HW5

Bhishan Poudel

142/150

HW Assignment 5 (Due by 10:30am on Nov 2)

1 Theory (110 points)

1. **[Properties of Linear Discriminants**, 20 points]
   We have proven in class that the distance between origin and the decision hyperplane $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$ is equal with $-w_0/\|\mathbf{w}\|$. Prove that the margin between a point $\mathbf{x}$ and the same decision hyperplane is equal with $h(\mathbf{x})/\|\mathbf{w}\|$.

2. **[Bonus**, 20 points]
   Prove the two properties above for the general n-dimensional case.

3. **[Fisher Criterion and Least Squares**, 30 points]
   Show that the Fisher criterion can be written in the vectorized form shown below:
   $$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}$$

4. **[Fisher Criterion (*)**, 20 points]
   Reference the PRML Chapter 4 material available on Blackboard under Content.
   Using the definitions of the between-class and within-class covariance matrices given by (4.27) and (4.28), respectively, together with (4.34) and (4.36) and the choice of target values described in Section 4.1.5, show that the expression (4.33) that minimizes the sum-of-squares error function can be written in the form (4.37).

5. **[Perceptrons**, 40 points]
   Consider a training set that contains the following 8 examples:

$\mathbf{x}$	x_1	x_2	x_3	$t(x)$
$\mathbf{x}^{(1)}$	0	0	0	+1
$\mathbf{x}^{(2)}$	0	1	0	+1
$\mathbf{x}^{(3)}$	1.5	0	-1.5	+1
$\mathbf{x}^{(4)}$	1.5	1	-1.5	+1
$\mathbf{x}^{(5)}$	1.5	0	0	−1
$\mathbf{x}^{(6)}$	1.5	1	0	−1
$\mathbf{x}^{(7)}$	0	0	-1.5	−1
$\mathbf{x}^{(8)}$	0	1	-1.5	−1

   (a) Prove that the perceptron algorithm does not converge on this dataset. Do not forget to include the bias.

   (b) Consider a kernel perceptron that uses a polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^d$. What is the smallest degree d for which the kernel perceptron would converge on this dataset?

6. **[Perceptrons**, 10 points]
   A kernel perceptron for binary classification is run for a number of epochs E on a training dataset containing N examples, resulting in the dual parameters α_1, α_2, ..., α_N. What is the total number of mistakes that are made during training?

7. [**Matrix Computations**, 10 points]
Let $U \in R_{k\times m}$ and $X \in R_{n\times m}$. Let u_i and x_i be the i-th columns of U and X, respectively, for $1 \leq i \leq m$. Prove that $UX^T = \sum_{i=1}^{m} u_i x_i^T$.

2 Submission

Turn in a hard copy of your homework report at the beginning of class on the due date. On this theory assignment, **clear and complete explanations and proofs of your results are as important as getting the right answer**.

