Statistical Machine Learning: Exercise 1

Refresher: Linear Algebra, Probabilities and Bayesian Decision Theory
Group 82: Alper Gece, Jinyao Chen

TECHNISCHE UNIVERSITÄT DARMSTADT

Summer Term 2021

Task 1: Machine Learning Introduction

1a) Model Fitting

We can use linear model classification as our first model.

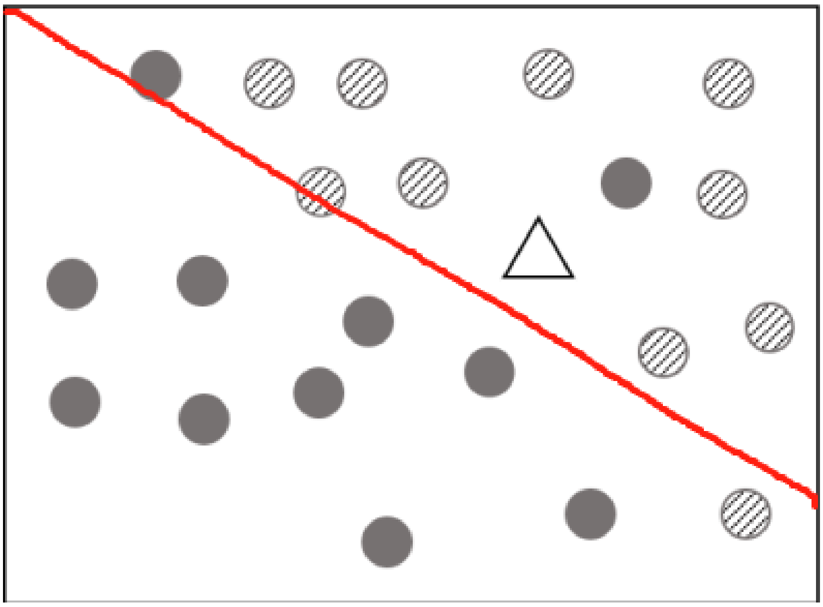

Figure 1: Classification of the first model

The right classification is given below.

striped circle	triangle	black circle
10	1	11

What we get from the first model is given below.

striped circle	black circle
11	9

The accuracy is for the striped circle is 10/10 = 1
The accuracy is for the black circle is 9/11 ≈ 0.82
As a result, the linear model (classification) cannot precisely classify the training set.
We can use overfit model classification as our second model.

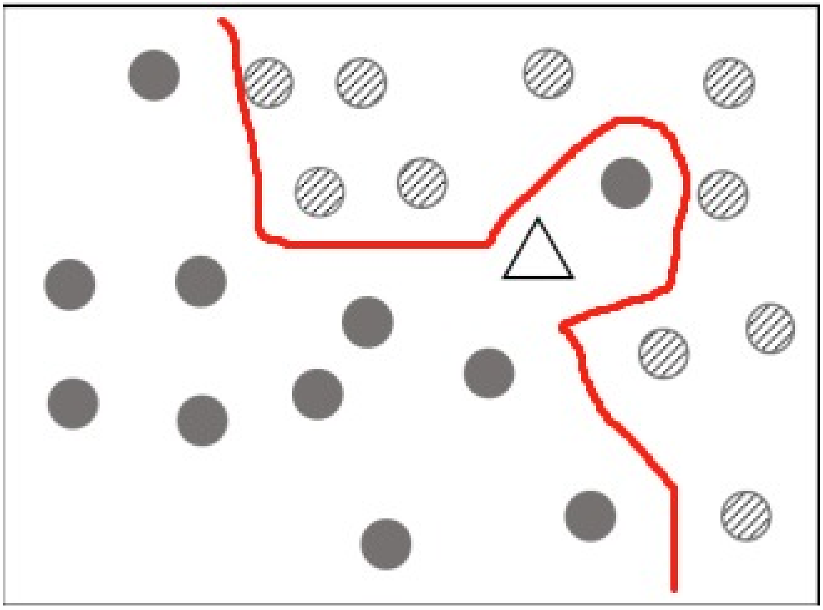

Figure 2: Classification of the second model

The right classification is given below.

striped circle	triangle	black circle
10	1	11

What we get from the first model is given below.

striped circle	black circle
10	12

The accuracy is for the striped circle is 10/10 = 1
The accuracy is for the black circle is 11/11 = 1

As a result, the overfit model classification gives us better results than the linear model classification.

