HW 6

YI NIAN
MPCS Machine learning, University of Chicago

5-25

Problem 1:

a:

Previously, we required that $(W^T x+b)y \geq 1$. In this case, no points will be allowed between the lines: $(W^T x+b)y = 1$ and $(W^T x + b)y = -1$. Now we do not have this strict assumption. Instead, we have a regularization term that tries to minimize the distance if there is any points between two lines.

b:

What we used for regularization:

$C \sum max(0, 1 - y(wx + b))$

What sklearn uses for regularization:

$C \sum(\epsilon)$

where $y(wx + b) \geq 1 - \epsilon$

This is equivalent to $\epsilon \geq 1 - y(wx + b)$

Since sklearn also has another restriction that $\epsilon \geq 0$ for any arbitrary ϵ. So the regularization is actually equivalent to $C \sum max(0, 1 - y(wx + b))$

i.e when y(wx+b) > 1, ϵ = 0 , and when y(wx+b) ≤ 1, $\epsilon \geq 1 - y(wx + b)$, but since we are focusing on the minimum of the regularization, we can take $\epsilon = 1 - y(wx + b)$, and therefore equivalent to our regularization term again.

c:

By observing the loss function, we know that the loss = 0 when yf(x) ≥ 1, so we just need to consider the case when yf(x) < 1.(one of the three cases)

Assume that $P(Y = 1|x) = p$ for simplicity, then we have the expected loss as :

(1-f(x))*p + (1+f(x))*(1-p) = (1-2p)f(x) + 1

which is the loss that we want to minimize.

Since argmin(1-2p)f(x) + 1 = argmin(1-2p)f(x)

and f(x) can only predict 1 or -1, we get

if (1-2p) $>$ 0, f(x) = -1

if (1-2p) $\leq$ 0, f(x) = 1

which is the anwer we want.

Problem 2:

The feature vector is :

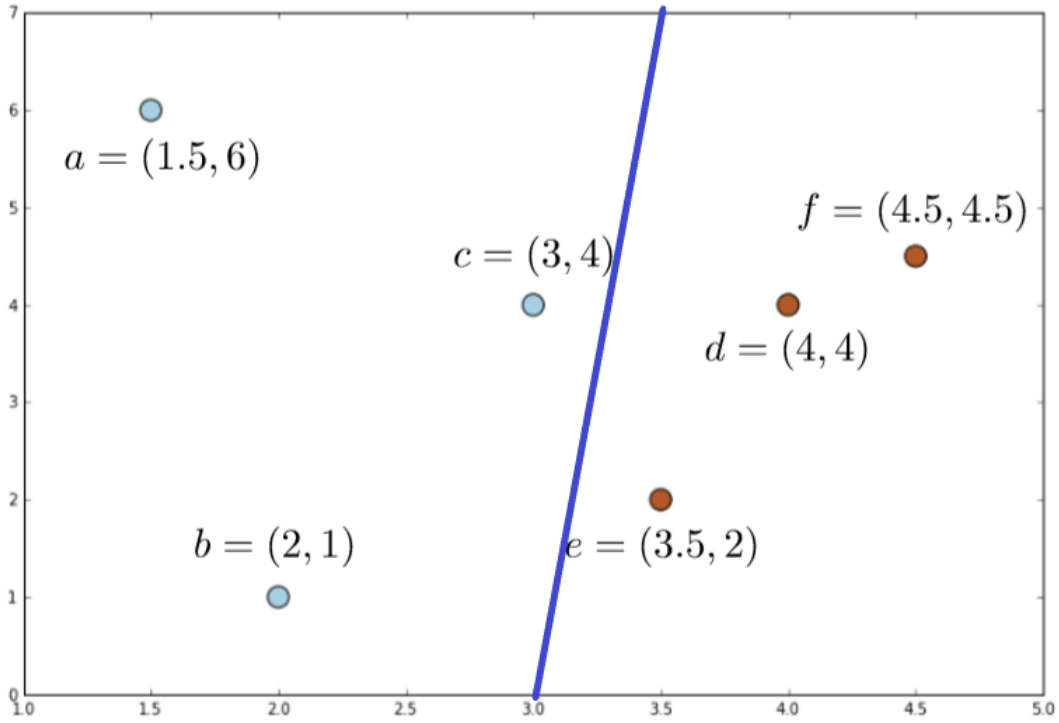
$\phi(x)$ = $[x_1^3, x_2^3, x_1^2x_2, x_2^2x_1, x_1^2, x_2^2, x_1x_2, x_1, x_2]$

