CS4044D Machine Learning
Assignemnt 1

Nikhil R, B180283CS

November 9, 2021

Intial Configuration

[77]:
```
import numpy as np

data = {
    "w1": {
        "x1": np.array([-5.01, -5.43, 1.08, 0.86, -2.67, 4.94, -2.51, -2.25, 5.56, 1.03]),
        "x2": np.array([-8.12, -3.48, -5.52, -3.78, 0.63, 3.29, 2.09, -2.13, 2.86, -3.33]),
        "x3": np.array([-3.68, -3.54, 1.66, -4.11, 7.39, 2.08, -2.59, -6.94, -2.26, 4.33])
    },
    "w2": {
        "x1": np.array([-0.91, 1.30, -7.75, -5.47, 6.14, 3.60, 5.37, 7.18, -7.39, -7.50]),
        "x2": np.array([-0.18, -2.06, -4.54, 0.50, 5.72, 1.26, -4.63, 1.46, 1.17, -6.32]),
        "x3": np.array([-0.05, -3.53, -0.95, 3.92, -4.85, 4.36, -3.65, -6.66, 6.30, -0.31])
    },
    "w3": {
        "x1": np.array([5.35, 5.12, -1.34, 4.48, 7.11, 7.17, 5.75, 0.77, 0.90, 3.52]),
        "x2": np.array([2.26, 3.22, -5.31, 3.42, 2.39, 4.33, 3.97, 0.27, -0.43, -0.36]),
        "x3": np.array([8.13, -2.66, -9.87, 5.19, 9.21, -0.98, 6.65, 2.41, -8.71, 6.43])
    }
}
```

Question 1

Write a function (in Python or any language of your choice) to calculate the discriminant function for the given normal density equation (as given below) and prior probabilities.

$g_i(x) = \frac{-1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}ln2\pi - \frac{1}{2}ln|\Sigma_i| + lnP(\omega_i)$

[78]:
```
def discriminant(x, mean, cov, prior):
  cov_inv = np.linalg.inv(cov)
  cov_det = np.linalg.det(cov)
  d = len(mean)

  p1 = -0.5*(x-mean).T.dot(cov_inv).dot(x - mean)
  p2 = -0.5*d*np.log(2*np.pi)
  p3 = -0.5*np.log(cov_det)
  p4 = np.log(prior)

  return p1 + p2 + p3 + p4
```

Question 2

Consider the problem of classifying 10 samples from the above table of data. Assume the that the underlying distributions are normal.

Setting up a Dichotomizer class that uses the above discriminant function and some data to output dichotomizers that is used in the upcoming questions.

[79]:

```
# A class that returns a dichotomizer by using the given data
class Dichotomizer():
  def __init__(self, category1, category2, prior):
    self.d = len(category1)

    self.cat1 = {
      "mean": np.mean(category1, axis=1).reshape((self.d, 1)),
      "cov": np.cov(category1).reshape(self.d, self.d)
    }

    self.cat2 = {
      "mean": np.mean(category2, axis=1).reshape((self.d, 1)),
      "cov": np.cov(category2).reshape(self.d, self.d)
    }

    self.prior = prior

  def __call__(self, x):
    a = self.discriminant(x, self.cat1["mean"], self.cat1["cov"], self.prior[0])
    b = self.discriminant(x, self.cat2["mean"], self.cat2["cov"], self.prior[1])
    return "w1" if a > b else "w2"

  def discriminant(self, x, mean, cov, prior):
    cov_inv = np.linalg.inv(cov)
    cov_det = np.linalg.det(cov)
    d = len(mean)

    p1 = -0.5*(x-mean).T.dot(cov_inv).dot(x - mean)
    p2 = -0.5*d*np.log(2*np.pi)
    p3 = -0.5*np.log(cov_det)
    p4 = np.log(prior)

    return p1 + p2 + p3 + p4
```

Question 2.a

Assume the prior probabilities of the first two categories are equal and is equal to 1/2 and that of the third category is zero. Design a dichotomizer for those two categories using the feature x1 alone.

[80]:

```
dichotomizer1 = Dichotomizer([data["w1"]["x1"]], [data["w2"]["x1"]], [1/2, 1/2])
```

Question 2.b

Determine the percentage of points misclassified.

