1.1. MATLAB code for this assingment is cotnained in the appendix. Mean vector, calculated by MATLAB's mean function, has the following value:

5.8000 5.0000 6.2000

Covariance matrix, calculated by MATLAB's cov function, has the following value:

3.2000 0.2500 -0.4500

0.2500 2.5000 -3.7500

-0.4500 -3.7500 5.7000

1.2. We modeled the data by normal distribution and calculated the values of probability density using the mvnpdf function. We got the following results:

0.0543

0.0006

0.0000

for the points [5 5 6], [3 5 7] and [4 6.5 1] respectively.

2.1. We have two vectors of two features: [a b] and [c d]. First feature has values a and c, and second feature has values b and d. Since we have two features, the dimensions of the covariance matrix will be 2x2.:

covariance.matrix = $\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$

If there are n features, covariance between features X and Y is calculated by formula:

$$\operatorname{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(X))(y_i - E(Y)).$$

Where E(X) and E(Y) are mean values of features X and Y.

In our case, σ_{ij} is calculated by formula: cov(X, Y) = $\frac{1}{2}\sum_{i=1}^{2}(x_i - E(x))(y_i - E(y))$.

The mean value of first feature is: (a + c) / 2.

The mean value of second feature is: (b + d) / 2.

From the above formula:

$$\sigma_{11} = \frac{1}{2}\sum_{i=1}^{2}(x_i - \left(\frac{a+c}{2}\right))^2$$

$$\sigma_{11} = \frac{1}{2}\left(\left(a-\left(\frac{a+c}{2}\right)\right)^2 + \left(c-\left(\frac{a+c}{2}\right)\right)^2\right)$$

$$\sigma_{11} = \frac{1}{2}\left(\left(\frac{a-c}{2}\right)^2 + \left(\frac{c-a}{2}\right)^2\right)$$

$$\sigma_{11} = \frac{a^2-2ac+c^2}{4}$$

$$\sigma_{22} = \frac{1}{2}\sum_{i=1}^{2}(y_i-\left(\frac{b+d}{2}\right))^2$$

$$\sigma_{22} = \frac{1}{2}\left(\left(\frac{b-d}{2}\right)^2 + \left(\frac{d-b}{2}\right)^2\right)$$

$$\sigma_{22} = \frac{b^2-2bd+d^2}{4}$$

$$\sigma_{12} = \sigma_{21} = \frac{1}{2}\sum_{i=1}^{2}\left(x_i-\left(\frac{a+c}{2}\right)\right)\left(y_i-\left(\frac{b+d}{2}\right)\right) =$$

$$= \frac{1}{2}\left(\left(\frac{a-c}{2}\right)\left(\frac{b-d}{2}\right) + \left(\frac{c-a}{2}\right)\left(\frac{d-b}{2}\right)\right) =$$

$$= \frac{ab-ad-cb+cd}{4}$$

So, this is what the covariance matrix looks like:

$$\text{covariance.matrix} = \begin{bmatrix} \frac{a^2-2ac+c^2}{4} & \frac{ab-ad-cb+cd}{4} \\ \frac{ab-ad-cb+cd}{4} & \frac{b^2-2bd+d^2}{4} \end{bmatrix}$$

2.2 If we inrease every value in the given dataset by k, we will get the following matrix:

$$\begin{bmatrix} \frac{(a+k)^2-2(a+k)(c+k)+(c+k)^2}{4} & \frac{(a+k)(b+k)-(a+k)(d+k)-(c+k)(b+k)+(c+k)(d+k)}{4} \\ \frac{(a+k)(b+k)-(a+k)(d+k)-(c+k)(b+k)+(c+k)(d+k)}{4} & \frac{(b+k)^2-2(b+k)(d+k)+(d+k)^2}{4} \end{bmatrix}$$

$$\begin{bmatrix} \dfrac{a^2+2ak+k^2-2(ac+ak+ck+k^2)+c^2+2ck+k^2}{4} & \dfrac{ab+ak+bk+k^2-ad-ak-dk-k^2-cb-ck-kb-k^2+cd+ck+kd+k^2}{4} \\ \dfrac{ab+ak+bk+k^2-ad-ak-dk-k^2-cb-ck-kb-k^2+cd+ck+kd+k^2}{4} & \dfrac{b^2+2bk+k^2-2(bd+bk+dk+k^2)+d^2+2dk+k^2}{4} \end{bmatrix}$$

After simplifying, we get exactly the same covariance matrix. So, increasing every value in the data set by the same amount does not affect the covariance matrix, or the way two features change together.