Its corresponding kernel function will be:

k(x,y) = $< \phi(x), \phi(y) >$ = $[x_1^3y_1^3, x_2^3y_2^3, x_1^2x_2y_1^2y_2, x_2^2x_1y_2^2y_1, x_1^2y_1^2, y_2^2x_2^2, x_1x_2y_1y_2, x_1y_1, x_2y_2]$ = $\phi(x^Ty)$

Problem 3:

a:

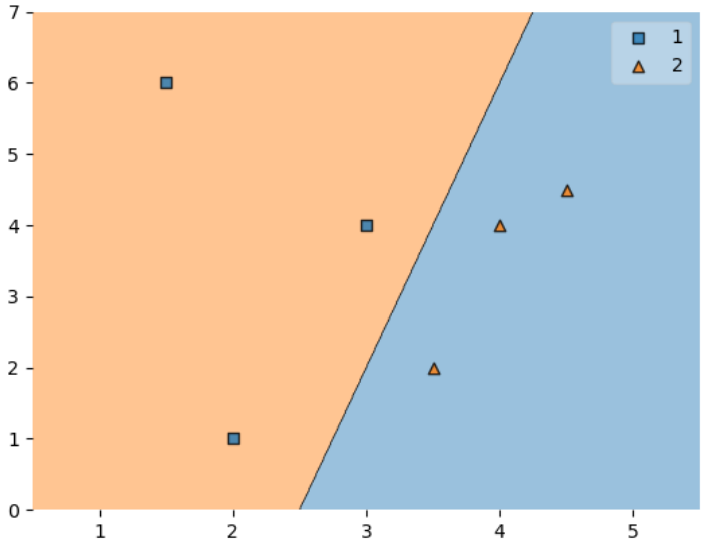


In this case the line function is y = wx+b = 14x-42

and the closest distance will be point c(3,4), which has distance = $|4 - 14 * 3 + 42|/\sqrt{14^2 + 1}$ = 0.285

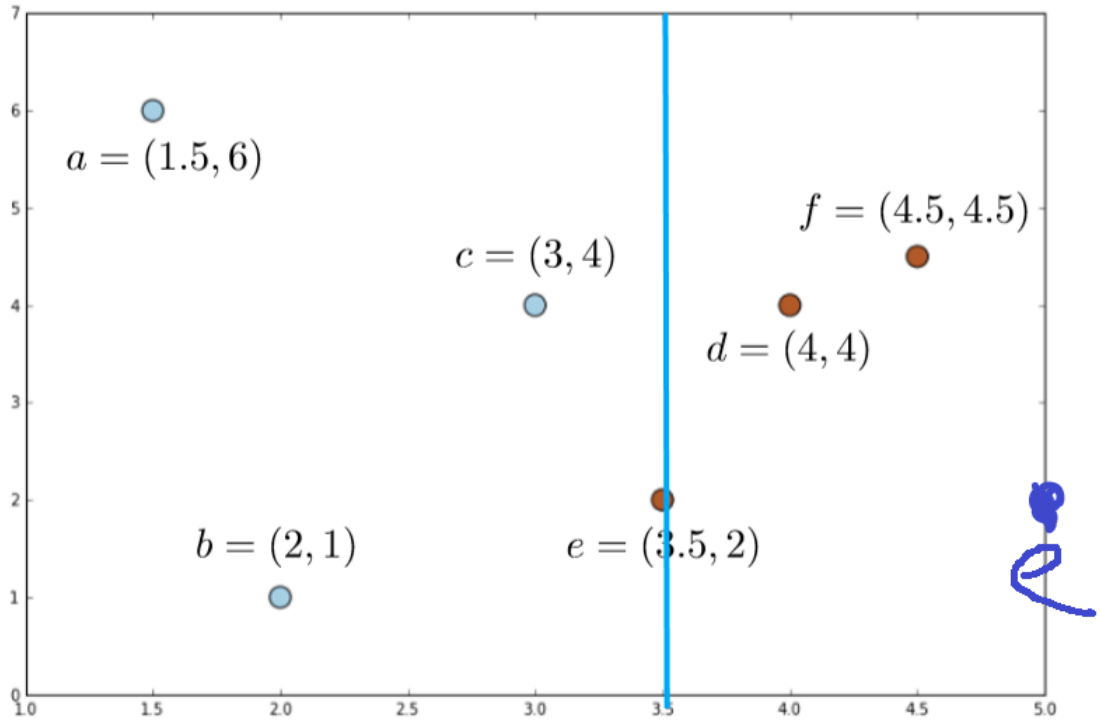
b:

see the code,result from sklearn



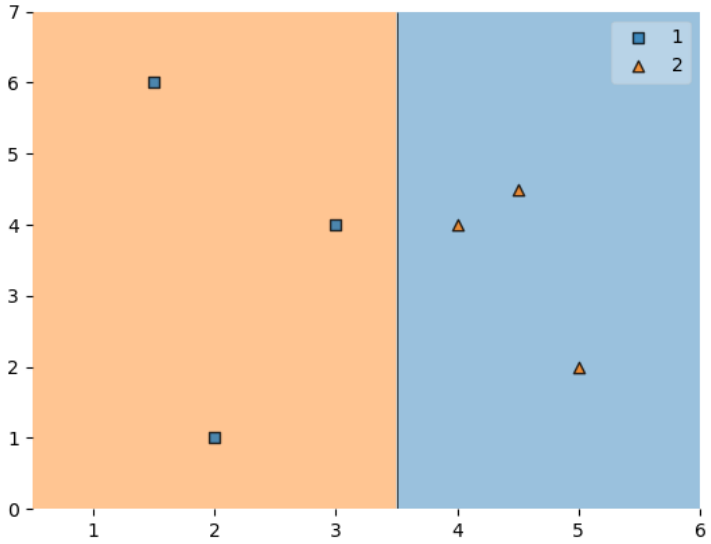
the closest point is: [3. 4.]with distance : 0.48503833264255575

c:



The closest distance from the graph is c or d, with distance 0.5.

