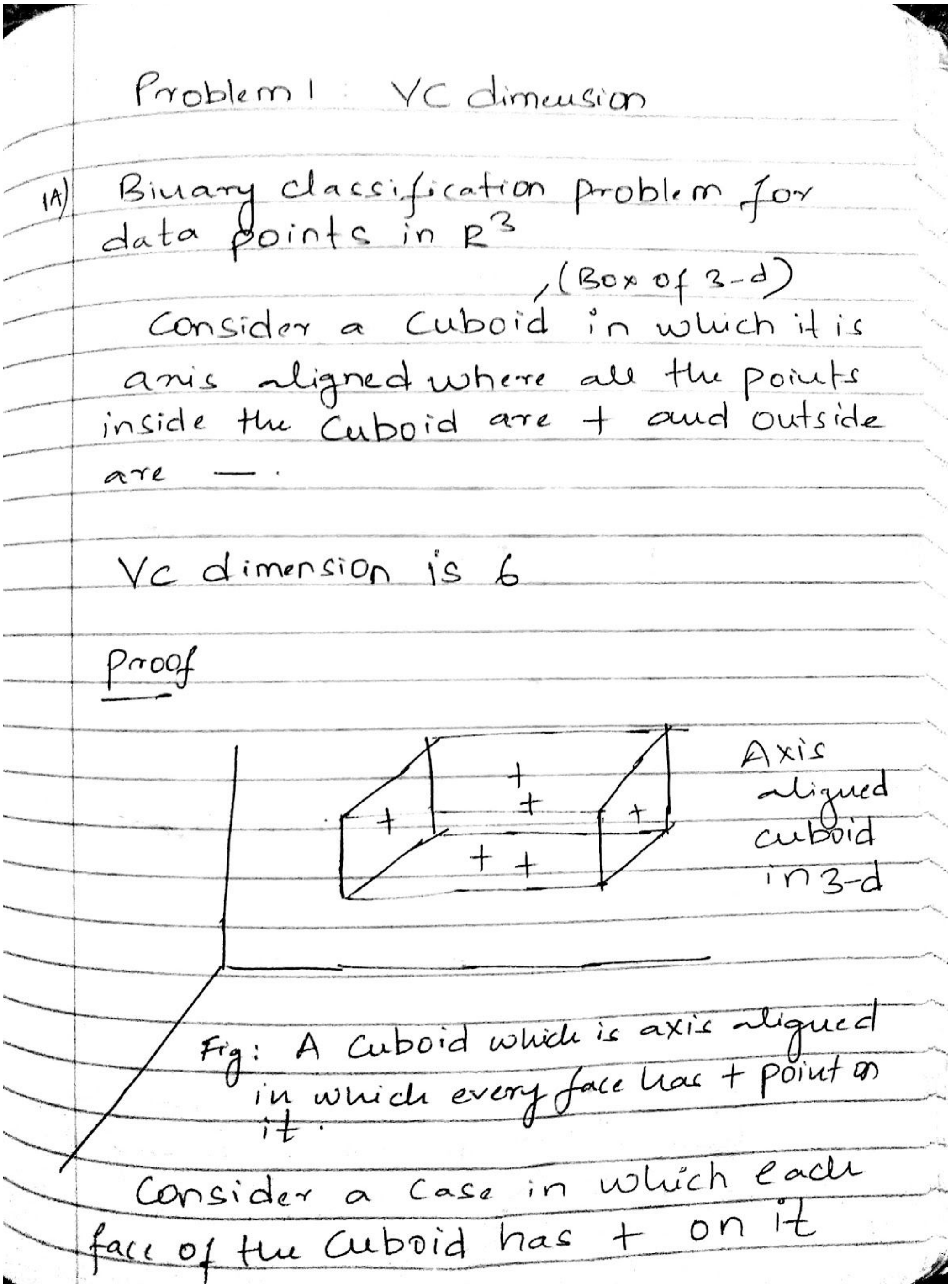
MACHINE LEARNING ASSIGNMENT 3

Problem 1: VC dimension

1a)



Problem 1 : VC dimension

1A) Binary classification problem for data points in R^3

Consider a Cuboid, (Box of 3-d) in which it is axis aligned where all the points inside the Cuboid are + and outside are — .

