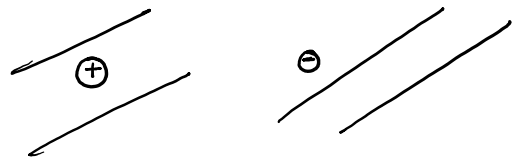
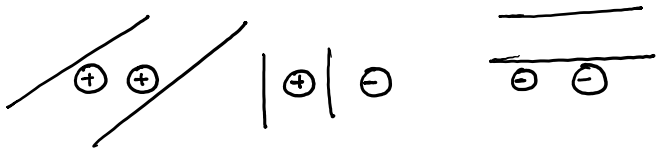
PROBLEM-1 : VC - DIMENSION :

1) Point b/w parallel lines is classified (+) and points outside as (-)

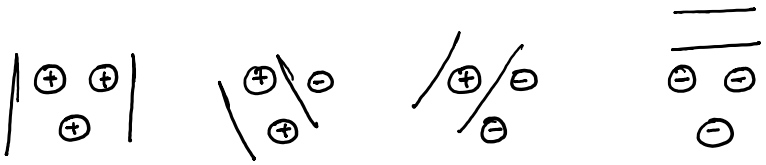
One point :



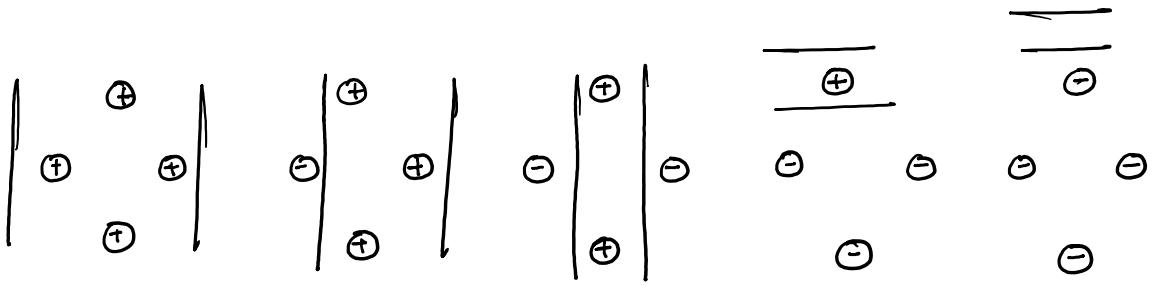
Two points :



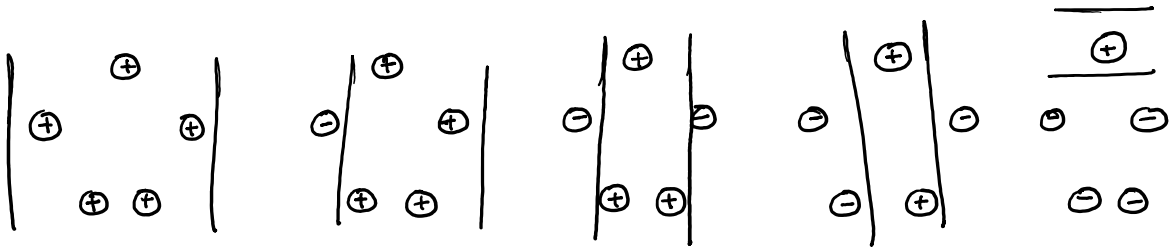
Three points :



Four points :



Five points :



Since they are linear separators, by pigeon hole principle $VC(H) < 7$

As for linear separator in $\mathbb{R}^2$ $VC(H) = 3$ → For two it should be < 7

i.e for greater than 4 one separator should shatter 4 points. Which is not possible.