Task 2: Linear Algebra Refresher

2a) Matrix Properties

1. Disprove of commutative properties of matrix
   For an order n square matrix D, let D′ be the square matrix of order n+1 defined as:

   $$d'_{ij} = \begin{cases} d_{ij}: & i < n+1 \wedge j < n+1 \\ 0: & i = n+1 \vee j = n+1 \end{cases}$$

   Thus D′ is just D with a zero row and zero column added at the ends. We have that D is a submatrix of D′.
   Now:

   $$(a'b')_{ij} = \begin{cases} \sum_{r=1}^{n+1} a'_{ir}b'_{rj}, & i < n+1 \wedge j < n+1 \\ 0, & i = n+1 \vee j = n+1 \end{cases}$$

   But:

   $$\sum_{r=1}^{n+1} a'_{ir}b'_{rj} = a'_{i(n+1)}b'_{(n+1)i} + \sum_{r=1}^{n} a'_{ir}b'_{rj} \qquad = \sum_{r=1}^{n} a_{ir}b_{rj}$$

   and so:
   $\mathbf{A}'\mathbf{B}'(n+1, n+1) = (\mathbf{AB})'(n+1, n+1)$
   $= \mathbf{AB}$
   $\neq \mathbf{BA}$
   $= (\mathbf{BA})'(n+1, n+1)$
   $= \mathbf{B}'\mathbf{A}'(n+1, n+1)$
   Thus it is seen that:
   $\exists \mathbf{A}', \mathbf{B}' \in \mathscr{M}_{n+1 \times n+1} : \mathbf{A}'\mathbf{B}' \neq \mathbf{B}'\mathbf{A}'$ So $P(k) \Rightarrow P(k+1)$ and the result follows by the Principle of Mathematical Induction
   Therefore:
   $\exists \mathbf{A}, \mathbf{B} \in \mathscr{M}_{\mathscr{R}}(n) : \mathbf{AB} \neq \mathbf{BA}$

2. Prove of distributive properties of matrix Let $\mathbf{A} = [a]_{nn}, \mathbf{B} = [b]_{nn}, \mathbf{C} = [c]_{nn}$ be matrices over a ring $(R, +, \circ)$
   Consider $\mathbf{A}(\mathbf{B} + \mathbf{Cd})$
   Let $\mathbf{R} = [r]_{nn} = \mathbf{B} + \mathbf{C}, \mathbf{S} = [r]_{nn} = \mathbf{A}(\mathbf{B} + \mathbf{C})$
   Let $\mathbf{G} = [g]_{nn} = \mathbf{AB}, \mathbf{H} = [h]_{nn} = \mathbf{AC}$
   Then:
   $s_{ij} = \sum_{k=1}^{n} a_{ik} \circ r_{kj}$
   $r_{kj} = b_{kj} + c_{kj}$
   $\Rightarrow s_{ij} = \sum_{k=1}^{n} a_{ik} \circ (b_{kj} + c_{kj})$
   $= \sum_{k=1}^{n} a_{ik}b_{kj} + \sum_{k=1}^{n} a_{ik}c_{kj}$
   $= g_{ij} + h_{ij}$
   Thus:
   $\mathbf{A}(\mathbf{B} + \mathbf{C}) = (\mathbf{AB}) + (\mathbf{AC})$ A similar construction shows that:
   $(\mathbf{B} + \mathbf{C})\mathbf{A} = (\mathbf{BA}) + (\mathbf{CA})$

