Μηχανική Μάθηση
7ο εξάμηνο Ακαδημαϊκό έτος 2023-2024

Ονοματεπώνυμο: Κούτρας Ιωάννης- Παναγιώτης
Α.Μ.: 03120130
Email-address: ioanniskoutras2002@gmail.com
1η Σειρά Ασκήσεων

Οι λύσεις είναι ως επί το πλείστον χειρόγραφες, ενώ συμπεριλαμβάνονται και κομμάτια κώδικα στα σημεία στα οποία χρησιμοποιήθηκαν.

Άσκηση 1.1 (Linear and Ridge Regression)

Όπως προβλέπεται και στη σημείωση της εκφώνησης για την άσκηση 1.1, για τη λύση παραθέτω τον κώδικα (Python) που χρησιμοποιήθηκε καθώς και τα ζητούμενα αποτελέσματα και σχόλια που τον συνοδεύουν.

- Normalize data:

```python
import pandas as pd

data = pd.read_csv('data.csv', sep=';')
quality = data['quality']
print(f' Quality range: {quality.min()} - {quality.max()}')

# normalize the data
normed = data.copy()
for attr in normed.columns.difference(['quality']):
    normed[attr] -= data[attr].mean()
    normed[attr] /= data[attr].std()

# save to csv
normed.to_csv('data_normed.csv')
```

➔ Quality range: 3 - 8

- Find the correlation coefficient between pH and sulphates (no need for division because of unit/normalized variances):

```
import pandas as pd
import numpy as np

data = pd.read_csv('data_normed.csv')

# with build ins
r9_10 = data.corr()['pH']['sulphates']

# without build ins
cov_ph_sul = np.mean((data['pH'] - data['pH'].mean()) * (data['sulphates'] - data['sulphates'].mean()))
cov_ph_sul, r9_10
```

➔ (-0.19652462068942095, -0.19664760230437037)

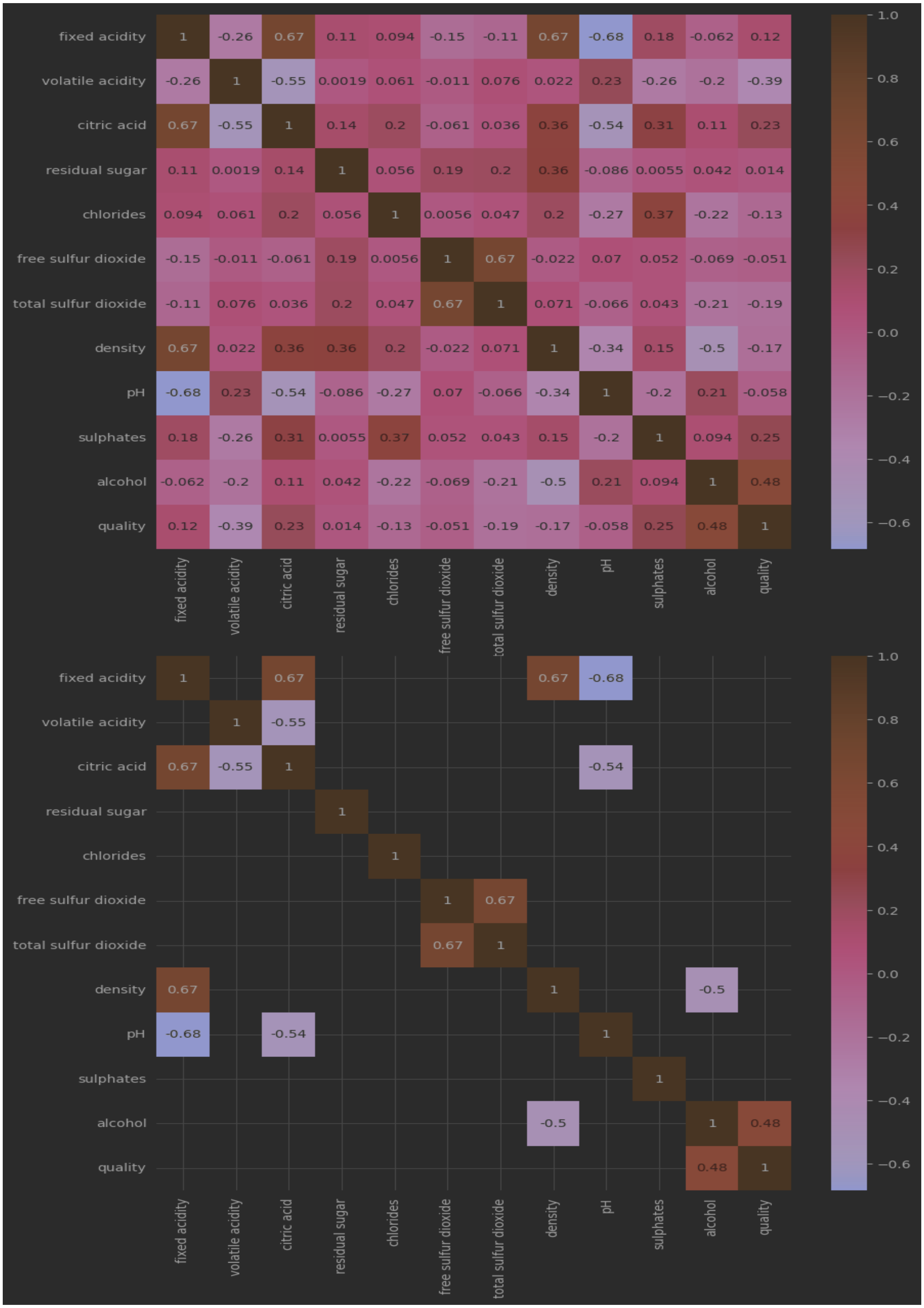
- Scatter plots of Cov using the seaborn library (specifically heatmaps):

```
import seaborn as sns
import matplotlib.pyplot as plt

corr = data.corr()

# create scatter plots of all the pairs
plt.figure(figsize=(10, 20))
plt.subplot(211)
sns.heatmap(corr, annot=True)

high_corr = corr.where(abs(corr) > 0.4)
plt.subplot(212)
sns.heatmap(high_corr, annot=True)
plt.show()
```


fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Σχόλια- Παρατηρήσεις: Από το heatmap παρατηρούμε ότι τα ζεύγη με υψηλό correlation είναι:

- (fixed acidity, citric acid, +), (fixed acidity, density, +), (fixed acidity, pH, -)
- (volatile acidity, citric acid, -)
- (pH, citric acid, -)
- (total sulfur dioxide, free sulfur dioxide, +)
- (alcohol, density, -)
- (quality, alcohol, +)

```
X = np.asarray(data.iloc[:100, :-1])
Y = np.asarray(data.iloc[:100, -1])
X_test = np.asarray(data.iloc[100:150, :-1])
Y_test = np.asarray(data.iloc[100:150, -1])
```

Για το επόμενο μέρος χρησιμοποιούμε τους τύπους για linear και ridge regression από τα slides του μαθήματος , δηλαδή:

Linear Regression weight formula:

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \Rightarrow \mathbf{w} = \mathbf{X}^+ \mathbf{y}$$

Ridge Regression weight formula:

$$\mathbf{w} = \left(\mathbf{X}^\top \mathbf{X} + \lambda \cdot \mathbf{I}\right)^{-1} \cdot \mathbf{X}^\top \cdot \mathbf{y}$$

- Setting up the calculations, data:

```
import numpy as np
from numpy.linalg import inv

# Linear Regression
w = inv(X.T @ X) @ X.T @ Y

# Ridge Regression
w_r = lambda _l: np.linalg.inv(X.T @ X + _l * np.eye(X.shape[1])) @ X.T @ Y

w_r_10, w_r_100, w_r_200 = w_r(10), w_r(100), w_r(200)
```

