Homework 3

Sarthak Maharana

EE 559

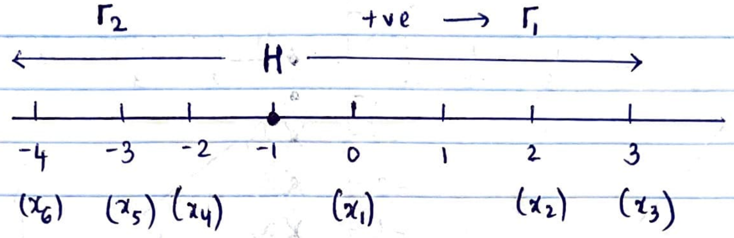
1. No. of features $(D) = 1$

Classes $(C) = 2$.

Data points : $x_1 = 0,\ x_2 = 2,\ x_3 = 3 \in S_1$

$x_4 = -2,\ x_5 = -3,\ x_6 = -4 \in S_2$

(a) (i)



[Non-augmented feature space]

$H = -1$. //

(ii) Given condition, $\|w\| = 1$.

Let the point on the decision boundary be x.

So, $g(x) = 0$

$w_0 + w_1 x = 0$

$x = -\frac{w_0}{w_1}$. if $x = -1$ or $w_0 = w_1$

Since, there's only one feature, $w = [w_0 \ w_1]$.

$\|\underline{w}\| = \sqrt{w_0^2 + w_1^2} = \sqrt{2}\, w_0$.

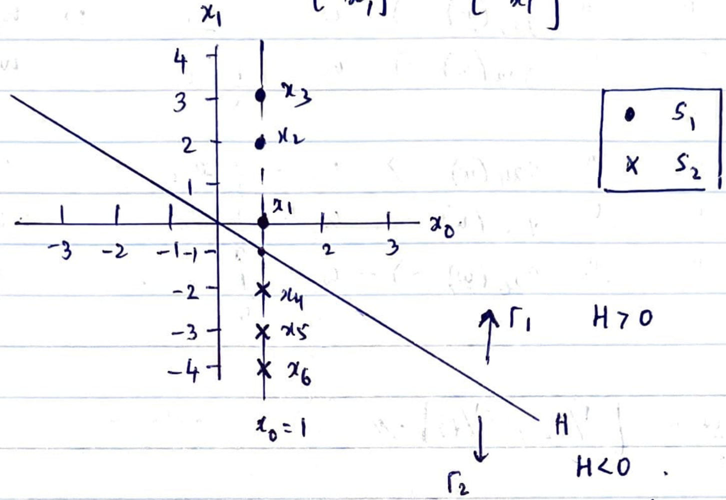
or $\underline{w} = [1/\sqrt{2},\ 1/\sqrt{2}]$.

Substituting it back, $x = -1$. //

$\therefore$ $H = g(x) = 0$ at $x = -1$.

$\underline{w} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$ //

(b) In augmented space, $x = \begin{bmatrix} 1 \\ x_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$.