Vc dimension is 6

Proof

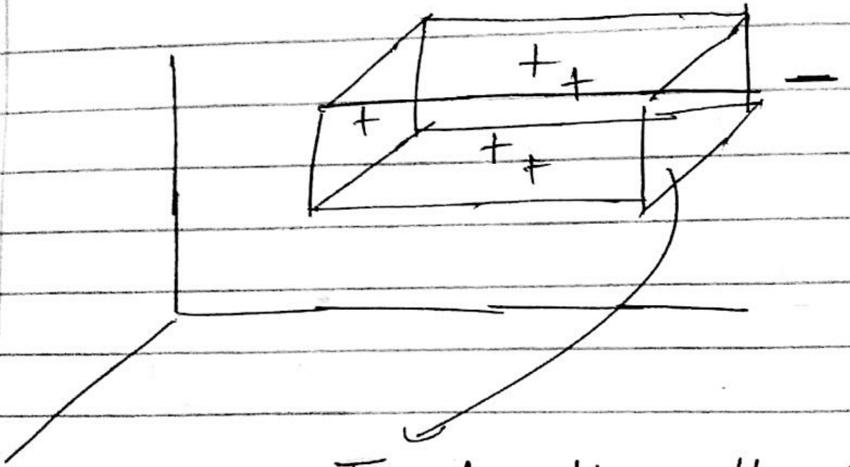
+
+
+
+
+
+
+
Axis aligned cuboid in 3-d

Fig: A Cuboid which is axis aligned in which every face has + point on it.

Consider a Case in which each face of the Cuboid has + on it

If the + is Changed to - on the cuboid shown, then the Cuboid can be shrinked in terms of length, width and height.

Example: —



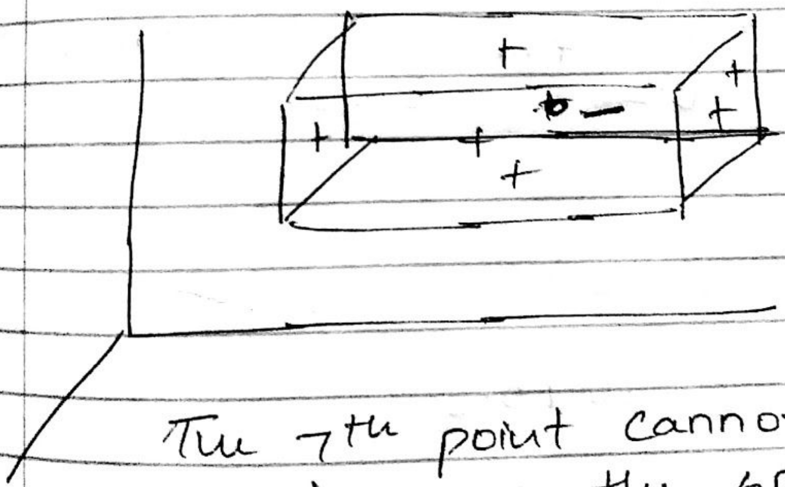
The length of the Cuboid is reduce in order to Separate the — out of the cuboid.

In this way if any point on the face of the cuboid is Changed, then it is adjusted in length, height & width.

why VC is not 7?

If 6 points are considered to be on each of the face of cuboid and 7th point should be inside the (center of the) cuboid.

Example Fig: —



All the 6 points occupy extremes and 7th points at the centre with −

The 7th point cannot be classified correctly as the 6 points occupy each face/ occupy extremes) and the 7th points should be with in the cuboid.