[81]:

```
# Utility function to classify for different features
# Useful for upcoming questions
def classify(d, dichotomizer, data = data):
  features = [f"x{x+1}" for x in list(range(d))]
  print("Using Features:", ','.join(features))
  print('='*20)
  overall = 0

  for cls in data:
    print("\nClass", cls)
    print('='*50)
    correct = 0

    for i in range(len(data[cls]["x1"])):
      point = [data[cls]["x1"][i], data[cls]["x2"][i], data[cls]["x3"][i]]
      selectFeatures = np.array([point[0:d]]).T
      res = dichotomizer(selectFeatures)

      if res == cls:
        correct += 1

      print("Point", point, "\tis classified as Class ", res)

    overall += 100 - correct*10
    print(f"\nPercentage of points missclassified: {100 - correct*10} %")

  print("\n" + '='*50)
  print(f"Overall Percantage missclassified: {np.round(overall/3, 2)}%")
  print('='*50)

classify(1, dichotomizer1)
```

```
Using Features: x1
====================

Class w1
==================================================
Point [-5.01, -8.12, -3.68]     is classified as Class  w1
Point [-5.43, -3.48, -3.54]     is classified as Class  w2
Point [1.08, -5.52, 1.66]       is classified as Class  w1
Point [0.86, -3.78, -4.11]      is classified as Class  w1
Point [-2.67, 0.63, 7.39]       is classified as Class  w1
Point [4.94, 3.29, 2.08]        is classified as Class  w2
Point [-2.51, 2.09, -2.59]      is classified as Class  w1
Point [-2.25, -2.13, -6.94]     is classified as Class  w1
Point [5.56, 2.86, -2.26]       is classified as Class  w2
Point [1.03, -3.33, 4.33]       is classified as Class  w1

Percentage of points missclassified: 30 %

Class w2
==================================================
Point [-0.91, -0.18, -0.05]     is classified as Class  w1
Point [1.3, -2.06, -3.53]       is classified as Class  w1
Point [-7.75, -4.54, -0.95]     is classified as Class  w2
Point [-5.47, 0.5, 3.92]        is classified as Class  w2
Point [6.14, 5.72, -4.85]       is classified as Class  w2
Point [3.6, 1.26, 4.36]         is classified as Class  w1
Point [5.37, -4.63, -3.65]      is classified as Class  w2
Point [7.18, 1.46, -6.66]       is classified as Class  w2
Point [-7.39, 1.17, 6.3]        is classified as Class  w2
Point [-7.5, -6.32, -0.31]      is classified as Class  w2

Percentage of points missclassified: 30 %

Class w3
==================================================
Point [5.35, 2.26, 8.13]        is classified as Class  w2
Point [5.12, 3.22, -2.66]       is classified as Class  w2
Point [-1.34, -5.31, -9.87]     is classified as Class  w1
Point [4.48, 3.42, 5.19]        is classified as Class  w2
Point [7.11, 2.39, 9.21]        is classified as Class  w2
Point [7.17, 4.33, -0.98]       is classified as Class  w2
Point [5.75, 3.97, 6.65]        is classified as Class  w2
Point [0.77, 0.27, 2.41]        is classified as Class  w1
Point [0.9, -0.43, -8.71]       is classified as Class  w1
Point [3.52, -0.36, 6.43]       is classified as Class  w1

Percentage of points missclassified: 100 %

==================================================
Overall Percantage missclassified: 53.33%
==================================================
```

Question 2.c

Repeat the above two steps, but now use the two features x1 and x2.

[82]:

```
dichotomizer2 = Dichotomizer(
    [data["w1"]["x1"], data["w1"]["x2"]],
    [data["w2"]["x1"], data["w2"]["x2"]],
    [1/2, 1/2]
    )

classify(2, dichotomizer2)
```

```
Using Features: x1,x2
====================

Class w1
==================================================
Point [-5.01, -8.12, -3.68]     is classified as Class  w1
Point [-5.43, -3.48, -3.54]     is classified as Class  w2
Point [1.08, -5.52, 1.66]       is classified as Class  w1
Point [0.86, -3.78, -4.11]      is classified as Class  w1
Point [-2.67, 0.63, 7.39]       is classified as Class  w2
Point [4.94, 3.29, 2.08]        is classified as Class  w2
Point [-2.51, 2.09, -2.59]      is classified as Class  w2
Point [-2.25, -2.13, -6.94]     is classified as Class  w1
Point [5.56, 2.86, -2.26]       is classified as Class  w2
Point [1.03, -3.33, 4.33]       is classified as Class  w1

Percentage of points missclassified: 50 %

Class w2
==================================================
Point [-0.91, -0.18, -0.05]     is classified as Class  w1
Point [1.3, -2.06, -3.53]       is classified as Class  w1
Point [-7.75, -4.54, -0.95]     is classified as Class  w2
Point [-5.47, 0.5, 3.92]        is classified as Class  w2
Point [6.14, 5.72, -4.85]       is classified as Class  w2
Point [3.6, 1.26, 4.36]         is classified as Class  w1
Point [5.37, -4.63, -3.65]      is classified as Class  w2
Point [7.18, 1.46, -6.66]       is classified as Class  w2
Point [-7.39, 1.17, 6.3]        is classified as Class  w2
Point [-7.5, -6.32, -0.31]      is classified as Class  w1

Percentage of points missclassified: 40 %

Class w3
==================================================
Point [5.35, 2.26, 8.13]        is classified as Class  w2
Point [5.12, 3.22, -2.66]       is classified as Class  w2
Point [-1.34, -5.31, -9.87]     is classified as Class  w1
Point [4.48, 3.42, 5.19]        is classified as Class  w1
Point [7.11, 2.39, 9.21]        is classified as Class  w2
Point [7.17, 4.33, -0.98]       is classified as Class  w2
Point [5.75, 3.97, 6.65]        is classified as Class  w2
Point [0.77, 0.27, 2.41]        is classified as Class  w1
Point [0.9, -0.43, -8.71]       is classified as Class  w1
Point [3.52, -0.36, 6.43]       is classified as Class  w1

Percentage of points missclassified: 100 %

==================================================
Overall Percantage missclassified: 63.33%
==================================================
```

Question 2.d

Repeat again, with all the three features taken.