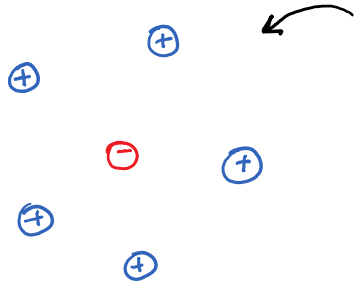
For 6 points : Configurations of 6 points can result in convex hulls of vertices ≤ 6

i) if configuration of convex hull is <6

then pick all other points (+) and the point inside (−)

As parallel lines are drawn so as to include all +ve's

−ve point which is inside is not shatterable



ii) If configuration of points results in convex hull of $\#V = 6$

To check whether this configuration can be shatterable or not

As it forms a convex polygon,

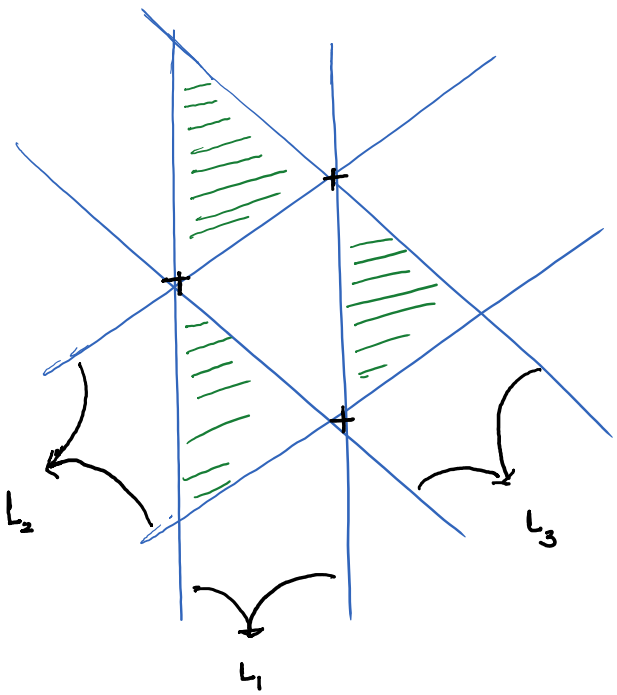
Let us try to build it with this configuration

Start by taking 3 +ve points.

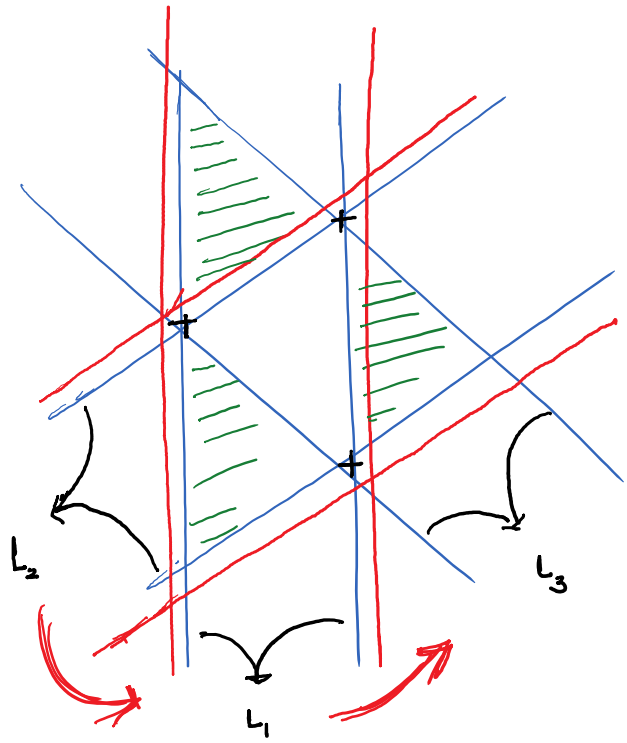
↳ To build a convex polygon of 6 points.

