Pattern Recognition
Department of Electrical and Information Engineering
University of Cassino and Southern Latium, Second Semester 2018
Homework Assignment 1
Assigned 10 April 2018; due 11:59pm, 25 April 2018

In preparing my solutions, I did not look at any old homework's, copy anybody's answers or let them copy mine.

NAME: FAKRUL ISLAM TUSHAR. Signature:

Problem 1.1 [5%]: Describe (max one page) an application of pattern recognition related to your research. What are the features? What is the decision to be made? Comment on how one might solve the problem.

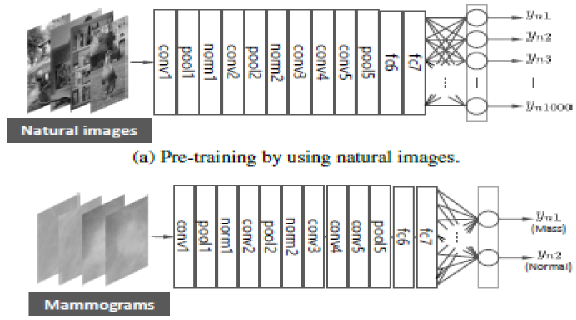
Answer:

From last few weeks I am exploring the broad world of pattern recognition and machine learning. Currently I am working on a research project Called "Segmentation and Classification of skin lesion of dermatological". The project involved the task of segmentation of the skin lesion from the dermatological and then classify them among three classes. It's a joint project that required the application of image processing and pattern recognition.

Defeating all other candidates skin cancer becomes most common cancer worldwide affecting 2-3 million cases yearly. To diagnose skin lesion dermatologists, use a dermatoscope (a non-invasive in vivo clinical examination which allows for a magnified and clear visualization of the morphological structures of the skin that are not visible to the naked eye).

General features of skin cancer are irregularity in shape, surface, color, structure, and may bleed. Since the shape and border of the skin lesion is extremely important in the diagnosis, to decide whether it is Melannorma, Nevus or Seborrheic keratosis.

One possible solution to the problem can be solved by using deep convolution neural network (DCNN) with transfer learning strategy. The DCNN can obtain feature representation and classification automatically without hand-designed feature extraction. Problem with the DCNN is that it requires a large amount of training data. Transfer Learning Strategy can solve the above training problem for the DCNN. This idea is inspired from the paper by Shintaro Suzuki on "Mass Detection Using Deep Convolutional Neural Network for Mammographic Computer-Aided Diagnosis"



(a) Pre-training by using natural images.

(b) Fine-tuning by using mammograms.

Fig. 1: Using DCNN for mammographic mess detecting with transfer learning strategy. [1]

Problem 1.2 [15%]: In a particular two class problem, the conditional densities for a scalar feature x are $P(x|\omega_1) = k_1 * exp^{(-\frac{x^2}{15})}$ and $P(x|\omega_1) = k_1 * exp^{(-\frac{(x-7)^n}{14})}$

(a) Find k_1 and k_2, and plot the two densities on a single graph using MATLAB.

(b) Find the decision regions which minimize the average probability of error and indicate them on the plot you made in part (a) for the following cases:

(1) $\frac{P(\omega_2)}{P(\omega_1)} = 1$

(2) $\frac{P(\omega_2)}{P(\omega_1)} = 0.5$

(3) $\frac{P(\omega_2)}{P(\omega_1)} = 4$

(a) Answer: From the Gaussian distribution we know, univariate gaussian us given by

$$P(x|\omega_i) = \frac{1}{\sqrt{2\pi}\,\sigma_i}\, exp^{(-\frac{(x-\mu_i)^2}{2\sigma_i^2})} \quad .... (1)$$

Now comparing Equation (1) with the class ω_1's conditional density,

$$\frac{1}{\sqrt{2\pi}\,\sigma_i}\, exp^{(-\frac{(x-\mu_i)^2}{2\sigma_i^2})} = k_1 * exp^{(-\frac{x^2}{15})}$$

Now comparing left side of both we get,

$$(x - \mu_i)^2 = x^2$$
$$\mu_1 = 0;$$
$$2\sigma_1^2 = 15$$
$$\sigma_1^2 = 7.5$$
$$\sigma_1 = \sqrt{7.5}$$
So, $$k_1 = \frac{1}{\sqrt{2\pi}\,\sigma_i} = \frac{1}{\sqrt{15\pi}}$$