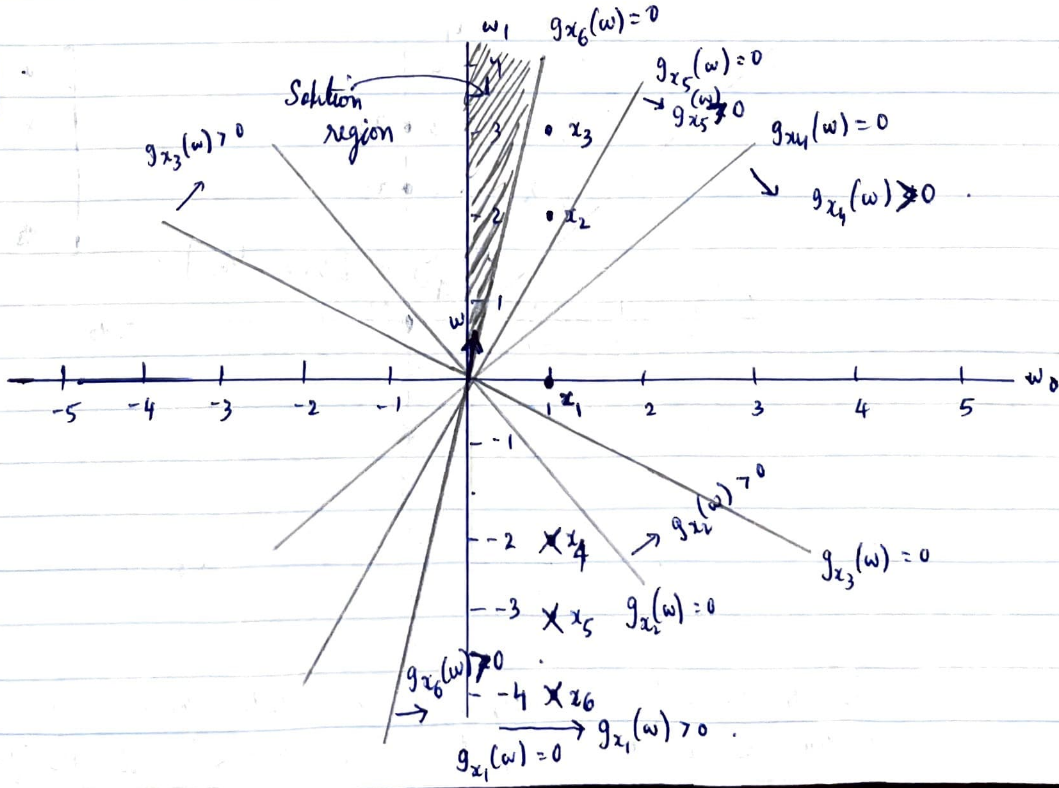
The decision boundary in the augmented space has to pass through the origin. The positive & negative regions have been marked.

(c) $D = 1$ and $w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}^T = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}^T$

and

(d) From the previous plot, $x_1 = (1,0)$, $x_2 = (1,2)$, $x_3 = (1,3)$
$x_4 = (1,-2)$, $x_5 = (1,-3)$, $x_6 = (1,-4)$.

[Weight space].



Equation of $g_{x_1}(\underline{w}) \Rightarrow \quad w_0 = 0$

" " $g_{x_2}(\underline{w}) \Rightarrow \quad w_1 = -\frac{w_0}{2}$ or $w_0 = -2w_1$

" " $g_{x_3}(\underline{w}) \Rightarrow \quad w_1 = -\frac{1}{3}w_0 \qquad w_0 = -3w_1$

" " $g_{x_4}(\underline{w}) \Rightarrow \quad w_1 = \frac{1}{2}w_0 \qquad w_0 = 2w_1$

" " $g_{x_5}(\underline{w}) \Rightarrow \quad w_1 = \frac{1}{3}w_0 \qquad w_0 = 3w_1$

