Homework 4

(Total 180 pts)

Due 11:59 pm on November 4, 2022 (Friday)

Canvas submission of your answers (with required plots and source codes attached) in a single PDF file ('hw4.pdf'), and then submit the following source code files:
Q1.py, Q2.m, Q4.m, and Q5.m

1. (30 pts) Classification using a reduced dataset derived by PCA.
   Load the wine dataset in sklearn:
   https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html
   https://archive.ics.uci.edu/ml/datasets/wine
   (a) Train a Naïve Bayes classifier using the above dataset. What is the score of the classifier?
   (b) Apply principal component analysis (PCA) on the dataset (and ignore the class labels). Keep only one component with the largest variance.
   (c) Regarding this principal component, what are its variance and variance ratio (i.e., the percentage of variance explained by this component)?
   (d) Train another Naïve Bayes classifier using the principle component obtained in (b), and the corresponding class labels in the original dataset.
   (e) What is the score of this new classifier?
   (f) Compare the scheme in (d) using PCA and the scheme in (a) without using PCA, in terms of both the classification accuracies and the sizes of the training data. Justify your answer.
   (g) Fill in the table below with your answers:

(a) Classifier Score	(c) Variance	(c) Variance Ratio	(e) Classifier Score

   (h) Attach and upload your python code 'Q1.py'.

2. (60 pts) Mini-Project: Investigation on the tradeoffs between classification accuracy and the number of PCA components to keep.
   In Matlab, load the wine dataset:
   >> [data, label] = wine_dataset
   where class labels are represented by a 1 in either row 1, 2 or 3 of the matrix 'label'.
   (a) Similar to Q1, train a Naïve Bayes classifier using the above dataset.
   Note: In order to use *fitcnb*, you might need to reshape the 'data' matrix and convert the 'label' matrix to class index numbers.
   (b) What is the accuracy of this classifier? Compare with Q1(a).
   (c) Apply principal component analysis (PCA) on the dataset (and ignore the class labels).
   (d) Keep only one component with the largest variance. Regarding this principal component, what is its variance? Compare with Q1(c).
   (e) Train another Naïve Bayes classifier using the principle component obtained in (d), and the corresponding class labels. What is the accuracy of this new classifier? Compare with Q1(e).
   (f) Repeat steps (d) and (e), but each time keep one additional PCA components (following the order of decreasing component variances). This way, we can determine how the classification accuracy will change with training data being the first principal component, first two principal components, first three components, …, all the way to all components.

(g) Plot the curve showing how the classification accuracies change as a function of the number (n) of PCA components to keep, where $n = 1, 2, \ldots, 13$. Attach the plot.

(h) What is the classification accuracy when training the classifier using all of the PCA components? How does this accuracy compare to the accuracy obtained in (b), by using the original dataset? Justify your answers.

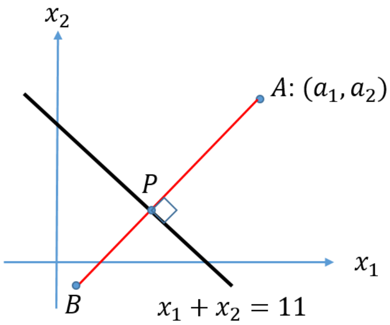
(i) Fill in the table below with the values:

(b) Accuracy	(d) Variance	(e) Accuracy (one component)	(h) Accuracy (all components)

(j) Attach and upload your script 'Q2.m'.

3. (20 pts) Geometry of discriminant function.

A dataset contains samples $x = (x_1, x_2)$ belonging to two linearly separable classes. Suppose a trained classifier has a decision boundary which is a straight line: $f(x_1, x_2) = x_1 + x_2 - 11 = 0$, as shown in the following figure.



For an arbitrary point A with coordinates (a_1, a_2), AB is a line segment perpendicular to the decision boundary line. AB intersects the decision boundary at point P.

(a) Determine analytically the coordinates (p_1, p_2), of the point P. Show your derivations. Simplify the expressions as much as possible. Fill in the blank below with your answer.
$(p_1, p_2) = ($ $\qquad\qquad$, $\qquad\qquad$ $)$.

(b) Determine analytically the Euclidean distance between point A and point P. Show your derivations. Simplify the expressions as much as possible. Fill in the blank below with your answer.
Distance between A and P = ($\qquad\qquad$).

4. (30 pts) Fisher's linear discriminant.

Read into Matlab the following dataset:
http://www.ece.uah.edu/~dwpan/course/ee610/hw/dataset_hw.csv

(a) Write a script to calculate the numerical values for the following terms associated with the Fisher's linear discriminant, where $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ are the mean vectors of Class 1 and Class 2, respectively, and $\boldsymbol{S_w}$ is the total within-class covariance matrix. Fill in the table below with your answers.

$\boldsymbol{m_2} - \boldsymbol{m_1}$ (2×1) vector	$\boldsymbol{S_w}$ (2×2) matrix	$\boldsymbol{W} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \boldsymbol{S_w^{-1}}(\boldsymbol{m_2} - \boldsymbol{m_1})$ (2×1) vector	$\left(-\frac{w_1}{w_2}\right)$ scalar
$\left(\qquad \right)$	$\left(\qquad \right)$	$\left(\qquad \right)$	($\qquad$)

(b) Attach and upload your script ‘Q4.m’.

5. (40 pts) Discriminant analysis classification in Matlab.
Read into Matlab the following dataset:
http://www.ece.uah.edu/~dwpan/course/ee610/hw/dataset_hw.csv
(a) Use the *gscatter* function to plot the dataset. Add legends to the scatter plot above to indicate the class indices for the samples. Specify the legend location to be ‘northwest’.
(b) Create a default (linear) discriminant analysis classifier using *fitcdiscr*.
(c) Retrieve the coefficients for the linear boundary between the first and second classes. Fill in the blanks below with the values for the line equation $ax_1 + bx_2 = c$, where
$a = (\qquad\qquad), \; b = (\qquad\qquad), \; c = (\qquad\qquad).$
The value of $\left(-\frac{a}{b}\right) = (\qquad\qquad).$
(d) Compare the value of $\left(-\frac{a}{b}\right)$ with the value of $\left(-\frac{w_1}{w_2}\right)$ in Q4(a). Are they the same? Why?
(e) On the scatter plot generated in (a), superimpose the line above (in green color) that separates the first and second classes. Attach the plot.
(f) Use *resubPredict* on the training data to predict the class labels, then display the confusion matrix chart.
(g) Use *resubLoss* to calculate the accuracy of this LDA classifier. Compare this accuracy with the accuracy of the Naïve Bayes classifier (Q2(d) in HW1). Which value is larger? Why?
(h) Attach and upload your script ‘Q5.m’.