Hence, VC dimension is 6

$\text{accuracy} = 0.8$

$\epsilon = \text{error} = 1 - 0.8 = 0.2$

$1 - \delta = 0.95$

$\delta = 0.05$

$M \geq \frac{1}{\epsilon}\left(4 \ln \frac{2}{\delta} + 8 * VC(H) \ln \frac{13}{\epsilon}\right.$

$M \geq \frac{1}{0.2}\left(4 \ln \frac{2}{0.05} + 8 * 6 * \log \frac{13}{0.2}\right.$

$M \geq \frac{1}{0.2}(215.126)$

$M \geq 1075.6$

$M \geq 1076$

Generalization to d dimensions

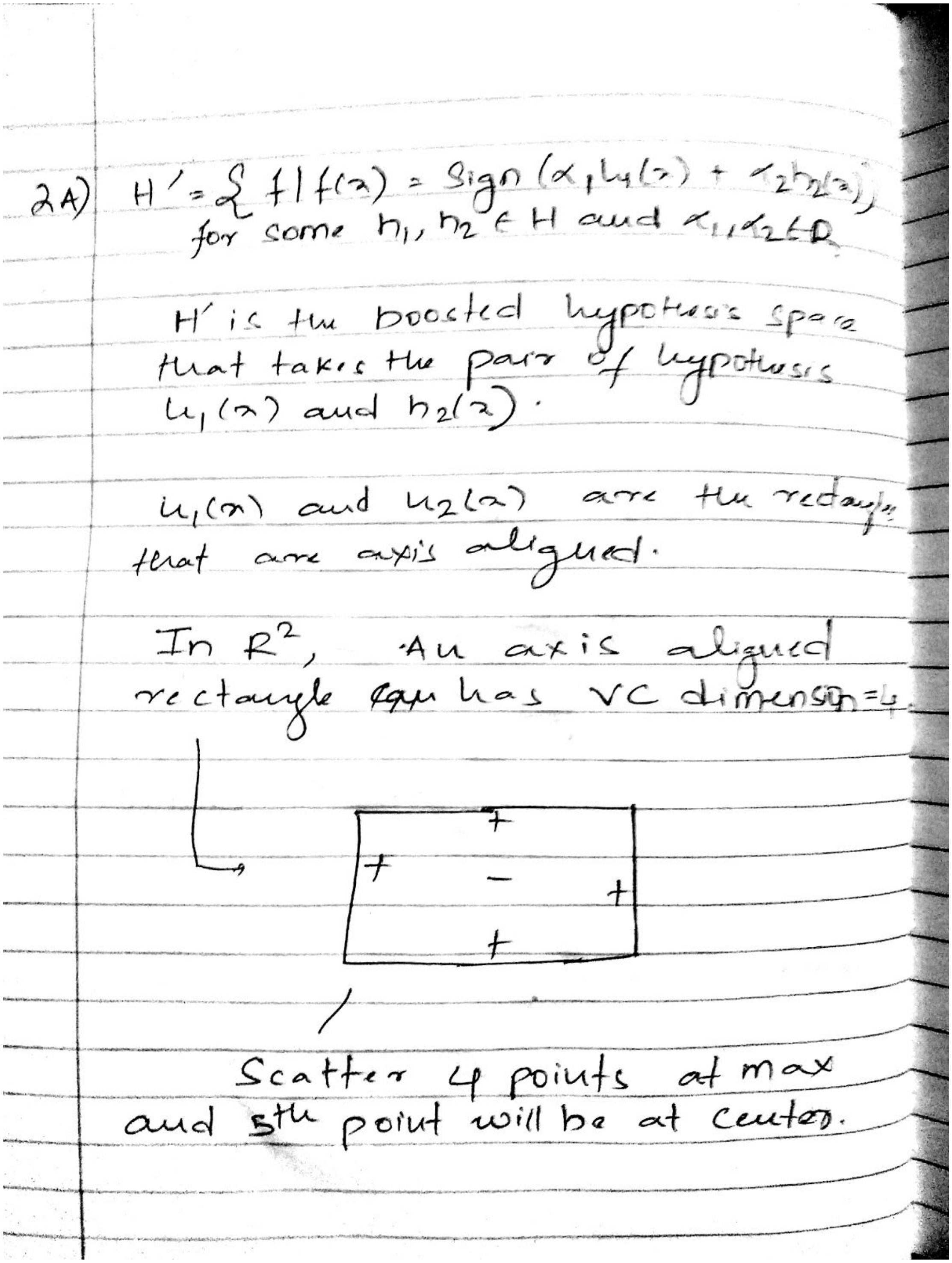
~~If~~ VC dimension will be twice the number of dimension as each face on each dimension can take ~~1~~point.