Now comparing Equation (1) with the class ω_2 conditional density,

$$\frac{1}{\sqrt{2\pi}\,\sigma_i}\, exp^{(-\frac{(x-\mu_i)^2}{2\sigma_i^2})} = k_2 * exp^{(-\frac{(x-7)^2}{14})}$$

Now comparing left side of both we get,

$$(x - \mu_i)^2 = (x - 7)^2$$
$$\mu_2 = 7;$$
$$2\sigma_2^2 = 14$$
$$\sigma_2^2 = 7$$
$$\sigma_2 = \sqrt{7}$$
So, $$k_2 = \frac{1}{\sqrt{2\pi}\,\sigma_i} = \frac{1}{\sqrt{14\pi}}$$

Both the densities $P(x|\omega_1)$ and $P(x|\omega_2)$ plotted below using MATLAB Shown in Figure-

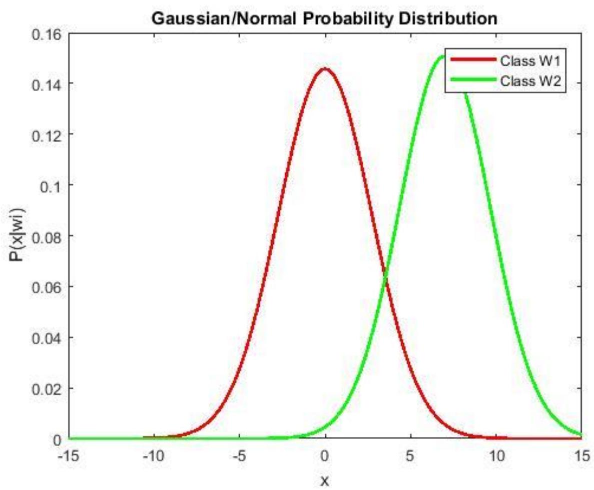


Fig. 2: Gaussian/ Normal Probability Distribution of two conditional densities.

(b) **<u>Answer:</u>** To find of the decision region that minimize the average probabilities of error the both densities we know from the Maximum A Posterior rule that the Error will be minimum when discriminate function of both the classes are equal,

$$g_1(x) = g_2(x)$$
$$g_1(x) - g_2(x) = 0$$

Where,

$$g_1(x) = Discriminate\ function\ of\ Class\ w1$$

$$g_2(x) = Discriminate\ function\ of\ Class\ w2$$

From minimum error rate classification, we know

$$g_1(x) = P(\omega_1|x) = P(x|\omega_1).P(\omega_1)$$

$$g_1(x) = P(\omega_2|x) = P(x|\omega_2).P(\omega_2)$$

Where,

$$P(\omega_1|x) = Posterior\ probalitity\ of\ class\ W1$$
$$P(\omega_2|x) = Posterior\ probalitity\ of\ class\ W2$$
$$P(\omega_1) = Prior\ probabily\ of\ Class\ W1$$
$$P(\omega_2) = Prior\ probabily\ of\ Class\ W2$$

In this case we compute the decision boundary by computing

$$g_1(x) = g_2(x)$$
$$P(\omega_1|x) = P(\omega_2|x)$$
$$P(x|\omega_1).P(\omega_1) = P(x|\omega_2).P(\omega_2)$$

$$P(x|\omega_1) = P(x|\omega_2).\frac{P(\omega_2)}{P(\omega_1)}$$

$$\frac{1}{\sqrt{15\pi}}\ exp^{(-\frac{x^2}{15})} = \frac{1}{\sqrt{14\pi}}\ exp^{(-\frac{(x-7)^2}{14})}.\frac{P(\omega_2)}{P(\omega_1)}$$

$$\ln(\frac{1}{\sqrt{15\pi}}) - \frac{x^2}{15} = \ln(\frac{1}{\sqrt{14\pi}}) - \frac{(x-7)^2}{14} + \ln(\frac{P(\omega_2)}{P(\omega_1)})$$

$$\frac{x^2 + 14x + 49}{14} - 0.034 - \frac{x^2}{15} = \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right)$$

$$x^2 - 14x + 49 - 0.476 - 0.93x^2 = 14 * \ln(\frac{P(\omega_2)}{P(\omega_1)})$$

$$0.07x^2 - 14x + 48.524 - 14 * \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right) = 0 \quad \textbf{......(2)}$$

Now Using equation (2) for Different Prior Probability ration the minimum error boundary can be found.

