Boston University
Department of Electrical and Computer Engineering

ENG EC 414 (Ishwar) Introduction to Machine Learning

HW 4

© 2015 – Spring 2022 Prakash Ishwar

Issued: Fri 4 Mar 2022 **Due:** 10:55pm Boston time Fri 18 Mar 2022 in Gradescope + Blackboard.
Required reading: Slides on LDA + OLS/ridge regression + your notes from lectures & Discussions 5, 6.

Important: Before you proceed, please read the documents pertaining to *Homework formatting and submission guidelines* and the *HW-grading policies* in the Homeworks section of Blackboard, especially guidelines for submitting reports in Gradescope. and code in Blackboard. In particular, for computer assignments *you are prohibited from using any online code or built-in MATLAB functions except as indicated in the problem or skeleton code (when provided).*
In order to receive full credit, all work should be supported by a concise explanation that is clear, relevant, specific, logical, and correct. In particular, for each part, you must clearly outline the key steps and provide proper justification for your calculations.

Note: Problem difficulty = number of coffee cups ☕

Problem 4.1 [15pts] *(LDA could perform poorly)* Consider a dataset $\mathcal{D}$ of 40 points in $\mathbb{R}^2$ with half the points in the positive class $\mathcal{D}_+$ and the remaining in the negative class $\mathcal{D}_-$. Specifically, let

$$\mathcal{D}_- = \left\{9 \text{ points at } \begin{pmatrix}-1\\0.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}-11\\0.5\end{pmatrix}, 9 \text{ points at } \begin{pmatrix}-1\\-1.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}-11\\-1.5\end{pmatrix}\right\}$$

$$\mathcal{D}_+ = \left\{9 \text{ points at } \begin{pmatrix}1\\1.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}11\\1.5\end{pmatrix}, 9 \text{ points at } \begin{pmatrix}1\\-0.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}11\\-0.5\end{pmatrix}\right\}$$

(a) Compute the mean vector and covariance matrix of each class and the average of the two class covariance matrices, i.e., compute $\hat{\boldsymbol{\mu}}_{\boldsymbol{x}-}, \hat{\boldsymbol{\mu}}_{\boldsymbol{x}+}, \hat{S}_{\boldsymbol{x}-}, \hat{S}_{\boldsymbol{x}+}, \hat{S}_{\boldsymbol{x}\cdot\text{avg}}$.

(b) Compute $\mathbf{w}_{\text{LDA}}$, b_{LDA}, and the CCR of the LDA classifier.

(c) Compute a linear binary classifier with the highest CCR and compare it with the LDA classifier.

Problem 4.2 [20pts] *(Linear Least Squares Regression)* Consider the following training set $\mathcal{D} = \{(x_1, y_1) = (-1, -1), (x_2, y_2) = (-1/2, -1/8), (x_3, y_3) = (0, 0), (x_4, y_4) = (1/2, 1/8), (x_5, y_5) = (1, 1)\}$. Hand-compute the following:

(a) [6pts] *Ordinary Least Squares*: $(w_{OLS}, b_{OLS}) = \arg\min_{w,b} \sum_{j=1}^{5} (y_j - w x_j - b)^2$.

(b) [14pts] *Polynomial Least Squares*: $(\mathbf{w}_{PLS}, b_{PLS}) = \arg\min_{\mathbf{w},b} \sum_{j=1}^{5} (y_j - \mathbf{w}^\top \boldsymbol{\phi}_j - b)^2$. Where $\boldsymbol{\phi}_j = ((x_j)^3, (x_j)^2)^\top$ and $\mathbf{w} = (w_3, w_2)^\top$.

Problem 4.3 [43pts] *(LDA)* In this problem we will implement LDA on a synthetic dataset and also develop geometric intuition for the eigenvalues and eigenvectors of the empirical feature covariance matrices. You are provided skeleton code to assist you in implementing LDA.

(a) [13pts] *(Synthetic dataset generation)* Write a Matlab function to generate as its output a 2-class labeled dataset consisting of 2D feature vectors each drawn independently from one of two Gaussian distributions (one distribution for each class). The function's inputs are (1) a specified number of examples n_1, n_2 for each distribution (class), (2) specified 2×1 mean vectors $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ for each class, and (3) a single **common** 2×2 covariance matrix for both classes specified via two real, nonnegative eigenvalues $\lambda_1, \lambda_2 \geq 0$, and a single orientation variable θ that defines their corresponding orthonormal eigenvectors $\mathbf{u}_1 = (\cos\theta, \sin\theta)^\top$ and $\mathbf{u}_2 = (\sin\theta, -\cos\theta)^\top$.

Use this function to generate $n_1 = 50$ class 1 examples and $n_2 = 100$ class 2 examples with $\boldsymbol{\mu}_1 = (1,2)^\top, \boldsymbol{\mu}_2 = (3,2)^\top$ and the following four choices of the λ's and θ: (i) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = 0$, (ii) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = \pi/6$, (iii) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = \pi/3$, (iv) $\lambda_1 = 0.25, \lambda_2 = 1, \theta = \pi/6$. For each of these four choices, create a scatter plot with class 1 examples shown as solid blue circles and class 2 examples shown as solid red triangles. Discuss how the eigenvalues and eigenvectors affect the geometry of the dataset.

Note: For the rest of this problem, we will work with the dataset created by the **second** choice, i.e., choice number (ii), out of the four choices in part (a).

(b) [17pts] Write a Matlab function which takes as inputs a 2D labeled dataset for binary classification and a unit direction (specified by an angle ϕ in radians) and returns the following three outputs (1) the squared Euclidean distance between the class means of the projections of the feature vectors along the direction $(\cos\phi, \sin\phi)^\top$, i.e., $(\hat{\mu}_{2z}(\phi) - \hat{\mu}_{1z}(\phi))^2$, where $\hat{\mu}_{1z}(\phi)$ and $\hat{\mu}_{2z}(\phi)$ are, respectively, the class 1 and class 2 empirical means along direction ϕ, (2) the average within-class variance of the projections of the feature vectors along direction $(\cos\phi, \sin\phi)^\top$, i.e., $\frac{n_1}{n}\hat{\sigma}^2_{1z}(\phi) + \frac{n_2}{n}\hat{\sigma}^2_{2z}(\phi)$, where $\hat{\sigma}^2_{1z}(\phi)$ and $\hat{\sigma}^2_{2z}(\phi)$ are, respectively, the class 1 and class 2 empirical variances along direction ϕ, and (3) their ratio, i.e., Signal-to-Noise Ratio (SNR).

Use this function to plot all three quantities as a function of ϕ, for ϕ ranging from 0 to π in steps of $\pi/48$, for the dataset of part(a)(ii). Determine (i) the value of ϕ which maximizes the squared distance between the class means of the projected values (ii) the value of ϕ which minimizes the average within-class variance, and (iii) the value of ϕ which maximizes the SNR. (iv) Also use Matlab's `ksdensity` function with default settings to plot estimates of the class 1 and class 2 probability densities of the projections of the feature vectors for $\phi = 0, \pi/6, \pi/3$. Discuss your findings.