[83]:

```
dichotomizer3 = Dichotomizer(
    [data["w1"]["x1"], data["w1"]["x2"], data["w1"]["x3"]],
    [data["w2"]["x1"], data["w2"]["x2"], data["w2"]["x3"]],
    [1/2, 1/2]
    )
classify(3, dichotomizer3)
```

```
Using Features: x1,x2,x3
====================

Class w1
==================================================
Point [-5.01, -8.12, -3.68]     is classified as Class  w1
Point [-5.43, -3.48, -3.54]     is classified as Class  w1
Point [1.08, -5.52, 1.66]       is classified as Class  w1
Point [0.86, -3.78, -4.11]      is classified as Class  w1
Point [-2.67, 0.63, 7.39]       is classified as Class  w2
Point [4.94, 3.29, 2.08]        is classified as Class  w1
Point [-2.51, 2.09, -2.59]      is classified as Class  w1
Point [-2.25, -2.13, -6.94]     is classified as Class  w1
Point [5.56, 2.86, -2.26]       is classified as Class  w2
Point [1.03, -3.33, 4.33]       is classified as Class  w1

Percentage of points missclassified: 20 %

Class w2
==================================================
Point [-0.91, -0.18, -0.05]     is classified as Class  w2
Point [1.3, -2.06, -3.53]       is classified as Class  w2
Point [-7.75, -4.54, -0.95]     is classified as Class  w2
Point [-5.47, 0.5, 3.92]        is classified as Class  w2
Point [6.14, 5.72, -4.85]       is classified as Class  w2
Point [3.6, 1.26, 4.36]         is classified as Class  w1
Point [5.37, -4.63, -3.65]      is classified as Class  w2
Point [7.18, 1.46, -6.66]       is classified as Class  w2
Point [-7.39, 1.17, 6.3]        is classified as Class  w2
Point [-7.5, -6.32, -0.31]      is classified as Class  w2

Percentage of points missclassified: 10 %

Class w3
==================================================
Point [5.35, 2.26, 8.13]        is classified as Class  w1
Point [5.12, 3.22, -2.66]       is classified as Class  w2
Point [-1.34, -5.31, -9.87]     is classified as Class  w1
Point [4.48, 3.42, 5.19]        is classified as Class  w1
Point [7.11, 2.39, 9.21]        is classified as Class  w1
Point [7.17, 4.33, -0.98]       is classified as Class  w2
Point [5.75, 3.97, 6.65]        is classified as Class  w1
Point [0.77, 0.27, 2.41]        is classified as Class  w1
Point [0.9, -0.43, -8.71]       is classified as Class  w1
Point [3.52, -0.36, 6.43]       is classified as Class  w1

Percentage of points missclassified: 100 %

==================================================
Overall Percantage missclassified: 43.33%
==================================================
```

Question 2.e

Compare your results and conclude.

- With only feature x1 selected, the misclassification rate was 53.33%
- Misclassification rate was increased to 63.33% when both feature x1 and x2 was taken
- Misclassification rate was decreased to 43.33% when all the features were taken
- We can conclude that feature selection is an important part in classification
- The best features can be selected by comparing the covariance

Question 2.f

Classify the points (1,2,1)t,, (5,3,2)t , (0,0,0)t , (1,0,0)t using each feature vector mentioned above and compare the results

[84]:

```python
points = [[1,2,1], [5,3,2], [0,0,0], [1,0,0]]

dichotomizers = [dichotomizer1, dichotomizer2, dichotomizer3]
outputs = {}
for point in points:
  for i in range(3):
    key = ",".join(map(str, point))
    selectFeatures = np.array([point[0:i+1]]).T
    cls = dichotomizers[i](selectFeatures)
    if ( key in outputs):
      outputs[key].append(f"Class {cls}")
    else:
      outputs[key] = [f"Class {cls}"]

print("Points\t\tOnly x1\t\tx1, x2\t\tx1, x2, x3")
print('='*60)
for key, value in outputs.items():
  print("{}\t\t{}".format(key, "\t".join(value)))
```

```
Points          Only x1         x1, x2          x1, x2, x3
============================================================
1,2,1           Class w1        Class w1        Class w2
5,3,2           Class w2        Class w2        Class w1
0,0,0           Class w1        Class w1        Class w1
1,0,0           Class w1        Class w1        Class w1
```