3. Prove of associative properties of matrix
   Let $\mathbf{A} = [a]_{nn}, \mathbf{B} = [b]_{nn}, \mathbf{C} = [c]_{nn}$ be matrices
   From inspection of the subscripts, we can see that both $(\mathbf{AB})\mathbf{C}$ and $\mathbf{A}(\mathbf{BC})$ are defined
   Consider$(\mathbf{AB})\mathbf{C}$
   Let $\mathbf{R} = [r]_{nn} = \mathbf{AB}, \mathbf{S} = [s]_{nn} = \mathbf{A}(\mathbf{BC})$
   Then: $s_{ij} = \sum_{k=1}^{n} r_{ik} \circ c_{kj}$
   $r_{ik} = \sum_{l=1}^{n} a_{il} \circ b_{lk}$
   $\Rightarrow s_{ij} = \sum_{k=1}^{n} (\sum_{l=1}^{n} a_{il} \circ b_{lk}) \circ c_{kj}$
   $= \sum_{k=1}^{n} \sum_{l=1}^{n} (a_{il} \circ b_{lk}) \circ c_{kj}$

Now consider $\mathbf{A}(\mathbf{BC})$
Let $\mathbf{R} = [r]_{nn} = \mathbf{BC}, \mathbf{S} = [s]_{nn} = \mathbf{A}(\mathbf{BC})$
Then: $s_{ij} = \sum_{l=1}^{n} a_{il} \circ r_{lj}$
$r_{lj} = \sum_{k=1}^{n} b_{lk} \circ c_{kj}$
$\Rightarrow s_{ij} = \sum_{l=1}^{n} a_{il}(\sum_{k=1}^{n} \circ b_{lk}) \circ c_{kj}$
$= \sum_{l=1}^{n} \sum_{k=1}^{n} a_{il} \circ (b_{lk} \circ c_{kj})$
Using Ring Axiom M1: Associativity of Product:
$s_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} (a_{il} \circ b_{lk}) \circ c_{kj} = \sum_{l=1}^{n} \sum_{k=1}^{n} a_{il} \circ (b_{lk} \circ c_{kj}) = s'_{ij}$
It is concluded that:
$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

2b) Matrix Inversion

$A^{-1} = \frac{1}{|A|} A^*$
$A^* = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}$
A_{xx} is algebraic complement. After calculating:
$A^* = \begin{bmatrix} c & -a & ad - bc \\ -1 & 1 & b - d \\ 0 & 0 & c - a \end{bmatrix}$
$|A| = c + 0 + 0 - (0 + 0 + a) = c - a$
only when $c - a \neq 0$, A is invertible, b can be any value.
so
$A^{-1} = \frac{1}{c-a} \begin{bmatrix} c & -a & ad - bc \\ -1 & 1 & b - d \\ 0 & 0 & c - a \end{bmatrix}$
If we change the matrix to
$A = \begin{bmatrix} 2 & 2 & 3 \\ 0 & 1 & 0 \\ 8 & 3 & 12 \end{bmatrix}$
$|A| = 2 \times 1 \times 2 + 0 + 0 - 3 \times 1 \times 8 - 0 - 0 = 0$
When $|A| = 0$, it is not invertible.

2c) Matrix Pseudoinverse

Moore-Penrose pseudoinverse allows taking the inverse of non-square matrices and Moore-Penrose pseudoinverse must fulfill the followings;

$$AA^+A = A \tag{1}$$

$$A^+AA^+ = A^+ \tag{2}$$

$$(AA^+)^* = AA^+ \tag{3}$$

$$(A^+A)^* = A^+A \tag{4}$$

Right inverse:

$$A^+ = A^*(AA^*)^{-1} \iff AA^+ = I \tag{5}$$

Left inverse:

$$A^{+} = (A^{*}A)^{-1}A^{*} \iff A^{+}A = I \tag{6}$$

2d) Basis Transformation

(1) $T = wv^{-1}$
$= \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1}$
$= \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$
(2) $v = Yw$
$Y = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$
$w = Y^{-1}v$
$= \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} -4 & 3 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 7 \\ -4 \end{bmatrix}$