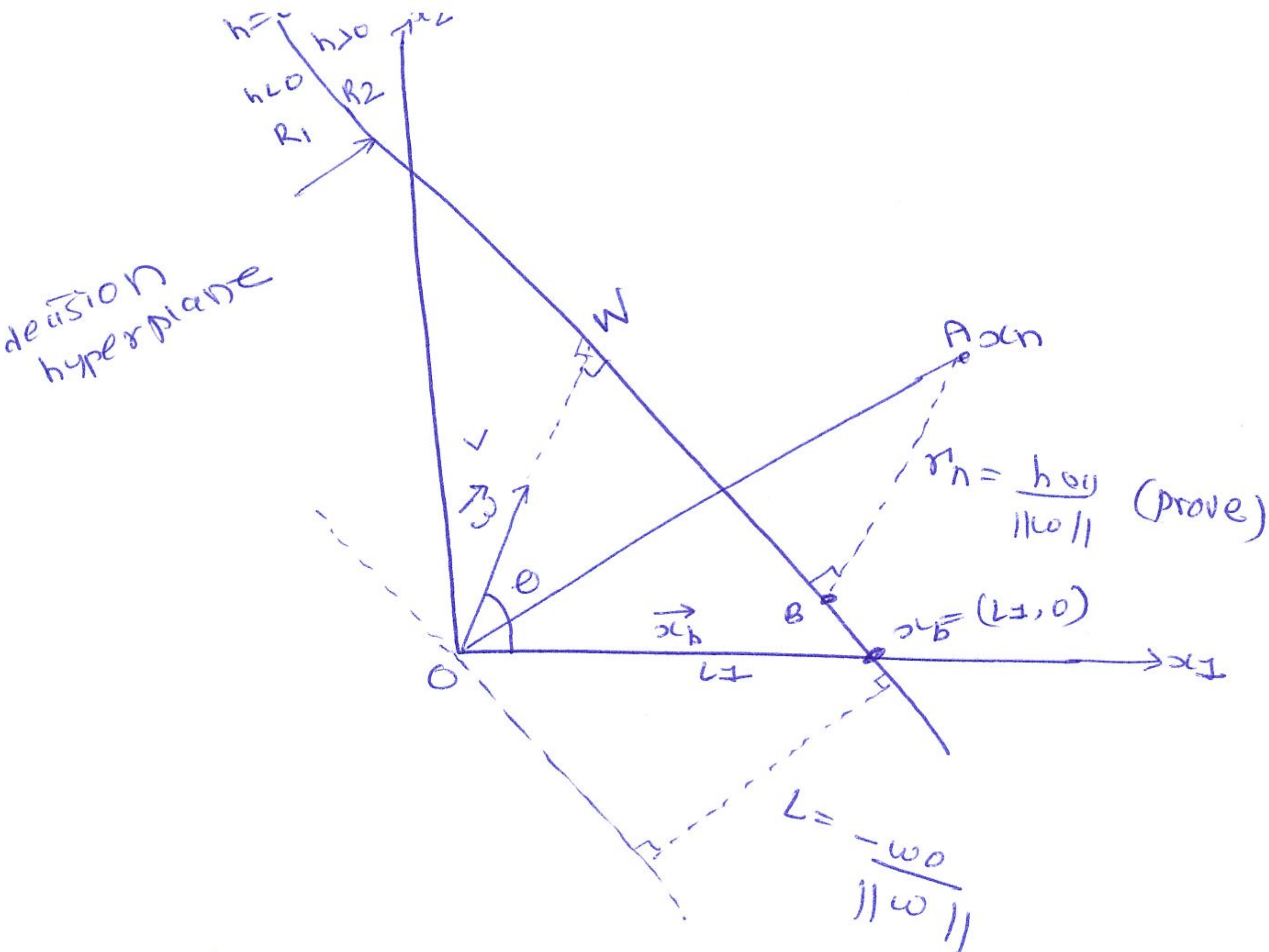
HW 05

Bhishan Poudel

<u>Qu 1</u>

prove:

dist betn <u>data point</u> & <u>decision hyp.</u> $= \dfrac{h(x)}{||w||}$



a) prove $L_1 = \frac{-w_0}{\|w\|}$

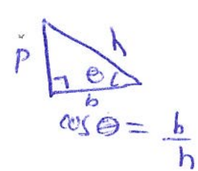
Here, $\vec{w} = [w_1, w_2]$

$\vec{x_b} = [x_1, x_2] = [L_1, 0]$

$\vec{w} \cdot \vec{x_b} + w_0 = 0 = w_1 x_1 + w_2 x_2 + w_0$

$0 = w_1 L_1 + w_2 \cdot 0 + w_0$

$$\boxed{L_1 = -\frac{w_0}{w_1}} \text{ — (1)}$$



Again, $\boxed{\cos\theta = \frac{\text{base}}{\text{hypotenuse}} = \frac{L}{L_1}}$

$\cos\theta = \frac{L}{-w_0/w_1}$

$$\boxed{L = -\frac{w_0}{w_1}\cos\theta} \text{ — (2)}$$

$$L = \frac{-w_0}{w_1} \cdot \frac{\vec{w} \cdot \vec{x}_b}{\|w\| \, \|x\|} \qquad \because \vec{a} \cdot \vec{b} = \|\vec{a}\| \, \|\vec{b}\| \cos\theta$$

$$= \frac{-w_0}{w_1} \cdot \frac{(w_1, w_2) \cdot (1, 0)}{\|w\| \, 1}$$

$$= \frac{-w_0}{w_1} \cdot \frac{w_1 \cdot 1}{\|w\| \, 1}$$

$$\boxed{L = \frac{-w_0}{\|w\|}} \quad \text{Ans}$$

Again,

To find the co-ordinate of point B,

$$\vec{OB} = \vec{OA} - \vec{AB}$$

$$\vec{B} = \vec{x}_n - r_n \hat{w} \longrightarrow \hat{w} \text{ is unit vector perpendicular to decision hyperplane}$$

$$\boxed{B = x_n - r_n \frac{w}{\|w\|}}$$

But, point B lies in decision boundary, so,

$$\vec{w} \cdot \vec{B} + w_0 = 0$$

$$w^T \left(x_n - r_n \frac{w}{\|w\|} \right) + w_0 = 0$$

$$\text{or, } w^T x_n - r_n \frac{w^T w}{\|w\|} + w_0 = 0 \qquad \because w^T w = \|w\|^2$$

$$\text{or, } w^T x_n - r_n \|w\| + w_0 = 0$$

or $w^T x_n + w_0 = r_n \|w\|$

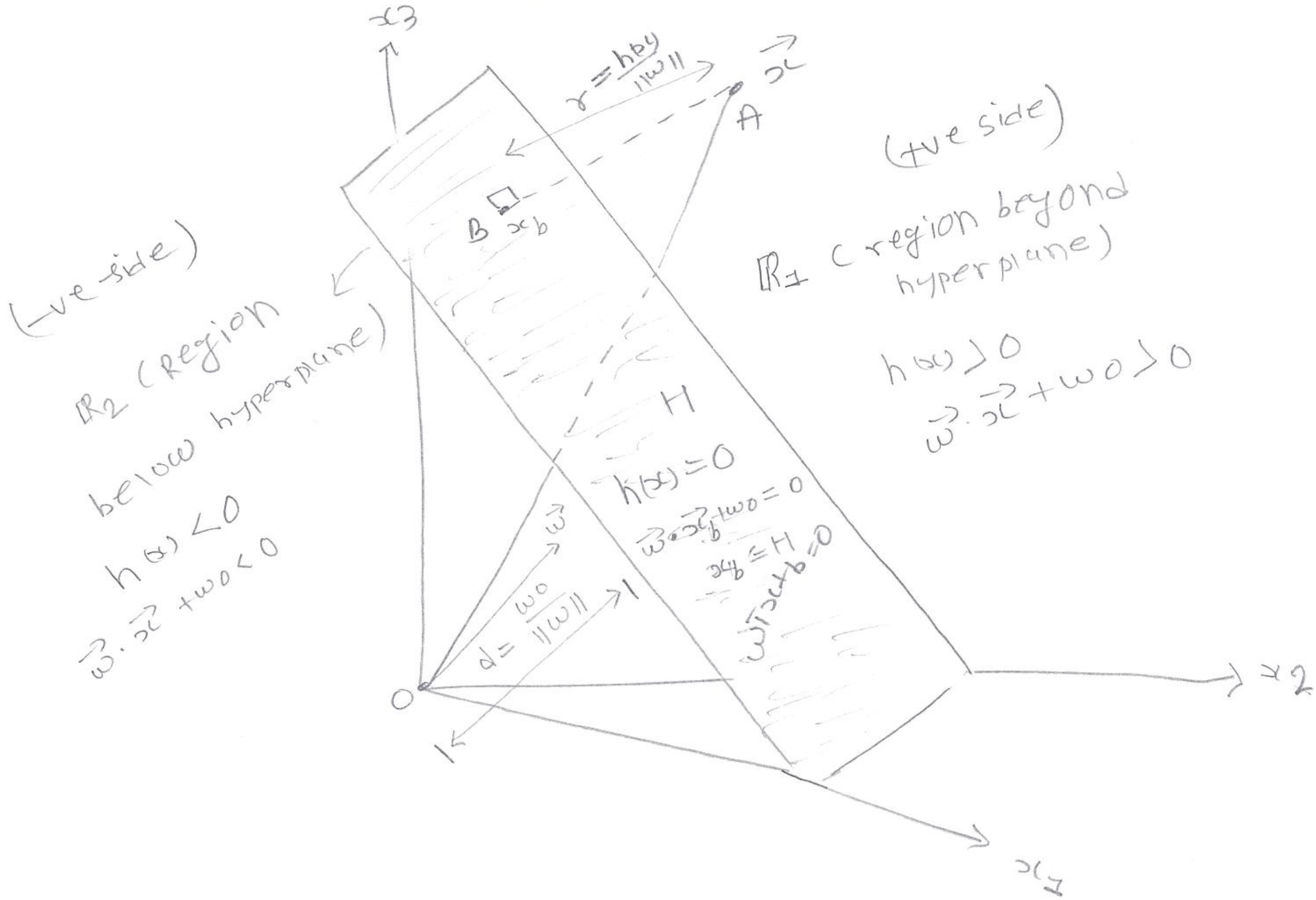
$\therefore \quad r_n = AB = \dfrac{w^T x_n + w_0}{\|w\|}$ where $h(x) = w^T x + w_0$