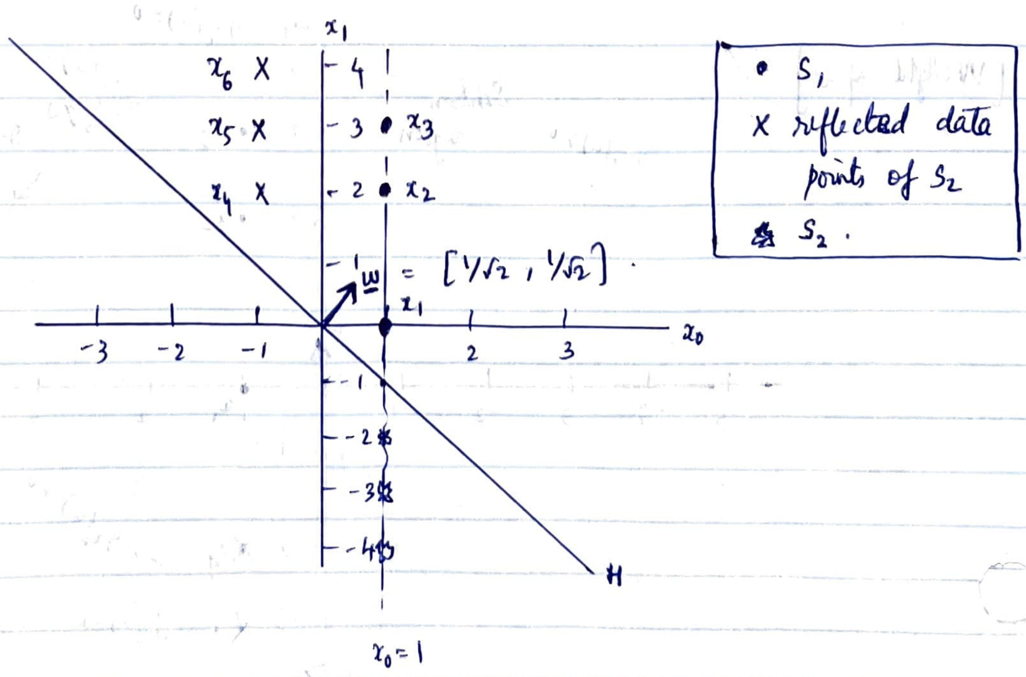
" " $g_{x_6}(\underline{w}) \Rightarrow \quad w_1 = \frac{1}{4}w_0 \qquad w_0 = 4w_1$

$\underline{w} = [1/\sqrt{2}, 1/\sqrt{2}]$ .//

Yes, it is in the solution region. It is marked as 'w' in the solution region.

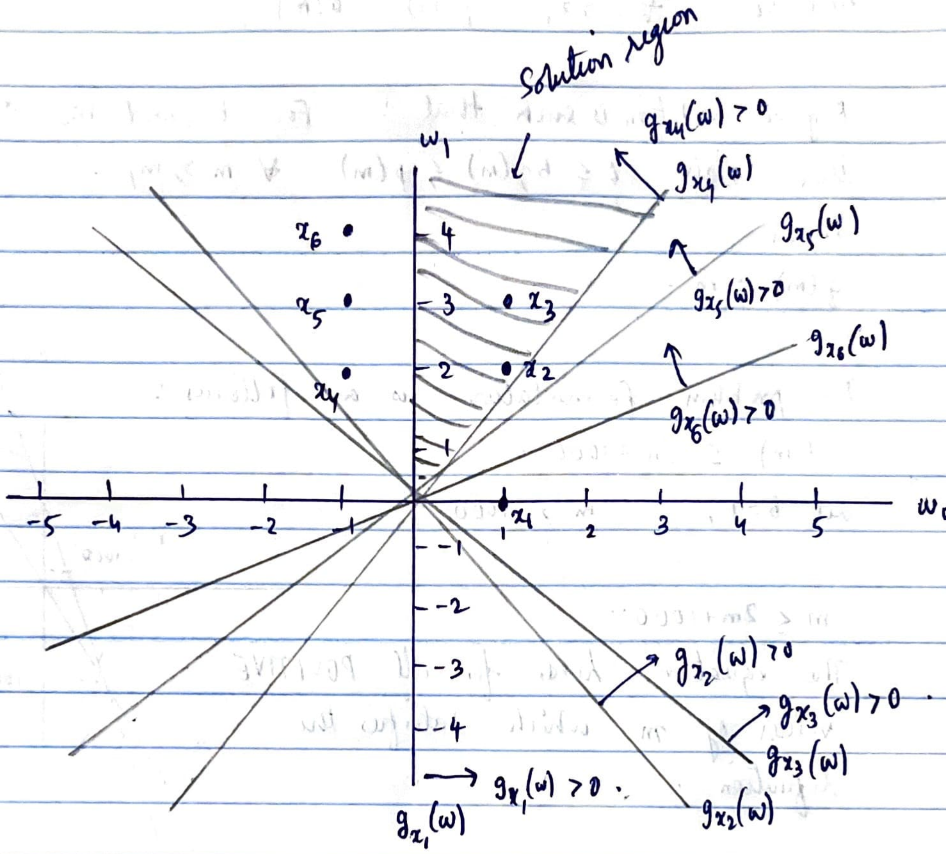
(e) The reflected data points are defined as follows, in the augmented feature space

$$z_n x_n = \begin{cases} x_n & x_n \in S_1 \\ -x_n & x_n \in S_2 \end{cases}$$



$\boxed{\text{No}}$, H doesn't classify all the data points. The weight vector is marked as 'w' in the plot. It is normal to the feature space.