2.3. If we multiply every value in the dataset by k, we get the following covariance matrix:

$$\begin{bmatrix} \dfrac{a^2k^2-2k^2ac+c^2k^2}{4} & \dfrac{k^2ab-k^2ad-k^2cb+k^2cd}{4} \\ \dfrac{k^2ab-k^2ad-k^2cb+k^2cd}{4} & \dfrac{b^2k^2-2k^2bd+d^2k^2}{4} \end{bmatrix}$$

So, the whole covariance matrix is multiplied by k^2:

$$k^2\begin{bmatrix} \dfrac{a^2-2ac+c^2}{4} & \dfrac{ab-ad-cb+cd}{4} \\ \dfrac{ab-ad-cb+cd}{4} & \dfrac{b^2-2bd+d^2}{4} \end{bmatrix}$$

3.1.

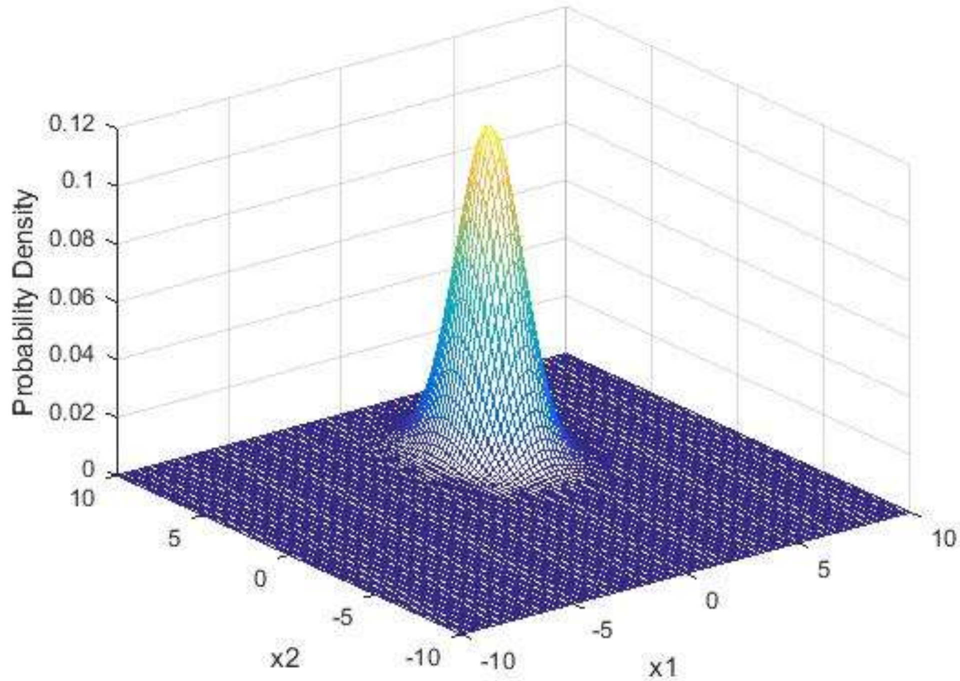


Figure 1: Plot of 2-D Gaussian pdf with mean and covariance matrix given in the assingment 3.

We generated the pdf using the MATLAB's mvnpdf function and plotted the function using MATLAB's mesh function. See code in the appendix.

3.2. The Mahalanobis distance of an observation $\vec{x} = (x_{1,} x_2, ..., x_n)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, \mu_2, ..., \mu_n)^T$ and covariance matrix S is defined as: $D_m(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$

We got the following results:

9.0554   3.4641   1.0000   4.8990

for points [10 10]', [0 0]', [3 4]' and [6 8]' respectively. Instead of calculating inverse of the covariance matrix S using the function inv(S) and multiplying $(\vec{x} - \vec{\mu})^T$ by it, we used $(\vec{x} - \vec{\mu})^T$ / S, since it is faster and more accurate generally. See the code in the appendix.

5.1. Discriminant functions that could be used for minimum error rate classification:

$$g_1(x) = \ln p(x \mid w_1) + \ln P(w_1)$$

$$g_2(x) = \ln p(x \mid w_2) + \ln P(w_2)$$

where p(x|w_i) is normal distribution associated with w_i.

We chose this types of discriminant function because it is very convenient to use when you have a normal distribution.