$$\boxed{r_n = \frac{h(x)}{\|w\|}}$$ proved!

✓ 20

— x — x — x — x — x

QN 2



QN 2 For n-dimensional case, prove:

dist betn <u>origin</u> and <u>decision hyperplane</u> $= \frac{-w_0}{||w||}$

& dist betn <u>datapoint</u> and <u>decision hyperplane</u> $= \frac{h(x)}{||w||}$

<u>soln</u> Let $\vec{x}$ be any point on the positive side of decision boundary (hyperplane), then,

$\vec{x} = \vec{x}_b + r\frac{\vec{w}}{||w||}$ where $\frac{w}{||w||}$ is unit vector perpendicular to decision hyperplane

$$\boxed{x_b = x - r\frac{w}{||w||}} \text{ --- (1)}$$

but, x_b lies in boundary, we need to find r,

$$\boxed{\vec{w} \cdot \vec{x}_b + w_0 = 0} \text{ --- (2)}$$

or, $w^T\left(x - r\frac{w}{||w||}\right) + w_0 = 0 \quad = w^T x - r\frac{w^T w}{||w||} + w_0 = 0$

or, $w^T x + w_0 = r||w|| \quad = h(x)$

$\Rightarrow \quad r = \frac{w^T x + w_0}{||w||}$

distance from datapoint to hyperplane $\boxed{r = \frac{h(x)}{||w||}}$ <u>Ans</u>

W

Also, we have to show the distance of hyperplane from the origin,

$$\vec{OB} = -\frac{w_0}{\|w\|} \hat{OB}$$

Here, the point x_b lies in decision hyperplane, so,

$$\vec{w} \cdot \vec{x_b} + w_0 = 0$$

$$w x_b + w_0 = 0$$

$$w x_b = -w_0$$

distance from origin to decision hyperplane

$$\boxed{x_b = \frac{-w_0}{\|w\|}}$$

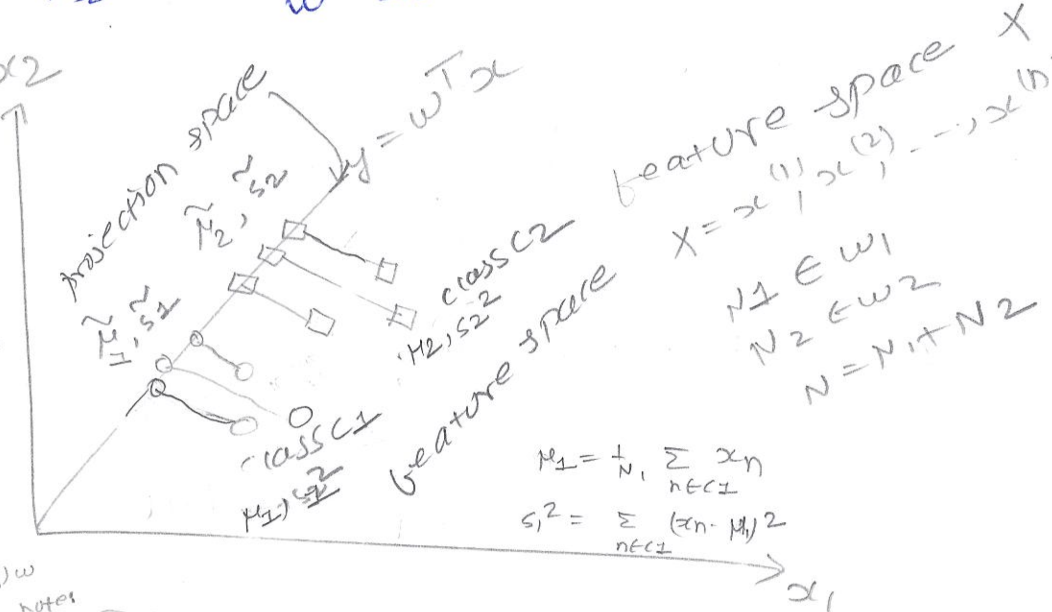
Ans

$\boxed{\text{QN3}}$ show that Fisher criterion

Bishop (ex 4.5) can be written as,

$$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

solution



$\tilde{\mu}_1 = \frac{1}{N_1} \sum_{n \in C_1} w^T x_n = w^T m_1$

$\tilde{s}_1^2 = \sum_{n \in C_1} (y_n - \tilde{\mu}_1)^2$

$= \sum_{n \in C_1} (w^T x_n - w^T \mu_1)^2$

$= \sum_n w^T (x_n - \mu_1)^T (x_n - \mu_1) w$

$\tilde{s}_1^2 = w^T S_1 w$

(big S)