Let us determine region of space where these points

can be placed such that CONVEXITY is not disturbed



Convexity of polygon is preserved only if other three points are placed in the following GREEN regions



Now, For any parallel lines to include all the positive points. they include a complete GREEN region inside them (which corresponds to a (-ve))

Parallel lines can be in range from [slope of L_2 to slope of L_3]

with distance b/w them > [distance b/w two points] i.e

$VC(H) = 5$

RED LINES can describe set of parallel lines including all the points

Thus the configuration of alternating +ve, -ve is not shattcrable.

SAMPLE COMPLEXITY :

Accuracy of 0.8

Probability at least 0.95

$\Rightarrow \delta = 0.05, \epsilon = 0.2, VC(H) = 5$

No. of samples (M):

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

$$\Rightarrow M \geq \frac{1}{0.2}\left(4 \ln \frac{2}{0.05} + (8 \cdot 5) \ln \frac{13}{0.2}\right)$$

$$\geq 5\left(4 \ln (40) + 40 \ln \left(65\right)\right)$$

$$\geq 5\left(14.75 + 166.975\right) \geq 5\,(181.72)$$

$M \geq 908.6 \rightarrow$ **M = 909**

2) H is the hypothesis space of intervals

All points inside the interval are put in as '+' and outside as '−'

⇒ VC for a single interval

3 points cannot be shattered as $+ - +$ is not shattarable

Hypothesis space [BOOSTED] → $H' = \{ f \mid f(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x)) \}$

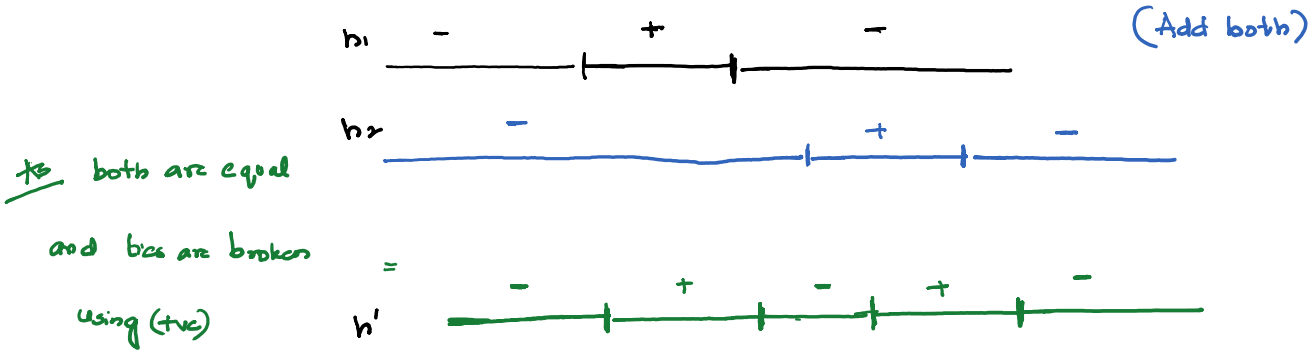
for some $h_1, h_2 \in H$ & $\alpha_1, \alpha_2 \in \mathbb{R}$

To break ties i.e if $\alpha_1 h_1(x) + \alpha_2 h_2(x) = 0$ → Hypothesis should return (+)

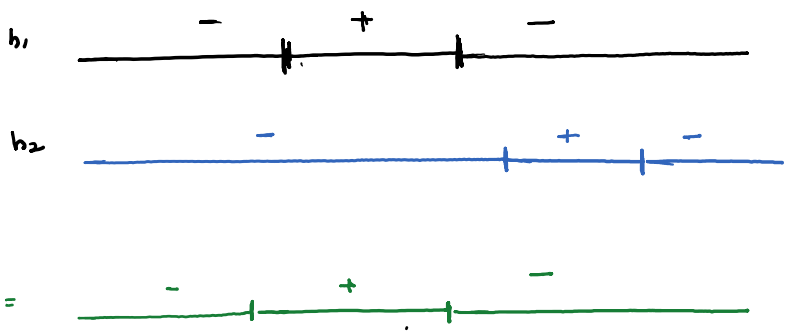
Configurations which can be producable with this hypothesis space :

i) if h_1, h_2 are non intersecting

ii) if $\alpha_1 = \alpha_2$



ii) if $\alpha_1 > \alpha_2$ (without loss of generality)