Task 3: Statistics Refresher

3a) Expectation and Variance

1. In given conditions, the definition of expectation and variance of f are given below.

$$E[f(w)] = \sum f(w)P(w) \qquad , \qquad w \in \Omega$$

$$Var[f(w)] = E[f(w) - E[f(w)^2] \qquad , \qquad w \in \Omega$$

The expectation is a linear operator but the variance is not a linear operator. It can be proved by multiplying f (w) by a constant ”a” and adding another constant ”b”.

$$E[af(w) + b] = aE[f(w)] + E[b]$$

$$Var(a * f(w) + b) = E[((af(w) + b - E[(af(w) + b)]^2] = a^2 Var[f(w)]$$

2. Expectation and the variance of the A is given below.

$$E[A] = 1 \times (1/18) + 2 \times (5/18) + 3 \times (6/18) + 4 \times (3/18) + 5 \times (2/18) + 6 \times (1/18) = 3.17 \tag{7}$$

$$Var[A] = \frac{(1-3.17)^2 \times 1 + (2-3.17)^2 \times 5 + (3-3.17)^2 \times 6 + (4-3.17)^2 \times 3 + (5-3.17)^2 \times 2 + (6-3.17)^2 \times 1}{18} = 1.58 \tag{8}$$

Expectation and the variance of the B is given below.

$$E[B] = 1 \times (6/18) + 2 \times (1/18) + 3 \times (1/18) + 4 \times (4/18) + 5 \times (1/18) + 6 \times (5/18) = 3.44 \tag{9}$$

$$Var[B] = \frac{(1-3.44)^2 \times 6 + (2-3.44)^2 \times 1 + (3-3.44)^2 \times 1 + (4-3.44)^2 \times 4 + (5-3.44)^2 \times 1 + (6-3.44)^2 \times 5}{18} = 4.14 \tag{10}$$

Expectation and the variance of the C is given below.

$$E[C] = 1 \times (3/18) + 2 \times (2/18) + 3 \times (3/18) + 4 \times (3/18) + 5 \times (4/18) + 6 \times (3/18) = 3.67 \tag{11}$$

$$Var[C] = \frac{(1-3.67)^2 \times 3 + (2-3.67)^2 \times 2 + (3-3.67)^2 \times 3 + (4-3.67)^2 \times 3 + (5-3.67)^2 \times 4 + (6-3.67)^2 \times 3}{18} = 2.89 \tag{12}$$

3. The definition of the Kullback-Leibler Divergence is given below.

$$D_{KL}(p||q) = -\sum_{i=1}^{N} p(x_i) ln(\frac{q(x_i)}{p(x_i)}) \tag{13}$$

Tables are formed to show dices are given in the table below.

Values generated for the random vector x						
A	1/18	5/18	6/18	3/18	2/18	1/18
B	6/18	1/18	1/18	4/18	1/18	5/18
C	3/18	2/18	3/18	3/18	4/18	3/18
Uniformly distributed dice	3/18	3/18	3/18	3/18	3/18	3/18

In order to estimate the KL-divergence between each die's distribution and the uniform distribution, we need to use the definition formula of the KL-divergence.

$$\begin{gathered} D_{KL}(p_A||q) = -[\frac{1}{18}.ln(\frac{\frac{3}{18}}{\frac{1}{18}}) + \frac{5}{18}.ln(\frac{\frac{3}{18}}{\frac{5}{18}}) + \frac{6}{18}.ln(\frac{\frac{3}{18}}{\frac{6}{18}}) + \\ \frac{3}{18}.ln(\frac{\frac{3}{18}}{\frac{3}{18}}) + \frac{2}{18}.ln(\frac{\frac{3}{18}}{\frac{2}{18}}) + \frac{1}{18}.ln(\frac{\frac{3}{18}}{\frac{1}{18}})] = 0.205825356 \end{gathered} \tag{14}$$