(f)



8. Given, $p(m) = 2m + 1000$. $m \in \mathbb{Z}^+$

(a) The big-Oh notations suggest that there exists +ve constants 'a' and m_0 such that $0 \le p(m) \le a\,q(m)$ $\forall\ m \ge m_0$.

Given claim: $p(m) = O(m)$

So, $q(m) = m$

or $2m + 1000 \le am$.

If $a = 1$,

$m \le -1000$ (m_0) but m_0 has to be positive. So, $a = 1$ does not make $p(m) = O(m)$.

$a = 2$, the equation does not satisfy.

$a = 3$, $m \ge 1000$ (m_0).

or, $p(m) = O(m)$ for $m_0 = 1000$ and $a = 3$. //

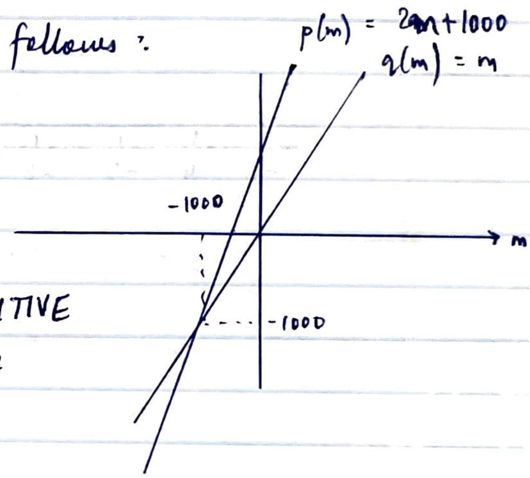
And so, $\forall a > 2, \quad p(m) = O(m)$ //

(b) Big Ω notation is such that : For b and m_0 as positive constants, there exists $0 \le b\,q(m) \le p(m) \quad \forall\, m \geq m_1$.

Here,

$q(m) = m$.

The problem formulation is as follows :



$3(m) \le 2m + 1000$.

Let $b = 1$, $\quad m \geq -1000$

$m \le 2m + 1000$ ..

This equation holds for all POSITIVE values of m which satisfies the definition .

Hence, $p(m) = \Omega(m) \;\forall\; m > 0$. and $b = 1$. The smallest positive integer m_1 that satisfies this is $\underline{1}$. //

(c) $p(m) = \Theta(q(m))$ if there exists +ve constants $a, b,$ and m_2 such that $0 \le b\,q(m) \le p(m) \le a\,q(m) \quad \forall\, m \geq m_2$.

Clearly, this is a combination of $O(m)$ and $\Omega(m)$. And this satisfies the governing inequalities . So, $p(m) = \Theta(m)$. //

3. Given, $p(m) = p_1(m) + p_2(m) + p_3(m)$.

(a) and $p_k(m) = O(q_k(m)) \qquad k = 1, 2, 3$.

To prove, $p(m) = O(q_1(m) + q_2(m) + q_3(m))$.

Condition for $p(m) = O(q(m))$: $0 \le p(m) \le a\,q(m) \quad a > 0, \;\forall\, m \geq m_0$.

Let,

$0 \le p_K(m) \le a\, q_K(m) \quad \forall\ K = 1,2,3$.

Summing up the three equations,

$0 \le p_1(m) + p_2(m) + p_3(m) \le a(q_1(m) + q_2(m) + q_3(m))$.

$0 \le p(m) = O(q_1(m) + q_2(m) + q_3(m))$. //

Proved . //

(b) Let $p_K(m) = \Omega(q_K(m)) \quad K = 1,2,3$.