- Weight plotting:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.plot(w, label='Linear Regression (w)')
```

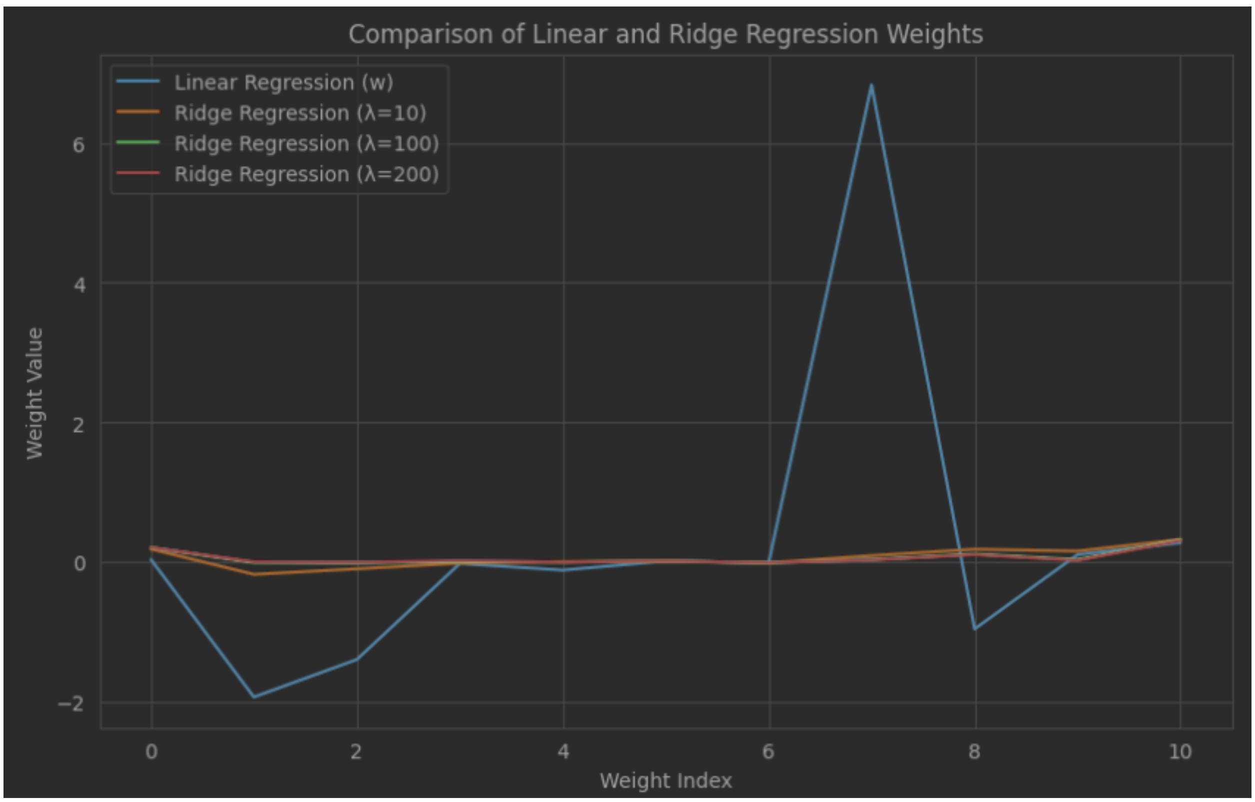
```
plt.plot(w_r_10, label='Ridge Regression (λ=10)')
plt.plot(w_r_100, label='Ridge Regression (λ=100)')
plt.plot(w_r_200, label='Ridge Regression (λ=200)')

plt.title('Comparison of Linear and Ridge Regression Weights')
plt.xlabel('Weight Index')
plt.ylabel('Weight Value')

plt.legend()

plt.show()
```

➔



Σχόλια- Παρατηρήσεις:
-Μικρότερο λ συνεπάγεται μικρότερες τιμές των βαρών. Αυτό συμβαίνει διότι το Ridge Regression χρησιμοποιεί L2 loss.
-Τα attributes που δεν έχουν υψηλό correlation έχουν προφανώς χαμηλές τιμές βαρών, συνεπώς δεν έχουν σημαντική επιρροή στο αποτέλεσμα.

- Calculation of RMSE:

Χρησιμοποιήσαμε τον τύπο

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

```python
def calc_rmse(_X, _Y):
    y_pred = [w_i @ _X.T for w_i in [w, w_r_10, w_r_100, w_r_200]]
    rmse = [np.sqrt(sum(_Y - y_i) ** 2)/len(y_pred) for y_i in y_pred]
    return rmse

