Gaussian Bayes classifier and NN classifier.

Subproblem b)

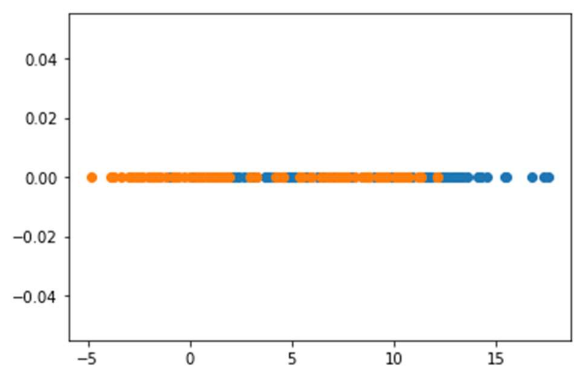

Fig – 4 Scatter plot of test data

The following are the parameters in case of 2 GMM –

Parameters	Class 1 Mixture 1	Class 1 Mixture 2	Class 2 Mixture 1	Class 2 Mixture 2
Weights (prior probabilities)	0.41708	0.5829	0.39788	0.60211
Means	0.1203	7.5835	3.8148	12.0442
Covariances	2.0951	5.1224	3.670	5.7477

The following are the parameters in case of a single Gaussian –

Parameters	Class 1 Mixture	Class 2 Mixture
Weights (prior probabilities)	0.5	0.5
Means	4.471	8.7698
Covariances	17.577	21.356

Type of classifier	2 GMM Bayes Classifier	Single Gaussian Bayes Classifier	Nearest Neighbour Classifier
Test set Accuracy	73.5%	57%	68%

Observations –

- The estimated weights, means, and covariances are approximately equal to the true values of the weights, means, and covariances.
- NN classifier performs the best followed by 2 GMM Bayes classifier.

Question 4

For sentiment classification using Naïve Bayes algorithms, two different feature vectors are available namely – Bag of Words (BOW) model and Term frequency – Inverse document frequency (TFIDF). The dataset consists of 2000 movie reviews. It is two-class problem with classes either 'pos' or 'neg'. For both feature vectors, three Naïve Bayes classifiers were implemented – Multinomial, Gaussian, and Bernoulli Naïve Bayes. Naïve Bayes classifier uses the Bayes theorem and assumes that all the features are independent of each other.

Bag of words – BOW model is used for feature extraction in text data. It returns a vector with all the words and number of times each word is repeated. It is known as BOW because it is only concerned with the number of times a word is repeated rather than order of words.

Term frequency – Inverse document frequency – Here, the product of term frequency and inverse document frequency is used. Term frequency is how frequently a word has appeared in a document. If a term appears f times in a document with d words.
Term Frequency (TF) = f/d
IDF is inverse document frequency. If a corpus contains N documents and the word of our interest appears only in D documents, then IDF is -
IDF = log(N/D)
TF-IDF is product of Term Frequency and Inverse Document Frequency. TF-IDF shows the rarity of a word in the corpus.
TF-IDF = TF * IDF

Before constructing the feature vectors, various data pre-processing and data cleaning steps were performed. These include the following.

- Converting all the reviews into lower-case.
- Removing white spaces, numbers, and punctuations.
- Removing stop words.

The dataset then was split into training and test sets. I have considered three cases wherein the test set consisted of 20%, 25% and 30% of the whole dataset, respectively. There were 39092 features for each review in the BOW model and 39363 features for each review in TF-IDF model. The results have been tabulated in the following table.

Type of algorithm	BOW (split – 80%-20%)	BOW (split – 75%-25%)	BOW (split – 70%-30%)	TF-IDF (split – 80%-20%)	TF-IDF (split – 75%-25%)	TF-IDF (split – 70%-30%)
Multinomial Naïve Bayes	81.00%	80.60%	80.67%	76.50%	78.40%	80.17%
Gaussian Naïve Bayes	65.50%	64.20%	63.33%	65.50%	64.60%	64.00%
Bernoulli Naïve Bayes	80.50%	79.20%	78.83%	79.75%	79.40%	78.67%

Observations –

- It can be observed that Multinomial Naïve Bayes model performs better than Gaussian Naïve Bayes or Bernoulli Naïve Bayes model for all the cases.
- Considering Multinomial Naïve Bayes model, BOW performs better than TF-IDF for all three kinds of train-test splits.