$$\text{VC dimension} = 2 * d$$

and

$$M \geq \frac{1}{\epsilon}\left(4 \ln \frac{2}{\delta} + 8 * \underbrace{2d}_{\text{dimension.}} * \ln \frac{13}{\epsilon}\right)$$

1b) Boosted Axis aligned Rectangles



2A) $H' = \{ f \mid f(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x)),$ for some $h_1, h_2 \in H$ and $\alpha_1, \alpha_2 \in R$

H' is the boosted hypothesis space that takes the pair of hypothesis $h_1(x)$ and $h_2(x)$.

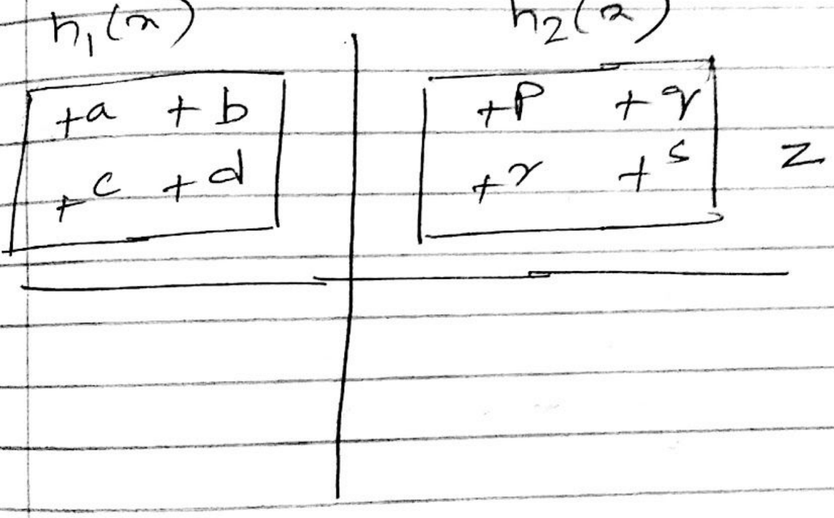
$h_1(x)$ and $h_2(x)$ are the rectangles that are axis aligned.

In R^2, An axis aligned rectangle ~~can~~ has VC dimension=4.

Scatter 4 points at max and 5th point will be at center.

consider a case where $\alpha_1 = \alpha_2$

Each hypothess has equal contributon ,



consider the data point 'p' in the above figure.

$H(x)$ will classify 'p' as −

$h_2(x)$ will classify 'p' as +

$f(x) = \text{Sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x))$

$\Rightarrow$ | $\alpha_1 = \alpha_2$ #

$\Rightarrow$ $\text{Sigu}(\alpha_1 h_1(x) + \alpha_1 h_2(x))$

$h_1(x) = -1$

$h_2(x) = +1$

$\Rightarrow$ $\text{Sign}(-\alpha_1 + \alpha_1)$

$= 0$

$\downarrow$

0 means f(x) classifies the point 'p' as positive

$\therefore$ $h_1(x)$ and $h_2(x)$ classifies all 8 positive points

Iconsider the point z in the same figure in which both the hypothesis classifies it either as + or −

consider $\alpha_1, \alpha_2 > 0$

$h_1(x) = h_2(x) = +$ Sign $(\alpha_1 + \alpha_2) = +$

or

$h_1(x) = h_2(x) = -$ Sign$(-\alpha_1 - \alpha_2) = -$

Hence, all the 8 points will be classified.