# Train
print(f'Train RMSE:\n{calc_rmse(X, Y)}\n\nTest RMSE:\n{calc_rmse(X_test,
Y_test)}')
```

➔ Train RMSE:
[0.00027317616972866166, 0.2433519846861174, 0.9709019133401842, 1.722949291019658]

Test RMSE:
[2.279020633087578, 1.5609199581132982, 1.826274308097586, 2.2584927200838147]

Άσκηση 1.2 (Multivariate Gaussian Distribution)

Άσκηση 1.2 (Multivariate Gaussian Distribution)

(α) $x=[x_1, x_2]^T$, $x \sim N(x \mid \mu, \Sigma)$

$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$

Θα δείξουμε ότι η υπό συνθήκη συνάρτηση πυκνότητας πιθανότητας (σππ) $p(x_1 \mid x_2 = a)$ είναι Gaussian $N(\mu', \sigma'^2)$

όπου $\mu' = \mu_1 + \dfrac{\sigma_{12}(a-\mu_2)}{\sigma_2^2}$, $\sigma' = \dfrac{\sigma_1^2\sigma_2^2 - \sigma_{12}^2}{\sigma_2^2}$

έχουμε: $p(x_1 \mid x_2 = a) \triangleq \dfrac{p(x_1, x_2)|_{x_2=a}}{p(x_2=a)}$

- $p(x_1, x_2)|_{x_2=a} = \dfrac{1}{2\pi\sqrt{|\Sigma|}} \cdot \exp\left\{-\dfrac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}\right\}$

με $|\Sigma| = \det\Sigma = \sigma_1^2\sigma_2^2 - \sigma_{12}^2$ και $\Sigma^{-1} = \dfrac{1}{\sigma_1^2\sigma_2^2 - \sigma_{12}^2} \cdot \begin{bmatrix} \sigma_2^2 & -\sigma_{12} \\ -\sigma_{12} & \sigma_1^2 \end{bmatrix}$

- $x_2 \sim N(\mu_2, \sigma_2^2)$, άρα $p(x_2 = a) = \dfrac{1}{\sigma_2\sqrt{2\pi}} \exp\left\{-\dfrac{(a-\mu_2)^2}{2\sigma_2^2}\right\}$

Συνδυάζοντας τις δύο σχέσεις παίρνουμε:

$$p(x_1 \mid x_2 = a) = \frac{1/2\pi\sqrt{|\Sigma|} \cdot \exp\left\{-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}\right\}}{1/\sigma_2\sqrt{2\pi} \cdot \exp\left\{-\frac{(a-\mu_2)^2}{2\sigma_2^2}\right\}}$$

$$\Rightarrow p(x_1 \mid x_2 = a) = \frac{1}{\sqrt{2\pi} \cdot \sqrt{\left(\frac{\sigma_1^2\sigma_2^2 - \sigma_{12}^2}{\sigma_2^2}\right)}} \cdot \exp\left\{-\frac{1}{2} \cdot \frac{1}{\sigma_1^2\sigma_2^2 - \sigma_{12}^2} \begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}^T \begin{bmatrix} \sigma_1^2 & -\sigma_{12} \\ -\sigma_{12} & \sigma_2^2 \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ a-\mu_2 \end{bmatrix}\right.$$

εκτελώντας τις πράξεις παίρνουμε:

$$p(x_1|x_2=a) = \frac{1}{\sqrt{\left(\frac{\sigma_1^2\sigma_2^2-\sigma_{12}^2}{\sigma_2^2}\right)}\cdot\sqrt{2\pi}} \cdot \exp\left\{ - \frac{\left[x_1 - \left(\mu_1 + \frac{\sigma_{12}(a-\mu_2)}{\sigma_2^2}\right)\right]^2}{2\cdot\left(\frac{\sigma_1^2\sigma_2^2-\sigma_{12}^2}{\sigma_2^2}\right)} \right\}$$

συνεπώς αν $\mu' = \mu_1 + \frac{\sigma_{12}(a-\mu_2)}{\sigma_2^2}$ και $\sigma'^2 = \frac{\sigma_1^2\sigma_2^2 - \sigma_{12}^2}{\sigma_2^2}$

τότε $p(x_1|x_2=a) = \frac{1}{\sigma'\sqrt{2\pi}} \cdot \exp\left(- \frac{(x_1-\mu')}{2\sigma'^2} \right)$

άρα η σππ $p(x_1|x_2=a)$ είναι Gaussian $N(\mu', \sigma'^2)$

(β) Χρησιμοποιώντας τις σχέσεις από τη σελ. 15 των slides του μαθήματος έχουμε: $(x_1|x_2=a) \sim N(\hat{\mu}, \hat{\Sigma})$, όπου

$\hat{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a-\mu_2)$, $\hat{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$

- $\hat{\mu} = \begin{bmatrix} -2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0,5 \\ 1 \end{bmatrix} \cdot 3^{-1} \cdot (-1) = \begin{bmatrix} -\frac{6,5}{3} \\ -\frac{1}{3} \end{bmatrix}$

- $\hat{\Sigma} = \begin{bmatrix} 1 & 0,8 \\ 0,8 & 2 \end{bmatrix} - \begin{bmatrix} 0,5 \\ 1 \end{bmatrix} \cdot 3^{-1} \begin{bmatrix} 0,5 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2,75}{3} & \frac{1,9}{3} \\ \frac{1,9}{3} & \frac{5}{3} \end{bmatrix}$.

δηλαδή $p(x_1, x_2 | x_3 = 1) = \frac{1}{2\pi\sqrt{|\hat{\Sigma}|}} \exp\left\{ -\frac{1}{2}(x-\hat{\mu})^T \hat{\Sigma}^{-1}(x-\hat{\mu}) \right\}$

με $x = [x_1, x_2]^T$, $\hat{\mu} = -\frac{1}{3}\begin{bmatrix} 6,5 \\ 1 \end{bmatrix}$, $\hat{\Sigma} = \frac{1}{3}\begin{bmatrix} 2,75 & 1,9 \\ 1,9 & 5 \end{bmatrix}$

(γ) Θα χρειαστούμε πρώτα έναν γραμμικό μετασχηματισμό $x' = Ax$ έτσι ώστε να μπορούμε να εργαστούμε όπως και στο (β) ερώτημα. Δηλαδή να είναι γνωστός ένας διαμερισμός $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ στο τυχαίο διάνυσμα στο οποίο δίνεται ότι $x_2 = a$. (Κοινώς πρέπει να εναλλάξουμε τα x_2, x_3).

Ορίζουμε λοιπόν

$$\underbrace{x' = \begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix}}_{x'} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{x}$$

έτσι ώστε

$$\mu' = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}}_{\mu} = \begin{bmatrix} \mu_1 \\ \mu_3 \\ \mu_2 \end{bmatrix}$$

και

$$\Sigma' = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1 & 0{,}8 & 0{,}5 \\ 0{,}8 & 2 & 1 \\ 0{,}5 & 1 & 3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{A^T}$$

(εδώ $A^T = A$ λόγω συμμετρίας)

Πραγματοποιούμε τώρα την ίδια διαδικασία με το (β) ερώτημα:

$$\hat{\mu}' = \begin{bmatrix} -2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0{,}8 \\ 1 \end{bmatrix} \cdot 2^{-1} \cdot 1 = \begin{bmatrix} -1{,}6 \\ 2{,}5 \end{bmatrix}$$

$$\hat{\Sigma}' = \begin{bmatrix} 1 & 0{,}5 \\ 0{,}5 & 3 \end{bmatrix} - \begin{bmatrix} 0{,}8 \\ 1 \end{bmatrix} \cdot 2^{-1} \cdot \begin{bmatrix} 0{,}5 & 1 \end{bmatrix} = \begin{bmatrix} 0{,}8 & 0{,}1 \\ 0{,}25 & 2{,}5 \end{bmatrix}$$

Σχόλιο: Ένας πιο ορθός τρόπος για να περιγράψουμε τη διαδικασία $x' = A\,x$ είναι ο όρος «μετασχηματισμός ομοιότητας». Γνωρίζουμε από τη γραμμική άλγεβρα ότι τα eigenvalues και το eigenspace του πίνακα δεν μεταβάλλονται καθώς ο πίνακας A είναι ορθογώνιος, δηλαδή $A^T = A^{-1}$ και τα columns, rows είναι unit vectors. Αυτό ισχύει και για τη διαδικασία $\Sigma' = A\,\Sigma\,A^T = A\,\Sigma\,A^{-1}$, όπου η προηγούμενη σχέση ισχύει αποκλειστικά λόγω της ιδιότητας της ορθογωνιότητας του πίνακα A.

$$p(x_1, x_3 | x_2 = 1) = \frac{1}{2\pi\sqrt{|\hat{\Sigma}'|}} \cdot \exp\left\{ -\frac{1}{2} (x - \hat{\mu}')(\hat{\Sigma}')^{-1}(x - \hat{\mu}') \right\}$$

οπου $x = [x_1, x_3]^T$, $\hat{\mu}' = \begin{bmatrix} -1,6 \\ 2,5 \end{bmatrix}$ και $\hat{\Sigma}' = \begin{bmatrix} 0,8 & 0,1 \\ 0,25 & 2,5 \end{bmatrix}$

(δ)

για το (β) ερώτημα:

$$(x - \hat{\mu})^T \hat{\Sigma}^{-1} (x - \hat{\mu}) = c$$

$$\Rightarrow \left[x_1 + \frac{6,5}{3} \quad x_2 + \frac{1}{3} \right] \cdot \frac{9}{(2,75 \cdot 5 - 1,9^2)} \begin{bmatrix} \frac{5}{3} & -\frac{1,9}{3} \\ -\frac{1,9}{3} & \frac{2,75}{3} \end{bmatrix} \begin{bmatrix} x_1 + \frac{6,5}{3} \\ x_2 + \frac{1}{3} \end{bmatrix} = c$$

$$\Rightarrow \frac{3}{10,14} \left\{ \left[5(x_1 + 6,5/3) - 1,9(x_2 + 1/3) \quad 2,75(x_2 + 1/3) - 1,9(x_1 + 6,5/3) \right] \begin{bmatrix} x_1 + 6,5/3 \\ x_2 + 1/3 \end{bmatrix} \right\} = c$$

$$\Rightarrow \frac{3}{10,14} \left[5x_1^2 + 20,4x_1 + 2,75x_2^2 - 6,4x_2 - 3,8x_1x_2 + \frac{63,1}{3} \right] = c$$

$$\Rightarrow \boxed{15x_1^2 + 61,2x_1 + 8,25x_2^2 - 19,2x_2 - 11,4x_1x_2 + (63,1 - 10,14c) = 0} \quad (1)$$

για το (γ) ερώτημα:

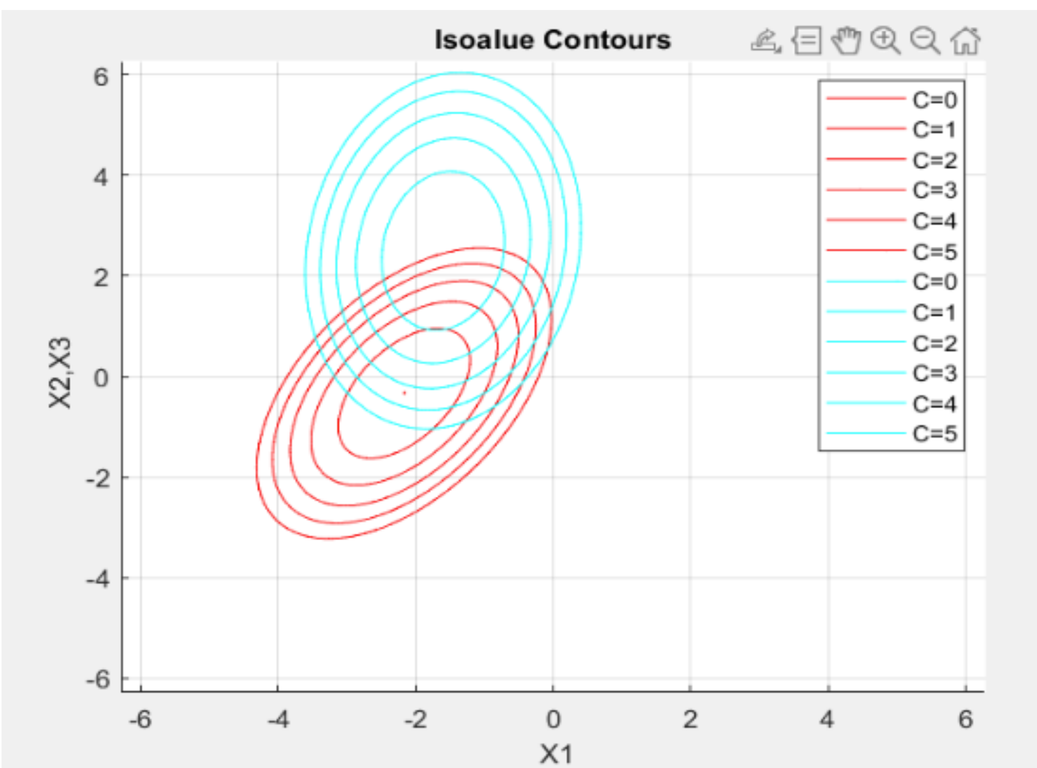
$$[x_1+1,6 \quad x_3-2,5] \begin{bmatrix}0,8 & 0,1\\0,25 & 2,5\end{bmatrix}^{-1} \begin{bmatrix}x_1+1,6\\x_3-2,5\end{bmatrix} = c$$

$$\Rightarrow \frac{1}{1,975}\left[(x_1+1,6)2,5-0,1(x_3-2,5) \quad 0,8(x_3-2,5)-0,25(x_1+1,6)\right]\begin{bmatrix}x_1+1,6\\x_3-2,5\end{bmatrix}=c$$

$$\Rightarrow \boxed{2,5x_1^2+0,8x_3^2+8,875x_1-4,56x_3-0,35x_1x_3+12,8-1,975c=0} \quad (2)$$

Οι (1) και (2) προσδιορίζουν τις εξισώσεις των ισοσταθμικών καμπύλων.

(δ) Ισοσταθμικές καμπύλες: Για καθένα από τα δύο σύνολα ισοσταθμικών καμπυλών έχει γίνει απεικόνιση με χρήση matlab για τιμές c = {0, 1, 2, 3, 4, 5}.



Κώδικας (Matlab) που χρησιμοποιήθηκε:

```
clc
clear
close all
%% Define variables &amp; equations
c=0;

Eqn= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=1;
Eqn1= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=2;
Eqn2= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=3;
Eqn3= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=4;
Eqn4= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=5;
Eqn5= @(x,y) 15*x.^2 + 61.2*x + 8.25*y.^2 -19.2*y - 11.41*x.*y + 63.1 -
10.14*c ;
c=0;
eqn= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2^2) + 8.87*x1 - 4.56*x2 -0.35*x1*x2 +
12.8 -1.975*c
c=1;
eqn1= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2.^2) + 8.87*x1 - 4.56*x2 -0.35*x1.*x2
+ 12.8 -1.975*c;
c=2;
eqn2= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2.^2) + 8.87*x1 - 4.56*x2 -0.35*x1.*x2
+ 12.8 -1.975*c;
c=3;
eqn3= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2.^2) + 8.87*x1 - 4.56*x2 -0.35*x1.*x2
+ 12.8 -1.975*c;
c=4;
eqn4= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2.^2) + 8.87*x1 - 4.56*x2 -0.35*x1.*x2
+ 12.8 -1.975*c;
c=5;
eqn5= @(x1, x2) 2.5*(x1.^2) + 0.8*(x2.^2) + 8.87*x1 - 4.56*x2 -0.35*x1.*x2
+ 12.8 -1.975*c
figure
grid on
hold on
b1=ezplot(Eqn);
set(b1,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=0&#39;)
b2=ezplot(Eqn1);
set(b2,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=1&#39;)
b3=ezplot(Eqn2);
set(b3,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=2&#39;)
b4=ezplot(Eqn3);
set(b4,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=3&#39;)
b5=ezplot(Eqn4);
set(b5,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=4&#39;)
```

```
b6=ezplot(Eqn5);
set(b6,&#39;Color&#39;,&#39;red&#39;,&#39;DisplayName&#39;,&#39;C=5&#39;)
c1=ezplot(eqn);
set(c1,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=0&#39;)
c2=ezplot(eqn1);
set(c2,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=1&#39;)
c3=ezplot(eqn2);
set(c3,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=2&#39;)
c4=ezplot(eqn3);
set(c4,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=3&#39;)
c5=ezplot(eqn4);

set(c5,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=4&#39;)
c6=ezplot(eqn5);
set(c6,&#39;Color&#39;,&#39;cyan&#39;,&#39;DisplayName&#39;,&#39;C=5&#39;)
title(&#39;Isoalue Contours&#39;)
xlabel(&#39; X1&#39;)
ylabel(&#39; X2,X3&#39;)
legend(&#39;show&#39;)
```

Άσκηση 1.3 (Bayes classifier)

Άσκηση 1.3 (Bayes classifier)

$p(\omega_1) = p(\omega_2)$

$p(x|\omega_1)$, $p(x|\omega_2)$: Gaussian $\mu_1 = [-2, 0]^T$

$\mu_2 = [2, 1]^T$

(α) κοινό $\Sigma = I_2$

$$\Rightarrow \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Από τη σελ. 27 των slides του μαθήματος παίρνουμε για το decision boundary:

$$g(x) = \frac{1}{2}\cancelto{0}{(x^T\Sigma^{-1}x - x^T\Sigma^{-1}x)} + \mu_1^T\Sigma^{-1}x - \mu_2^T\Sigma^{-1}x - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2$$

$$+ \cancelto{0}{\ln\frac{P(\omega_1)}{P(\omega_2)}} + \cancelto{0}{\frac{1}{2}\ln\frac{|\Sigma|}{|\Sigma|}} = 0$$

το οποίο λόγω του κοινού πίνακα Σ και επειδή $p(\omega_1) = p(\omega_2)$ γίνεται:

$$g(x) \underset{\mu_i^T I = \mu_i^T}{\overset{\mu_i^T I = \mu_i^T}{=}} (\mu_1^T - \mu_2^T)\,x - \frac{1}{2}\cdot(\mu_1^T\mu_1 - \mu_2^T\mu_2) = 0$$

$$\Rightarrow g(x) = [-4 \;\; -1]\,x - \frac{1}{2}(4-5) = 0$$

$$\Rightarrow g(x) = -4x_1 - x_2 + \frac{1}{2} = 0$$

Δηλαδή η βέλτιστη ευθεία απόφασης είναι εκείνη με εξίσωση

$$\boxed{8x_1 + 2x_2 - 1 = 0}$$

(β) κοινό $\hat{\Sigma} = \begin{bmatrix} 1 & -0,6 \\ -0,6 & 1 \end{bmatrix}$

ομοίως, $g(x) = (\mu_1^T - \mu_2^T)\hat{\Sigma}^{-1}x - \frac{1}{2}(\mu_1^T\hat{\Sigma}^{-1}\mu_1 - \mu_2^T\hat{\Sigma}^{-1}\mu_2) = 0$

όπου $\hat{\Sigma}^{-1} = \begin{bmatrix} 1 & -0,6 \\ -0,6 & 1 \end{bmatrix}^{-1} = \frac{1}{1-0,36}\begin{bmatrix} 1 & 0,6 \\ 0,6 & 1 \end{bmatrix} = \begin{bmatrix} 1,5625 & 0,9375 \\ 0,9375 & 1,5625 \end{bmatrix}$

$g(x) = [-7,1875 \quad -5,3125]x - \frac{1}{2}\left([-3,125 \quad -1,875]\begin{bmatrix} -2 \\ 0 \end{bmatrix} - [4,0625 \quad 3,4375]\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right)$
$= 0$

$\Rightarrow g(x) = -7,1875x_1 - 5,3125x_2 - \frac{1}{2}(-6,25 - (11,5625)) = 0$

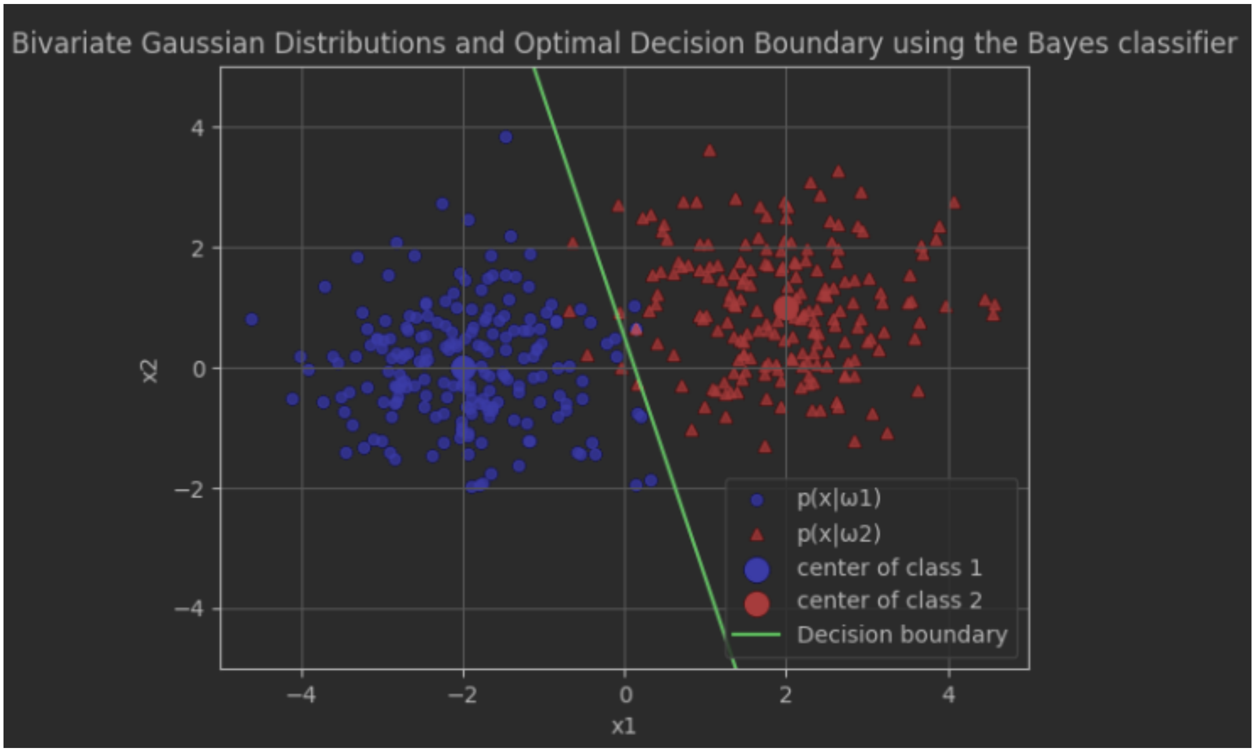
$\Rightarrow g(x) = -7,1875x_1 - 5,3125x_2 + 8,90625 = 0$

Η βέλτιστη ευθεία απόφασης είναι εκείνη με εξίσωση
$\boxed{7,1875x_1 + 5,3125x_2 - 8,90625 = 0}$

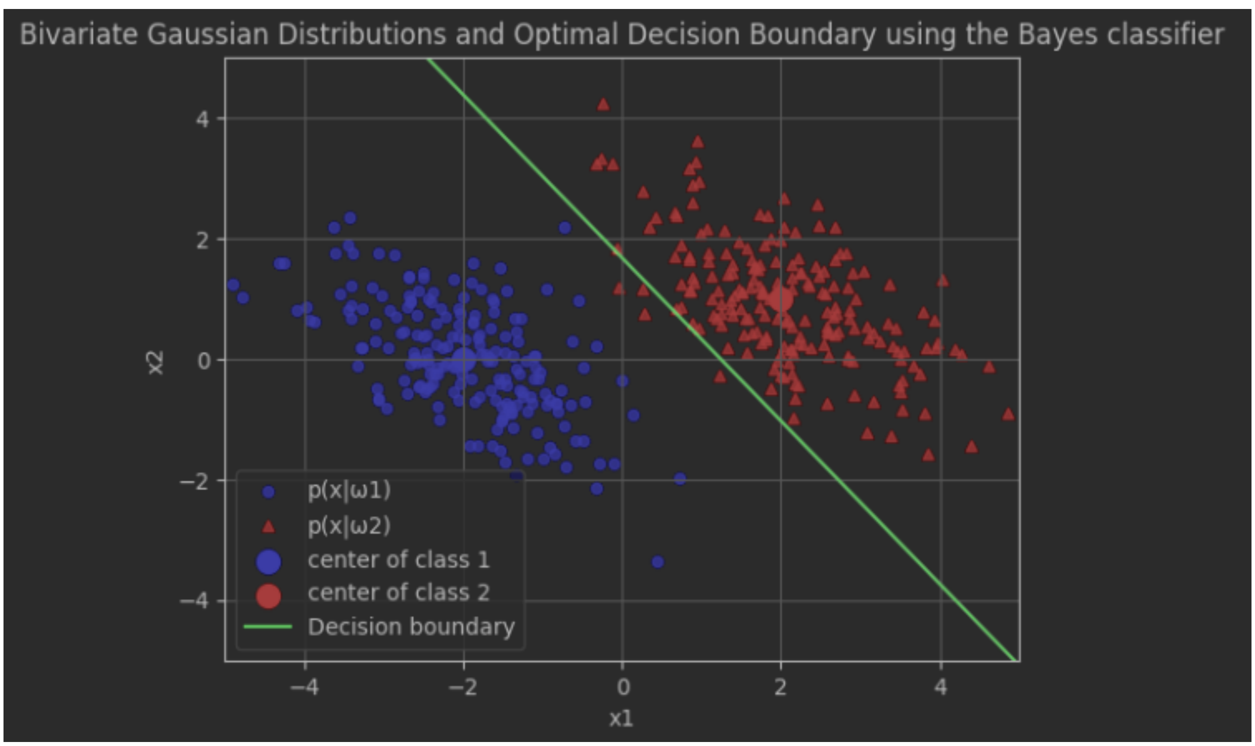
(γ) κοινό $\hat{\Sigma}$, $\lambda_{12} = 1$ και $\lambda_{21} = 1/2$

Το decision boundary επιλέγεται τώρα ως:
$\ln(\lambda_{12}\,p(x|\omega_1)) = \ln(\lambda_{21}\,p(x|\omega_2)) \Rightarrow \ln(p(x|\omega_1)) = \ln(0,5\,p(x|\omega_2))$
$\Rightarrow \ln(p(x|\omega_1)) = \ln(0,5) + \ln(p(x|\omega_2))$
όμως η εξίσωση $\ln(p(x|\omega_1)) = \ln(p(x|\omega_2))$ δεν είναι παρά η $g(x)$ του (β) ερωτήματος.
Συνεπώς έχουμε: $-7,1875x_1 - 5,3125 + 8,90625 = -0,69314 = g(x)$
$\Rightarrow g(x) = -7,1875x_1 - 5,3125x_2 + 9,6 = 0$
Η βέλτιστη ευθεία απόφασης είναι εκείνη με εξίσωση
$\boxed{7,1875x_1 + 5,3125x_2 - 9,6 = 0}$

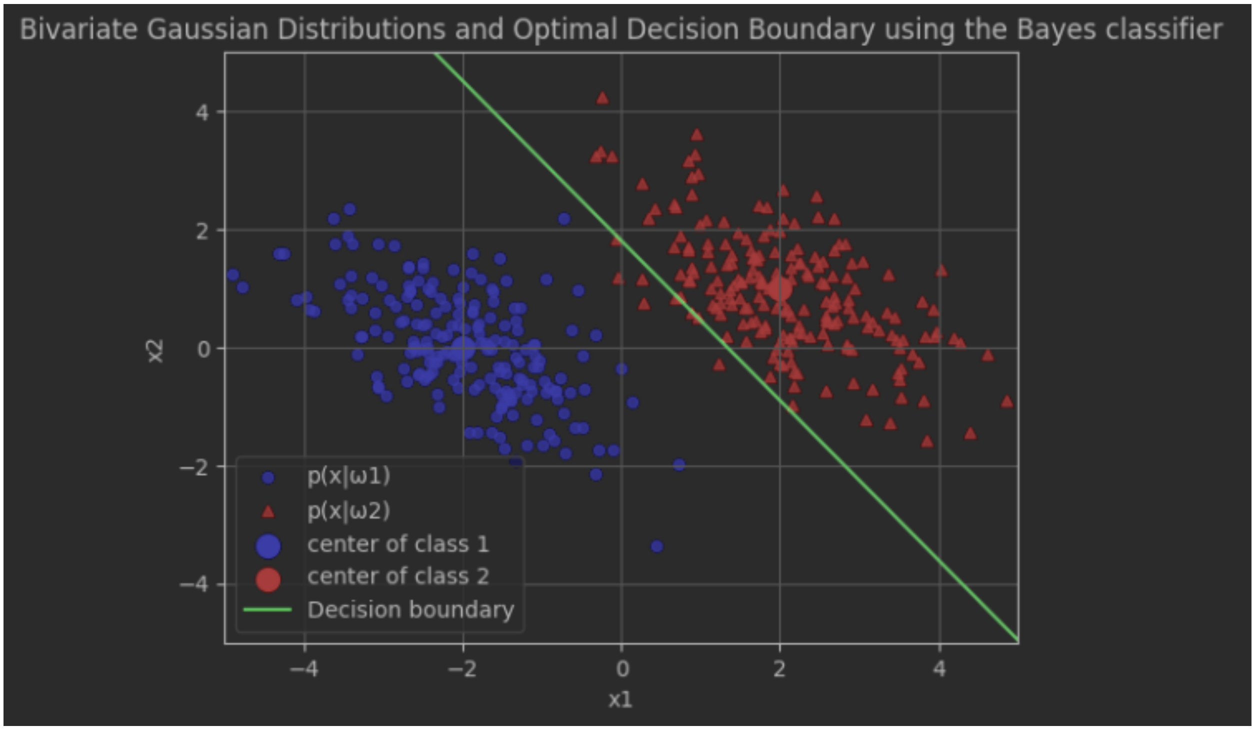
(δ)
για το (α) ερώτημα:



για το (β) ερώτημα:



για το (γ) ερώτημα:



Κώδικας (Python) που χρησιμοποιήθηκε:

```
import numpy as np
import matplotlib.pyplot as plt

# Set seed for reproducibility
np.random.seed(42)

# Parameters for the first distribution
mu1 = np.array([-2, 0])
cov1 = np.array([[1, 0], [0, 1]])

# Parameters for the second distribution
mu2 = np.array([2, 1])
cov2 = np.array([[1, 0], [0, 1]])

# Number of points to generate for each distribution
num_points = 200

# Generate points for the first distribution
points1 = np.random.multivariate_normal(mu1, cov1, num_points)
```

```python
# Generate points for the second distribution
points2 = np.random.multivariate_normal(mu2, cov2, num_points)

# Plot the points for the first distribution
plt.scatter(points1[:, 0], points1[:, 1], label='p(x|ω1)', alpha=0.7, c='b',
marker='o', s=20)

# Plot the points for the second distribution
plt.scatter(points2[:, 0], points2[:, 1], label='p(x|ω2)', alpha=0.7, c='r',
marker='^', s=20)

# Plot the means as red points
plt.scatter(mu1[0], mu1[1], color='blue', marker='o', label='center of class
1', s=100)
plt.scatter(mu2[0], mu2[1], color='red', marker='o', label='center of class
2', s=100)

# Plot the decision boundary line
x_line = np.linspace(-5, 5, 100)
y_line = (1 - 8 * x_line) / 2
plt.plot(x_line, y_line, color='green', label='Decision boundary')

# Set labels and title
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Bivariate Gaussian Distributions and Optimal Decision Boundary
using the Bayes classifier')

# Add legend
plt.legend()

# Set limits to zoom in
plt.xlim([-5, 5])
plt.ylim([-5, 5])

# Show the plot
plt.grid(True)
plt.show()
```

Το παραπάνω code cell αντιστοιχεί στο πρώτο plot, προφανώς ο κώδικας για τα άλλα 2 είναι ακριβώς ο ίδιος με μοναδική διαφορά στα:

```python
 y_line = (8.90625 - 7.1875 * x_line) / 5.3125. και
y_line = (9.6 - 7.1875 * x_line) / 5.3125 αντίστοιχα.
```

Σχόλια: Παρατηρούμε ότι τα decision boundaries είναι και στις 3 περιπτώσεις ευθείες. Αυτό συμβαίνει διότι ο πίνακας συνδιακύμανσης είναι κοινός και ως αποτέλεσμα η εξίσωση g(x) δεν περιέχει quadratic terms. Ακόμα, η υπερεπιφάνεια απόφασης στο (γ) ερώτημα δεν είναι παρά μια μετατοπισμένη μορφή εκείνης του (β) ερωτήματος: μας έχουν δοθεί τα λ_{12} και λ_{21} με $\lambda_{12} > \lambda_{21}$, το οποίο σημαίνει ότι η κλάση ω_1 είναι πιο «σημαντική» ,άρα πρέπει να αυξήσουμε τον σταθερό όρο της ευθείας, ώστε να μεγαλώσουμε την περιοχή στην οποία αποφασίζουμε υπέρ της πιο σημαντικής κλάσης ω_1.

Άσκηση 4 (Perceptron - MultiLayer Perceptron)

Άσκηση 14 (Perceptron - MultiLayer Perceptron)

(a) activation function: step function

$\vec{w} = [1, 0, 0, 0]$, $y = [0, 1, 1, 0]$

και $X = \left[\begin{bmatrix} 1 \\ 4 \\ 3 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ -2 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \\ 2 \\ 3 \end{bmatrix} \right]$

Εκπαίδευση σύμφωνα με $w_{i+1} \to w_i + n(y-\hat{y}) \cdot X^T[i]$, $i \in \{0,1,2,3\}$

$n=1$ (training step), δηλαδή $w \pm (y-\hat{y})x$,

$\hat{y}_i = f(\vec{w}_{i-1} X^T[i-1])$, εδώ f=activation function: step function

οπότε $f(\cdot) = u(\cdot)$

1ο iteration:

$\hat{y} = u(\vec{w} X[0]) = u(1) = 1$ FP

$\vec{w} += (y-\hat{y}) \cdot X^T[0] = [1\ 0\ 0\ 0] + (0-1)[1\ 4\ 3\ 6] = [0\ -4\ -3\ -6]$

$\hat{y} = u(\vec{w} \cdot X[1]) = u([0\ -4\ -3\ -6] \cdot [1\ 2\ -2\ 3]^T) = u(-19) = 0$ TN

$\vec{w} += (y-\hat{y}) X^T[1] = [0\ -4\ -3\ 6] + (1-0)[1\ 2\ -2\ 3] = [1\ -2\ -5\ 9]$

$\hat{y} = u(\hat{w} X[2]) = u([1\ -2\ -5\ 9] \cdot [1\ 1\ 0\ -3]^T) = u(-28) = 0$ TN

$\vec{w} += (y-\hat{y}) X^T[2] = [1\ -2\ -5\ 9] + [1\ 1\ 0\ -3] = [2\ -1\ -5\ -6]$

$\hat{y} = u(\hat{w} X[3]) = u([2\ -1\ -5\ 6] \cdot [1\ 4\ 2\ 3]^T) = u(6) = 1$ FP

$\vec{w} += (y-\hat{y}) X^T[3] = [2\ -1\ -5\ 6] - [1\ 4\ 2\ 3] = [1\ -5\ -7\ 3]$

2º iteration:

$\hat{y} = u(\vec{w}x(0)) = u([1 \; -5 \; -7 \; 3][1 \; 4 \; 3 \; 6]^T) = u(-22) = 0$ FN

$\hat{y} = u(\vec{w}x(1)) = u([1 \; -5 \; -7 \; 3][1 \; 2 \; -2 \; 3]^T) = u(14) = 1$ TP

$\hat{y} = u(\vec{w}x(2)) = u([1 \; -5 \; -7 \; 3][1 \; 1 \; 0 \; -3]^T) = u(-13) = 0$ TN

$\vec{w} = [1 \; -5 \; -7 \; 3] + [1 \; 1 \; 0 \; -3] = [2 \; -4 \; -7 \; 0]$

$\hat{y} = u(\vec{w}x(3)) = u([2 \; -4 \; -7 \; 0][1 \; 4 \; 2 \; 3]^T) = u(-32) = 0$ FN

3º iteration:

$\hat{y} = u(\vec{w}x(0)) = u([2 \; -4 \; -7 \; 0][1 \; 4 \; 3 \; 6]^T) = u(-25) = 0$ FN

$\hat{y} = u(\vec{w}x(1)) = u([2 \; -4 \; -7 \; 0][1 \; 2 \; -2 \; 3]^T) = u(8) = 1$ TP

$\hat{y} = u(\vec{w}x(2)) = u([2 \; -4 \; -7 \; 0][1 \; 1 \; 0 \; -3]^T) = u(-2) = 0$ TN

$\vec{w} = [2 \; -4 \; -7 \; 0] + [1 \; 1 \; 0 \; 3] = [3 \; -3 \; -7 \; 3]$

$\hat{y} = u(\vec{w}x(3)) = u([3 \; -3 \; -7 \; 3][1 \; 4 \; 2 \; 3]^T) = u(-14) = 0$ FN

4º iteration:

$\hat{y} = u(\vec{w}x(0)) = u([3 \; -3 \; -7 \; 3][1 \; 4 \; 3 \; 6]^T) = u(-12) = 0$ FN

$\hat{y} = u([3 \; -3 \; -7 \; 3][1 \; 2 \; -2 \; 3]^T) = u(11) = 1$ TP

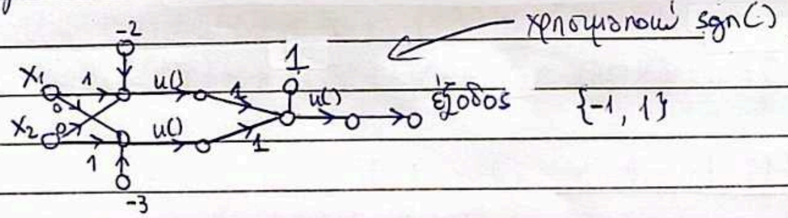
$\hat{y} = u([3 \; -3 \; -7 \; 3][1 \; 1 \; 0 \; 3]^T) = u(9) = 1$ TP

$\hat{y} = u([3 \; -3 \; -7 \; 3][1 \; 4 \; 2 \; 3]^T) = u(-14) = 0$ FN αφού στόχος = έξοδος φτάσαμε σε convergence, το οποίο έγινε σε $4 \cdot 4 = 16$ βήματα

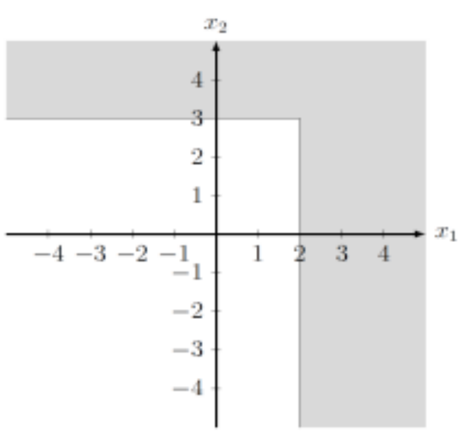
Έχουμε γραμμικά διαχωρίσιμα data, οπότε ο αριθμός των updates (iterations) k είναι άνω φραγμένος έτσι ώστε $k \le \frac{R^2}{\gamma^2}$, όπου $y_n x_n^T w^* \ge \gamma$ και $\|x_n\| \le R$

(β) Για την εικόνα του επιπέδου $\mathbb{R}^2$ έχουμε:
Για το χωρίο στο οποίο $x \in C_p$: $\{x \in C_p \Leftrightarrow x_1 \ge 2$ OR $x_2 \ge 3\}$ (δύο ευθείες).
Οι 2 πρώτοι (παράλληλοι) perceptrons έχουν p1: $(-2, 1, 0)$ και p2: $(-3, 0, 1)$
Ο p_3 συνδέεται με τους παραπάνω p_1, p_2 με λογική AND, άρα έχει p3: $(1, 1, 1)$

Τοπολογία δικτύου



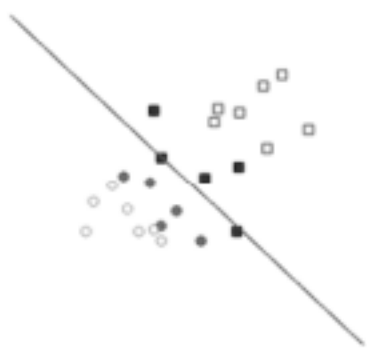
Εικόνα (β) ερωτήματος:



Άσκηση 1.5 (Support Vector Machines – Kernels)

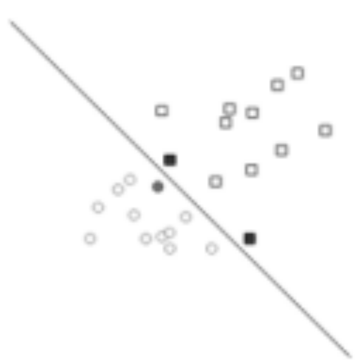
Η αντιστοίχιση γίνεται ως εξής:

(α) → 4 : έχουμε να επιλέξουμε ανάμεσα σε 2 linear SVMs (3 και 4). Αναμένουμε με μικρότερη τιμή C να έχουμε περισσότερα misclassifications και μεγαλύτερο margin, γι’αυτό επιλέγουμε το διάγραμμα 4. Παρατηρούμε επίσης περισσότερα support vectors.



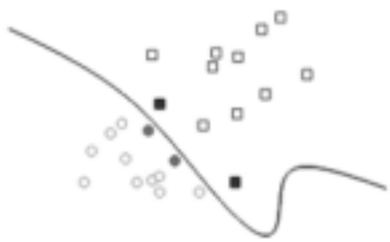
4

(β) → 3 : το μοναδικό άλλο linear SVM. Έχει μικρότερο tolerance σε misclassifications καθώς έχει μεγαλύτερη τιμή C αλλά και μικρότερο margin.