notes: $S_W = S_1 + S_2$

In feature space X, we have input data,

$$x = \left(x^{(1)}, x^{(2)}, \dots, x^{(n)}\right)$$

feature space X, mean $\boxed{\mu_i = \frac{1}{N_i} \sum_{x \in C_i} x}$

projection space y, mean $\tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in C_i} y$

$$= \frac{1}{N_i} \sum_{x \in C_i} w^T x$$

$$\boxed{\tilde{\mu}_i = w^T \mu_i}$$

variance

Also,

variance, (feature space)

$$\boxed{\sigma_i^2 = \frac{1}{N_i} \sum_{x \in e_i} (x - \mu_i)^2}$$

variance (projection space)

$$\boxed{\tilde{\sigma}_i^2 = \frac{1}{N_i} \sum_{y \in e_i} (y - \tilde{\mu}_i)^2}$$

Also, scatter in feature space,

$$s_i^2 = \sum_{x \in e_i} (x - \mu_i)^2 = \sum_{x \in e_i} (x - \mu_i)(x - \mu_i)^T$$

scatter in projection space,

scatter

$$\boxed{\tilde{s}_i^2 = \sum_{y \in e_i} (y - \tilde{\mu}_i)^2}$$

Also, within-class scatter $S_w = S_1 + S_2$

$$\tilde{S}_w = \tilde{S}_1 + \tilde{S}_2$$

Also,

Now, we define Fisher's criterion,

$$\boxed{J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}}$$

Fisher's linear discriminant

Now,

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T\mu_1 - w^T\mu_2)^2$$

$$= w^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w$$

$$\boxed{(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = w^T S_B w} \quad \text{(A)}$$

where, $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ is between-class scatter.

Also,

$$\tilde{S}_i^2 = \sum_{y \in e_i} (y - \tilde{\mu}_i)^2$$

$$= \sum_{x \in e_i} (w^T x - w^T \mu_i)^2$$

$$= \sum_{x \in e_i} w^T (x - \mu_i)(x - \mu_i)^T w$$

$$\boxed{\tilde{S}_i^2 = w^T S_i w}$$

where, feature-space scatter

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$\Rightarrow \tilde{s}_1^2 = w^T S_1 w$$

$$\tilde{s}_2^2 = w^T S_2 w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T(S_1 + S_2)w = w^T S_w w$$

where $S_w = S_1 + S_2$ is within-class scatter.

$$\Rightarrow \boxed{\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w} \longleftarrow (B)$$

using equations (A) & (B), we have,

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

$$\boxed{J(w) = \frac{w^T S_B w}{w^T S_W w}}$$

Q.E.D.!

30

QN4 In Bishop BOOK, (p. 190 or 208)

Bishop (Ex 4.6) p.

use equations: 4.27, 4.28, 4.34, 4.36, 4.33 to prove eq 4.37,

Derive:
$$\left(S_W + \frac{N_1 N_2}{N} S_B\right) w = N(m_1 - m_2) \quad \text{—— } (4.37)$$

Here, Given, SSE is given by

$$\boxed{E(w) = \frac{1}{2}\sum_{n=1}^{N}(w^T x_n + w_0 - t_n)^2} \quad \text{—— } (4.31) \quad p. 190/208$$

$$\boxed{\frac{\partial E}{\partial w} = 0 = \sum_{n=1}^{N}(w^T x_n + w_0 - t_n)\, x_n} \quad \text{—— } (4.33)$$

(4.34)
$$\boxed{w_0 = -w^T m}$$

$$\boxed{t_n = \begin{cases} \frac{N}{N_1} & \text{if } x_n \in C_1 \\ -\frac{N}{N_2} & \text{if } x_n \in C_2 \end{cases}}$$

so that $\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2}$

$\sum_{n=1}^{N} t_n = 0$

mean of the total dataset (population mean)

(4.36)
$$\boxed{m = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}\left(N_1 m_1 + N_2 m_2\right)}$$

where, (sample mean of class C_1)

$m_1 = \frac{1}{N_1}\sum_{n \in C_1} x_n$ (mean of x_n belonging to class C_1) $\Rightarrow \boxed{\sum_{n \in C_1} x_n = N_1 m_1}$

(4.21) p. 205

$m_2 = \frac{1}{N_2}\sum_{n \in C_2} x_n \Rightarrow \boxed{\sum_{n \in C_2} x_n = N_2 m_2}$

we define within-class scatter as,

eq 4.28 p.207
$$\boxed{S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T}$$

and, between-class scatter as,

eq 4.27 (p.207)
$$\boxed{S_B = (m_2 - m_1)(m_2 - m_1)^T}$$

we have to prove,

TO prove
$$\boxed{\left(S_W + \frac{N_1 N_2}{N} S_B\right) w = N(m_1 - m_2)} \quad (\text{eq } 4.37)$$

$$\Rightarrow \frac{N(m_1 - m_2)}{w} = S_W + \frac{N_1 N_2 S_B}{N} \quad \text{——— (1)}$$