So, VC dimension is $\underline{8}$.

why not 9?

(Scatter)

A rectangle can classify only 4 points at max.

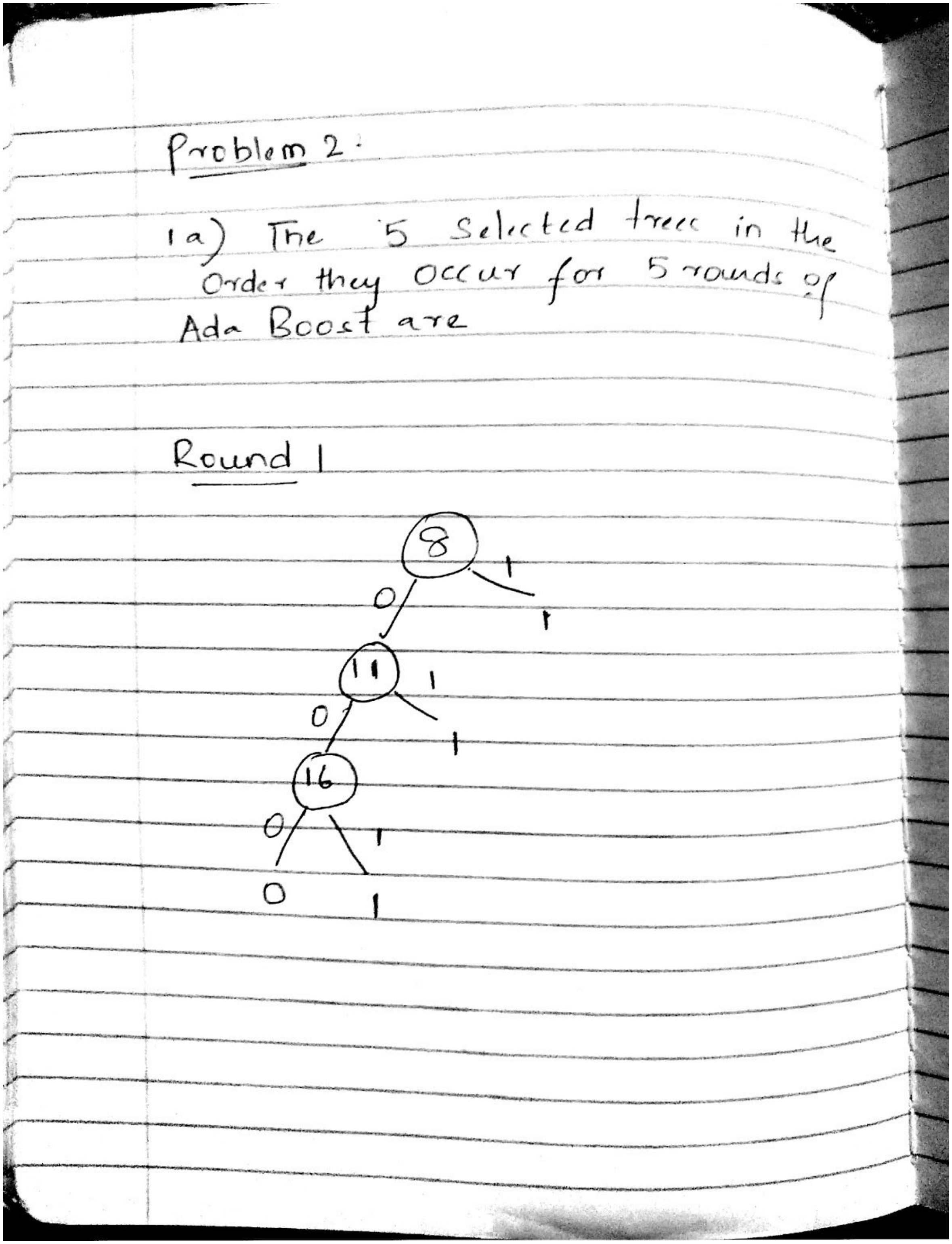
So, 2 axis aligned rectangles can scatter 8.