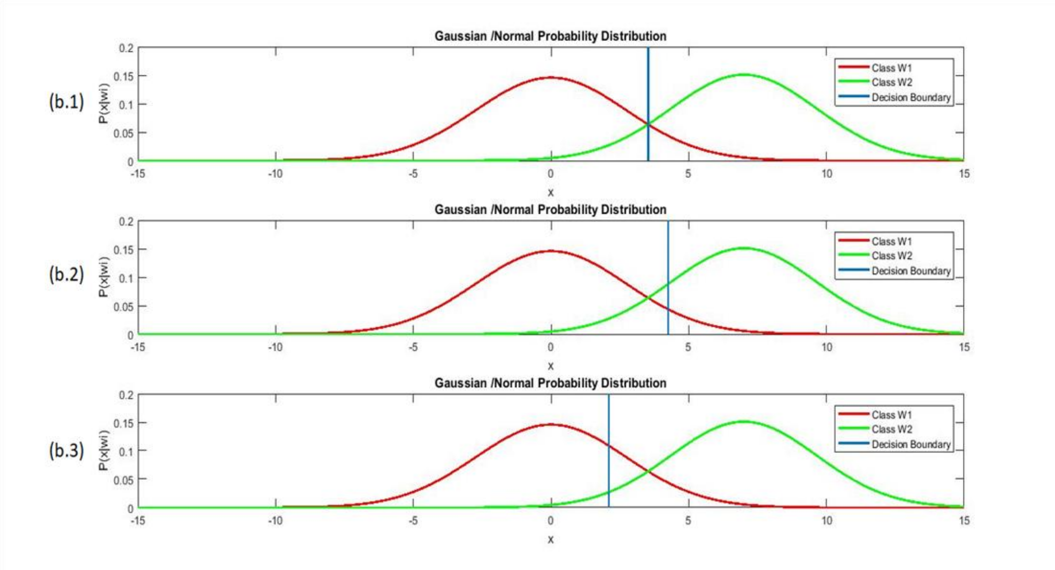


Fig.3: Decision boundary for minimum error of densities for different Prior probability ration (b.1) $\frac{P(\omega_2)}{P(\omega_1)} = 1, Probabilities\ are\ equal$ (b.2) $\frac{P(\omega_2)}{P(\omega_1)} = 0.5$, $P(\omega_1)\ is\ higher\ than\ P(\omega_2)$ (b.3) $\frac{P(\omega_2)}{P(\omega_1)} = 4, P(\omega_2)\ is\ higher\ than\ P(\omega_1).$

TABLE I below shown the Computation od decision boundaries.

TABLE I: Decision Boundaries for Different Priors ratio

(b.1)	(b.2)	(b.3)
• In this case $\frac{P(\omega_2)}{P(\omega_1)} = 1$ • Using this ration in equation 2 we get $0.07x^2 - 14x + 48.524 - 14 * \ln(1) = 0$ After solving this we get, x=3.528 and x=196.472 The decision boundary is at **x= 3.528** which lies between the intersection between the two density functions. The decision boundary is shown in the Figure 2(b.1).	• In this case $\frac{P(\omega_2)}{P(\omega_1)} = 0.5$ • Using this ration in equation 2 we get $0.07x^2 - 14x + 48.524 - 14 * \ln(0.5) = 0$ After solving this we get, x=4.249 and x=195.751 The decision boundary is **at x=4.249** which lies between the intersection between the two density functions. The decision boundary is shown in the Figure 2(b.2).	• In this case $\frac{P(\omega_2)}{P(\omega_1)} = 4$ • Using this ration in equation 2 we get $0.07x^2 - 14x + 48.524 - 14 * \ln(4) = 0$ After solving this we get, x=2.102 and x=197.898 The decision boundary is at **x= 2.102** which lies between the intersection between the two density functions. The decision boundary is shown in the Figure 2(b.3).