we start from gradient zero equation (4.33),

$$0 = \sum_{n=1}^{N} (w^T x_n - w_0 - t_n) x_n$$

$$= \sum_{n=1}^{N} (w^T x_n - w^T m - t_n) x_n$$

$$= \sum_{n=1}^{N} w^T (x_n x_n^T - m x_n) - \sum_{n=1}^{N} t_n x_n$$

$$= \sum_{n \in C_1} (x_n x_n^T - x_n m^T) w - \sum_{n \in C_1} t_n x_n \quad \text{use } t_n = \begin{cases} \frac{N}{N_1} & \text{for } C_1 \\ -\frac{N}{N_2} & \text{for } C_2 \end{cases}$$

$$+ \sum_{n \in C_2} (x_n x_n^T - x_n m^T) w - \sum_{n \in C_2} t_n x_n$$

$$m_2 = \frac{1}{N_2} \sum_{n \in C_2} x_n \Rightarrow \sum_{n \in C_2} x_n = N_2 m_2$$

$$0 = \left(\sum_{n\in C_1} x_n x_n^T - N_1 m_1 m^T\right) w \quad - \frac{N}{\cancel{N_1}} \cancel{N_1} m_1$$

$$\left(\sum_{n\in C_2} x_n x_n^T - N_2 m_2 m^T\right) w \quad + \frac{N}{\cancel{N_2}} \cancel{N_2} m_2$$

$$0 = \left(\sum_{n\in C_1} x_n x_n^T + \sum_{n\in C_2} x_n x_n^T - (N_1 m_1 + N_2 m_2) m^T\right) w \quad - N(m_1 - m_2)$$

$$\frac{N(m_1 - m_2)}{w} = \underbrace{\sum_{n\in C_1} x_n x_n^T + \sum_{n\in C_2} x_n x_n^T}_{\text{write in terms of } S_W} - (N_1 m_1 + N_2 m_2) m^T \quad ②$$

$$m = \frac{N_1 m_1 + N_2 m_2}{N}$$

Now, we expand the first term in S_W,

$$\sum_{n\in C_1} (x_n - m_1)(x_n - m_1)^T = \sum_{n\in C_1} \left(x_n x_n^T - x_n m_1^T - m_1 x_n^T + m_1 m_1^T\right)$$

$$= \sum_{n\in C_1} x_n x_n^T - m_1^T \sum_{n\in C_1} x_n - m_1 \sum_{n\in C_1} x_n^T + \sum_{n\in C_1} m_1 m_1^T$$

$$= \sum_{n\in C_1} x_n x_n^T - N_1 m_1 m_1^T - \cancel{N_1 m_1 m_1^T} + \cancel{N_1 m_1 m_1^T}$$

$$\boxed{\sum_{n\in C_1} (x_n - m_1)(x_n - m_1)^T = \sum_{n\in C_1} x_n x_n^T - N_1 m_1 m_1^T}$$

so, $$S_W = \sum_{n\in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n\in C_2} (x_n - m_2)(x_n - m_2)^T$$

$$\boxed{S_W = \sum_{n\in C_1} x_n x_n^T + \sum_{n\in C_2} x_n x_n^T - N_1 m_1 m_1^T - N_2 m_2 m_2^T}$$

$$\Rightarrow \sum_{n\in C_1} x_n x_n^T + \sum_{n\in C_2} x_n x_n^T = S_W + N_1 m_1 m_1^T + N_2 m_2 m_2^T$$

Then eqn ② becomes,

$$\frac{N(m_1-m_2)}{w} = S_w + N_1 m_1 m_1^T + N_2 m_2 m_2^T - (N_1 m_1 + N_2 m_2)\cdot\frac{(N_1 m_1 + N_2 m_2)^T}{N}$$

$$= S_w + m_1 m_1^T\left(N_1 - \frac{N_1^2}{N}\right) + m_2 m_2^T\left(N_2 - \frac{N_2^2}{N}\right)$$

$$- \frac{N_1 N_2}{N}(m_1 m_2^T + m_2 m_1^T)$$

$N = N_1 + N_2$

$N_1(\cancel{N_1} + N_2 - \cancel{N_1}) = N_1 N_2$

$N_2(N_1 + N_2 - N_2) = N_1 N_2$

$$= S_w + \frac{(N_1+N_2)N_1 - N_1^2}{N} m_1 m_1^T + \frac{(N_1+N_2)N_2 - N_2^2}{N} m_2 m_2^T$$

$$- \frac{N_1 N_2}{N}(m_1 m_2^T + m_2 m_1^T)$$

$$= S_w + \frac{N_1 N_2}{N}(m_1 m_1^T + m_2 m_2^T)$$

$$- \frac{N_1 N_2}{N}(m_1 m_2^T + m_2 m_1^T)$$

$$= S_w + \frac{N_1 N_2}{N}\left(m_1 m_1^T + m_2 m_2^T - m_1 m_2^T - m_2 m_1^T\right)$$

$S_B = (m_2 - m_1)(m_2 - m_1)^T$
$= m_2 m_2^T - m_2 m_1^T - m_1 m_2^T + m_1 m_1^T$
$= m_1 m_1^T + m_2 m_2^T - m_1 m_2^T - m_2 m_1^T$

$$\frac{N(m_1 - m_2)}{w} = S_w + \frac{N_1 N_2}{N} S_B$$

$$\therefore \boxed{\left(S_w + \frac{N_1 N_2}{N} S_B\right) w = N(m_1 - m_2)}$$

Q.E.D.

20 ✓

QN5

Given training set,

x	x_1	x_2	x_3	$t(x)$
$x^{(1)}$	0	0	0	+1
$x^{(2)}$	0	1	0	+1
$x^{(3)}$	1.5	0	-1.5	+1
$x^{(4)}$	1.5	1	-1.5	+1
$x^{(5)}$	1.5	0	0	-1
$x^{(6)}$	1.5	1	0	-1
$x^{(7)}$	0	0	-1.5	-1
$x^{(8)}$	0	1	-1.5	-1

a) prove that the perceptron algorithm does not converge on this dataset. (Note: Include bias term)

b) consider a kernel perceptron that uses a polynomial kernel $K(x,y) = (1+x^T y)^d$.