(c) [6pts] Write a Matlab function which takes as input a 2D labeled dataset for binary classification and outputs the LDA vector $\mathbf{w}_{LDA} = \hat{S}^{-1}_{x.\text{avg}}(\hat{\boldsymbol{\mu}}_{2x} - \hat{\boldsymbol{\mu}}_{1x})$ where $\hat{S}_{x.\text{avg}} = \frac{n_1}{n}\hat{S}_{1x} + \frac{n_2}{n}\hat{S}_{2x}$ and $\hat{S}_{1x}$ and $\hat{S}_{2x}$ are the empirical 2×2 covariance matrices of classes 1 and 2 respectively.

Use this function to compute $\mathbf{w}_{LDA}$ for the dataset of part(a)(ii). Compare it with the difference between the class 2 and class 1 empirical mean vectors $(\hat{\boldsymbol{\mu}}_{2x} - \hat{\boldsymbol{\mu}}_{1x})$. Overlay both these vectors on the scatter plot of the dataset with the vectors represented as arrows starting at the location of $\hat{\boldsymbol{\mu}}_{1x}$. Discuss what you observe. Also compare the direction of $\mathbf{w}_{LDA}$ with the value of ϕ from part (b) which maximizes the SNR.

(d) [7pts] Consider the following separating hyperplane decision rule for binary classification based on thresholding a linear (affine) function of the feature vector: $h_{\mathbf{w},b}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^\top\mathbf{x} + b \leq 0 \\ 2 & \text{else} \end{cases}$. Write a Matlab function which takes as inputs a 2D labeled dataset for binary classification, $\mathbf{w}$, and b as inputs and outputs the value of CCR that results from applying $h_{\mathbf{w},b}(\mathbf{x})$ to classify all feature vectors in the dataset into class 1 or class 2.

Use this function to create a plot of CCR as a function of b for the dataset of part(a)(ii) and $\mathbf{w} = \mathbf{w}_{LDA}$. Compute the value of the offset parameter b which maximizes the CCR and the resulting CCR value.

Problem 4.4 [22pts] *(ridge regression)* In this problem we will implement ridge regression and apply it to a real-world 8-dimensional (8 features) prostate cancer dataset contained in the file `prostateStnd.mat`. In this dataset, 8 medically relevant features named lcavol, lweight, age, lbph, svi, lcp, gleason, and pgg45 are used to estimate lpsa (log prostate specific antigen). The training and test data are provided as `(xtrain, ytrain)` and `(xtest, ytest)` respectively. The first 8 features correspond to the first 8 entries of `names`. The ninth entry of `names` (the last one) is the label to be predicted whose values are in `(ytest, ytrain)`.

(a) [4pts] As a first step, write Matlab code to normalize the **training dataset** so that post-normalization, each of the 8 features and the label in the normalized training dataset has zero mean and unit variance. This requires determining a *pair* of offset and scaling parameters, one pair for each feature and one pair for the label. These parameters must be computed only from the training dataset, but they must be applied to both the training and test datasets, i.e., we normalize both the training and test data, but we are only allowed to normalize the test data using parameters derived from the training data. In other words we must apply identical operations to training and test data. Only the training data will be actually normalized by the operation. If the test data is statistically similar to the training data, it too will be approximately normalized. **Report** the mean and variance of each feature and the label *before* normalization.

(b) [6pts] Next, write Matlab code to use the normalized data to train a ridge regression model for each of the following values of the quadratic regularization penalty parameter λ: $\{e^{-5}, e^{-4}, e^{-3}, ..., e^{10}\}$.

(c) [9pts] In a single figure, plot the ridge regression coefficient of each feature (8 in total) as a function of $\ln \lambda$ (8 curves in total) for $\ln \lambda$ ranging from -5 to 10 in steps of 1. Use suitable colors and/or markers to distinguish between the 8 curves and label them appropriately in a legend. Discuss what happens to the coefficients as λ becomes larger.

(d) [3pts] In another figure, plot the mean-squared-error (MSE) of both the training and the test data as a function of $\ln \lambda$. Discuss your observations.

Code-submission via Blackboard: Create two dot-m files. One named `<yourBUemailID>_hw4_3.m` for Problem 4.3 and one named `<yourBUemailID>_hw4_4.m` for Problem 4.4. All local functions and scripts pertaining to one problem should appear within the single dot-m file for that problem. When run, your scripts should be able to display in the command window whatever you are asked to compute and report in each part. One skeleton code is provided for your reference: `skeleton_hw4_3.m`. Reach-out to the TA via Piazza and office hours for questions related to coding. When submitting code, please include a 'Readme' text file in your source code directory describing approximate running times of different parts, any additional comments (as needed) explaining how to use your code, and any dependencies between different parts. Please do not include into the directory, any data files that are already provided. Write your code under the assumption that all data files are in the same directory as your source code.

Problem 4.1 [15pts] *(LDA could perform poorly)* Consider a dataset $\mathcal{D}$ of 40 points in $\mathbb{R}^2$ with half the points in the positive class $\mathcal{D}_+$ and the remaining in the negative class $\mathcal{D}_-$. Specifically, let

$$\mathcal{D}_- = \left\{9 \text{ points at } \begin{pmatrix}-1\\0.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}-11\\0.5\end{pmatrix}, 9 \text{ points at } \begin{pmatrix}-1\\-1.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}-11\\-1.5\end{pmatrix}\right\}$$

$$\mathcal{D}_+ = \left\{9 \text{ points at } \begin{pmatrix}1\\1.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}11\\1.5\end{pmatrix}, 9 \text{ points at } \begin{pmatrix}1\\-0.5\end{pmatrix},\ \text{one point at } \begin{pmatrix}11\\-0.5\end{pmatrix}\right\}$$

(a) Compute the mean vector and covariance matrix of each class and the average of the two class covariance matrices, i.e., compute $\hat{\mu}_{x-}, \hat{\mu}_{x+}, \hat{S}_{x-}, \hat{S}_{x+}, \hat{S}_{x\text{-avg}}$.

(b) Compute $\mathbf{w}_{\text{LDA}}$, b_{LDA}, and the CCR of the LDA classifier.

(c) Compute a linear binary classifier with the highest CCR and compare it with the LDA classifier. ← find a better line... eyeball it.

a) $X_- = \begin{vmatrix} 9(-1) & -11 & 9(-1) & -11 \\ 9(1/2) & 1/2 & 9(-3/2) & -3/2 \end{vmatrix}$

Remember $\hat{\mu} = \frac{1}{n}\sum_{j:y_j=1}^{n} x_j$

$\hat{\mu}_- = \frac{1}{20}\begin{vmatrix} -40 \\ -10 \end{vmatrix} = \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$

$\hat{\mu}_- = \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$

$X_+ = \begin{vmatrix} 9(1) & 11 & 9(1) & 11 \\ 9(3/2) & 3/2 & 9(-1/2) & -1/2 \end{vmatrix}$

$\hat{\mu}_+ = \frac{1}{20}\begin{vmatrix} 40 \\ 10 \end{vmatrix} = \begin{vmatrix} 2 \\ 1/2 \end{vmatrix}$

$\hat{\mu}_+ = \begin{vmatrix} 2 \\ 1/2 \end{vmatrix}$

$S = \frac{1}{n}\sum_{j=1}^{n} \bar{x}_j \cdot \bar{x}_j^T$

$\bar{x}_1 = \begin{vmatrix} -1 \\ 1/2 \end{vmatrix} - \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$