$$\begin{gathered} D_{KL}(p_B||q) = \frac{6}{18}.ln(\frac{\frac{3}{18}}{\frac{6}{18}}) + \frac{1}{18}.ln(\frac{\frac{3}{18}}{\frac{1}{18}}) + \frac{1}{18}.ln(\frac{\frac{3}{18}}{\frac{1}{18}}) + \\ \frac{4}{18}.ln(\frac{\frac{3}{18}}{\frac{4}{18}}) + \frac{1}{18}.ln(\frac{\frac{3}{18}}{\frac{1}{18}}) + \frac{5}{18}.ln(\frac{\frac{3}{18}}{\frac{5}{18}}) = 0.253772367 \end{gathered} \tag{15}$$

$$\begin{gathered} D_{KL}(p_C||q) = \frac{3}{18}.ln(\frac{\frac{3}{18}}{\frac{3}{18}}) + \frac{2}{18}.ln(\frac{\frac{3}{18}}{\frac{2}{18}}) + \frac{3}{18}.ln(\frac{\frac{3}{18}}{\frac{3}{18}}) + \\ \frac{3}{18}.ln(\frac{\frac{3}{18}}{\frac{3}{18}}) + \frac{4}{18}.ln(\frac{\frac{3}{18}}{\frac{4}{18}}) + \frac{3}{18}.ln(\frac{\frac{3}{18}}{\frac{3}{18}}) = 0.018877671 \end{gathered} \tag{16}$$

According to the KL-Divergence calculation, C is the closest one to the uniform distribution.

3b) It is a Cold World

1. p(back): Probability that a person has back pain
   p(cold): Probability that a person has a cold

2. The domain of each random variable can be defined as ”true” or ”false”.

3. Statement A: $p(back = true|cold = true) = 25/100$
   Statement B: $p(cold = true) = 4/100$
   Statement C: $p(back = true|cold = false) = 10/100$

4. Bayes rules can be used to find people who have backpain and cold at the same time.

$$\begin{aligned} P(cold = true|back = true) &= P(back = true|cold = true) * P(cold = true)/P(back = true) \\ &= [(0.25 * 0.04)/((0.25 * 0.04) + (0.975 * 0.1))] = 0.093 \end{aligned} \tag{17}$$

3c) Cure the Virus

1.

$$\vec{s_t} = [m, \widetilde{m}]^T = [0.026, 0.974]^T \tag{18}$$

$$\vec{s_{t+1}} = [m, \widetilde{m}]^T = [0.42, 0.58]^T \tag{19}$$

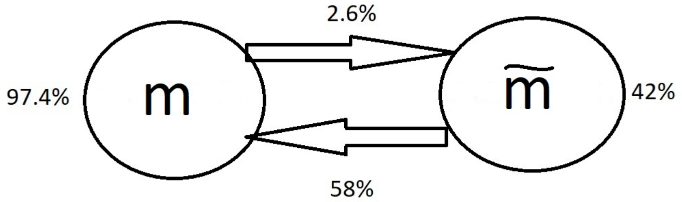

Figure 3: Markov Chain State Diagram

2. Cure the Virus
   (1)
   $mutated = 42\% + 2.6\% = 44.6\%$
   $unmutated = 58\% - 2.6\% = 55.4\%$

State	m	$\tilde{m}$
$\tilde{m}$	44.6%	55.4%

   (2)

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

# State space
states = ["Mutated","Unmutated"]

# Possible events
transitionName = [["UM","UU"],["MU","MM"]]

# Probilistic matrix
```

```
transitionMatrix = [0.446,0.554]