Problem 1.3 [20%]: Consider a medical diagnosis problem where a biochemical test is used for screening patients. The test returns a result close to 0 for healthy patients and close to 1 for sick patients, according to the following likelihood functions: p(x|ω_1) = N (0.0; 0.1) and p(x|ω_2) = N (1.5, 0:3) where N(μ,σ) is the Gaussian univariate density. Assume that, on average, 1 out of 10,000 patients is sick, and the following costs:

1. $\lambda_{21} = 1500$ euros for improductive test cost in the case of wrongly diagnosed disease;
2. λ_{12}= 800000 euros for reimbursement to the patient in the case of not diagnosed disease;
3. $\lambda_{11} = \lambda_{22}$= 0 in the case of correct diagnosis

and define the decision rule that minimizes the Conditional Risk.

Answer: Above given a two-class medical diagnosis problem of class ω_1 and class ω_2 where test returns,

$$\omega_1 = 0 \text{ for a healthy patient}$$
$$\omega_2 = 1 \text{ for a sick patient}$$

Also given the gaussian density of likelihood functions of the two classes,

$$for\ class\ \omega_1,\ \mu_1 = 0\ and\ \sigma_1 = 0.1$$
$$for\ class\ \omega_2,\ \mu_2 = 1.5\ and\ \sigma_2 = 0.3$$

Given on average 1 out of 10,000 patients is sick, and cost associate with the decision.

$\lambda_{21} = 1500$ euros for wrongly diagnosed disease;
λ_{12} = 800000 euros, for not diagnosed disease;
$\lambda_{11} = \lambda_{22} = 0$ = Right Decision

From that we can get the cost matrix

$$\lambda = \begin{bmatrix} 0 & 800000 \\ 1500 & 0 \end{bmatrix}$$

And the probability for $for\ class\ \omega_1\ and\ for\ class\ \omega_2$ is given by

$$P(\omega_1) = \frac{9999}{10000} = 0.9999, \text{ Probability of being healthy}$$

$P(\omega_2) = \frac{1}{10000} = 0.0001$, Probability of being sick

Now we need to define decision rule that define the minimum condition risk.

So the conditional risk $R(\alpha_1|\omega_1)$ for $class\ \omega_1\ and\ for\ class\ \omega_2$ is given by

$$R(\alpha_1|\omega_1) = \lambda_{11}\, P(\omega_1|x) + \lambda_{12}\, P(\omega_2|x)$$

$$R(\alpha_2|\omega_2) = \lambda_{21}\, P(\omega_1|x) + \lambda_{22}\, P(\omega_2|x)$$

Now from the cost matrix we can put the cost associated with the corresponding decision.

$$R(\alpha_1|\omega_1) = \lambda_{11}\, P(\omega_1|x) + \lambda_{12}\, P(\omega_2|x) = 0 * P(\omega_1|x) + 800000 * P(\omega_2|x)$$

$$R(\alpha_1|\omega_1) = 800000 * P(\omega_2|x) \ldots (3)$$

$$R(\alpha_2|\omega_2) = \lambda_{21}\, P(\omega_1|x) + \lambda_{22}\, P(\omega_2|x) = 1500 * P(\omega_1|x) + 0 * P(\omega_2|x)$$

$$R(\alpha_2|\omega_2) = 1500 * P(\omega_1|x) \ldots.. (4)$$

Decision Region will have the minimum cost when the condition risk/cost $R(\alpha_1|\omega_1)$ and $R(\alpha_2|\omega_2)$ will be same. So, we can find the decision boundary by solving the equation (3) and (4)

$$R(\alpha_1|\omega_1) = R(\alpha_2|\omega_2)$$

$$800000 * P(\omega_2|x) = 1500 * P(\omega_1|x)$$

$$800000 * P(x|\omega_2).P(\omega_2) = 1500 * P(x|\omega_1) * P(\omega_1)$$

$$(800000 * 0.0001)P(x|\omega_2) = 1500 * (0.9999)P(x|\omega_1) * P(\omega_1)$$

$$80 * P(x|\omega_2) = 1499.85 * P(x|\omega_1)$$

$$P(x|\omega_2) = 18.748 * P(x|\omega_1)$$

$$\frac{1}{\sqrt{2\pi}\,\sigma_2}\ exp^{(-\frac{(x-\mu_2)^2}{2\sigma_2^2})} = \frac{18.748}{\sqrt{2\pi}\,\sigma_1}\ exp^{(-\frac{(x-\mu_1)^2}{2\sigma_1^2})}$$

$$\frac{1}{\sqrt{2\pi}\,0.3}\ exp^{(-\frac{(x-1.5)^2}{0.18})} = \frac{18.748}{\sqrt{2\pi}\,0.1}\ exp^{(-\frac{(x)^2}{0.02})}$$

$$-\frac{(x-1.5)^2}{0.18} = \ln(56.244) - \frac{x^2}{0.02}$$

$$-\frac{(x-1.5)^2}{0.18} + \frac{x^2}{0.02} = 4.029$$

$$8x^2 + 3x - 6.279 = 0$$

Solution of this equation will give the decision boundary for the minimum risk.

<u>Problem 1.4 [30%]</u>: The _le hw1data.zip available in the homework folder contains samples coming from a two-class problem, each made of 10 numerical features and a binary label (±1). Split the data into training and test sets by randomly selecting 25% of the examples from each class for the test set.

(a) Using the training data, implement a linear classifier and a quadratic classifier and evaluate on the test set the True Positive Rate obtained for a False Positive Rate of 0.1. Repeat the above steps several times. What is the average obtained TPR for each of the implemented classifiers? Discuss your results.

(b) Repeat the previous exercise, but this time implementing a KNN classifier. Experiment with several values for k. Discuss your results and compare them with those obtained in the previous exercise.

<u>Answer 1.4(a):</u>

Linear Classifier:

Two Separate Matlab Script was prepared for the linear and quadratic Classifier. The procedure of making the linear and Quadratic Classifier is shown below in Table 2. and Table 4. respectively.

TABLE 2: Details about the procedure of making linear classifier

I.	Load the data provided for the home work
II.	Divide the data into positive and negative class
III.	Then split the positive and negative data in 75% for training and 25% for the testing. The split was performed complete randomly.
IV.	Then we combined the data positive and negative testing and training set into one Train_Data and Test_Data.
V.	Calculate Mean for Positive and negative class using training data.
VI.	Compute the sigma for positive and negative class using training data.
VII.	Compute the probability of positive and negative class using the training data.

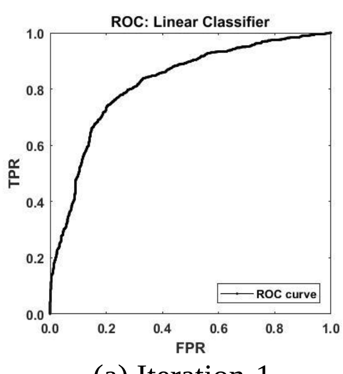
VIII. Now making one sigma multiplying probabilires of respective classes with their sigma and adding them to make Sigma for linear classifier.
IX. Afterward testing period started.
X. Calculated the discriminant function for positive and negative classes.
XI. Decision of taken based on discriminate function.
XII. Make one 2x1 matrix for calculating the ROC storing the true labels in 1st column and decision results in column 2.
XIII. Using the provided function **EvalROC** and **Rchplot** ROC and AUC (Area Under Curve) was calculated.
XIV. TPR for FPR=0.1 was calculated.
XV. Finally, accuracy was calculated.

The procedure was repeated for the 10 iterations. The table below shown the Parameters of each iteration and average.

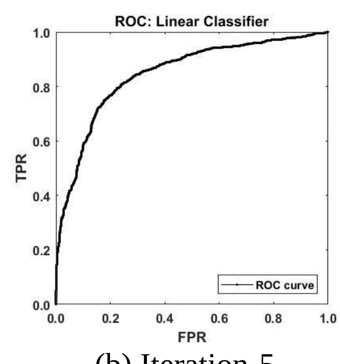
TABLE 3: Linear Classifier Performance Parameters.