What is the smallest degree d for which the kernel perceptron would converge on this dataset?

5a

for a given example, example belongs to class +1 if

$$w_0 + \vec{w} \cdot \vec{x}_n \geq 0$$

and, example belongs to class -1 if

$$w_0 + \vec{w} \cdot \vec{x}_n < 0$$

now, $x^{(1)} = \begin{array}{c} \begin{array}{ccc} w_1 & w_2 & w_3 \end{array} \\ [0, 0, 0] \end{array}$ $t^{(1)} = +1$

$$w_0 + w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 0 \geq 0$$

$$w_0 \geq 0$$

similarly,

20

+ve

1. $w_0 + 0 + 0 + 0 \geq 0$ —— (1)
2. $w_0 + 0 + w_2 + 0 \geq 0$ —— (2)
3. $w_0 + 1.5w_1 + 0 - 1.5w_3 \geq 0$ —— (3)
4. $w_0 + 1.5w_1 + w_2 - 1.5w_3 \geq 0$ —— (4)

$(4) - (3) \Rightarrow w_2 \geq 0$

-ve

5. $w_0 + 1.5w_1 + 0 + 0 < 0$ —— (5)
6. $w_0 + 1.5w_1 + w_2 + 0 < 0$ —— (6)
7. $w_0 + 0 + 0 - 1.5w_3 < 0$ —— (7)
8. $w_0 + 0 + w_2 - 1.5w_3 < 0$ —— (8)

$(6) - (5) \Rightarrow w_2 < 0$

Here, (4) - (3) gives $w_2 \geq 0$

but (6) - (5) gives $w_2 < 0$

This is contradictory so perceptron fails.

(5b) Here, the given dataset is <u>not</u> linearly separable in x-space.

we shall propose a perceptron that uses polynomial kernel,

the polynomial kernel is,

$K(x, x') = (a x^T x' + b)^d$

for $a = b = 1$

$$\boxed{K(x,x') = (1 + x^T x')^d}$$

a = scaling factor
b = bias term
d = degree of polynomi
x = augmented dataset of shape N rows m+1 cols
x' = one example of dataset with m+1 features

when d = 1 : The kernel is linear classifier, but our dataset is <u>NON</u> linear, so it cannot classify correctly.

CANNOT

12

When d = 2: when degree is 2, quadratic kernel can classify linearly non-separable dataset.

CAN

need find a vector to seperate the data set.

when d > 2 : higher degree polynomial kernels can fit the linearly non-separable dataset good, but, they may overfit the test dataset.

may overfit

∴ smallest degree d = 2 <u>Ans</u>

qu 6

For a Kernel perceptron,

from lecture note, (Lecture 06)

the algorithm is

1. $f(x) = w^T x = \sum_n \alpha_n t_n x_n^T x = \sum_n \alpha_n t_n \underbrace{K(x_n, x)}_{\text{kernel}}$

2. initialize α : $\alpha_n = 0$

for e in E epochs

3. for $n = 1 \cdots N$

$h_n = \text{sgn}(f(x_n))$

if $h_n \neq t_n$ then

$\alpha_n = \alpha_n + 1$

Now, from ~~step~~ 3, if we run perceptron E number of epochs,

then total number of mistakes made so far is,

$$\boxed{M = \sum_{n=1}^{N} \alpha_n}$$

Ans

10

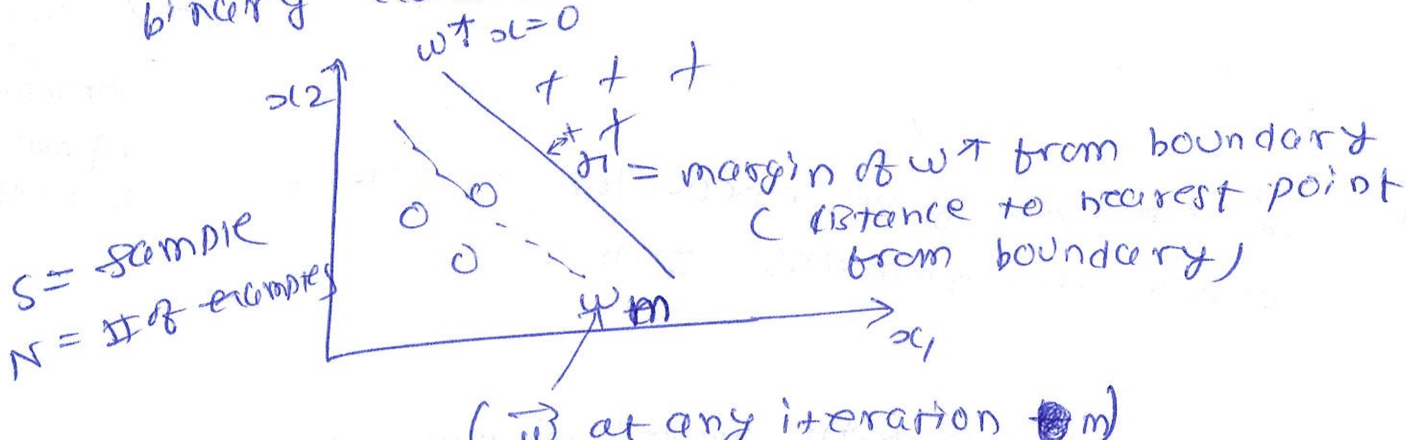
Q N6B ~~Method 2~~ (Extra practice)

A Kernel perceptron for binary classification is run for a number of epochs E on a training dataset containing N examples, resulting in the dual parameters $\alpha_1, \alpha_2, \dots, \alpha_N$

What is the total number of mistakes that are made during training?

maximum Error bound

<u>Sol</u> consider a Kernel perceptron for binary classification.



