**Άσκηση 1.3**

**(α)** Python Code για το υπολογισμό πιθανοτήτων  
import numpy as np

from scipy.stats import multivariate\_normal

# Given data

mu1 = np.array([3, 2])

mu2 = np.array([4, 3])

mu3 = np.array([6, 7])

cov\_matrix = np.array([[1, 0.5], [0.5, 1]])

priors = np.array([0.2, 0.5, 0.3])

x = np.array([3, 5])

# Calculate likelihoods

p\_x\_given\_w1 = multivariate\_normal.pdf(x, mean=mu1, cov=cov\_matrix)

p\_x\_given\_w2 = multivariate\_normal.pdf(x, mean=mu2, cov=cov\_matrix)

p\_x\_given\_w3 = multivariate\_normal.pdf(x, mean=mu3, cov=cov\_matrix)

# Calculate posteriors using Bayes' rule

evidence = p\_x\_given\_w1 \* priors[0] + p\_x\_given\_w2 \* priors[1] + p\_x\_given\_w3 \* priors[2]

p\_w1\_given\_x = (p\_x\_given\_w1 \* priors[0]) / evidence

p\_w2\_given\_x = (p\_x\_given\_w2 \* priors[1]) / evidence

p\_w3\_given\_x = (p\_x\_given\_w3 \* priors[2]) / evidence

print("Posterior probabilities:")

print(f"P(ω1 | x) = {p\_w1\_given\_x:.4f}")

print(f"P(ω2 | x) = {p\_w2\_given\_x:.4f}")

print(f"P(ω3 | x) = {p\_w3\_given\_x:.4f}")

Output:

Posterior probabilities:

P(ω1 | x) = 0.0618

P(ω2 | x) = 0.5864

P(ω3 | x) = 0.3518

**(γ)** Παρακάτω φαίνονται οι καμπύλες απόφασης που βρήκαμε.

A graph with lines and numbers

Description automatically generated

A graph with lines and numbers

Description automatically generated with medium confidence

import numpy as np

import matplotlib.pyplot as plt

x\_vals = np.linspace(0, 30, 1000)

# Boundary line between ω1 and ω2: 3x1 - 9x2 = 7.3744

boundary12\_y = (3 \* x\_vals - 7.3744) / 9

# Boundary line between ω2 and ω3: x2 = 5.1277 (horizontal line)

boundary23\_y = np.full\_like(x\_vals, 5.1277)

# Boundary line between ω1 and ω3: -3x1 + 15x2 = 23.3918

boundary13\_y = (3 \* x\_vals + 23.3918) / 15

# Plotting

plt.figure(figsize=(10, 8))

# Plot decision boundaries

plt.plot(x\_vals, boundary12\_y, 'r', label='Boundary ω1-ω2')

plt.plot(x\_vals, boundary23\_y, 'g', label='Boundary ω2-ω3')

plt.plot(x\_vals, boundary13\_y, 'b', label='Boundary ω1-ω3')

# Additional plot settings

plt.xlabel('x1')

plt.ylabel('x2')

plt.title('Class Distributions with Decision Boundaries')

plt.legend()

plt.grid(True)

plt.xlim(0, 30)

plt.ylim(-5, 10)

# Plotting

plt.figure(figsize=(10, 8))

# Plot decision boundaries

plt.plot(x\_vals, boundary12\_y, 'm', label='Boundary ω1-ω2')

plt.plot(x\_vals, boundary23\_y, 'k', label='Boundary ω2-ω3')

plt.plot(x\_vals, boundary13\_y, 'c', label='Boundary ω1-ω3')

# # Fill areas based on classification regions

plt.fill\_between(x\_vals, boundary23\_y, 10, color='lightcoral', alpha=0.3, label='Class ω3')

plt.fill\_between(x\_vals, boundary12\_y, boundary23\_y, where=(boundary12\_y <= boundary23\_y),color='lightblue', alpha=0.3, label='Class ω2')

plt.fill\_between(x\_vals, -10, boundary12\_y, color='lightgreen', alpha=0.3, label='Class ω1'

# Additional plot settings

plt.xlabel('x1')

plt.ylabel('x2')

plt.title('Class Distributions with Decision Boundaries')

plt.legend()

plt.grid(True)

plt.xlim(0, 17.84126)

plt.ylim(-5, 10)

plt.show()

A diagram of a class distribution

Description automatically generated**(δ)** Δίνεται το ζητούμενο σχήμα:

Για το ζητούμενο σχήμα χρησιμοποιήθηκε ο εξής κώδικας:

import numpy as np

import matplotlib.pyplot as plt

# Define parameters for each class

mean1 = np.array([3, -2])

mean2 = np.array([4, 3])

mean3 = np.array([6, 7])

cov\_matrix = np.array([[1, 0.5], [0.5, 1]])  # Shared covariance matrix

priors = np.array([0.2, 0.5, 0.3])

# Generate 500 points for each class

np.random.seed(0)  # For reproducibility

p\_x\_given\_w1 = np.random.multivariate\_normal(mean1, cov\_matrix, 500)

p\_x\_given\_w2 = np.random.multivariate\_normal(mean2, cov\_matrix, 500)

p\_x\_given\_w3 = np.random.multivariate\_normal(mean3, cov\_matrix, 500)

# Define decision boundaries as lines based on given equations

x\_vals = np.linspace(-3, 20, 1000)