The general multivariate normal density in d dimensions is written as:

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1} (x-\mu) \right]$$

where x is a d dimensional column vector, μ is a d-dimensional mean vector, ∑ is d-by-d covariance matrix, |∑| and $\sum^{-1}$ are its determinant and inverse, respectively.

In this case, d=2 and for w_1 $\mu_1 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$, $|\Sigma_1| = 4$ and $\Sigma_1^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix}$.

$$g_1(x,y) = \ln \frac{1}{2\pi\sqrt{4}} + \ln \exp\left[-\frac{1}{2}[x-3, y-5] \begin{bmatrix} 1 & 0 \\ 0 & 1/4 \end{bmatrix} \begin{bmatrix} x-3 \\ y-5 \end{bmatrix} \right] + \ln 0.3$$

$$g_1(x,y) = \ln \frac{3}{10\pi} - \frac{1}{2}[x-3, (y-5)/4] \begin{bmatrix} x-2 \\ y-5 \end{bmatrix}$$

$$g_1(x,y) = \ln \frac{3}{10\pi} - \frac{4x^2 - 24x + 61 + y^2 - 10y}{8}$$

For w_2, $\mu_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, $|\Sigma_1| = 2$ and $\Sigma_2^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$.

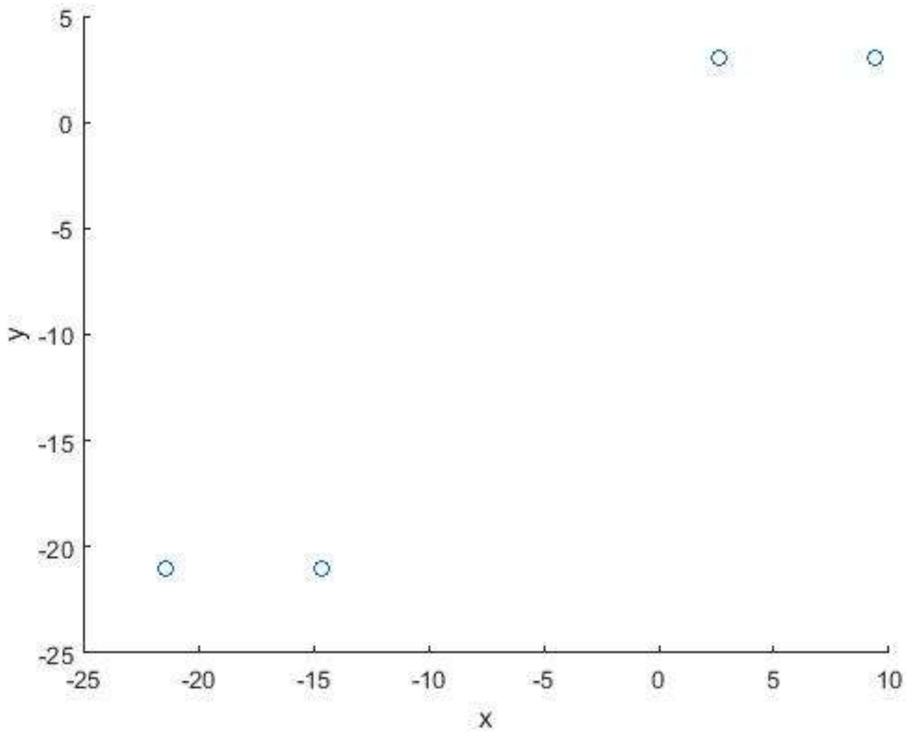
$$g_2(x,y) = \ln \frac{1}{2\pi\sqrt{2}} - \frac{1}{2}[x-2, y-1] \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x-2 \\ y-1 \end{bmatrix} + \ln 0.7$$

$$g_2(x,y) = \ln \frac{7}{20\pi\sqrt{2}} - \frac{1}{2}\left[\frac{x-2}{2}, y-1 \right] \begin{bmatrix} x-2 \\ y-1 \end{bmatrix}$$

$$g_2(x,y) = \ln \frac{7}{20\pi\sqrt{2}} - \frac{1}{2}\left(\frac{(x-2)^2}{2} + (y-1)^2 \right)$$

$$g_2(x,y) = \ln \frac{7}{20\pi\sqrt{2}} - \frac{x^2 - 4x + 2y^2 - 4y + 6}{4}$$

5.2. Solving the equation g1(x, y) – g2(x, y) = 0 and filtering out the imaginary results, we got the results shown in the figure below:



These results give us the decision boundary shown in the figure below: