Com S 574		Spring Semester 2021
Intro Machine Learning		
	Homework 4	
Instructor: Forrest Bao		Submitted by: Ibne Farabi Shihab

1 No Question

Feature vectors with labels -

$x_1 = [1.1764 \quad 4.2409 \quad 0.9750];$

$\mathrm{y}_1 = 1$

$x_2 = [1.0400 \quad 3.8676 \quad 0.4243];$

$\mathrm{y}_2 = 1$

$x_3 = [1.0979 \quad 1.0227 \quad 0.4484];$

$\mathrm{y}_3 = 1$

$x_4 = [2.0411 \quad 4.7610 \quad 0.6668];$

$\mathrm{y}_4 = -1$

$x_5 = [2.0144 \quad 4.1217 \quad 1.2470];$

$\mathrm{y}_5 = -1$

$x_6 = [2.1454 \quad 4.4439 \quad 0.3974];$

$\mathrm{y}_5 = -1$

By multiplying features with labels, we get :

$x_1y_1 = [1.1764 \quad 4.2409 \quad 0.9750]$

$x_2y_2 = [1.0400 \quad 3.8676 \quad 0.4243]$

$x_3y_3 = [1.0979 \quad 1.0227 \quad 0.4484]$

$x_4y_4 = [-2.0411 \quad -4.7610 \quad -0.6668]$

$x_5y_4 = [-2.0144 \quad -4.1217 \quad -1.2470]$

$x_6y_4 = [-2.1454 \quad -4.4439 \quad -0.3974]$

We know that,

$w = \sum_k -1^k \lambda_k y_k x_k$

So, from there we can say that:

$w_1 = 1 * [1.1764 \quad 4.2409 \quad 0.9750]$

$= [1.1764 \quad 4.2409 \quad 0.9750]$

$w_2 = w_1 + 0.7383 * [1.0400 \quad 3.8676 \quad 0.4243]$

$= [1.944232 \quad 7.09634908 \quad 1.28826069]$

$w_3 = w_2 + 0 * [1.0979 \quad 1.0227 \quad 0.4484]$

$= [1.944232 \quad 7.09634908 \quad 1.28826069]$

$w_4 = w_3 + 0.0411 * [-2.0411 \quad -4.7610 \quad -0.6668]$

$= [1.86034279 \quad 6.90067198 \quad 1.26085521]$

$w_5 = w_4 + 1 * [-2.0144 \quad -4.1217 \quad -1.2470]$

$= [-0.15405721 \quad 2.77897198 \quad 0.01385521]$

$w_6 = w_5 + 0.6972 * [-2.1454 \quad -4.4439 \quad -0.3974]$

$= [-1.64983009 \quad -0.3193151 \quad -0.26321207]$

After all the updates, we get the final:

w= - $[-1.64983009 \quad -0.3193151 \quad -0.26321207]$

2 No Question

Here given that, $w_b = 3.3149$

$x = [1, 1, 0]^T$

We calculated in the following way:

$w^T x + w_b$

$= (-1.64983009 - 0.3193151 - 0.26321207) * [1, 1, 0]^T + 3.3149$

$= 1.346 > 0$

From the calculation we can see that the new sample belongs to class $C\epsilon(+1)$

3 No Question

Given that,

$x = [1, 1, 0]^T$

We know that :

$W^T + W_b = 1$

$$[-1.64983009 - 0.3193151 - 0.26321207] * \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} + 3.3149 = \pm 1$$

For +1,

$[-1.64983009x_a - 0.3193151x_b - 0.26321207x_c] + 2.3149 = 0$

For -1,

$1.64983009x_a + 0.3193151x_b + 0.26321207x_c = 2.3149$

4 No Question

The calucation for every sample is given below:

Sample 1:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 1.174 \\ 4.2409 \\ 0.975 \end{bmatrix} + 1$$

$= -2.55$

Verdict:Point outside gutter

Sample 2:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 1.04 \\ 3.8676 \\ 0.4243 \end{bmatrix} + 1$$

$= -2.06$

Verdict:Point outside gutter

Sample 3:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 1.0979 \\ 1.0227 \\ 0.4484 \end{bmatrix} + 1$$

$= -1.256$

Verdict:Point outside gutter

Sample 4:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 2.04 \\ 4.761 \\ 0.666 \end{bmatrix} + 1$$

$= -4.063$

Verdict:Point outside gutter

Sample 5:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 2.0144 \\ 4.1217 \\ 1.247 \end{bmatrix} + 1$$

$= -3.9677$

Verdict:Point outside gutter

Sample 6:

$$W^T + w_b = (-1.64983009 - 0.3193151 - 0.26321207) * \begin{bmatrix} 2.1454 \\ 4.4439 \\ .3974 \end{bmatrix} + 1$$

$= -4.063157$

Verdict: Point outside gutter

5 No Question

From lecture we know that, mis-classified point falls in the opposite site of the gutter.

So the sample must fulfill the following condition:

$-1 \leq y_i(w^T + w_b)$

or, $y_i(w^T + w_b) \geq -1$ when a value is less than -1 must be less than 1 or 0;

Hence, the sample must fulfill the following 3 conditions-

$y_i(w^T + w_b) \geq -1$

$y_i(w^T + w_b) \geq 1$

$y_i(w^T + w_b) \geq 0$

On the other hand, the value which is greater than 0 or 1 has to be greater than -1;

So the false choices should be -

$y_i(w^T + w_b) \leq 1$

$y_i(w^T + w_b) \leq -1$

$y_i(w^T + w_b) \leq 0$

6 No Question

For training a machine learning model, we first divide the data into train and test set. For validation phase, during each validation we split the data into k subset and use k-1 subset as training and other one as validation set. We do this for k-fold times. So, we do not get same training set every time.

The testing set remains the same as we do not do anything with that.