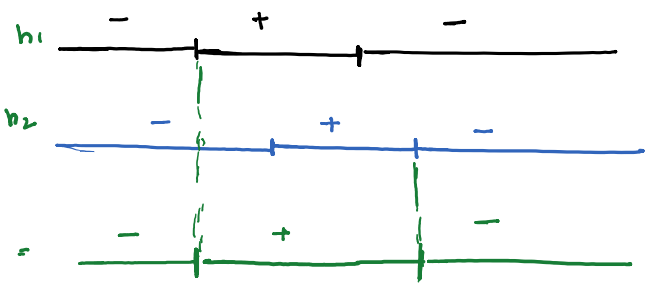
2) If h_1, h_2 are intersecting (overlapping)

i) $\alpha_1 = \alpha_2$

$+h_1 - h_2 = (+ve)$

$-h_1 + h_2 = (+ve)$

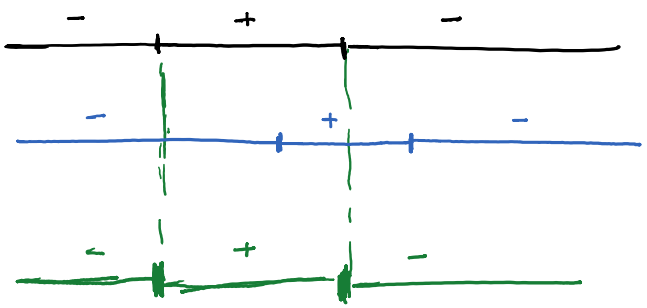
Breaking this



ii) $\alpha_1 > \alpha_2$ (without loss of generality)

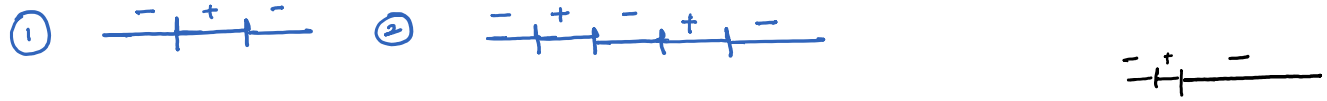
$+h_1 - h_2 > 0$ +ve

$-h_1 + h_2 < 0$ -ve



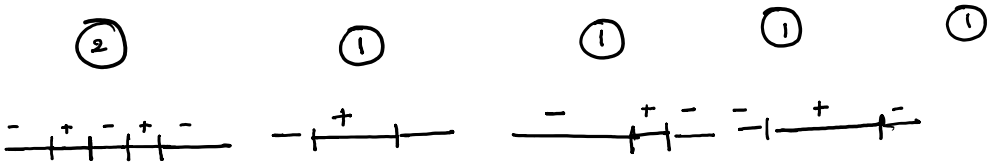
So, configurations possible : $(-,+,-)$, $(-,+,-,+,-)$

