Syed Ai: COSC 729 Assignment due 3/15/2021

(a) $\mu = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{pmatrix} = A$

$A = \det(A - \lambda I) = 0$

$A = \det\left(\begin{pmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right)$

$$A = \det \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 3-\lambda & 0 & 0 \\ 2 & 4-\lambda & 0 \\ 0 & 1 & 3-\lambda \end{pmatrix} \end{array}$$

$\Rightarrow a \det\left(\begin{vmatrix} e & f \\ h & i \end{vmatrix}\right) - b\det\left(\begin{vmatrix} d & f \\ g & i \end{vmatrix}\right) + c\det\left(\begin{vmatrix} d & e \\ g & h \end{vmatrix}\right)$

$\Rightarrow aei + bfg + cdh - afh - bdi - ceg$

$\Rightarrow [(3-\lambda)(4-\lambda)(3-\lambda)] + 0 + 0 - 0 - 0$

$\Rightarrow -\lambda^3 + 10\lambda^2 - 28\lambda + 21$

$$\Rightarrow \boxed{\begin{aligned} \lambda_1 &= 3 \\ \lambda_2 &= \frac{7+\sqrt{21}}{3} = 3.861 \\ \lambda_3 &= \frac{7-\sqrt{21}}{3} = 0.806 \end{aligned}}$$

Syed Ali

Eigenvectors :- $\Lambda_1, \Lambda_2, \Lambda_3$

$Ax = \Lambda X$

$$\begin{bmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \overset{3}{\Lambda} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \Rightarrow \boxed{\begin{Bmatrix} -1 \\ 0 \\ 2 \end{Bmatrix}} = \Lambda_1$$

$$\begin{bmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = 3.86 \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \Rightarrow \boxed{\begin{Bmatrix} 4 \\ 5.583 \\ 2 \end{Bmatrix}} = \Lambda_2$$

$$\begin{bmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 0 & 1 & 3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = 0.806 \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \Rightarrow \boxed{\begin{Bmatrix} 4 \\ -3.583 \\ 2 \end{Bmatrix}} = \Lambda_3$$

(b) To see if the vectors are mutually orthogonal

$\Lambda_1 \cdot \Lambda_2 = 0$

$$\begin{Bmatrix} -1 \\ 0 \\ 2 \end{Bmatrix} \begin{Bmatrix} 4 \\ 5.583 \\ 2 \end{Bmatrix} \rightarrow -4 + 4 = \boxed{0}$$

$$\Lambda_2 \cdot \Lambda_3 = \begin{Bmatrix} 4 \\ 5.583 \\ 2 \end{Bmatrix} \begin{Bmatrix} 4 \\ -3.583 \\ 2 \end{Bmatrix}$$

$$16 - 20 + 4 = \boxed{0}$$

$$\Lambda_1 \cdot \Lambda_3 = \begin{Bmatrix} 4 \\ -3.583 \\ 2 \end{Bmatrix} \begin{Bmatrix} -1 \\ 0 \\ 2 \end{Bmatrix} = -4 + 4 = \boxed{0}$$

$\boxed{\text{Thus proving that the eigen vectors are mutually orthogonal}}$

Scanned with CamScanner

Syed Ali

Part c and d done in Python

(2) Given the following equations

(a) $P(x_1|w_1) = K_1 \exp\left(\frac{-y^2}{10}\right)$

$P(x_2|w_2) = K_2 \exp\left(-\frac{(y-2)^2}{2}\right)$

$P(w|d) \Rightarrow \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$

$P(x_1|w_1) = K_1 \cancel{\exp}\left(\frac{-y^2}{10}\right) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \cancel{\exp}\left(\frac{-(y-\mu)^2}{2\sigma^2}\right)$

$K_1\left(\frac{\cancel{-}y^2}{10}\right) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \cancel{-} \frac{(y-\mu)^2}{2\sigma_1^2}$ Where $\mu_1 = 0$; $\sigma_1^2 = 5$