$\bar{x}_1 = \begin{vmatrix} 1 \\ 1 \end{vmatrix}$

$\bar{x}_1 \cdot \bar{x}_1^T = \underset{2\times 1}{\begin{vmatrix} 1 \\ 1 \end{vmatrix}} \cdot \underset{1\times 2}{\begin{vmatrix} 1 & 1 \end{vmatrix}} = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$

$9(\bar{x}_1 \cdot \bar{x}_1^T) = \begin{vmatrix} 9 & 9 \\ 9 & 9 \end{vmatrix}$

$\bar{x}_2 = \begin{vmatrix} -11 \\ 1/2 \end{vmatrix} - \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$

$\bar{x}_2 = \begin{vmatrix} -9 \\ 1 \end{vmatrix}$

$\bar{x}_2 \cdot \bar{x}_2^T = \begin{vmatrix} -9 \\ 1 \end{vmatrix} \cdot \begin{vmatrix} -9 & 1 \end{vmatrix} = \begin{vmatrix} 81 & -9 \\ -9 & 1 \end{vmatrix}$

$\bar{x}_3 = \begin{vmatrix} -1 \\ -3/2 \end{vmatrix} - \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$

$\bar{x}_3 = \begin{vmatrix} 1 \\ -1 \end{vmatrix}$

$\bar{x}_3 \cdot \bar{x}_3^T = \begin{vmatrix} 1 \\ -1 \end{vmatrix} \cdot \begin{vmatrix} 1 & -1 \end{vmatrix} = \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}$

$9(\bar{x}_3 \cdot \bar{x}_3^T) = \begin{vmatrix} 9 & -9 \\ -9 & 9 \end{vmatrix}$

$$\bar{x}_4 = \begin{vmatrix} -11 \\ -3\,|2 \end{vmatrix} - \begin{vmatrix} -2 \\ -1/2 \end{vmatrix}$$

$$\bar{x}_4 = \begin{vmatrix} -9 \\ -1 \end{vmatrix}$$

$$\bar{x}_4 \cdot \bar{x}_4^T = \begin{vmatrix} -9 \\ -1 \end{vmatrix} \cdot \begin{vmatrix} -9 & -1 \end{vmatrix} = \begin{vmatrix} 81 & 9 \\ 9 & 1 \end{vmatrix}$$

$$\sum_{j=1}^{n} \bar{x}_j \cdot \bar{x}_j^T = \begin{vmatrix} 9 & 9 \\ 9 & 9 \end{vmatrix} + \begin{vmatrix} 81 & -9 \\ -9 & 1 \end{vmatrix}$$

$$+ \begin{vmatrix} 9 & -9 \\ -9 & 9 \end{vmatrix} + \begin{vmatrix} 81 & 9 \\ 9 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} 9+81+9+81 & 0 \\ 0 & 20 \end{vmatrix}$$

$$\begin{array}{r} 81 \\ +81 \\ \hline 162 \\ +18 \\ \hline 180 \end{array}$$

$$= \begin{vmatrix} 180 & 0 \\ 0 & 20 \end{vmatrix}$$

$$\hat{S}_- = 1/20 \begin{vmatrix} 180 & 0 \\ 0 & 20 \end{vmatrix} = \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}$$

$$\boxed{\hat{S}_- = \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}}$$

$$x_+ = \begin{vmatrix} 9(1) & 11 & 9(1) & 11 \\ 9(3/2) & 1/2 & 9(-1/2) & -1/2 \end{vmatrix}$$

$$\bar{x}_1 = \begin{vmatrix} 1 \\ 3/2 \end{vmatrix} - \begin{vmatrix} 2 \\ 1/2 \end{vmatrix}$$

$$\bar{x}_1 = \begin{vmatrix} -1 \\ 1 \end{vmatrix}$$

$$\bar{x}_1 \cdot \bar{x}_1^T = \begin{vmatrix} -1 \\ 1 \end{vmatrix} \cdot \begin{vmatrix} -1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}$$

$$9 \left(\bar{x}_1 \cdot \bar{x}_1^T \right) = \begin{vmatrix} 9 & -9 \\ -9 & 9 \end{vmatrix}$$

$$\bar{x}_2 = \begin{vmatrix} 11 \\ 3/2 \end{vmatrix} - \begin{vmatrix} 2 \\ 1/2 \end{vmatrix} = \begin{vmatrix} 9 \\ 1 \end{vmatrix}$$

$$\bar{x}_2 \cdot \bar{x}_2^T = \begin{vmatrix} 9 \\ 1 \end{vmatrix} \cdot \begin{vmatrix} 9 & 1 \end{vmatrix} = \begin{vmatrix} 81 & 9 \\ 9 & 1 \end{vmatrix}$$

$$\bar{x}_3 = \begin{vmatrix} 1 \\ -1/2 \end{vmatrix} - \begin{vmatrix} 2 \\ 1/2 \end{vmatrix} = \begin{vmatrix} -1 \\ -1 \end{vmatrix}$$

$$\bar{x}_3 \cdot \bar{x}_3^T = \begin{vmatrix} -1 \\ -1 \end{vmatrix} \cdot \begin{vmatrix} -1 & -1 \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$$

$$9 \left(\bar{x}_3 \cdot \bar{x}_3^T \right) = \begin{vmatrix} 9 & 9 \\ 9 & 9 \end{vmatrix}$$

$$\bar{x}_4 = \begin{vmatrix} 11 \\ -1/2 \end{vmatrix} - \begin{vmatrix} 2 \\ 1/2 \end{vmatrix} = \begin{vmatrix} 9 \\ -1 \end{vmatrix}$$

$$\bar{x}_4 \cdot \bar{x}_4^T = \begin{vmatrix} 9 \\ -1 \end{vmatrix} \cdot \begin{vmatrix} 9 & -1 \end{vmatrix} = \begin{vmatrix} 81 & -9 \\ -9 & 1 \end{vmatrix}$$

$$\sum_{j=1}^{n} \bar{x}_j \cdot \bar{x}_j^T = \begin{vmatrix} 9 & 9 \\ 9 & 9 \end{vmatrix} + \begin{vmatrix} 81 & -9 \\ -9 & 1 \end{vmatrix}$$

$$+ \begin{vmatrix} 9 & -9 \\ -9 & 9 \end{vmatrix} + \begin{vmatrix} 81 & 9 \\ 9 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} 9+81+9+81 & 0 \\ 0 & 20 \end{vmatrix}$$

$$= \begin{vmatrix} 180 & 0 \\ 0 & 20 \end{vmatrix}$$

$$\hat{S}_+ = 1/20 \begin{vmatrix} 180 & 0 \\ 0 & 20 \end{vmatrix} = \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}$$

$$\boxed{\hat{S}_+ = \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}}$$

$\hat{P}_- = 20/40 = 1/2$

$\hat{P}_+ = 1/2$

$b_{LDA} := \operatorname{argmax}_{b \in \mathbb{R}} \left\{ \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\left(h_{WLDA,b}(x_j) = y_j\right) \right\}$

$\hat{S}_{x\cdot avg} = 1/2 \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix} + 1/2 \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}$ Need to project w^T to even point.