① ②

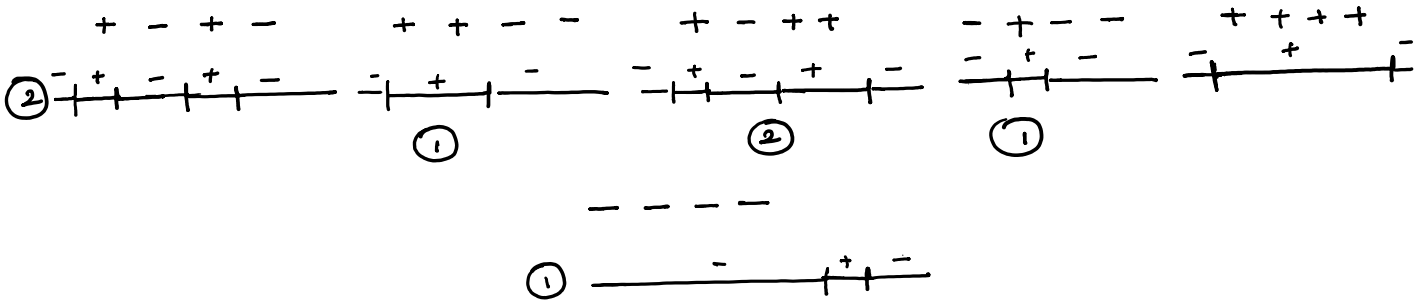


For 8 points → + − + , + + − , − − + , + + + , − − −

Using configurations possible



for 4 points :



For 5 points : Configuration : + − + − +

is not shatterable :

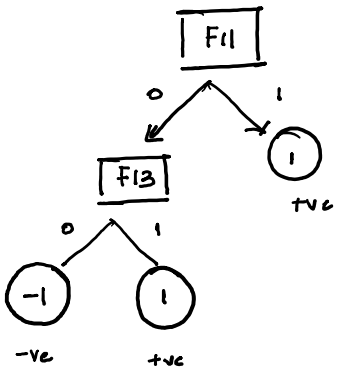
So, $\boxed{VC(H) = 4}$

PROBLEM 2 MEDICAL DIAGNOSTICS

1) (a) Adaboost for 5 rounds, trees and corresponding error, alpha values

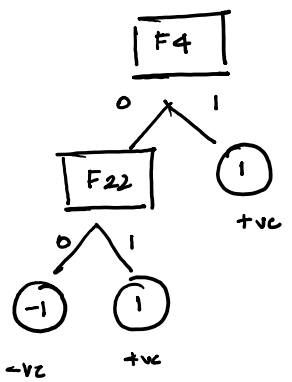
ITERATION 1

ERROR 0.21518987341772153
ALPHA 0.6469605204944376
ACCURACY ON TRAINING SET [78.48101266]
ACCURACY ON TEST SET : [71.50537634]



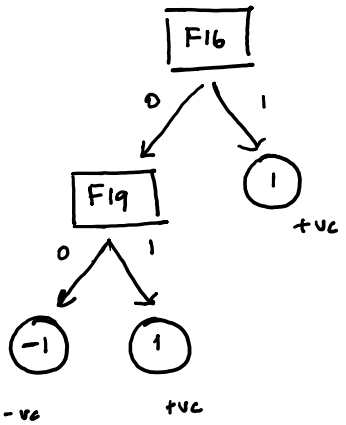
ITERATION 2

ERROR 0.32163187855787473
ALPHA 0.3731412176567788
ACCURACY ON TRAINING SET [78.48101266]
ACCURACY ON TEST SET : [71.50537634]



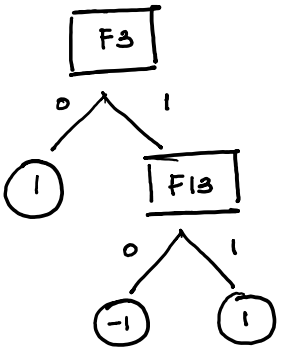
ITERATION 3

ERROR 0.36383852136064526
ALPHA 0.2793711408869234
ACCURACY ON TRAINING SET [81.01265823]
ACCURACY ON TEST SET : [66.12903226]



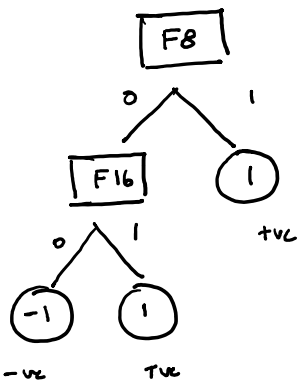
ITERATION 4

ERROR 0.32858000310790286
ALPHA 0.3573072595191374
ACCURACY ON TRAINING SET [79.74683544]
ACCURACY ON TEST SET : [74.19354839]