So, VC dimension is 8 not 9.

Problem 2: Medical Diagnostics

Refer to program 2.1a.py for 5 rounds of boosting

1a)



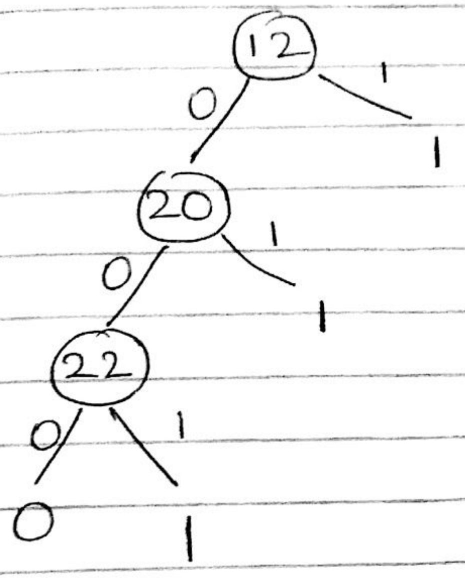
Problem 2:

1a) The 5 Selected trees in the Order they occur for 5 rounds of Ada Boost are

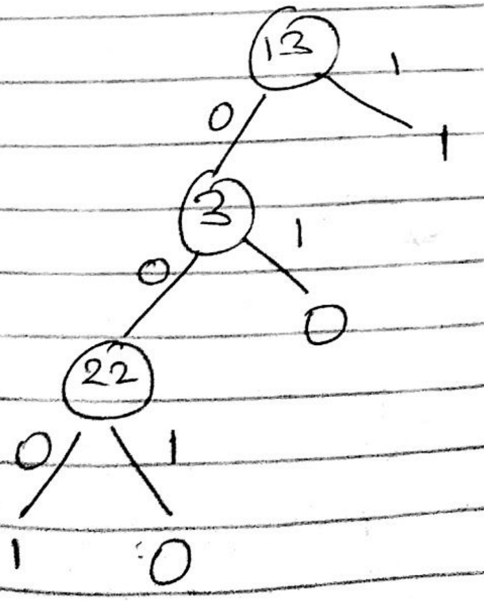
Round 1

8
0
1
1
11
0
1
1
16
0
1
0
1

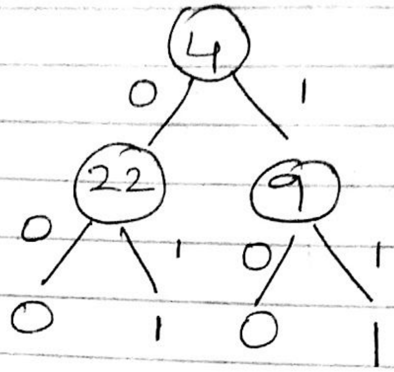
Round 2



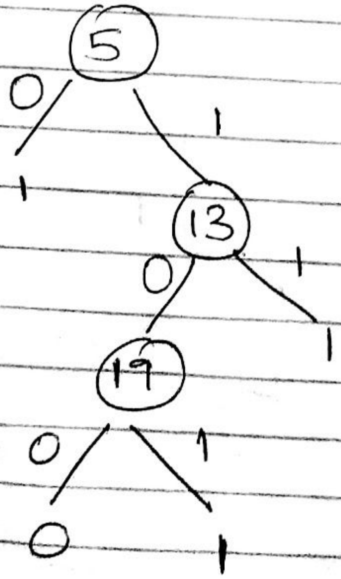
Round 3



Round 4



Round 5



Values of Alpha and Epsilon generated in the ordering

1 . The value of alpha and epsilon for round 1 is 0.7331685343967135, 0.18750000000000003

2 . The value of alpha and epsilon for round 2 is 0.49926441505556, 0.26923076923076916