$K_1\left(\frac{y^2}{10}\right) = \frac{1}{\sqrt{2\pi \cdot 5}} \left(\frac{y-0}{2 \cdot 5}\right)$

$$\boxed{K_1 = \frac{1}{\sqrt{10\pi}}}$$

Similarly $P(x_2|w_2) = K_2 \exp\left(-\frac{(y-2)^2}{2}\right) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(y-2)^2}{2(1)}\right)$

Where $\mu_2 = 2$ and $\sigma_2^2 = 1$

$K_2 \cancel{\exp}\left(\cancel{-\left(\frac{y-2}{2}\right)^2}\right) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \cancel{\exp}\left(\cancel{-\frac{(y-2)^2}{2}}\right)$

$$\boxed{K_2 = \frac{1}{\sqrt{2\pi}}}$$

Plots done in python

Syed Ali

(b) The Conditional risk for two Category classification is Where we can see that the formulas for the risk associated with the action α_i of classifying a featurevector y as class w_i is different for each action.

$$\begin{cases} R(\alpha_1/y) = \lambda_{11} P(\omega_1/y) + \lambda_{12} P(\omega_2/y) \\ R(\alpha_2/y) = \lambda_{21} P(\omega_1/y) + \lambda_{22} P(\omega_2/y) \end{cases}$$

The prior probabilities of the two classes are equal

$P(\omega_1) = P(\omega_2) = 1/2$

The assumption that are cost for choosing correctly is 0.

$\therefore \lambda_{11} = 0$ and $\lambda_{22} = 0$

the costs of choosing incorrectly are given as C_{12} and C_{21}:

$\lambda_{12} = C_{12} = 1$

$\lambda_{21} = C_{21} = \sqrt{5}$

thus the expression for the conditional risk of α_1 is

$$R(\alpha_1|y) = \lambda_{11} P(\omega_1|y) + \lambda_{12} P(\omega_2|y)$$
$$= 0\, P(\omega_1|y) + \sqrt{1}\, P(\omega_2|y)$$
$$\Rightarrow \underbrace{P(\omega_2|y)}$$

Similary the expression of α_2

$$R(\alpha_2|y) = \lambda_{21} P(\omega_1|y) + \lambda_{22} P(\omega_2|y)$$
$$= \sqrt{5}\, P(\omega_1|y) + 0\, P(\omega_2|y)$$
$$= \underbrace{\sqrt{5}\, P(\omega_1|y)}$$

The Bayes risk is the integral of the conditional risk when we use the optimal decision regions R_1 and R_2, So, solving for the optimal decision boundary is a matter of solving for the roots of the eqn.

Syed Ali

$$R(\alpha_1|y) = R(\alpha_2|y)$$

$$\frac{P(y|\omega_2)P(\omega_2)}{P(y)} = \sqrt{5}\,\frac{P(y|\omega_1)P(\omega_1)}{P(y)}$$

Given that the priors are equal $P(y|\omega_2) = \sqrt{5}\,P(y|\omega_1)$

Now using the values of k_1 and k_2 from part a we have expressions for $P\,y|\omega_1$ and $P\,y|\omega_2$

$$\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(y-2)^2}{2}\right) = \sqrt{5}\,\frac{1}{\sqrt{10\pi}} \exp\left(\frac{-y^2}{10}\right)$$

$$\therefore \frac{-(y-2)^2}{2} = \frac{-y^2}{10}$$

$$10(y-2)^2 = 2y^2$$
$$10(y^2-2y+4) = 2y^2$$

$$10y^2 - 20y + 40 = 2y^2$$
$$2y^2 - 5y + 10$$

$$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

$$\frac{5 \pm \sqrt{5^2-4(2)(10)}}{4}$$

$$\frac{5 \pm \sqrt{25-80}}{4}$$

$$\underbrace{y \Rightarrow \frac{5+\sqrt{55}\,i}{2} \text{ or } \frac{5-\sqrt{55}\,i}{2}}$$