Condition for $p(m) = \Omega(q(m))$

is $0 \le b\,q(m) \le p(m) \quad b > 0, \ \forall\ m \ge m_1$.

Let $0 \le b\,q_K(m) \le p_K(m) \quad \forall\ K = 1,2,3$.

Summing up the three equations,

$0 \le b(q_1(m) + q_2(m) + q_3(m)) \le p(m)$.

or $p(m) = \Omega(q_1(m) + q_2(m) + q_3(m))$ //

Proved . //

4. Given, $p(m) = m^2 \log_2 m + \frac{2m^3}{\log_2 m} + (\log_2 m)^3$.

(a)

Let $p_1(m) = m^2 \log_2 m$

$p_2(m) = \frac{2m^3}{\log_2 m}$

$p_3(m) = (\log_2 m)^3$

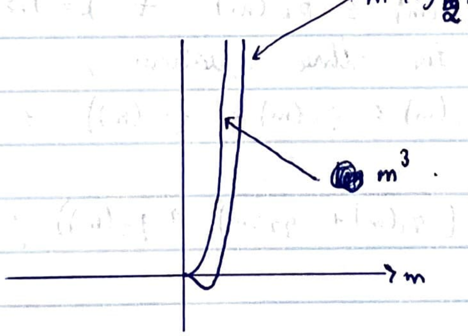
$q(m) = m^3$.

$p(m) = O(q(m)) \longrightarrow 0 \le p(m) \le a\,q(m)$, $a > 0$, $\forall m > m_0$

Let $a = 1$, without any loss of generality :

$0 \le p_1(m) \le a\,q(m) = q(m)$.

$p_1(m) = m^2 \log_2 m \rightarrow$



$p_1(m) = m^2 \log_2 m$;

$\boxed{m_0 > 0}$

or, $m_0 \ge 1$.

Clearly, $\forall m > 0$, $m^2 \log_2 m = O(m^3)$ //

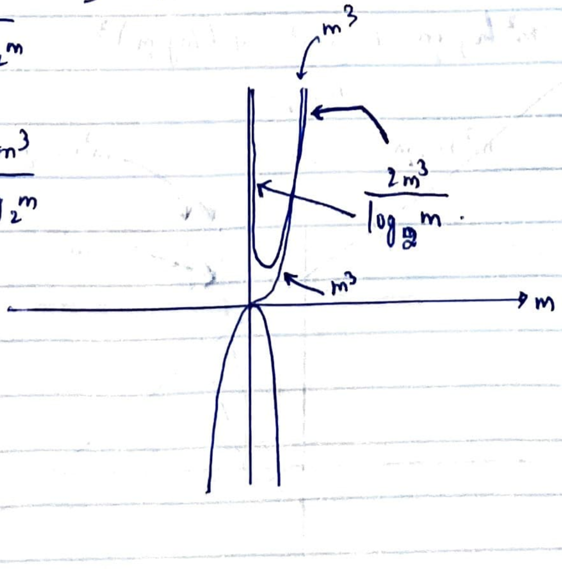
$p_2(m) = \frac{2m^3}{\log_2 m}$. $\rightarrow$ $0 \le \frac{2m^3}{\log_2 m} \le am^3$.

$2 \le \log_2 m$

or $m \ge 4$.

i.e $\boxed{m_0 \ge 4}$.