# Boundary line between ω1 and ω2: 3x1 - 9x2 = 7.3744

boundary12\_y = (3 \* x\_vals - 7.3744) / 9

# Boundary line between ω2 and ω3: x2 = 5.1277 (horizontal line)

boundary23\_y = np.full\_like(x\_vals, 5.1277)

# Boundary line between ω1 and ω3: -3x1 + 15x2 = 23.3918

boundary13\_y = (3 \* x\_vals + 23.3918) / 15

# Plotting

plt.figure(figsize=(10, 8))

# Scatter plot for each class

plt.scatter(p\_x\_given\_w1[:, 0], p\_x\_given\_w1[:, 1], color='red', label='Class ω1', alpha=0.6)

plt.scatter(p\_x\_given\_w2[:, 0], p\_x\_given\_w2[:, 1], color='blue', label='Class ω2', alpha=0.6)

plt.scatter(p\_x\_given\_w3[:, 0], p\_x\_given\_w3[:, 1], color='green', label='Class ω3', alpha=0.6)

# Plot decision boundaries

plt.plot(x\_vals, boundary12\_y, 'm--', label='Boundary ω1-ω2')

plt.plot(x\_vals, boundary23\_y, 'k--', label='Boundary ω2-ω3')

plt.plot(x\_vals, boundary13\_y, 'c--', label='Boundary ω1-ω3')

# Plot centers of each class

plt.plot(mean1[0], mean1[1], 'ro', marker='x', markersize=10, label='Center ω1')

plt.plot(mean2[0], mean2[1], 'bo', marker='x', markersize=10, label='Center ω2')

plt.plot(mean3[0], mean3[1], 'go', marker='x', markersize=10, label='Center ω3')

# Additional plot settings

plt.xlabel('x1')

plt.ylabel('x2')

plt.title('Class Distributions with Decision Boundaries')

plt.legend()

plt.grid(True)

plt.xlim(-3, 17.84126)

plt.ylim(-5, 10)

plt.show()

**(ε)** Υπολογίστηκε με δύο τρόπους. Αρχικά, πήρα 10000 δείγματα από την κατανομή και βλέπουμε το ποσοστό που αυτά σύμφωνα με τον ταξινομιτή Bayes γίνονται misclassified.

Αποτέλεσμα: Probability of misclassification of Class w2 is: 0.0191

Κώδικας:

import numpy as np

from scipy.stats import multivariate\_normal

# Data

mu1 = np.array([3, -2])

mu2 = np.array([4, 3])

mu3 = np.array([6, 7])

sigma = np.array([[1, 0.5], [0.5, 1]])

priors = [0.2, 0.5, 0.3]

# Creating 10000 samples

num\_samples = 10000

samples = multivariate\_normal.rvs(mean=mu2, cov=sigma, size=num\_samples)

# Miclassification initialization

misclassified\_count = 0

# Compute posterior probability and classify them with Bayes

for x in samples:

    # Computation of PDF for every distribution

    p\_x\_given\_w1 = multivariate\_normal.pdf(x, mean=mu1, cov=sigma) \* priors[0]

    p\_x\_given\_w2 = multivariate\_normal.pdf(x, mean=mu2, cov=sigma) \* priors[1]

    p\_x\_given\_w3 = multivariate\_normal.pdf(x, mean=mu3, cov=sigma) \* priors[2]

    # Classification with Bayes classifier

    predicted\_class = np.argmax([p\_x\_given\_w1, p\_x\_given\_w2, p\_x\_given\_w3])

    # If misclassification then increase variable

    if predicted\_class != 1:

        misclassified\_count += 1

# Final Output

error\_probability = misclassified\_count / num\_samples

print("Probability of misclassification of Class w2 is:", error\_probability)

Ο δεύτερος τρόπος είναι ο υπολογισμός του ολοκληρώματος στις περιοχές που δεν ανήκει στην κλάση .

Αποτέλεσμα: Probability of Misclaffication is: 0.01909033805483369

Κώδικας: import numpy as np

from scipy.stats import multivariate\_normal

from scipy.integrate import dblquad

# Given parameters

mean\_w2 = [4, 3]

cov\_matrix = [[1, 0.5], [0.5, 1]]

# Define the distribution for p(x|w2)

rv\_w2 = multivariate\_normal(mean=mean\_w2, cov=cov\_matrix)

# Define the boundaries

boundary12 = lambda x1: (3 \* x1 - 7.3744) / 9  # Boundary ω1-ω2

boundary23 = 5.1277  # Boundary ω2-ω3

# Misclassification region for ω3 (x2 > boundary23)

def integrand\_w3(x2, x1):

    return rv\_w2.pdf([x1, x2])

# Misclassification region for ω1 (x2 < boundary12)

def integrand\_w1(x2, x1):

    return rv\_w2.pdf([x1, x2])

# Integration limits

x1\_min, x1\_max = -np.inf, 17.84126

# Compute the probability for ω3 region

P\_error\_w3, \_ = dblquad(

    integrand\_w3,

    x1\_min, x1\_max,

    lambda x1: boundary23,  # Lower bound for x2

    lambda x1: np.inf       # Upper bound for x2

)

# Compute the probability for ω1 region

P\_error\_w1, \_ = dblquad(

    integrand\_w1,

    x1\_min, x1\_max,

    lambda x1: -np.inf,            # Lower bound for x2

    lambda x1: boundary12(x1)      # Upper bound for x2

)

# Total misclassification probability

P\_misclassification = P\_error\_w1 + P\_error\_w3

print(f"Probability of Misclassification is: {P\_misclassification}")