Syed Ali

(i) for $x < a$ the decision rule will choose ω_1

(ii) for $a < x < b$ the decision rule will choose ω_2

(iii) for $b < x$ the decision will choose ω_1

$\therefore$ the decision region R_1 is $x < a$ and $x > b$

the decision region R_2 is $a < x < b$

Bayes Risk $= \int_{R_1} [\lambda_{11} P(\omega_1 | y) + \lambda_{12} P(\omega_2 | y)]$

$+ \int_{R_2} [\lambda_{21} P(\omega_1 | y) + \lambda_{22} P(\omega_2 | y)]$

$$\boxed{\Rightarrow \int_{y=\infty}^{a} N(2,1) \frac{1}{2} + \int_{0}^{\infty} N(2,1) \frac{1}{2} + \int_{a}^{b} \sqrt{5}\, N(0,5) \frac{1}{2}}$$

Scanned with CamScanner

Syed Ali HMWK4 Problem 1 c and d

In [1]:
```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
```

Initialize a covariance matrix for a three dimensional data

In [2]:
```
X = np.array([[3,2,0],[2,4,1],[0,1,3]])
print(X)
```

```
[[3 2 0]
 [2 4 1]
 [0 1 3]]
```

find eigenvalues and eigenvectors using numpy.linalg

In [3]:
```
eg, egv = la.eig(X)
print(eg)
```

```
[5.79128785+0.j 1.20871215+0.j 3.        +0.j]
```

In [4]:
```
print(eg[0])
```

```
(5.791287847477921+0j)
```

In [5]:
```
print(egv[0])
print(egv[1])
print(egv[2])
```

```
[-0.55920734  0.69805956 -0.4472136 ]
[-7.80454320e-01 -6.25212808e-01  8.06936742e-17]
[-0.27960367  0.34902978  0.89442719]
```

Distance between two vectors

In [6]:
```
y1 = np.array([1,2,1])
y2 = np.array([2,1,3])
```

In [7]:
```
d1 = la.norm(y1-y2)
print(d1)
```

```
2.449489742783178
```

Let us project y1 and y2 on eigenvectors

In [8]:
```
y1p = np.dot(egv,y1)
y2p = np.dot(egv,y2)
```

```
print(y1p)
print(y2p)
```

```
[ 0.3896982  -2.03087994  1.31288309]
[-1.76199589 -2.18612145  2.47310402]
```

Distance between the two vectors in the projected space

In [9]:
```
d2 = la.norm(y1p-y2p)
print(d2)
```

```
2.4494897427831774
```

Based on this we can see that the transformation of space R^3 and R^3 is distance preserving

Now let us ignore the eigenvalues with the least magnitude which is egv[1]. We will set this row to zero

In [10]:
```
egv1 = egv
egv1[1,] = 0
print(egv1)
```

```
[[-0.55920734  0.69805956 -0.4472136 ]
 [ 0.          0.          0.        ]
 [-0.27960367  0.34902978  0.89442719]]
```

Now let us project y1 and y2

In [11]:
```
y1pp = np.dot(egv1, y1)
y2pp = np.dot(egv1, y2)
print(y1pp, y2pp)
```

```
[0.3896982  0.         1.31288309] [-1.76199589  0.          2.47310402]
```

In [12]:
```
d3 = la.norm(y1pp-y2pp)
print(d3)
```

```
2.4445654159683534
```

So based on this we can see that dimensionality reduction is possible from three - two from the covariance matrix and find the eigen vectors (principal components). Ignore or assign zeros to the eigenvector

corresponding to the least value. Then project the original data on new space and eliminate the values in the least eigenvector row.

Problem 2 a

In [13]:

```
%matplotlib inline
k1 = 1.0/np.sqrt(10*np.pi)
k2 = 1.0/np.sqrt(2*np.pi)

x = np.arange(-10,30,0.1)
y1 = k1*np.exp((-x**2)/10)
y2 = k2*np.exp((-(x-2)**2)/2)

plt.plot(x, y1, 'blue', x, y2, 'red')
plt.legend(['p(x|w1)', 'p(x|w2)'])
plt.xlabel('y')
plt.ylabel('p')
plt.yticks(np.arange(0, 0.5, step=0.04))
plt.show()
```