Let there is a unit vector $\vec{w^T}$ that can separate N training examples into two classes with margin γ.

Let R = norm of maximum value of $\vec{x}$ in dataset

$$\|\vec{R}\| = \max_{x \in S} \|\vec{x}\|$$

perceptron Algorithm:

1. initialize w to zero. $w_1 = [0\ 0\ 0\ 0]$

2. for e in epochs

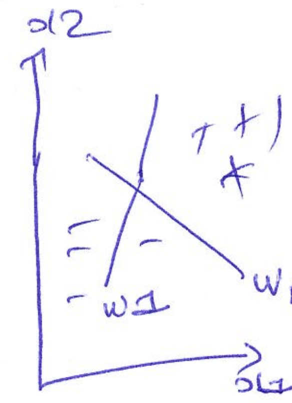
    for $n = 1$ to N

        if $(w\, x_n) < 0$

            $w = w + t_n x_n$

    if converged w:

        break



$w^* = \frac{\vec{w}_M}{|w_M|}$

for total M mistakes

$1 \le m \le M$

* $\vec{w}$ changes if we encounter mistake, otherwise remains same.

* suppose perceptron makes a mistake at iteration m, then,

$$\vec{w}_m \cdot w^* = (\vec{w}_{m-1} + t\vec{x}) \cdot w^*$$

$$= \vec{w}_{m-1} \cdot w^* + t(\vec{x} \cdot \vec{w}^*)$$

$$w_m \cdot w^* \ge \vec{w}_{m-1} \cdot \vec{w}^* + \gamma$$

from definition of γ. each data point x is $\ge \gamma$ from the boundary w^*

$\therefore \boxed{w_m \cdot w^* \geq w_{m-1} \cdot w^* + \gamma}$ — (A)

Also,

$\|w_m\|^2 = \|w_{m-1} + tx\|^2$

$= \|w_{m-1}\|^2 + \underbrace{2t(w_{m-1} \cdot x)}_{(a)} + \underbrace{\|t\vec{x}\|^2}_{(b)}$

(b) maximum of $\|x\|$ is R

(a) Here we have mistake at iteration m, so the previous iteration has hypothesis $t(w_{m-1} \cdot x) < 0$

$\boxed{\|w_m\|^2 \leq \|w_{m-1}\|^2 + R^2}$ — (B) (from (a) & (b))

Here, inequalities (A) & (B) hold true for all the iterations m,

$w_m \cdot w^* \geq w_{m-1} \cdot w^* + \gamma$

$\|w_m\|^2 \leq \|w_{m-1}\|^2 + R^2$

$\|w_m\|^2 - \|w_{m-1}\|^2 \leq R^2$

(If there is a mistake then difference of square of weight vectors is less than R^2).

(for each mistake $w^2 - w_{m-1}^2$ cannot be larger than R^2)

~~after M mistakes,~~

~~$w_M \cdot w_* \geq$~~

For all iterations,

$$w_m \cdot w_* \geq w_{m-1} \cdot w_* + \gamma$$

after M mistakes,

$$\boxed{\vec{w}_M \cdot \vec{w}_* \geq \gamma M} \quad \text{— (C)}$$

also, for all iterations,

$$\|w_m\|^2 \leq \|w_{m-1}\|^2 + R^2$$

after M mistakes

$$\|w_M\|^2 \leq R^2 M$$

$$\boxed{\|w_M\| \leq R\sqrt{M}} \quad \text{— (D)}$$

so, from (C) & (D),

$$\gamma M \leq \vec{w}_M \cdot \vec{w}_*$$

$$\leq \|\vec{w}_m\| \, \|\vec{w}_*\|$$

∵ $\vec{a} \cdot \vec{b} = ab\cos\theta$

$\cos\theta$ is less than or equal to 1

$$\leq \|w_M\|$$

w^* is unit vector of $\vec{w}_M$

$\|w^*\| = 1$

$$\gamma M \leq R\sqrt{M}$$

$$\gamma\sqrt{M} \leq R$$

$$\gamma^2 M \leq R^2$$

$$\boxed{M \leq \left(\frac{R}{\gamma}\right)^2}$$

Ans

Hence,

Let the data (training) sample S is,

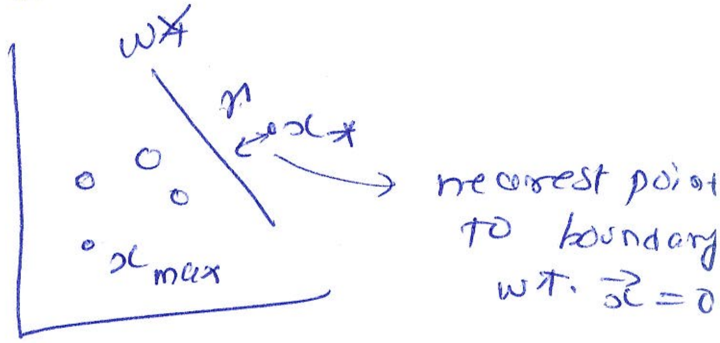
$x_1 \quad t_1$

$x_2 \quad t_2$

$x_3 \quad t_3$

$\vdots \quad \vdots$

$x_N \quad t_N$



$\|x_{max}\| = R$

example,

$x_1: \; 0 \; 0 \; 1 \; 2 \; 3 \qquad t_1 = +$

$x_2: \; 5 \; 0 \; 3 \; 2 \; 6 \qquad t_2 = -$

$x_N: \; 10 \; 20 \; 30 \; 40 \qquad t_N = +$

Then, if we use Perceptron to train our model, the total number of mistakes (M) made is,

$$\boxed{M \leq \left(\frac{R}{\gamma}\right)^2}$$

$\boxed{\text{QN7}}$ Let $U \in R_{k \times m}$ u_i, x_i = ith column of

$X \in R_{n \times m}$ U and X respectively

$1 \le i \le m$.