$\hat{S}_{x\cdot avg} = \begin{vmatrix} 9 & 0 \\ 0 & 1 \end{vmatrix}$

$z_i = w^T x_i$

(4) + $z_A = |4/9| \left|\begin{matrix} -1 \\ 0.5 \end{matrix}\right| = -4/9 + 1/2 =$

(4) + $z_A = 1/18 = 0.05$

+ $z_B = |4/9| \left|\begin{matrix} -11 \\ 1/2 \end{matrix}\right| = -11(4/9) + 1/2$

+ $z_B = -4.38$

b) $(\hat{S}_{x\cdot avg})^{-1} = \frac{1}{9} \begin{vmatrix} 1 & 0 \\ 0 & 9 \end{vmatrix}$

$= \begin{vmatrix} 1/9 & 0 \\ 0 & 1 \end{vmatrix}$

(4) + $z_C = |4/9| \left|\begin{matrix} -1 \\ -3/2 \end{matrix}\right| = 4/9 + -3/2$

(4) + $z_C = -1.05$

$\hat{\mu}_{x+} - \hat{\mu}_{x-} = \begin{vmatrix} 2 \\ 1/2 \end{vmatrix} - \left|\begin{matrix} -2 \\ -1/2 \end{matrix}\right|$

$= \begin{vmatrix} 4 \\ 1 \end{vmatrix}$

+ $z_D = |4/9| \left|\begin{matrix} -11 \\ -3/2 \end{matrix}\right| = 4/9(-11) + -3/2$

+ $z_D = -6.38$

$w_{LDA} = (\hat{S}_{x\cdot avg})^{-1} (\hat{\mu}_{x_2} - \hat{\mu}_{x_1})$

(4) $z_E = |4/9| \left|\begin{matrix} 1 \\ 3/2 \end{matrix}\right| = 4/9 + 3/2$

(4) $z_E = 1.94$

$w_{LDA} = \begin{vmatrix} 1/9 & 0 \\ 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 4 \\ 1 \end{vmatrix}$ $z_A = |4/9| \left|\begin{matrix} 11 \\ 3/2 \end{matrix}\right| = 4/9(11) + 3/2$

2x2 2x1

$z_A = 6.38$

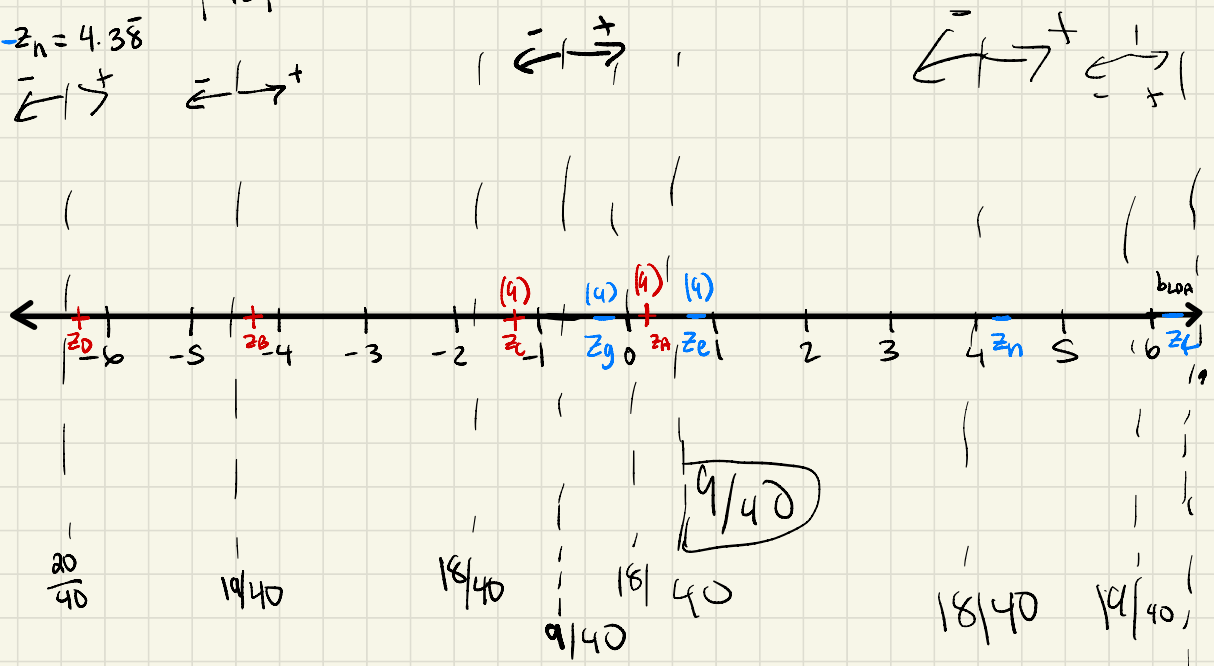
$w_{LDA} = \begin{vmatrix} 4/9 \\ 1 \end{vmatrix}$

(4) $z_G = |4/9| \left|\begin{matrix} 1 \\ -1/2 \end{matrix}\right| = 4/9 + -1/2$

(4) $z_G = -0.05$

$z_h = |4|9 \; 1 \; | \begin{vmatrix} 11 \\ -1/2 \end{vmatrix} = 4|9(11) + -1/2$

$z_h = 4.3\overline{8}$

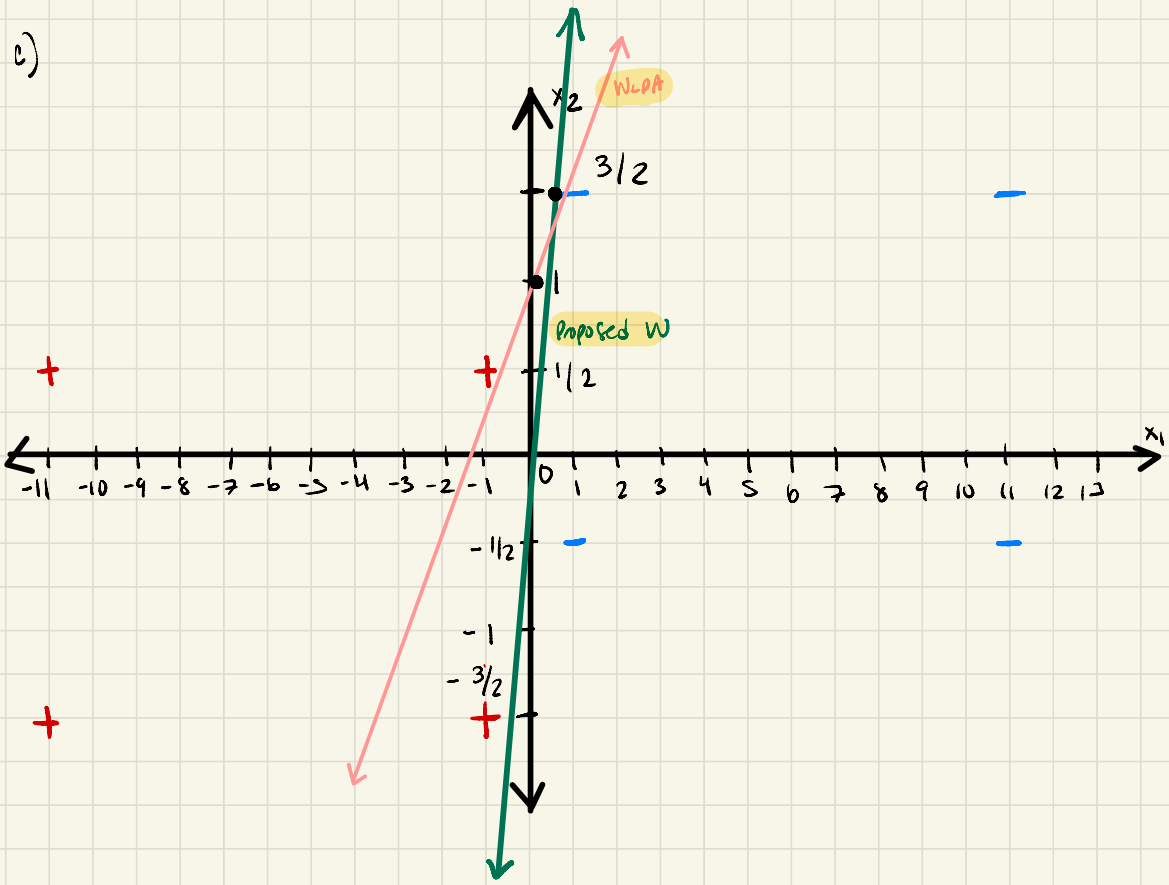


Best CCR ≡ 1/2 (half right / half wrong)

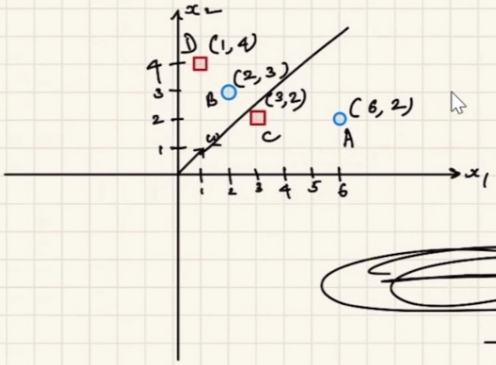
$b_{LOA} = \begin{cases} b_{LOA} > 6.38 \\ b_{LOA} < -6.38 \\ 0 \text{ else} \end{cases}$

$b_{LOA} = [-8, 8]$

c)



$W = \begin{vmatrix} 1/2 \\ 3/2 \end{vmatrix}$ all points are correct CCR = 1



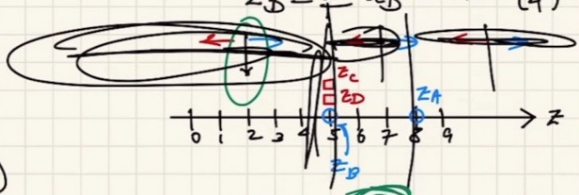
$$\underline{w} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$z_A = \underline{w}^T \underline{x}_A = (1\ 1)\begin{pmatrix} 6 \\ 2 \end{pmatrix} = 8$$

$$z_B = \underline{w}^T \underline{x}_B = (1\ 1)\begin{pmatrix} 2 \\ 3 \end{pmatrix} = 5$$

$$z_C = \underline{w}^T \underline{x}_C = (1\ 1)\begin{pmatrix} 3 \\ 2 \end{pmatrix} = 5$$

$$z_D = \underline{w}^T \underline{x}_D = (1\ 1)\begin{pmatrix} 1 \\ 4 \end{pmatrix} = 5$$



$$h(x) = \text{sign}(\underline{w}^T \underline{x} + b)$$

$$= \begin{cases} + & \text{if } \underline{w}^T \underline{x} + b > 0 \\ - & \underline{w}^T \underline{x} + b < 0 \end{cases}$$

CCR $= \frac{1}{2} = \frac{2}{4}$ $\quad \frac{3}{4}$ $\quad \frac{2}{4} = \frac{1}{2}$

Problem 4.2 [20pts] *(Linear Least Squares Regression)* Consider the following training set $\mathcal{D} = \{(x_1, y_1) = (-1, -1), (x_2, y_2) = (-1/2, -1/8), (x_3, y_3) = (0, 0), (x_4, y_4) = (1/2, 1/8), (x_5, y_5) = (1, 1)\}$. Hand-compute the following:

(a) [6pts] *Ordinary Least Squares*: $(w_{OLS}, b_{OLS}) = \arg\min_{w,b} \sum_{j=1}^{5} (y_j - wx_j - b)^2$.

(b) [14pts] *Polynomial Least Squares*: $(\mathbf{w}_{PLS}, b_{PLS}) = \arg\min_{\mathbf{w},b} \sum_{j=1}^{5} (y_j - \mathbf{w}^\top \boldsymbol{\phi}_j - b)^2$. Where $\boldsymbol{\phi}_j = ((x_j)^3, (x_j)^2)^\top$ and $\mathbf{w} = (w_3, w_2)^\top$.

a) $X = \begin{vmatrix} -1 & -1/2 & 0 & 1/2 & 1 \end{vmatrix}$

$Y = \begin{vmatrix} -1 & -1/8 & 0 & 1/8 & 1 \end{vmatrix}$

$\hat{\mu}_x = (-1) + -1/2 + 1/2 + 1$

$\hat{\mu}_x = \begin{vmatrix} 0 \end{vmatrix}$

$\hat{\mu}_y = \begin{vmatrix} -1 & -1/8 & 0 & 1/8 & 1 \end{vmatrix}$ 1/5

$\hat{\mu}_y = 0$

$\bar{X} = X$

$\bar{Y} = Y$

$\hat{S}_x = \frac{1}{n} \bar{X} \cdot \bar{X}^T$

$\hat{S}_x = \frac{1}{5} \begin{vmatrix} -1 & -1/2 & 0 & 1/2 & 1 \end{vmatrix} \begin{vmatrix} -1 \\ -1/2 \\ 0 \\ 1/2 \\ 1 \end{vmatrix}$ (1×5, 5×1)

$\hat{S}_x = 1/2$

$\hat{S}_{xy} = \frac{1}{n} \bar{X} \bar{Y}^T$

$\hat{S}_{xy} = \frac{1}{5} \begin{vmatrix} -1 & -1/2 & 0 & 1/2 & 1 \end{vmatrix} \begin{vmatrix} -1 \\ -1/8 \\ 0 \\ 1/8 \\ 1 \end{vmatrix}$

$\hat{S}_{xy} = 17/40$

$w_{OLS} = (S_x)^{-1} \cdot S_{xy}$

$w_{OLS} = 2 \cdot \frac{17}{40} = \frac{34}{40}$

$w_{OLS} = 34/40$

$b_{OLS} = \hat{\mu}_Y - (w_{OLS}^T) \hat{\mu}_X$

$b_{OLS} = 0$

b) $w^T \phi_j + b = w_3 (x_j)^3 + w_2 (x_j)^2 + b$

$\phi_1 = ((-1)^3, (-1)^2) = (-1, 1)$

$\phi_1 = (-1, 1)$

$\phi_2 = ((-1/2)^3, (-1/2)^2)$

$\phi_2 = (-1/8, 1/4)$

$\phi_3 = (0, 0)$

$\phi_4 = (1/8, 1/4)$

$\phi_5 = (1, 1)$

$(w_{PLS}, b_{PLS}) = \arg\min_{w,b} \sum_{j=1}^{5} (y_j - w^T \phi_j - b)^2$

Need to find w_3

$$X = \begin{vmatrix} -1^3 & -1/2^3 & 0^3 & 1/2^3 & 1^3 \end{vmatrix}$$

$$x = \begin{vmatrix} -1 & -1/8 & 0 & 1/8 & 1 \end{vmatrix}$$

$$\hat{\mu}_x = 0$$

$$\hat{S}_x = \frac{1}{5} \begin{vmatrix} -1 & -1/8 & 0 & 1/8 & 1 \end{vmatrix} \begin{vmatrix} -1 \\ -1/8 \\ 0 \\ 1/8 \\ 1 \end{vmatrix}$$