3. The value of alpha and epsilon for round 3 is 0.3308654921632833, 0.3403508771929824

4. The value of alpha and epsilon for round 4 is 0.33790736963862866, 0.33719604863221897

5. The value of alpha and epsilon for round 5 is 0.26672793237717274, 0.3697112237623083

1b) **Refer to program Boost10.py for 10 rounds of boosting**

Accuracies on the training data set are [81.25, 81.25, 81.25, 87.5, 80.0, 90.0, 90.0, 91.25, 92.5, 91.25]

Accuracies on the test data set are [75.93582887700535, 75.93582887700535, 81.81818181818183, 79.14438502673798, 81.81818181818183, 75.93582887700535, 75.93582887700535, 75.40106951871658, 73.2620320855615, 74.33155080213903]

Plot in next page

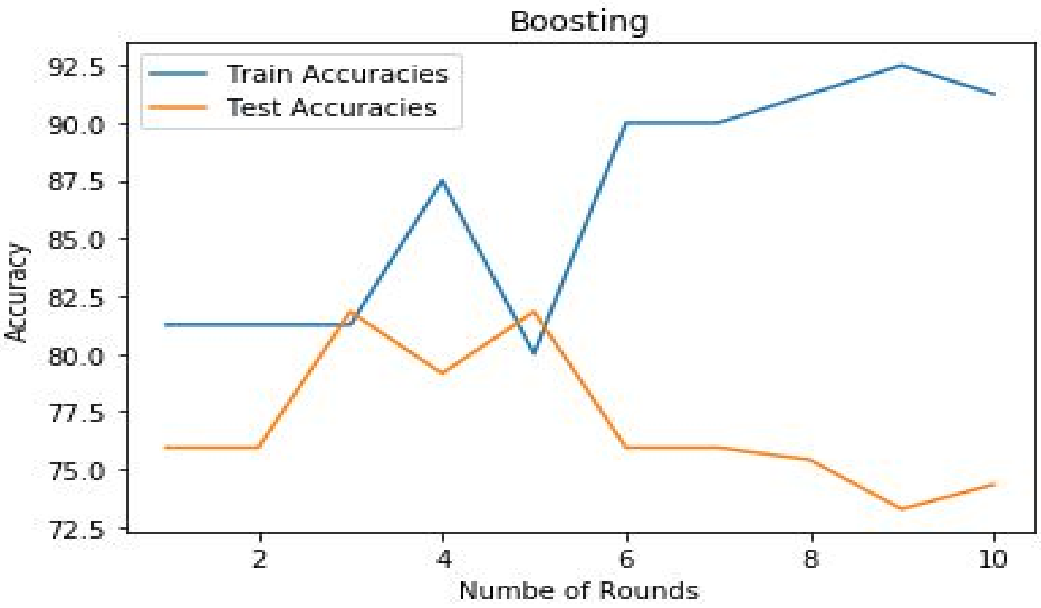


Fig showing plot between accuracies(Train and Test) and number of rounds

2a)

Refer to program CD.py

The optimal value of alpha is
[0.49999999999998485, 3.0520049917641083, 0.5000000000000137, 1.4272912997155005,
0.5000000000000022, 0.9846274408891119, 0.5000000000000575, 0.9083741643393392,
0.4999999999999967, 5.82069821673548, 0.4999999999999574, 3.927068128328222,
0.5000000000000545, -0.31327938260149085, 0.5000000000000268,
-0.027262278811380623, 0.5000000000000151, 0.18110783587078969,
0.5000000000000097, 0.3581248541102072, 0.5000000000000075, 0.3693825581692475,
0.4999999999999952, 0.39691529356031935, 0.5000000000000016, -0.10747220704336483,
0.4999999999999964, 0.1775608822633549, 0.5000000000000057, -4.430126382183389,
0.49999999999994965, -2.816293053719661, 0.49999999999997113, 4.123582657426387,
0.4999999999999723, 3.304148086075774, 0.5000000000000279, -1.9183964328200258,
0.5000000000000041, -0.41728601733444876, 0.49999999999997524,
0.10916053815682555, 0.5000000000000189, 0.24125011435686555, 0.4999999999999985,
0.9559936171721358, 0.5000000000000105, 0.778707556693998, 0.500000000000014,
-0.2936930607830476, 0.49999999999999334, 0.11438150124808323, 0.5000000000000168,
-2.5898146610479498, 0.5000000000000038, -1.865577276509186, 0.5000000000000098,
-0.731752508501147, 0.5000000000000026, -0.7078556733844109, 0.4999999999999985,
-0.23672825294232136, 0.4999999999999824, -0.09203008957241801,
0.4999999999999883, -2.7077759008087483, 0.500000000000022, -2.6456124597762845,
0.49999999999999023, -1.8883984888578393, 0.4999999999999945, -1.8687375716619468,

