1. **5 PONTOS** Suponha que o vetor aleatório $\mathbf{X} = (X_1, X_2)$ seja observado em vários indivíduos ou itens que compõem uma amostra. Existem duas classes de itens, classe 0 e classe 1. Em cada classe, a densidade de probabilidade de $\mathbf{X}$ segue uma Gaussiana bivariada:

$$(\mathbf{X}| \in 0) \sim N_2\left(\boldsymbol{\mu}_0, \sum\nolimits_0\right) = N_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 4\end{bmatrix}\right)$$

and

$$(\mathbf{X}| \in 1) \sim N_2\left(\boldsymbol{\mu}_1, \sum\nolimits_1\right) = N_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4 & 0\\0 & 1\end{bmatrix}\right)$$

Suponha que as duas classes sãao igualmente frequentes na população. Isto é, que $\pi_0 = \mathbb{P}(\mathbf{X} \in 0)$ seja igual a $\pi_1 = \mathbb{P}(\mathbf{X} \in 1)$.

Suponha também que os custos c_0 e c_1 de má classificação sejam iguais onde c_0 é o custo de classificar como 0 um item da classe 1 e c_1 é o custo de classificar como 1 um item da classe 0.

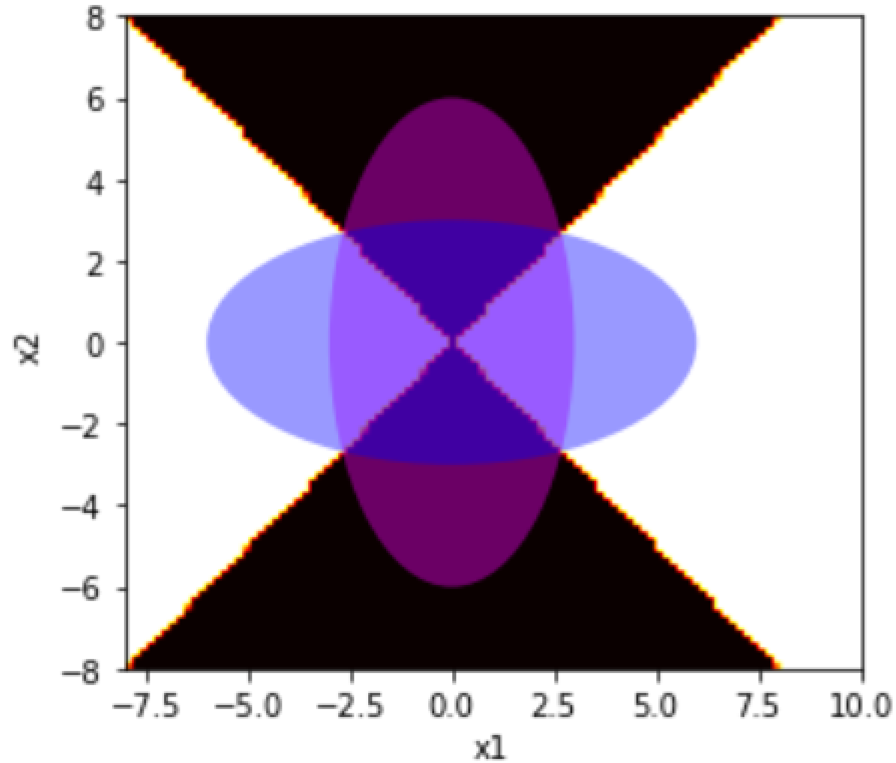
Produza um código R ou python que desenhe algumas curvas de nível das densidades de $\mathbf{X}$ em cada classe e esboce a fronteira de decisšao determinada pela regra ótima de Bayes. O resultado deverá ser uma imagem como a da Figura 1.

A seguir, refaça a figura assumindo que a classe 0 aparece 3 vezes mais frequentemente que a classe 1 (isto é, que $\pi_0 = 3\pi_1$). Faça uma terceira figura supondo adicionalmente que os custos também são diferentes, com $c_0 = 5c_1$.

Solução:

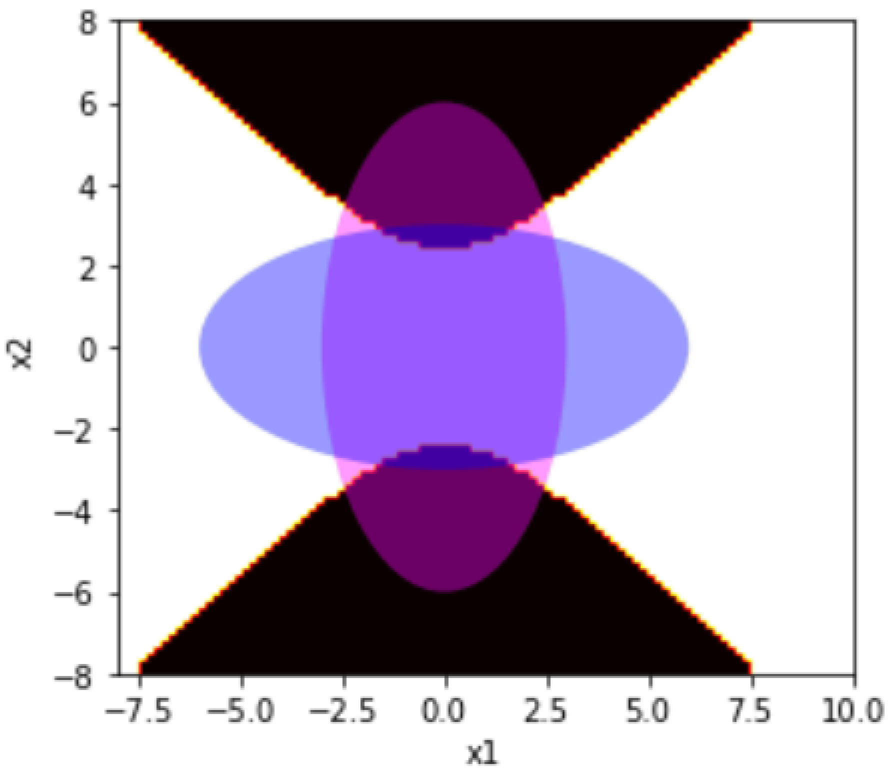
```
import numpy as np
import math as math
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse

def get_cov_ellipse(cov, centre, nstd, **kwargs):
    eigvals, eigvecs = np.linalg.eigh(cov)
    order = eigvals.argsort()[::-1]
    eigvals, eigvecs = eigvals[order], eigvecs[:, order]

    vx,vy = eigvecs[:,0][0], eigvecs[:,0][1]
    theta = np.arctan2(vy,vx)

    width, height = 2 *nstd *np.sqrt(eigvals)
    return Ellipse(xy=centre, width = width, height = height, angle = np.degrees(theta), **kwargs)

#parametros da classe 1
mu1 = np.array([0,0])
sigma1 = np.array([[1,0],[0,4]])

#parametros da classe 2
mu2 = np.array([0,0])
sigma2 = np.array([[4,0],[0,1]])
```

```
#Supondo que as classes são balanceadas
pi1 = 0.5
pi2 = 1 - pi1

#custos

c1 = 1
c2 = 1

#a linha de decisão é quadrática pois as matrizes são diferentes
A1 = np.linalg.inv(sigma1)
A2 = np.linalg.inv(sigma2)
A = A1-A2

beta = np.dot(A1, mu1) - np.dot(A2,mu2)

invsigma = np.linalg.inv(sigma1)
d1 = np.dot (np.dot (mu1, invsigma), mu1)
d2 = np.dot (np.dot (mu2, invsigma), mu2)
alpha = (d1-d2) - 2*(np.log(np.linalg.det(A2)) - np.log(np.linalg.det(A1)) - 2*np.log(pi1/pi2) -
2*np.log(c2/c1))

x, y = np.meshgrid(np.linspace(-8,10, 100), np.linspace (-8,8, 100))

z = np.sign(A [0,0]*x**2 + 2*A[0,1]*x*y + A[1,1]*y**2 -2 *beta[0]*x-2 * beta[1]*y + alpha)

xmin, xmax, ymin, ymax  = -8,10,-8,8

#visualizando a população
labels = ['Class 1', 'Class 2']
colours = ['magenta', 'blue']
fig, ax = plt.subplots()
plt.imshow (z, extent=[xmin,xmax,ymin,ymax], cmap ='hot', origin='lower')
e1 = get_cov_ellipse(sigma1, mu1, 3, fc = colours[0], alpha=0.4)
ax.add_artist(e1)
e2 = get_cov_ellipse(sigma2,mu2, 3, fc= colours[1], alpha = 0.4)
ax.add_artist(e2)

ax.set_xlim(-8,10)
ax.set_ylim(-8,8)
ax.set_xlabel('x1')
ax.set_ylabel('x2')
plt.show()
```

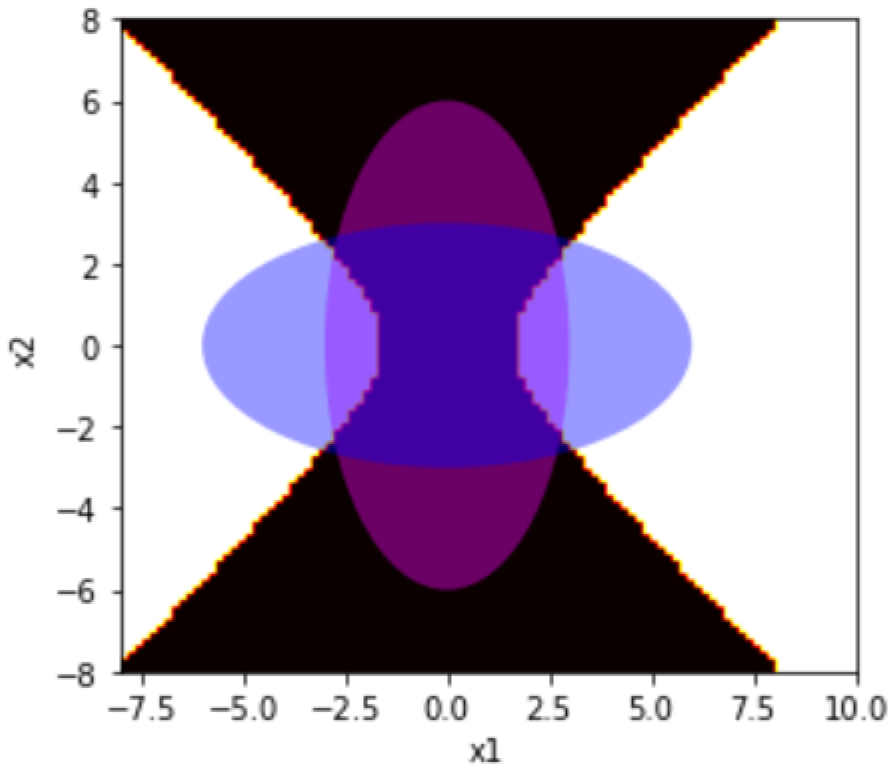



```
-----------------------------------------
#Supondo que a classe 1 é 3 vezes mais comum que a classe 2
pi1 = 0.75
pi2 = 0.25

#custos

c1 = 1
c2 = 1

#a linha de decisão é quadrática pois as matrizes são diferentes
A1 = np.linalg.inv(sigma1)
A2 = np.linalg.inv(sigma2)
A = A1-A2

beta = np.dot(A1, mu1) - np.dot(A2,mu2)

invsigma = np.linalg.inv(sigma1)
d1 = np.dot (np.dot (mu1, invsigma), mu1)
d2 = np.dot (np.dot (mu2, invsigma), mu2)
alpha = (d1-d2) - 2*(np.log(np.linalg.det(A2)) - np.log(np.linalg.det(A1)) - 2*np.log(pi1/pi2) -
2*np.log(c2/c1))
```

```
x, y = np.meshgrid(np.linspace(-8,10, 100), np.linspace (-8,8, 100))

z = np.sign(A [0,0]*x**2 + 2*A[0,1]*x*y + A[1,1]*y**2 -2 *beta[0]*x-2 * beta[1]*y + alpha)

xmin, xmax, ymin, ymax  = -8,10,-8,8

#visualizando a população
labels = ['Class 1', 'Class 2']
colours = ['magenta', 'blue']
fig, ax = plt.subplots()
plt.imshow (z, extent=[xmin,xmax,ymin,ymax], cmap ='hot', origin='lower')
e1 = get_cov_ellipse(sigma1, mu1, 3, fc = colours[0], alpha=0.4)
ax.add_artist(e1)
e2 = get_cov_ellipse(sigma2,mu2, 3, fc= colours[1], alpha = 0.4)
ax.add_artist(e2)

ax.set_xlim(-8,10)
ax.set_ylim(-8,8)
ax.set_xlabel('x1')
ax.set_ylabel('x2')
plt.show()
```



```
#Supondo que a classe 1 é 3 vezes mais comum que a classe 2
pi1 = 0.75
pi2 = 0.25

#Supondo que os custos também são diferentes

c1 = 5
c2 = 1

#a linha de decisão é quadrática pois as matrizes são diferentes
A1 = np.linalg.inv(sigma1)
A2 = np.linalg.inv(sigma2)
A = A1-A2

beta = np.dot(A1, mu1) - np.dot(A2,mu2)

invsigma = np.linalg.inv(sigma1)
d1 = np.dot (np.dot (mu1, invsigma), mu1)
d2 = np.dot (np.dot (mu2, invsigma), mu2)
alpha = (d1-d2) - 2*(np.log(np.linalg.det(A2)) - np.log(np.linalg.det(A1)) - 2*np.log(pi1/pi2) -
2*np.log(c2/c1))

x, y = np.meshgrid(np.linspace(-8,10, 100), np.linspace (-8,8, 100))

z = np.sign(A [0,0]*x**2 + 2*A[0,1]*x*y + A[1,1]*y**2  -2 *beta[0]*x-2 * beta[1]*y + alpha)

xmin, xmax, ymin, ymax  = -8,10,-8,8

#visualizando a população
labels = ['Class 1', 'Class 2']
colours = ['magenta', 'blue']
fig, ax = plt.subplots()
plt.imshow (z, extent=[xmin,xmax,ymin,ymax], cmap ='hot', origin='lower')
e1 = get_cov_ellipse(sigma1, mu1, 3, fc = colours[0], alpha=0.4)
ax.add_artist(e1)
e2 = get_cov_ellipse(sigma2,mu2, 3, fc= colours[1], alpha = 0.4)
ax.add_artist(e2)

ax.set_xlim(-8,10)
ax.set_ylim(-8,8)
ax.set_xlabel('x1')
ax.set_ylabel('x2')
plt.show()
```



----

2. **5 PONTOS**

   Considere a seguinte tabela de dados, representando uma amostra composta por quatro pontos amostrais, cada um dos pontos $\mathbf{X} = (X_1, X_2)$ com dois atributos ou features:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 2 & 2 \\ -2 & -2 \end{bmatrix}$$

----

3. **5 PONTOS** Suponha que os pontos da amostra venham de uma distribuição normal gaussiana multivariada p-dimensional com vetor esperado $\boldsymbol{\mu}$ e com uma matriz de covariãncia $p \times p$ representada por Σ.

   - Mostre que, se Σ for uma matriz diagonal, a densidade conjunta será o produto de p densidades gaussianas univariadas. Pode assumir $\boldsymbol{\mu} = \mathbf{0}$ para simplificar suas contas.
   - Suponha agora que Σ não é diagonal mas ainda é simétrica e definida positiva. Faąa a decomposição espectral de Σ e mostre que podemos escrever a densidade gaussiana multivariada como um produto de densidades gaussianas univariadas, cada uma delas associada com um dos autovetores de Σ.

Solução:

- Se a matriz for uma matriz diagonal, a densidade é o produto das p densidades gaussianas univariadas, como na fórmula abaixo

$$f\left(x_1,\ldots,x_p\right)=f\left(X\right)=\frac{1}{\left(2\pi\right)^{\frac{p}{2}}\left|\Sigma\right|^{\frac{1}{2}}}e^{-\frac{1}{2}\left(X-\mu\right)^T\Sigma^{-1}\left(X-\mu\right)}$$

e com u = 0, podemos ver com não muita dificuldade que a densidade conjunta nada mais é que o produto das p densidade gaussianas univariadas.

----

4. **5 PONTOS** Suponha que você tenha uma distribuição normal multivariada com uma matriz de covariância definida positiva Σ. Considere uma segunda distribuição gaussiana multivariada com o mesmo valor esperado mas cuja matriz de covariância seja $\cos(\theta)\ \Sigma$. Isto é, a matriz anterior multiplicada por um escalar (um número real) positivo $\cos(\theta) > 0$ onde θ é um certo ângulo. Você aprendeu como as curvas de nível da desnidade de probabilidade estão associados com os autovetores da matriz de covariância.

   - Os autovetores da nova matriz de covariância são rotacionados pelo ângulo θ? Justifique sua resposta.
   - Os autovalores da nova matriz de covariância são alterados? Justifique sua resposta.
   - Como serão alteradas as curvas de nível da segunda distribuição em comparação com aquelas da primeira distribuição?

Solução

- Não. os autovetores são ortogonais entre si e pelo teorema espectral. podemos dizer que a matriz que está sendo rotacionada é age como uma matriz diagonal e as novas coordenadas da matriz após a rotação nada mais é do que os autovetores multiplicado pelo ângulo aplicado.
- Sim. Os autovalores são os inversos de 1/lambda dos autovalores de lambda antes da rotação (lambda = a matriz).
- Os pontos na curva de nível tendem a estar a igual distância do perfil esperado ao da primeira distribuição. A maneira de medir a distância o perfil esperado é pela forma quadrática

$$d^2(\boldsymbol{y},\boldsymbol{\mu}) = (\boldsymbol{y}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$$