$$\hat{S}_x = (1 + 1/8(-1/8) + 1/8(1/8) + 1)^{1/5}$$

$$\hat{S}_x = (2 + 2/16)^{1/5} = 17/40$$

$$\hat{S}_{xy} = \frac{1}{5} \begin{vmatrix} -1 & -1/8 & 0 & 1/8 & 1 \end{vmatrix} \begin{vmatrix} -1 \\ -1/8 \\ 0 \\ 1/8 \\ 1 \end{vmatrix}$$

$$\hat{S}_{xy} = 17/40$$

$$w_3 = 40/17 \cdot \frac{17}{40} = 1$$

$$\boxed{w_3 = 1}$$

Need to find w_2

$$X = \begin{vmatrix} -1^2 & -1/2^2 & 0^2 & 1/2^2 & 1^2 \end{vmatrix}$$

$$x = \begin{vmatrix} 1 & 1/4 & 0 & 1/4 & 1 \end{vmatrix}$$

$$\hat{\mu}_x = 1/5\,(2\,1/4) = 1/2$$

$$\bar{X} = \hat{X} - \hat{\mu}_x = \begin{vmatrix} 1/2 & -1/4 & -1/2 & -1/2 & 1/2 \end{vmatrix}$$

$$\hat{S}_x = \frac{1}{5} \begin{vmatrix} 1/2 & -1/4 & -1/2 & -1/2 & 1/2 \end{vmatrix} \begin{vmatrix} 1/2 \\ -1/4 \\ -1/2 \\ -1/2 \\ 1/2 \end{vmatrix}$$

$$\hat{S}_x = \Big(1/2(1/2) + -1/4(-1/4) + -1/2(-1/2) + -1/2(-1/2) + 1/2(1/2)\Big)^{1/5}$$

$$\hat{S}_x = 17/40$$

$$\hat{S}_{xy} = \frac{1}{5} \begin{vmatrix} 1/2 & -1/4 & -1/2 & -1/2 & 1/2 \end{vmatrix} \begin{vmatrix} -1 \\ -1/8 \\ 0 \\ 1/8 \\ 1 \end{vmatrix}$$

$$\hat{S}_{xy} = 0$$

$$\boxed{w_2 = 0}$$

$$\phi_1 : \left(-1 - \begin{vmatrix} w_3 & w_2 \end{vmatrix} \begin{vmatrix} -1 \\ 1 \end{vmatrix} - b\right)^2$$

$$(-1 + w_3 - w_2 - b)^2$$

$$(\cancel{-1} + \cancel{1} - b)^2$$

$$(-b)^2$$

$$\phi_2 : \left(-1/8 - \begin{vmatrix} w_3 & w_2 \end{vmatrix} \begin{vmatrix} -1/8 \\ -1/4 \end{vmatrix} - b\right)^2$$

$$(\cancel{-1/8} + 1/8\cancel{w_3} + w_2/4 - b)^2$$

$$(-b)^2$$

$$\phi_3 : \left(-1 \begin{vmatrix} w_3 & w_2 \end{vmatrix} \begin{vmatrix} -1/8 \\ -1/4 \end{vmatrix} - b\right)^2$$

$$(w_3/8 + w/4 - b)^2$$

$$(1/8 - b)^2$$

θ_4: $\left(1/8 - \begin{vmatrix} w_3 & w_2 \end{vmatrix} \begin{vmatrix} 1/8 \\ 1/4 \end{vmatrix} - b\right)^2$

$\left(1/8 - w_3/8 - w_2/4 - b\right)^2$

$(-b)^2$

θ_5: $\left(1 - \begin{vmatrix} w_3 & w_2 \end{vmatrix} \begin{vmatrix} 1 \\ 1 \end{vmatrix} - b\right)^2$

$(1 - w_3 - w_2 - b)^2$

$(-b)^2$

$$\underset{w,b \in \mathbb{R}^2}{\arg\min} \left[(-b)^2 + (-b)^2 + (1/8 - b^2) + (-b)^2 + (-b)^2\right]$$

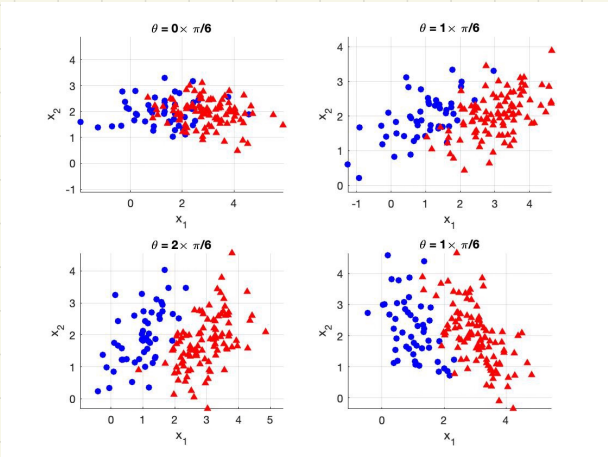
$$\frac{1/8 - 5b}{5} = 0 \rightarrow 1/8 = 5b$$

$$b = 1/40$$

Problem 4.3 [43pts] *(LDA)* In this problem we will implement LDA on a synthetic dataset and also develop geometric intuition for the eigenvalues and eigenvectors of the empirical feature covariance matrices. You are provided skeleton code to assist you in implementing LDA.

(a) [13pts] *(Synthetic dataset generation)* Write a Matlab function to generate as its output a 2-class labeled dataset consisting of 2D feature vectors each drawn independently from one of two Gaussian distributions (one distribution for each class). The function's inputs are (1) a specified number of examples n_1, n_2 for each distribution (class), (2) specified 2×1 mean vectors $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ for each class, and (3) a single **common** 2×2 covariance matrix for both classes specified via two real, nonnegative eigenvalues $\lambda_1, \lambda_2 \geq 0$, and a single orientation variable θ that defines their corresponding orthonormal eigenvectors $\mathbf{u}_1 = (\cos\theta, \sin\theta)^\top$ and $\mathbf{u}_2 = (\sin\theta, -\cos\theta)^\top$.

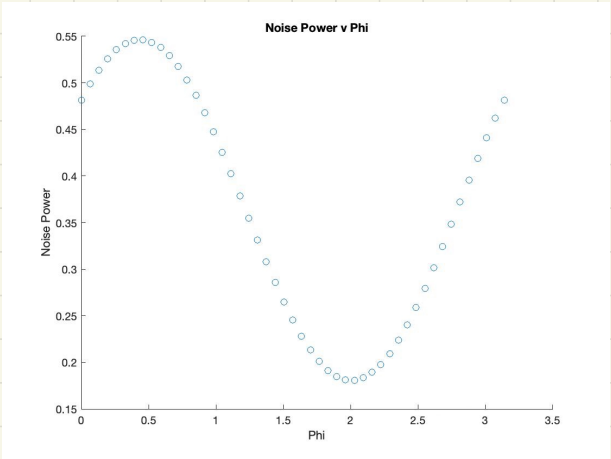
Use this function to generate $n_1 = 50$ class 1 examples and $n_2 = 100$ class 2 examples with $\boldsymbol{\mu}_1 = (1,2)^\top, \boldsymbol{\mu}_2 = (3,2)^\top$ and the following four choices of the λ's and θ: (i) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = 0$, (ii) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = \pi/6$, (iii) $\lambda_1 = 1, \lambda_2 = 0.25, \theta = \pi/3$, (iv) $\lambda_1 = 0.25, \lambda_2 = 1, \theta = \pi/6$. For each of these four choices, create a scatter plot with class 1 examples shown as solid blue circles and class 2 examples shown as solid red triangles. Discuss how the eigenvalues and eigenvectors affect the geometry of the dataset.