0.5000000000000143, 0.44890840867857745, 0.49999999999997335, 0.4665160908476605, 0.49999999999999223, 0.17483555592347, 0.5000000000000089, 0.2852521956566988, 0.49999999999998274, 0.33602463191962806, 0.49999999999999256, 0.38706995225494073, 0.5000000000000003, 0.12270429156607024, 0.49999999999999506, 0.3246482293480271]

Exponential Loss on Training data set is 39.4751466595816

2b)

Refer to program CD.py

Accuracy on the test data set on the resulting classifier is 69.5187165775401

2c)

Refer to Program Boost20.py

Accuracy on the test data set after 20 rounds is 66.84491978609626

Alpha learned by AdaBoost 20 rounds is
[0.48470027859405157, 0.23296515192905, 0.3084191516166043, 0.269059960403377, 0.12707489951194617, 0.1939136051934971, 0.16646581973436514, 0.20484093019665991, 0.11233780854211395, 0.16915310437537498, 0.12721184797605228, 0.15386377069176382, 0.09807054208811221, 0.15796706354327916, 0.13458519734012808, 0.130356664737319, 0.10144574639261557, 0.12621641283923687, 0.10744964087976837, 0.11081674484994171]

The alpha learned by AdaBoost differs from Coordinate descent because the adaboost is run for 20 rounds and has only 20 values of alpha. On the other hand the total hypothesis space in the Coordinate descent is 88 so alpha has 88 values in it.
Apart from the difference in the length of alpha values, In the Coordinate descent only takes fixed subset of values into consideration while AdaBoost takes epsilon, weight and Hypothesis output .

The main difference is that in the alpha values of the Coordinate descent there are some negative values while alpha values in the AdaBoost are positive. From this we can conclude that all the hypothesis are contributing to the output in Adaboost which is not the case in the Coordinate descent .

2d) Bagging

Refer to program Bagging.py

Accuracy on test data set is 60.42780748663101

Classifiers are [{17: {0: 0, 1: 1}}, {8: {0: 0, 1: 1}}, {13: {0: 0, 1: 1}}, {16: {0: 0, 1: 1}}, {22: {0: 0, 1: 1}}, {7: {0: 0, 1: 1}}, {11: {0: 0, 1: 1}}, {12: {0: 0, 1: 1}}, {18: {0: 0, 1: 1}}, {20: {0: 0, 1: 1}}, {3: {0: 0, 1: 1}}, {9: {0: 0, 1: 1}}, {21: {0: 0, 1: 1}}, {4: {0: 0, 1: 1}}, {6: {0: 0, 1: 1}}, {14: {0: 0, 1: 1}}, {5: {0: 0, 1: 1}}, {2: {0: 0, 1: 1}}, {15: {0: 0, 1: 1}}, {19: {0: 0, 1: 1}}]

Bagging test accuracy is lower(60% accuracy) compared to the remaining classifiers AdaBoost 20 rounds(66.8%) and Coordinate Descent(69.5%)

As bagging takes random data sampling, the accuracy turns out to be lower compared to the remaining classifiers.

2e)

Out of the 3 classifiers, Coordinate Descent is preferred as it has the highest accuracy of 69.5 on test data out of all the 3 classifiers and has faster computation as it takes only subset of values into consideration.

Hence, Coordinate Descent is preferred