def activity_forecast(days):
    # Choose initial state
    activityToday = "Unmutated"
    print("Start state: " + activityToday)
    # Initial state list
    activityList = [activityToday]
    prob_list = []
    i = 0
    # Calculate the probability of activityList
    prob = 1
    while i != days:
        if activityToday == "Unmutated":
            change = np.random.choice(transitionName[0],replace=True,p=transitionMatrix)
            if change == "UM":
                prob = prob * 0.446
                activityToday = "Mutated"
                activityList.append("Mutated")
                prob_list.append(prob)
                pass
            elif change == "UU":
                prob = prob * 0.554
                activityList.append("Unmutated")
                prob_list.append(prob)
        elif activityToday == "Mutated":
            change = np.random.choice(transitionName[1], replace=True, p=transitionMatrix)
            if change == "MU":
                prob = prob * 0.446
                activityToday = "Unmutated"
                activityList.append("Unmutated")
                prob_list.append(prob)
                pass
            elif change == "MM":
                prob = prob * 0.554
                activityList.append("Mutated")
                prob_list.append(prob)
        i += 1
    print("Possible states: " + str(activityList))
    print("End state after "+ str(days) + " days: " + activityToday)
    print("Probability of the possible sequence of states: " + str(prob))

    x = np.arange(0, 18, 1)
    prob_list = np.array(prob_list)
    plt.plot(x,prob_list)
    plt.show()

# predict states after 18 days
activity_forecast(18)
```

The result is:

```
Start state: Unmutated
Possible states: ['Unmutated', 'Unmutated', 'Mutated', 'Unmutated', 'Unmutated',
    'Mutated', 'Mutated', 'Unmutated', 'Unmutated', 'Unmutated', 'Unmutated', 'Mutated',
    'Unmutated', 'Mutated', 'Mutated', 'Mutated', 'Mutated', 'Unmutated', 'Mutated']
End state after 18 days: Mutated
```