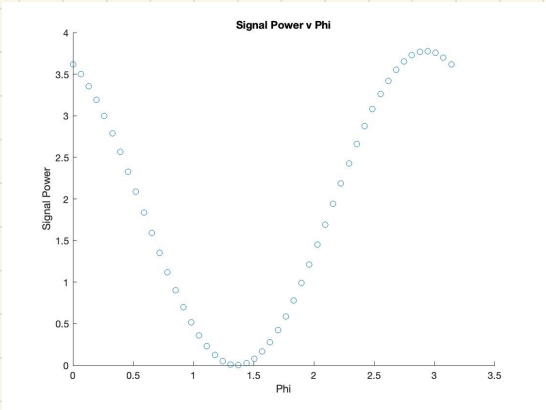


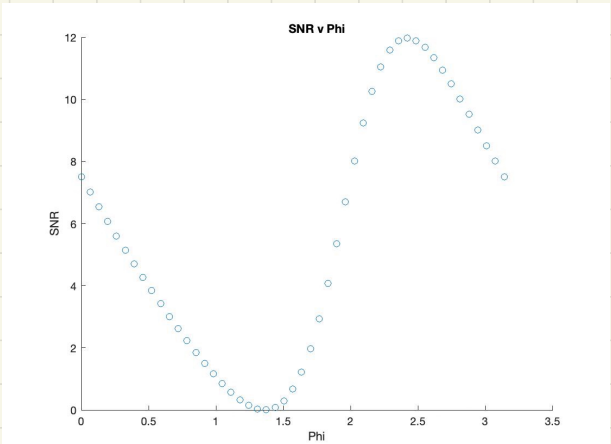
Eigenvalues change the orientation of the points. It adjusts the slope and placement of data.

(b) [17pts] Write a Matlab function which takes as inputs a 2D labeled dataset for binary classification and a unit direction (specified by an angle ϕ in radians) and returns the following three outputs (1) the squared Euclidean distance between the class means of the projections of the feature vectors along the direction $(\cos\phi, \sin\phi)^\top$, i.e., $(\hat{\mu}_{2z}(\phi) - \hat{\mu}_{1z}(\phi))^2$, where $\hat{\mu}_{1z}(\phi)$ and $\hat{\mu}_{2z}(\phi)$ are, respectively, the class 1 and class 2 empirical means along direction ϕ, (2) the average within-class variance of the projections of the feature vectors along direction $(\cos\phi, \sin\phi)^\top$, i.e., $\frac{n_1}{n}\hat{\sigma}^2_{1z}(\phi) + \frac{n_2}{n}\hat{\sigma}^2_{2z}(\phi)$, where $\hat{\sigma}^2_{1z}(\phi)$ and $\hat{\sigma}^2_{2z}(\phi)$ are, respectively, the class 1 and class 2 empirical variances along direction ϕ, and (3) their ratio, i.e., Signal-to-Noise Ratio (SNR).

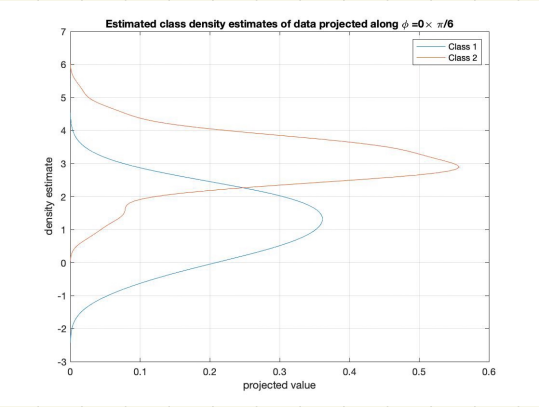
Use this function to plot all three quantities as a function of ϕ, for ϕ ranging from 0 to π in steps of $\pi/48$, for the dataset of part(a)(ii). Determine (i) the value of ϕ which maximizes the squared distance between the class means of the projected values (ii) the value of ϕ which minimizes the average within-class variance, and (iii) the value of ϕ which maximizes the SNR. (iv) Also use Matlab's `ksdensity` function with default settings to plot estimates of the class 1 and class 2 probability densities of the projections of the feature vectors for $\phi = 0, \pi/6, \pi/3$. Discuss your findings.

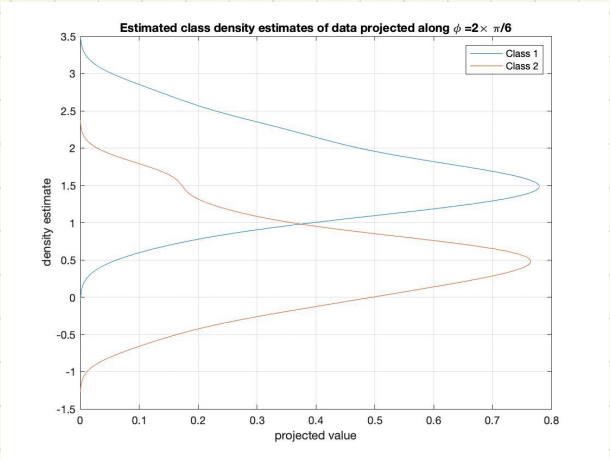


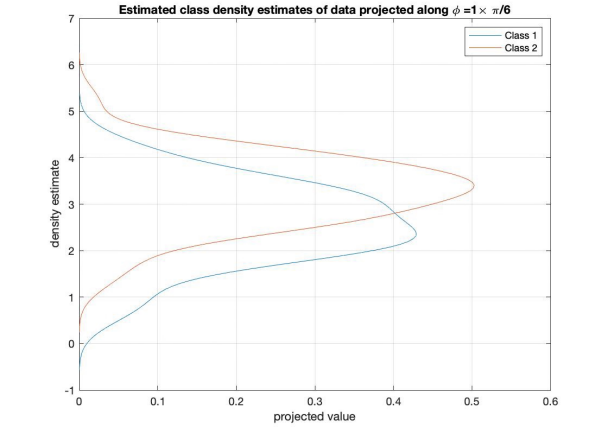




Noise, and signal are highly dependent on phi as the keep the shape of the cos/sine waves.