Iteration	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average**
Accuracy	0.7580	0.7540	0.7635	0.7735	**0.7785**	0.7705	0.7685	0.7685	0.7745	0.7640	**0.7667**
TPR	0.4935	0.5595	0.5290	0.5325	0.5825	0.5170	0.5270	0.5395	0.5675	0.5210	**0.5369**
AUC	0.8226	0.8288	0.8302	0.8424	0.8471	0.8369	0.8343	0.8309	0.8414	0.8355	**0.8350**

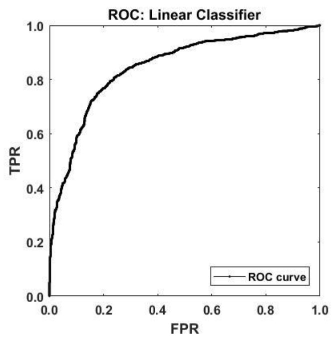
From the Table 3. above, we can see the Accuracy of the Accuracy, TPR and AUC for different iteration. The parameters vary as the train and test samples are randomly varied during each iteration. The average accuracy is 76.67%, whereas maximum accuracy achieved was 77.85% in Iteration-5. Average AUC and TPR was 0.8350 and 0.5350 respectively.



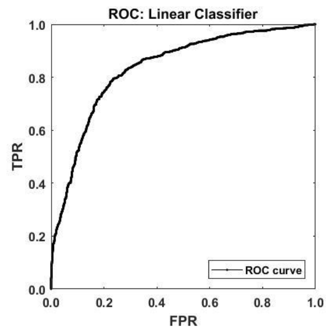
(a) Iteration-1



(b) Iteration-5



(c) Iteration-8



(d) Iteration-10

Fig. 4: ROC of Linear Classifier for different iteration.

Quadratic Classifier:

TABLE 4 : Details about the procedure of making Quadratic classifier

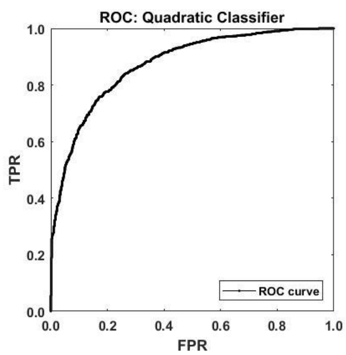
I.	Load the data provided for the home work
II.	Divide the data into positive and negative class
III.	Then split the positive and negative data in 75% for training and 25% for the testing. The split was performed complete randomly.
IV.	Then we combined the data positive and negative testing and training set into one Train_Data and Test_Data.
V.	Calculate Mean for Positive and negative class using training data.
VI.	Compute the sigma for positive and negative class using training data.
VII.	Compute the probability of positive and negative class using the training data.
VIII.	Afterward testing period started.
IX.	Calculated the discriminant function for positive and negative classes.
X.	Decision of taken based on discriminate function.
XI.	Make one 2x1 matrix for calculating the ROC storing the true labels in 1^{st} column and decision results in column 2.
XII.	Using the provided function **EvalROC** and **Rchplot** ROC and AUC (Area Under Curve) was calculated.
XIII.	TPR for FPR=0.1 was calculated.
XIV.	Finally, accuracy was calculated.

The procedure was repeated for the 10 iterations. The table below shown the Parameters of each iteration and average.

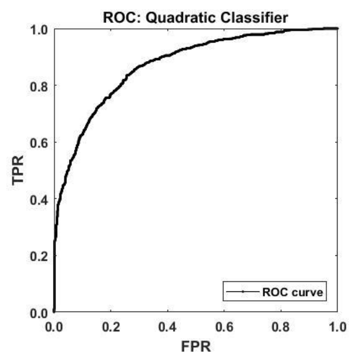
TABLE 5: Quadratic Classifier Performance Parameters.

Iteration	1	2	3	4	5	6	7	8	9	10	Average
Accuracy	**0.7385**	0.7205	0.7225	0.7225	**0.7385**	0.7280	0.7220	0.7205	0.7345	0.7170	**0.7262**
TPR	0.6465	0.6195	0.6260	0.6260	0.6250	0.6290	0.5900	0.5865	0.6300	0.5690	**0.6127**
AUC	0.8742	0.8624	0.8699	0.8760	0.8760	0.8715	0.8683	0.8646	0.8767	0.8725	**0.8703**

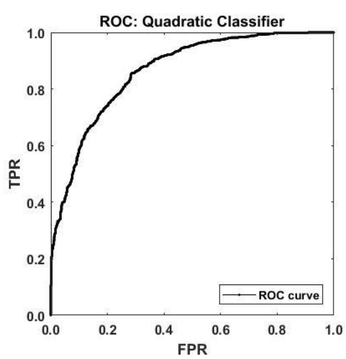
From the Table 5. above, we can see the Accuracy of the Accuracy, TPR and AUC for different iteration. The parameters vary as the train and test samples are randomly varied during each iteration. The average accuracy is 72.62%, whereas maximum accuracy achieved was 73.85% in Iteration-1 and 5. Average AUC and TPR was 0.8703 and 0.6127 respectively.