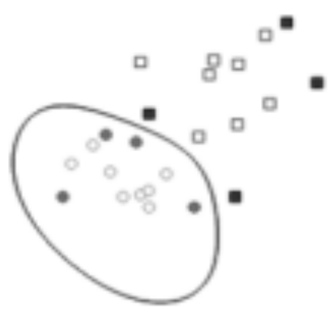
3

(γ) → 2 : αντιστοιχεί σε ένα πολυώνυμο 2ου βαθμού λόγω του όρου $(u^T v)^2$ και δεν μπορεί να αναπαραστήσει υπερβολή (διάγραμμα 5) ή μια εκ των κλειστών καμπύλων όπως έχουν τα διαγράμματα 1 και 6. Το πολυώνυμο 2ου βαθμού αντιστοιχεί λοιπόν στο διάγραμμα 2.



2

(δ) → 1 : υπάρχουν 2 διαγράμματα που αντιστοιχούν σε RBF: τα διαγράμματα 1 και 6 τα οποία διαφέρουν αποκλειστικά ως προς την παράμετρο γ. Συγκεκριμένα γνωρίζουμε ότι μεγαλύτερη τιμή της παραμέτρου γ αντιστοιχεί σε πιο «στενό» και περίπλοκο decision boundary, οπότε για τιμή γ = 0.25 επιλέγουμε το διάγραμμα 1.

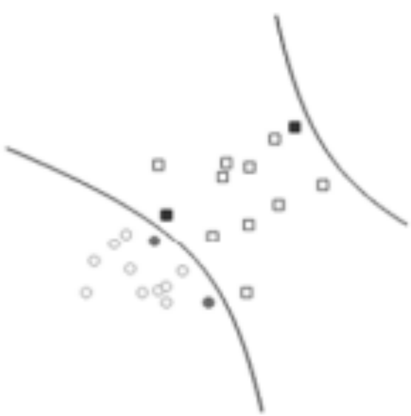


1

(ε) → 6 : με βάση το παραπάνω επιχείρημα επιλέγουμε το διάγραμμα 6.

6

(στ) → 5 : κανένα από τα υποψήφια SVMs δεν μπορεί να αναπαραστήσει το συγκεκριμένο decision boundary (υπερβολή).



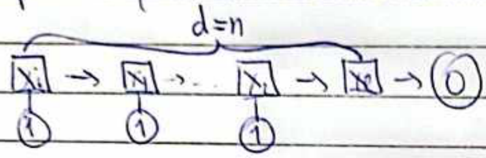
5

Άσκηση 1.6 (Decision Trees)

Άσκηση 1.6 (Decision Trees)

(a) 1. x: vector k Boolean χαρ/ών
fn classifier με βάση n χαρ/κά.

depth δέντρου d ΤΟΥΛΑΧΙΣΤΟΝ n :



Θεωρούμε το δέντρο → $h_d: \{0,1\}^d \to \{0,1\}$
Διακρίνουμε τώρα τα εξής:

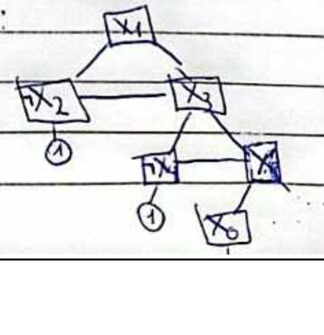
· $d<n$: ενδεχομένως $im\, h_d \subset im\, f_n$ και αντιστοιχεί στην περίπτωση στην οποία: x not in h AND x ελείπει από το fn (αλλάζει η απεικόνιση του fn)

· $d \geq n$ και $dom(f_n) \subseteq dom(h_d)$: τότε πέρα πάσης αμφιβολίας
$im(f_n) \subseteq im(h_d)$

2. fn εκφράζεται με n διαζεύξεις συζεύξεων ή αρνήσεών τους.
(unique appearance)

Θα σχεδιάσουμε το δέντρο με την εξής ιδέα:
· διαζεύξεις → κανόνες που φτάνουν στα φύλλα
Ως εκ τούτου η μέγιστη απόσταση από κάθε κόμβο σε φύλλο είναι 2. Δηλαδή (n,x):

$f_n = (x_1 \wedge \neg x_2) \vee (x_3 \wedge \neg x_4)$
$\vee (x_5 \wedge x_6)$



(β)

1. Με κριτήριο Gini:

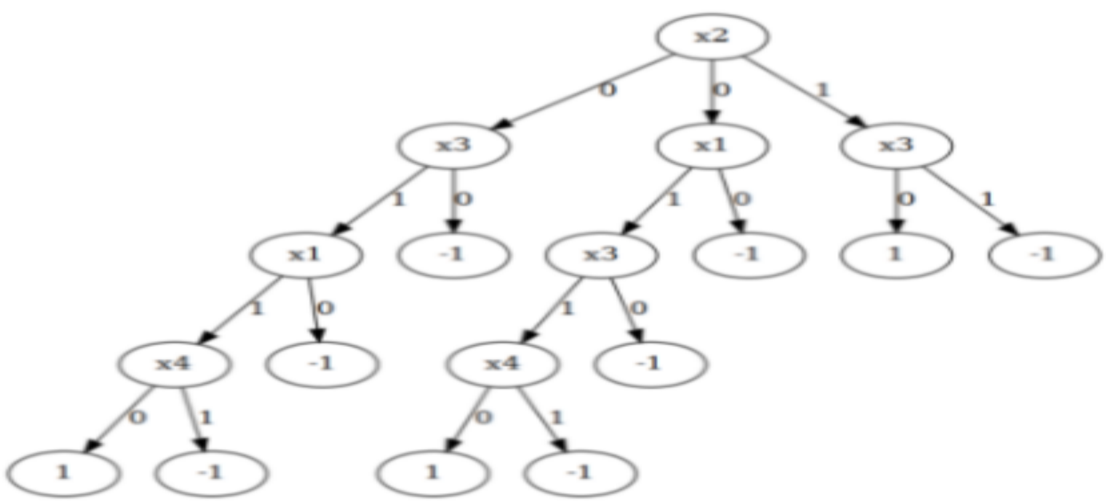
χρησιμοποιώντας τον αλγόριθμο από τα slides έχουμε:

$$gini(Root) = 1 - \left(\frac{|y=1|}{10}\right)^2 - \left(\frac{|y=0|}{10}\right)^2$$

$$gini(other) = \frac{|other=0|}{|Root|} gini(other=0) + \frac{|other=1|}{|Root|} gini(other=1),$$

$$best\_attr = \underset{other \text{ in } attr}{\operatorname{argmax}} \{gini(Root) - gini(other)\}$$

Κατά αυτόν τον τρόπο παράγονται τα εξής δέντρα:



Με κριτήριο Εντροπίας: Ως γνωστόν, $E = \sum - p_i \log_2 p_i$

$E(Root) = -\frac{5}{10} \log_2\left(\frac{5}{10}\right) \cdot 2 = 1$ $\quad E(x_1) = -\frac{5}{10} \cdot \log_2\left(\frac{5}{10}\right) \cdot 2 = 1$

$E(x_2) = -\frac{3}{10} \log_2\left(\frac{3}{10}\right) - \frac{7}{10} \log_2\left(\frac{7}{10}\right) = 0,88$

$E(x_3) = -\frac{6}{10} \log_2\left(\frac{6}{10}\right) - \frac{4}{10} \log_2\left(\frac{4}{10}\right) = 0,97$

$E(x_4) = 0,88$

$ig(x_1) = E(Root) - E(x_1) \cdot 5/10 = 0,5$ $\quad ig(x_2) = E(Root) - E(x_2) \cdot 3/10 = 0,74$

$ig(x_3) = E(Root) - E(x_3) \cdot 6/10 = 0,42$ $\quad ig(x_4) = E(Root) - E(x_4) \cdot 3/10 = 0,74$

Επιλέγουμε αυθαίρετα το x_2 (θα μπορούσαμε και το x_4)

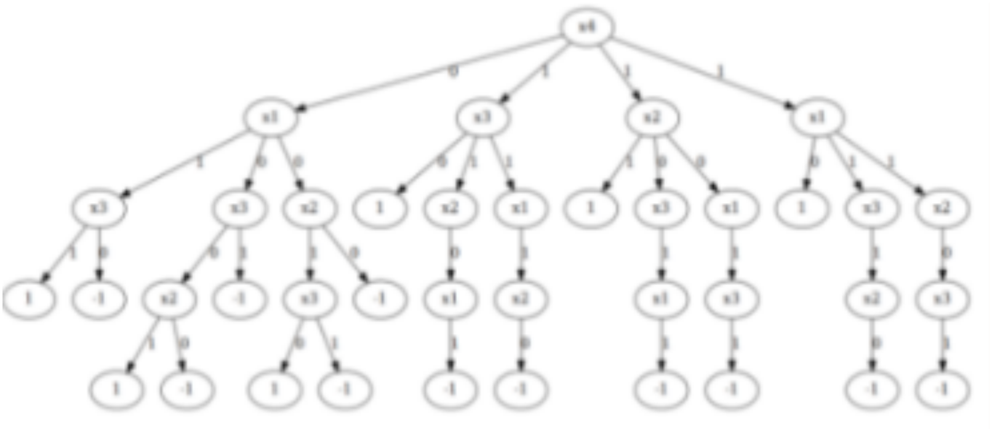
Αριστερή μεριά: $\#(x_2=0) = 7$,

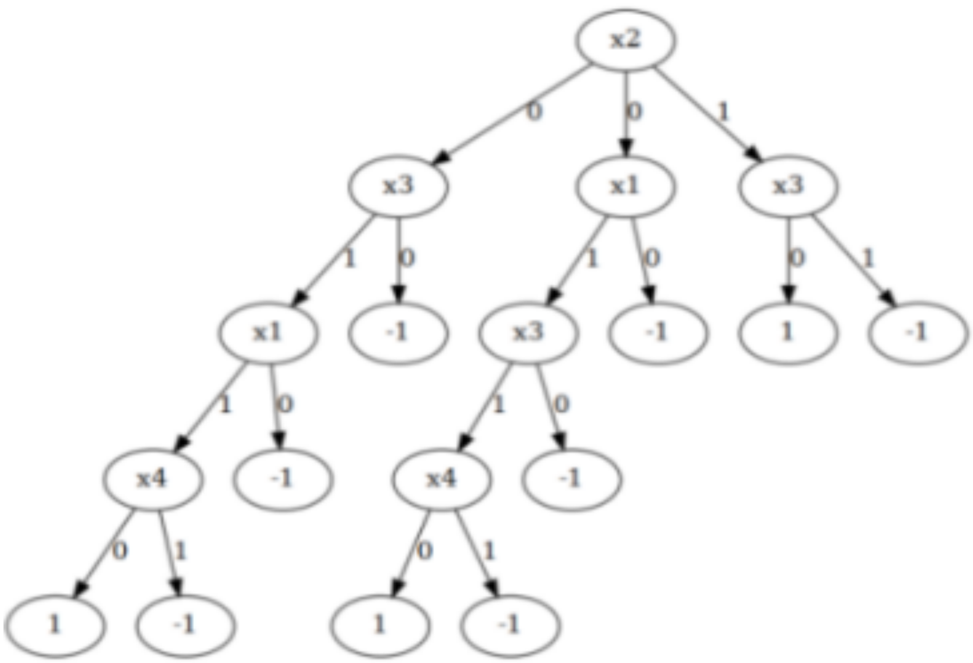
$E(x_1) = -\frac{5}{7}\left(\frac{5}{7} \log_2 \frac{5}{7} + \frac{2}{7} \log_2 \frac{2}{7}\right) = 0,61$

$E(x_3) = -\frac{5}{7}\left(\frac{5}{7} \log_2 \frac{5}{7} + \frac{2}{7} \log_2 \frac{2}{7}\right) = 0,61$

$E(x_4) = -\frac{2}{7}\left(\frac{5}{7} \log_2 \frac{5}{7} + \frac{2}{7} \log_2 \frac{2}{7}\right) = 0,246$

Συνεχίζουμε με τον ίδιο τρόπο επιλέγοντας το x_4 και προκύπτουν τα παρακάτω





2.
Τα βέλτιστα , ως προς την ακρίβεια, δέντρα απόφασης, αν έχουμε τον επιπλέον περιορισμό ότι το δέντρο πρέπει να έχει ύψος το πολύ 2 αντιστοιχούν στα δέντρα που έχουν ως ρίζες τους κόμβους που βρίσκονται σε ύψος 1 στις εικόνες.

3.
Προφανώς ένας αποδοτικός αλγόριθμος για τον ζητούμενο υπολογισμό θα ήταν δύσκολο να επινοηθεί για οποιοδήποτε dataset και για οποιουδήποτε ύψους δέντρα (σίγουρα με τις γνώσεις που έως τώρα διαθέτουμε).
Μια καλή πρακτική είναι γενικά το κλάδεμα, όπως και μικρός (ελάχιστος) αριθμός φύλλων και samples για split. Θα μπορούσαμε ακόμα, υπό κάποιες προϋποθέσεις να αντικαταστήσουμε συγκεκριμένους κόμβους (ίσως από έναν αριθμό από κλαδέματα και μετά) με την πιο συχνή κλάση, ή με μία κλάση που πληρεί -ανάλογα με το dataset- συγκεκριμένες διαφορετικές προδιαγραφές πέρα από τη συχνότητα (πχ class distribution σε διάφορα features, feature importance κλπ)

Οι παρατηρήσεις που χρησιμοποιήθηκαν στο (β) ερώτημα (χαρακτηριστικά και κλάση ταξινόμησης):

x_1	x_2	x_3	x_4	y
1	0	1	0	1
0	1	0	1	1
1	0	1	0	1
1	0	1	1	1
0	1	0	0	1
1	0	1	1	-1
0	1	1	0	-1
0	0	0	0	-1
0	0	1	0	-1
1	0	0	0	-1