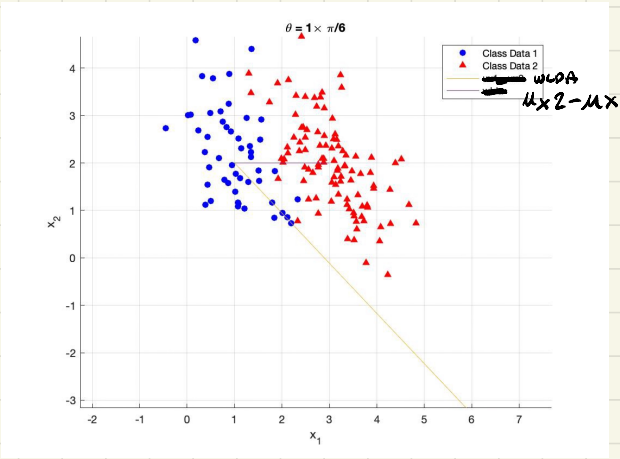




The density plots are adjusted profundly according to Phi. Projecting points to a line is sensitive and can easily be influenced with the slightest change.

(c) [6pts] Write a Matlab function which takes as input a 2D labeled dataset for binary classification and outputs the LDA vector $\mathbf{w}_{LDA} = \hat{S}_{x.\text{avg}}^{-1}(\hat{\mu}_{2x} - \hat{\mu}_{1x})$ where $\hat{S}_{x.\text{avg}} = \frac{n_1}{n}\hat{S}_{1x} + \frac{n_2}{n}\hat{S}_{2x}$ and $\hat{S}_{1x}$ and $\hat{S}_{2x}$ are the empirical 2×2 covariance matrices of classes 1 and 2 respectively.

Use this function to compute $\mathbf{w}_{LDA}$ for the dataset of part(a)(ii). Compare it with the difference between the class 2 and class 1 empirical mean vectors $(\hat{\mu}_{2x} - \hat{\mu}_{1x})$. Overlay both these vectors on the scatter plot of the dataset with the vectors represented as arrows starting at the location of $\hat{\mu}_{1x}$. Discuss what you observe. Also compare the direction of $\mathbf{w}_{LDA}$ with the value of ϕ from part (b) which maximizes the SNR.

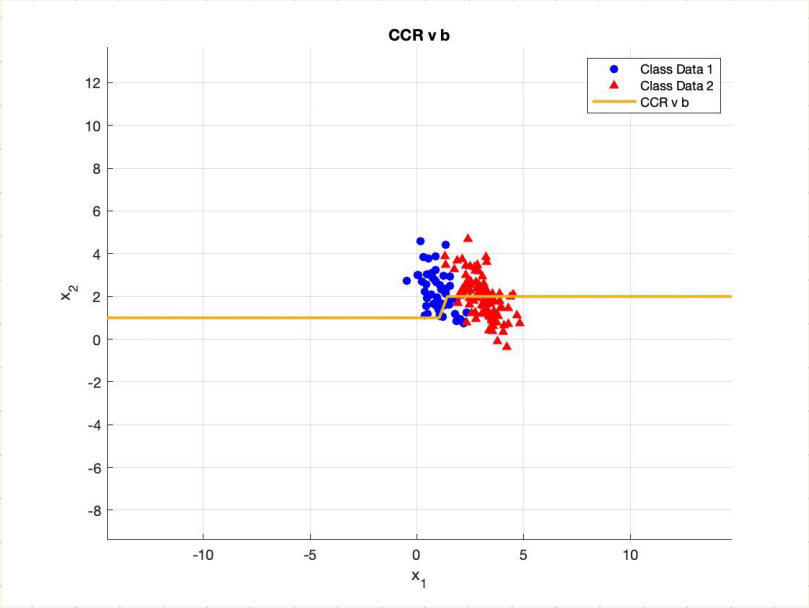


(Typo in code)

The direction of w_{LDA} is very similar to the value of ϕ which maximizes the SNR. w_{LDA} has a negative slope.

(d) [7pts] Consider the following separating hyperplane decision rule for binary classification based on thresholding a linear (affine) function of the feature vector: $h_{\mathbf{w},b}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^\top \mathbf{x} + b \leq 0 \\ 2 & \text{else} \end{cases}$. Write a Matlab function which takes as inputs a 2D labeled dataset for binary classification, $\mathbf{w}$, and b as inputs and outputs the value of CCR that results from applying $h_{\mathbf{w},b}(\mathbf{x})$ to classify all feature vectors in the dataset into class 1 or class 2.

Use this function to create a plot of CCR as a function of b for the dataset of part(a)(ii) and $\mathbf{w} = \mathbf{w}_{LDA}$. Compute the value of the offset parameter b which maximizes the CCR and the resulting CCR value.




```
B that maximizes CCR: 1.428
```

Problem 4.4 [22pts] *(ridge regression)* In this problem we will implement ridge regression and apply it to a real-world 8-dimensional (8 features) prostate cancer dataset contained in the file `prostateStnd.mat`. In this dataset, 8 medically relevant features named lcavol, lweight, age, lbph, svi, lcp, gleason, and pgg45 are used to estimate lpsa (log prostate specific antigen). The training and test data are provided as `(xtrain, ytrain)` and `(xtest, ytest)` respectively. The first 8 features correspond to the first 8 entries of `names`. The ninth entry of `names` (the last one) is the label to be predicted whose values are in `(ytest, ytrain)`.

(a) [4pts] As a first step, write Matlab code to normalize the **training dataset** so that post-normalization, each of the 8 features and the label in the normalized training dataset has zero mean and unit variance. This requires determining a *pair* of offset and scaling parameters, one pair for each feature and one pair for the label. These parameters must be computed only from the training dataset, but they must be applied to both the training and test datasets, i.e., we normalize both the training and test data, but we are only allowed to normalize the test data using parameters derived from the training data. In other words we must apply identical operations to training and test data. Only the training data will be actually normalized by the operation. If the test data is statistically similar to the training data, it too will be approximately normalized. **Report** the mean and variance of each feature and the label *before* normalization.

```
The mean and variance of training data before normalization

The mean of feature column lcavol is: -0.031
The variance of feature column lcavol is: 1.123

The mean of feature column lweight is: -0.054
The variance of feature column lweight is: 0.931

The mean of feature column age is: 0.119
The variance of feature column age is: 1.026

The mean of feature column lbph is: -0.020
The variance of feature column lbph is: 1.028

The mean of feature column svi is: 0.018
The variance of feature column svi is: 1.040

The mean of feature column lcp is: -0.025
The variance of feature column lcp is: 1.014

The mean of feature column gleason is: -0.030
The variance of feature column gleason is: 0.974

The mean of feature column pgg45 is: 0.067
The variance of feature column pgg45 is: 1.091

The mean of label column lpsa is: 2.452
The variance of label column lpsa is: 1.459
```

before normalization.

(b) [6pts] Next, write Matlab code to use the normalized data to train a ridge regression model for each of the following values of the quadratic regularization penalty parameter λ: $\{e^{-5}, e^{-4}, e^{-3}, ..., e^{10}\}$.

(c) [9pts] In a single figure, plot the ridge regression coefficient of each feature (8 in total) as a function of $\ln \lambda$ (8 curves in total) for $\ln \lambda$ ranging from -5 to 10 in steps of 1. Use suitable colors and/or markers to distinguish between the 8 curves and label them appropriately in a legend. Discuss what happens to the coefficients as λ becomes larger.

(d) [3pts] In another figure, plot the mean-squared-error (MSE) of both the training and the test data as a function of $\ln \lambda$. Discuss your observations.

I HAVE CODE FOR 4.4b & c
DID GET TO FINISH & TEST.

DID NOT GET A CHANCE
TO COMPLETE 4.4d.