ITERATION 5

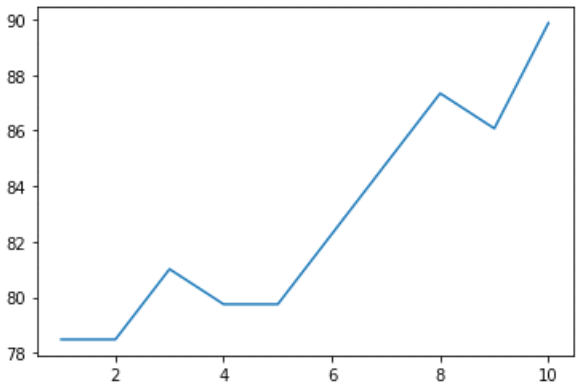
ERROR 0.3552761689793555
ALPHA 0.2979633141608934
ACCURACY ON TRAINING SET [79.74683544]
ACCURACY ON TEST SET : [74.7311828]



1) (b) Adaboost for 10 rounds, plots of training and test datasets accuracies

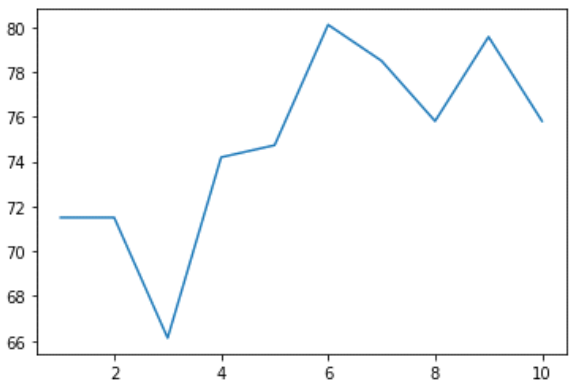
TRAINING ACCURACIES

[78.481, 78.481, 81.013, 79.747, 79.747, 82.278, 84.81 , 87.342, 86.076, 89.873]

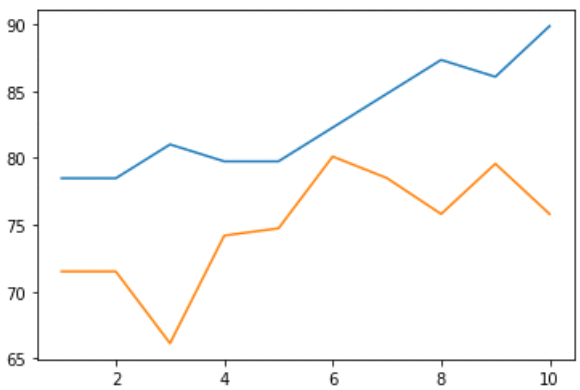


TEST ACCURACIES

[71.505, 71.505, 66.129, 74.194, 74.731, 80.108, 78.495, 75.806, 79.57 , 75.806]



COMBINED :



Blue: TRAINING
Orange : TEST

2) (a) Using coordinate descent to minimize the exponential loss function

HYPOTHESIS SPACE :
Here the total hypothesis space consists of 88 hypothesis (4 * 22)

But (+,+) , (-,-) for each of the attribute is same as ALL POSITIVE and ALL NEGATIVE
So, 44 of them can be replaced with 2

Thus, 46 -> [All positive, All negative, 2*22 (attributes)] - Total 46

INITIALIZATION :
[0, 0, 0, 0 ……0] of length 46

STRATEGY :
Performing coordinate descent in a round robin way, taking norm of alpha difference 0.0001

OPTIMAL VALUES OF ALPHA :

[-7.572, -0.013, 2.21 , 0. , 0.487, 0. , 5.279, 0. , -0.811, 0. , -0.316, 0. , -0.131, 0. , -0.6 , 0. , -4.888, 0. , 3.589, 0. , -2.079, 0. , -0.392, 0. , 0.447, 0. , -0.794, 0. , -3.06 , 0. , -1.192, 0. , -0.744, 0. , -2.952, 0. , -1.898, 0. , -0.05 , 0. , -0.321, 0. , -0.159, 0. , -0.377, 0.]