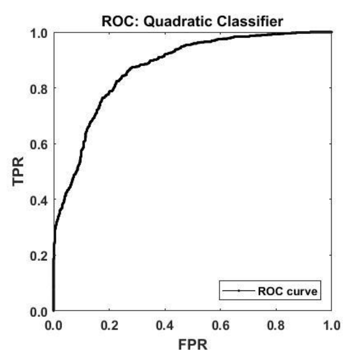
(a) Iteration-1



(b) Iteration-5



(c) Iteration-8



(d) Iteration-10

Fig. 5: ROC of Quadratic Classifier for different iteration.

Discussion: After performing same number of iteration for both Linear and Quadratic classifier, the average accuracy of linear classifier is 76.67%, which is higher than the average accuracy 72.62% of quadratic classifier. We know linear classifier learns from the linear boundary between different classes, quadratic classifier learns from the quadratic boundary. As our data is linear in nature and quadratic classifier has model bias, so that's why the linear classifier has a higher average accuracy. On the other hand, Average TPR and AUC is higher for the quadratic classifier which is 0.6127 and 0.8703respectively.

<u>KNN Classifier:</u>

KNN classifier is non-parametric approach, where classifier don't have any assumption about the form of the densities. Using the same data provides KNN classifier was build. ROC was plotted, TPR and Accuracy was calculated. As asked in the question to try different k value. We have varied the k from 1 to 100 and calculated the average values.

Details about making the KNN classifier MATLAB code given below in the table-

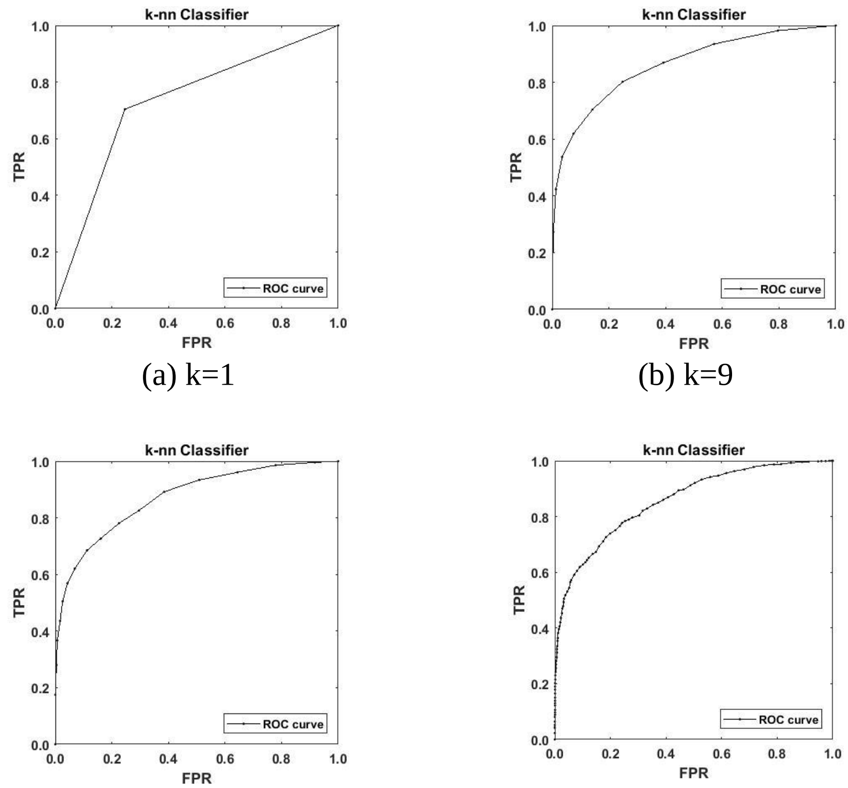
TABLE 6 : Details about the procedure of making KNN classifier