```
Probability of the possible sequence of states: 3.432429382297346e-06
```

Plot:

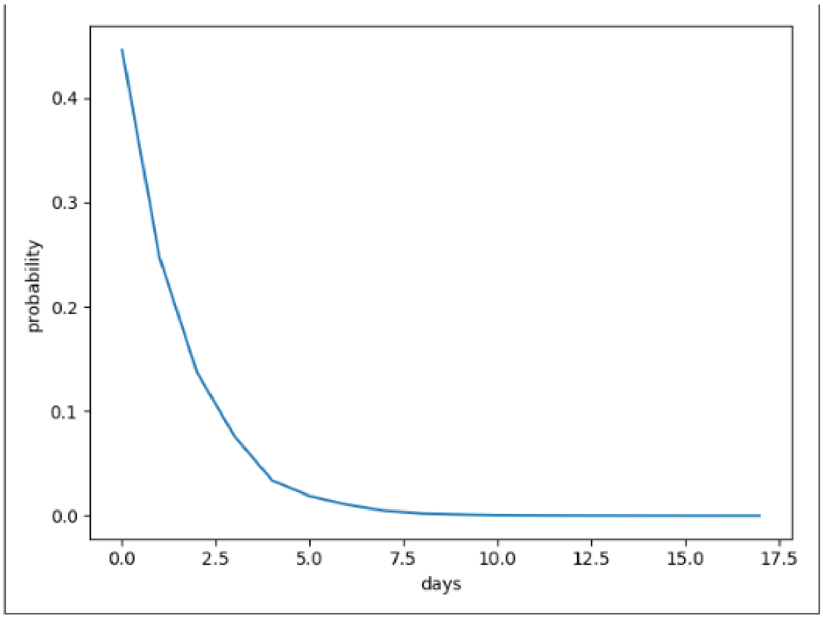

Figure 4: Plot Result

3. We continue to run the code for 40 generations. Results were not much different from 20 days. As can be seen in the figure above, the probability does not change much after the 10th day. The fixed probability is about 0.04. The mathematical expression of stable probability is found below.

$$x + y = 1$$

$$0.58x - 0.026y = 0$$

Finally, the result of the equation solution and the result of the simulation are the same.

Task 4: Information Theory

4a) Entropy

1. Entropy of a source S is defined below.

$$H(S) = -\sum_{n=1}^{N} p_n.ld(p_n) \tag{20}$$

Since we have 4 different symbols (N=4), bits of information can be transmitted on average per symbol under this distribution. The entropy of the source is calculated below.

$$\begin{aligned} H(S) &= -[0.04log_2(0.04) + 0.22log_2(0.22) + 0.67log_2(0.67) + 0.07log_2(0.07)] \\ &= 0.18575 + 0.4806 + 0.3871 + 0.26855 = 1.322[Bits/Symbol] \end{aligned} \tag{21}$$

2. Uniform distribution over the symbols is required to achieve the maximum number of bits per symbol that can be transmitted. Uniform distribution calculation is given below.

$$H(S)_{uniform} = log_2(N) = 2bits \tag{22}$$

Task 5: Bayesian Decision Theory

5a) Optimal Boundary

Bayes Decision Theory describes the probability of an event, based on prior knowledge of conditions that might be related to the event.
Its Goal is to minimize the misclassification rate.
Bayes optimal classification based on probability distributions $p(x|C_k)p(C_k)$.Posterior should be calculated in order to find the optimal decision boundary.
We compare the result among the magnitude of $p(x|C_1)$ and $p(x|C_2)$. If they are equal we decide for class C1 over C2. The higher one should be chosen. For example, when $p(x|C_1)$ is bigger, then we choose C_1

$$\frac{p(x|C_1)}{p(x|C_2)} > \frac{p(C_2)}{p(C_1)} \tag{23}$$

5b) Decision Boundaries

$p(x|C_1) = \frac{p(xC_1)}{p(C_1)} = \frac{p(C_1|x)p(x)}{p(C_1)}$
$p(x|C_2) = \frac{p(xC_2)}{p(C_2)} = \frac{p(C_2|x)p(x)}{p(C_2)}$
Because $p(C_1) = p(C_2)$ and $\sigma_1 = \sigma_2$
According to Gaussian Distribution so $\mu_1 = \mu_2$
So $p(x|C_1) = p(x|C_2)$

$$p(x|C_1) = \frac{1}{\sigma\sqrt{(2\pi)}}e^{\frac{-(x-\mu_1)^2}{2\sigma^2}} \; p(x|C_2) = \frac{1}{\sigma\sqrt{(2\pi)}}e^{\frac{-(x-\mu_2)^2}{2\sigma^2}} \tag{24}$$

The decision boundary is derived below.

$$e^{\frac{-(x-\mu_1)^2}{2\sigma^2}} = e^{\frac{-(x-\mu_2)^2}{2\sigma^2}}$$

$$(x-\mu_1)^2 = (x-\mu_2)^2$$

$$\mathrm{x}^2 - 2x\mu_1 + \mu_1^2 = x^2 - 2x\mu_2 + \mu_2^2$$

$$(\mu_1+\mu_2)(\mu_1-\mu_2) = 2x(\mu_1-\mu_2)$$

$$\mathrm{x}^* = x = \frac{\mu_1+\mu_2}{2} \tag{25}$$

5c) Different Misclassification Costs

No cost for correctly classifying samples means that $\lambda_{11} = \lambda_{22} = 0$. Since the class C1 is four times more expensive than the opposite, the decision boundary change as $\lambda_{12} = 1$ and $\lambda_{21} = 0.25$. The risks are given in below from the loss function.

$$R(\sigma_1 \mid x) = \lambda_{12} p(c_2 \mid x)$$

$$R(\sigma_2 \mid x) = \lambda_{21} p(c_1 \mid x) \tag{26}$$

$\sigma_1 = \sigma_2$ and $p(C1) = p(C2)$;

$$R(\sigma_1 \mid x) = R(\sigma_2 \mid x)$$

$$0.25 \text{ .e}^{\frac{-(x-\mu_1)^2}{2\sigma^2}} = e^{\frac{-(x-\mu_2)^2}{2\sigma^2}}$$

$$\ln(0.25 \text{ .e}^{\frac{-(x-\mu_1)^2}{2\sigma^2}}) = ln(e^{\frac{-(x-\mu_2)^2}{2\sigma^{2)}}})$$

(27)

$\mu_1 = 2\mu_2$;

$$\text{x}^* = \frac{\sigma^2 * ln(4)}{\mu_2} + \frac{3\mu_2}{2} \tag{28}$$