$p_2(m) = \frac{2m^3}{\log_2 m}$



$\therefore$ $\forall m \ge 4$, $\frac{2m^3}{\log_2 m} = O(m^3)$

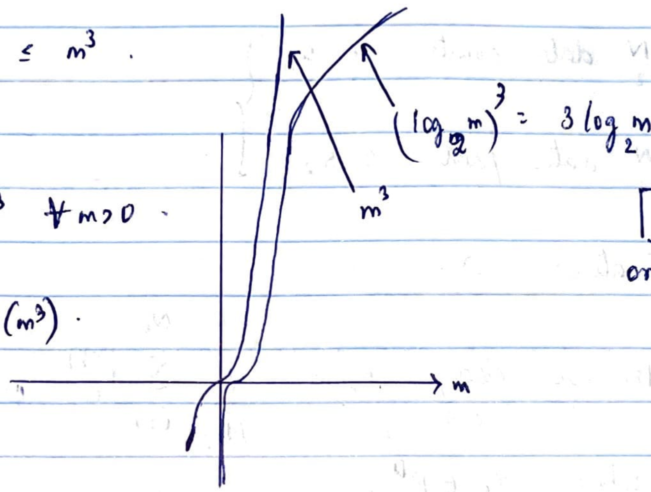
$p_3(m) = (\log_2 m)^3$

$p_3(m) = (\log_2 m)^3 \rightarrow \quad 0 \leq (\log_2 m)^3 \leq m^3$.

Clearly,

$(\log_2 m)^3 \leq m^3 \quad \forall m > 0$.

$(\log_2 m)^3 = O(m^3)$.



$\boxed{m_0 > 0}$

or $m_0 \geq 1$

Since, $p_1(m) = O(m^3) = p_2(m) = p_3(m)$

By the property proved in 3(a), $p(m) = p_1(m) + p_2(m) + p_3(m)$
$= O(q_1(m) + q_2(m) + q_3(m))$.

and so, $p(m) = m^2 \log_2 m + \dfrac{2m^3}{\log_2 m} + (\log_2 m)^3 = O(m^3)$ //.

Yes, $p(m) = O(m^3)$ //

(b) No, $p(m) \neq \Omega(m^3)$. //

For example, in $p_1(m) \rightarrow \quad q(m) = m^3 > p_1(m) \quad \forall m > 0$.
The definition of Ω suggests: There must exist positive constants b & m_1 such that $0 \leq b q(m) \leq p(m)$, $\forall m \geq m_1$.

This fails in the case of $p_1(m)$ and $p_3(m)$. There is no lower bound for such a functions.

$p(m) \neq \Omega(m^3)$ //.

5. Given, no. of classes $= C = 2$.

$\frac{N}{2}$ data points $\in S_1$

$\frac{N}{2}$ data points $\in S_2$

Features $= D$.

(a) For each class, $\mu_k = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i^{(k)}$ for $k = 1,2$.

where $x_i \in \mathbb{R}^D$.

Here, $N_c = N/2$.

Time complexity for each class (i.e. per mean vector):

Additions : $(\frac{N}{2}-1) D$.

Multiplications : D.

Divisions : 1.

Total time complexity : $(\frac{N}{2}-1) DT_A + DT_M + 1 \cdot T_D = T$. //

where T_A = Time for addition

T_M = Time for multiplications

T_D = Time for division.

$$T = (\frac{N}{2}-1) D T_A + DT_M + T_D.$$

$$\approx \frac{N}{2} D T_A + DT_M = O(\frac{ND}{2}).$$

$\therefore$ $\boxed{T \approx O(\frac{ND}{2})}$ // for each mean-vector.

Reason : For each x_i, there are D features. Total additions are D and since $\frac{N}{2}$ data points (out of the ~~data phase~~ training data) are of a certain class, total additions $= (\frac{N}{2}-1) D$. For each class, there are D multiplications and only one division (by $\frac{N}{2}$).

The algorithm to calculate the mean vector for each class :

Initialize x_sum = 0 .

for each i in $1 \ldots D$:

for each j in $1 \ldots N/2$:

load $x_{ij}^{(k)}$ $\quad$ $k \rightarrow$ class number .

$(x\_sum)_j = (x\_sum)_j + x_{ij}^{(k)}$

$\mu = \left(\frac{1}{\frac{N}{2}}\right)(x\_sum)_j$ $\quad$ $\mu, x_{ij} \rightarrow$ scalar ,

output μ . $\quad$ x_sum $\rightarrow$ scalar .

In the above, we make use of scalars and do not use any other data structure that will have a fixed size . Hence, space complexity $\approx O(1)$. Even though we iterate through two loops, data is not being stored .

$\boxed{\text{Space complexity} \rightarrow O(1)}$ //

(b) M data points have to be classified .