prove $UX^T = \sum_{i=1}^{m} u_i x_i^T$

solution:

$$U = U_{k,m} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots \\ u_{k1} & u_{k2} & \cdots & u_{km} \end{pmatrix}_{k,m} = \begin{pmatrix} u_1 & u_2 & \cdots & u_m \end{pmatrix}_{1,m}$$

k rows
m cols

$$X = X_{n,m} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}_{n,m} = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \end{pmatrix}_{1,m}$$

n rows
m cols

~~x_1~~ column vector

$$V = X^T = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & & x_{n2} \\ \vdots & \vdots & & \vdots \\ x_{1m} & x_{2m} & & x_{nm} \end{pmatrix} = \begin{pmatrix} x_1^T \\ x_2^T \\ x_m^T \end{pmatrix}_{m,1}$$

From definition of matrix multiplication

if $U = U_{km}$

and $V = V_{mn}$

then,

$$(UV)_{pq} = \sum_{i=1}^{m} U_{pi} V_{iq} \qquad \forall\ 1<p<K \text{ and } 1<q<n$$

$1<p<K$
$1<q<n$

where $X = \begin{pmatrix} x_{11} & x_{12} & x_{1m} \\ x_{21} & x_{22} & x_{2m} \\ x_{n1} & x_{n2} & x_{nm} \end{pmatrix}_{n,m}$

n rows
m cols

$x_1 = \begin{pmatrix} x_{11} \\ x_{21} \\ x_{m1} \end{pmatrix}_{n,1}$

$$= \begin{pmatrix} u_{11} & u_{12} & u_{1m} \\ u_{21} & u_{22} & u_{2m} \\ u_{k1} & u_{k2} & u_{km} \end{pmatrix}_{k,m} \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & & x_{n2} \\ x_{1m} & x_{2m} & & x_{nm} \end{pmatrix}_{m,n}$$

$$= \begin{pmatrix} u_{11}x_{11} + u_{12}x_{12} + u_{1m}x_{1m} & u_{11}x_{21} + u_{12}x_{22} + u_{1m}x_{2m} & u_{11}x_{n1} + u_{12}x_{n2} + u_{1m}x_{nm} \\ u_{21}x_{11} + u_{22}x_{12} + u_{2m}x_{1m} & u_{21}x_{21} + u_{22}x_{22} + u_{2m}x_{2m} & u_{21}x_{n1} + u_{22}x_{n2} + u_{2m}x_{nm} \\ u_{k1}x_{11} + u_{k2}x_{12} + u_{km}x_{1m} & u_{k1}x_{21} + u_{k2}x_{22} + u_{km}x_{2m} & u_{k1}x_{n1} + u_{k2}x_{n2} + u_{km}x_{nm} \end{pmatrix}_{k,n}$$

$$= \begin{pmatrix} u_{11}x_{11} & u_{11}x_{21} & u_{11}x_{n1} \\ u_{21}x_{11} & u_{21}x_{21} & u_{21}x_{n1} \\ u_{k1}x_{11} & u_{k1}x_{21} & u_{k1}x_{n1} \end{pmatrix}_{k,n} + \begin{pmatrix} u_{12}x_{12} & u_{12}x_{22} & u_{12}x_{n2} \\ u_{22}x_{12} & u_{22}x_{22} & u_{22}x_{n2} \\ u_{k2}x_{12} & u_{k2}x_{22} & u_{k2}x_{n2} \end{pmatrix}_{k,n} + \begin{pmatrix} u_{1m}x_{1m} & u_{12}x_{22} & u_{1m}x_{nm} \\ u_{2m}x_{1m} & u_{22}x_{22} & u_{2m}x_{nm} \\ u_{km}x_{1m} & u_{k2}x_{22} & u_{km}x_{nm} \end{pmatrix}_{k,n}$$

$$= u_1 x_1^T + u_2 x_2^T + u_m x_m^T$$

10 ✓

since,

$$u_1 x_1^T = \begin{pmatrix} u_{11} \\ u_{21} \\ u_{k1} \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{21} \\ x_{n1} \end{pmatrix}^T$$

$$= \begin{pmatrix} u_{11} \\ u_{21} \\ u_{k1} \end{pmatrix}_{k,1} \begin{pmatrix} x_{11} & x_{21} & x_{n1} \end{pmatrix}_{1,n}$$

$$u_1 x_1^T = \begin{pmatrix} u_{11}x_{11} & u_{11}x_{21} & u_{11}x_{n1} \\ u_{21}x_{11} & u_{21}x_{21} & u_{21}x_{n1} \\ u_{k1}x_{11} & u_{k1}x_{21} & u_{k1}x_{n1} \end{pmatrix}_{k,n} \quad \text{——— (a)}$$

similarly,

$$u_2 x_2^T = \begin{pmatrix} u_{12}x_{12} & u_{12}x_{22} & u_{12}x_{n2} \\ u_{22}x_{12} & u_{22}x_{22} & u_{22}x_{n2} \\ u_{k2}x_{12} & u_{k2}x_{22} & u_{k2}x_{n2} \end{pmatrix}_{k,n} \quad \text{——— (b)}$$

and,

$$u_m x_m^T = \begin{pmatrix} u_{1m}x_{1m} & u_{12}x_{22} & u_{1m}x_{nm} \\ u_{2m}x_{1m} & u_{22}x_{22} & u_{2m}x_{nm} \\ u_{km}x_{1m} & u_{k2}x_{22} & u_{km}x_{nm} \end{pmatrix} \quad \text{——— (c)}$$

$$\therefore \; u x^T = u_1 x_1^T + u_2 x_2^T + \cdots + u_m x_m^T$$

$$\boxed{u x^T = \sum_{i=1}^{m} u_i x_i^T} \quad \text{Q.E.D.}$$