EXPONENTIAL LOSS :

np.exp(-1*Y*Y_pred).sum() = **39.509027085598426**

(b) **ACCURACY :**

ACCURACY ON TRAINING DATA : [83.5443038]
ACCURACY ON TEST DATA : **[68.8172043]**

(c) Adaboost for 20 rounds

ATTR : 0 F13
ATTR : 1 F11
ATTR : 2 POSITIVE
ATTR : 3 F7
ATTR : 4 F22
ATTR : 5 POSITIVE
ATTR : 6 F8
ATTR : 7 F3
ATTR : 8 F16
ATTR : 9 POSITIVE
ATTR : 10 F20
ATTR : 11 POSITIVE
ATTR : 12 F11
ATTR : 13 POSITIVE
ATTR : 14 F8
ATTR : 15 F3
ATTR : 16 F22
ATTR : 17 F8
ATTR : 18 POSITIVE
ATTR : 19 F17

ACCURACY ON TRAINING SET [82.27848101]
ACCURACY ON TEST SET : **[68.8172043]**

OBSERVATIONS ON ALPHAS

[0.476, 0.245, 0.304, 0.275, 0.13 , 0.183, 0.19 , 0.177, 0.167, 0.143, 0.16 , 0.098, 0.13 , 0.083, 0.117, 0.118, 0.131, 0.069, 0.107, 0.106]

Alphas found from both the algorithms are very different.
Alphas calculated from adaboost **doesn't have negative values.**
But, there are negative values in alphas calculated from coordinate descent.
Coordinate descent also has many **zero alphas** compared to none in adaboost 20.

(d) Bagging 20 samples to produce an average classifier

BAGGED SAMPLE : 1 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 2 , SPLIT ATTRIBUTE : F22
BAGGED SAMPLE : 3 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 4 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 5 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 6 , SPLIT ATTRIBUTE : F16
BAGGED SAMPLE : 7 , SPLIT ATTRIBUTE : F11
BAGGED SAMPLE : 8 , SPLIT ATTRIBUTE : F16
BAGGED SAMPLE : 9 , SPLIT ATTRIBUTE : F8
BAGGED SAMPLE : 10 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 11 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 12 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 13 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 14 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 15 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 16 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 17 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 18 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 19 , SPLIT ATTRIBUTE : F13
BAGGED SAMPLE : 20 , SPLIT ATTRIBUTE : F13

ACCURACY ON TRAINING SET [72.15189873]
ACCURACY ON TEST SET : **[61.82795699]**

Bagged average classifier **performs poorly on test set** compared to the other methods

(d) Following are the accuracies on training and test sets for three methods

COORDINATE DESCENT :

Training accuracy : [83.5443038]
Test accuracy : **[68.8172043]**

For alpha difference norm computational cost using round robin : 6+ lakh iterations

ADABOOST :

@20 rounds
Training accuracy : [82.27848101]
Test accuracy : **[68.8172043]**

@50 rounds
Training accuracy : [83.5443038]
Test accuracy : **[72.58064516]**

@100 rounds
Training accuracy : [84.81012658]
Test accuracy : **[70.43010753]**

BAGGING :

Training accuracy : [72.15189873]
Test accuracy : **[61.82795699]**

For this dataset, **adaboost** should be preferred, as it produced same accuracy as coordinate descent on the test set with much less computational cost. (adaboost - 20 rounds)

If adaboost is run for few more iterations like 50, it produced better results on the test set.