I.	Load the data provided for the home work
II.	Divide the data into positive and negative class
III.	Then split the positive and negative data in 75% for training and 25% for the testing. The split was performed complete randomly.
IV.	Then we combined the data positive and negative testing and training set into one Train_Data and Test_Data..
V.	Compute the probability of positive and negative class using the training data.
VI.	Afterward testing period started.
VII.	K value was set from 1 to 100 using for loop.
VIII.	Calculated the Euclidean distance between the test sample and all the samples in the training set, and make a distance matrix.
IX.	Afterward select and take the k shortest distances.
X.	Count was done to find how many belongs to Positive and Negative classes among the k distances.
XI.	According to the majority of the vote the class of the samples is selected. In case of even number of k , if number of positive and negative class candidates are same then we decide based of the probabilities of the classes.
XII.	Using the provided function **EvalROC** and **Rchplot** ROC and AUC (Area Under Curve) was calculated.
XIII.	TPR for FPR=0.1 was calculated.
XIV.	Finally, accuracy was calculated.

TABLE 7: KNN Classifier Performance Parameters.

Value of K	K=1	K=3	K=9	K=17	K=28	K=49	K=65	K=81	K=95	K=100	Average
Accuracy	0.7290	0.7755	0.7810	**0.7880**	0.7865	0.7760	0.7745	0.7685	0.7670	0.7640	**0.7757**
TPR	0.2862	0.5755	0.6507	0.6692	0.6712	0.6596	0.6502	0.6366	0.6330	0.6272	**0.6450**
AUC	0.72.90	0.8278	0.8611	0.8704	0.8701	0.8651	0.8646	0.8619	0.8594	0.8575	**0.8619**

From the table above, we can see the Accuracy of the Accuracy, TPR and AUC for different iteration. The parameters vary as the value of k varied during each iteration. The average accuracy is 77.57%, whereas maximum accuracy achieved was 78.80% for k=17. Average AUC and TPR was 86.19% and 64.50% respectively.



(a) k=1

(b) k=9

(c) k=16

(d) k=45

Fig. 6: ROC of Quadratic Classifier for different iteration.

Discussion: Among all the three classifier we implemented in this question KNN given the highest average accuracy 77.57%, approximately 0.9% higher than the average accuracy of the linear classifier which was 76.67%. Among all the classifiers quadratic classifier gives the lowest average accuracy.

According to the average AUC quadratic classifier lead the theme with average AUC of 0.8703 and lowest average AUC was given by linear classifier which was 0.8350.

Again, highest average TPR was given by KNN classifier which is 0.6450. Table 8. below shown the comparative performance ranking of the above classifier for this test.

TABLE 8: Comparative performance ranking of Classifiers

Ranking	Average Accuracy	Average TRP	AUC
1st	KNN Classifier **77.57%**	KNN Classifier **0.7757**	Quadratic Classifier **0.8703**
2nd	Linear Classifier **76.67%**	Quadratic Classifier **0.6127**	KNN Classifier **0.8619**
3rd	Quadratic Classifier 72.62%	Linear Classifier **0.5369**	Linear Classifier **0.8350**

Problem 1.5 [30%] Use the dataset of Problem 1.4 and, using a model among the ones you know (linear, quadratic, k NN), build the classifier with the best accuracy on that classi_cation problem. To this aim, you can consider several training/test splits with different percentages for training and test sets. Prepare a Matlab fuction called test.m (function y=test(A)) that implements your best choice classi_er. The function will accept a matrix A and return a vector y having the same number of rows in A. The matrix A will containseveral samples (one for each row) organized in the same way described in Problem 1.4, butwithout label (i.e. each row will contain containing a sample with 10 numerical features).The function will classify each of the samples A(i,:), providing the predicted class in y(i).Your function must be submitted and will be run on a separate matrix containing new testdata. Your grade will be based on the performance of your classi_er on the new test data,which will contain a very large number of examples generated from the same distribution.

Answer: Analyzing the performance of the classifier made earlier, KNN showed the best performance in terms of accuracy. So for this section KNN classifier is implemented in the function test.m, All the instruction given in the question was followed.

After trying different splitting percentage of data set and different value of k, k=5 is been implemented for a train and test set split of 75% and 25% respectively.

Code and Matlab m.files are attached in the assignment folder.

Reference:

[1] S. Suzuki et al., "Mass detection using deep convolutional neural network for mammographic computer-aided diagnosis," 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016, pp. 1382-1386.
doi: 10.1109/SICE.2016.7749265