Let the evaluation metric be ~~Manhattan~~ Euclidean distance .

✓ Each data point has D features .

✓ For each data point, two distances have to be computed since there are 2 classes . $\longrightarrow$ $\|x_i - \mu_1\|_2$ & $\|x_i - \mu_2\|_2$

Distance $\longrightarrow$ $\|x_{11} - \mu_1\|_2 + \|x_{12} - \mu_1\|_2 + \cdots + \|x_{1D} - \mu_1\|_2$.

Total additions $\rightarrow M(D-1)$. $\quad$ Total multiplications $\rightarrow MD$.

Total subtractions $\rightarrow MD$.

Total comparisons $\rightarrow 2$.

Time complexity $= M(D-1)T_A + MDT_S + 2$ comparisons $+ MDT_M$

$\approx \boxed{O(MD)}$ //

Algorithm (classification for each class)

```
for each i in 1 . . . . M
    load x_i
    for each k in 1,2
        compute ||x_i - μ_k||_2
        class = k
        if ||x_i - μ_{k+1}||_2 < ||x_i - μ_k||_2 → update class.
output class.
```

In the above, we make use of scalars and not data structures of a certain size. "class" is either used to store 1 or 2 depending on the if condition. Hence, space complexity = O(1) i.e it happens in linear space. Finding the minimum is O(1).

$\therefore$ $\boxed{\text{Space complexity} = O(1)}$

(c) No. of classes = C

No. of data points in each class = N/C

$$\mu_k = \frac{1}{N_c}\sum_{i=1}^{N_c} x_i^{(k)} \qquad k = 1, 2, .. C$$

$x_i^{(k)} \rightarrow$ i^{th} data point of class C.

$N_c \rightarrow$ # data of class C.

Time complexity for each class:

Additions = $\left(\frac{N}{C} - 1\right) D$.

Multiplications = D.

Divisions = 1

Total time complexity = $\left(\frac{N}{C} - 1\right) D\, T_A + D T_M + T_D$

or $T \approx O\left(\frac{ND}{C}\right)$.

$\therefore$ Time complexity = $\boxed{O\left(\frac{ND}{C}\right)}$ //

The algorithm $\rightarrow$ to compute mean vector for each class

Init $x\text{-sum} = 0$

for each i in $1 : \dots D$

for each j in $1 \dots N/C$

load $x_{ij}^{(k)}$

$(x\text{-sum})_j = (x\text{-sum})_j + x_{ij}^{(k)}$ of class K

$\mu = \frac{1}{\frac{N}{C}} (x\text{-sum})_j$ $K = 1, 2, \dots, C$.

Output μ

Space complexity $= \boxed{O(1)}$ //

Since we're making use of scalars, and no data structures of a certain size, space complexity $= O(1)$ i.e happens in linear space.

(d) # of points to be classified $= M$

Let the evaluation metric be ~~[illegible]~~ Euclidean distance.

Each data point has D features.

For each data point, C distances have to be computed

i.e $\|x_i - \mu_K\|_2^2$ $\forall K = 1, 2, \dots, C$.

Distance for one class: $\|x_1 - \mu_1\|_2 + \|x_2 - \mu_2\|_2 \dots + \|x_{10} - \mu_1\|_2$

Total additions : $M(D-1)C$.

Total subtractions : MDC. Total multiplications : MD

Total comparisons : $C \approx O(C)$.

Time complexity $= M(D-1)C\,T_A + C\,T_C + MDC\,T_S + MD\,T_m$

$\approx \boxed{O(MDC)}$ //

Algorithm (classification of each class):

Initialise d_sum = 0 , mean_classes = []

for each k in 1, ..., C

  for each i in 1, ..., M

    for each j in 1, ..., D

      load x_{ij}, μ_K

      d_sum = d_sum + $\| x_{ij} - \mu_K \|_2$

  → mean_class.append [d_sum] .   # to store distances of each class

output min (mean_classes)

The minimum distance can be found in O(1) from mean_classes but the size of mean_classes is C and so, space complexity $\boxed{= O(C)}$ in my algorithm.