Result from Sklearn

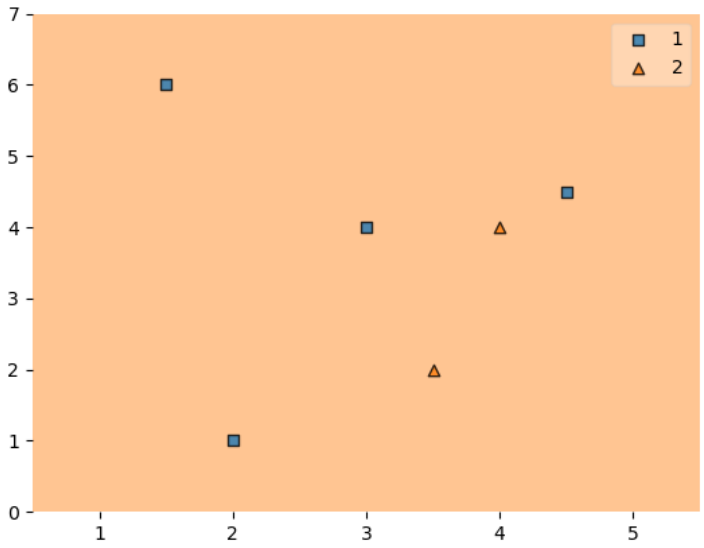


the closest point is: [3. 4.]with distance : 0.5

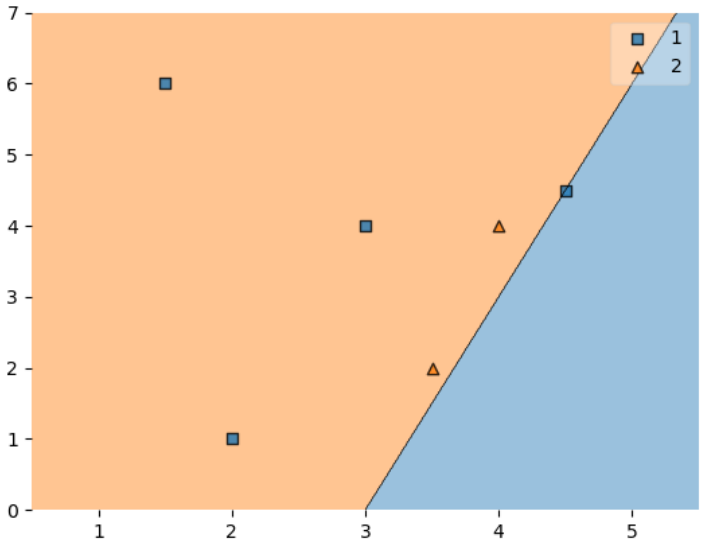
d:

C=0.1

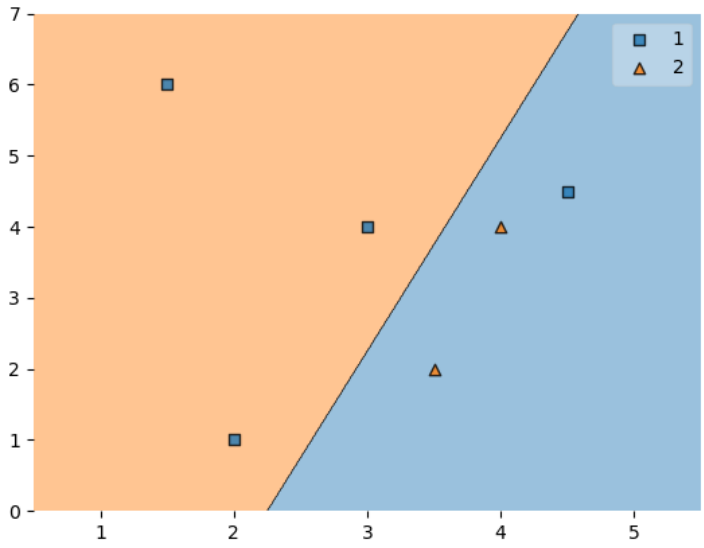
All the points are classified into single class:



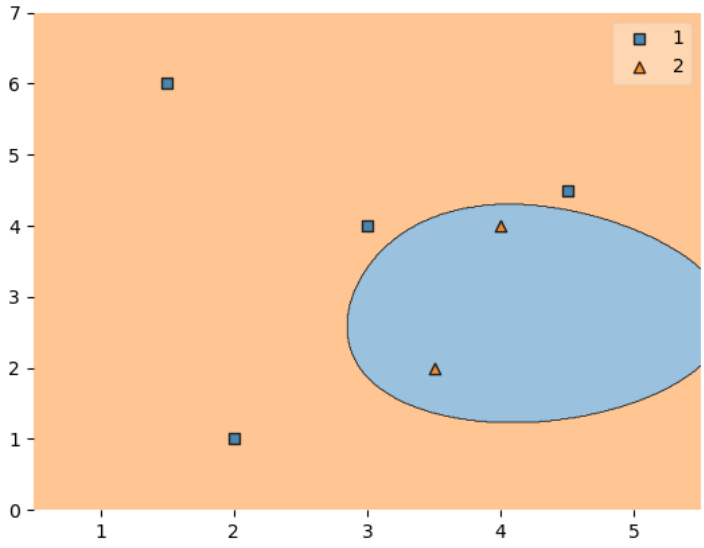
C=1



C=10



e



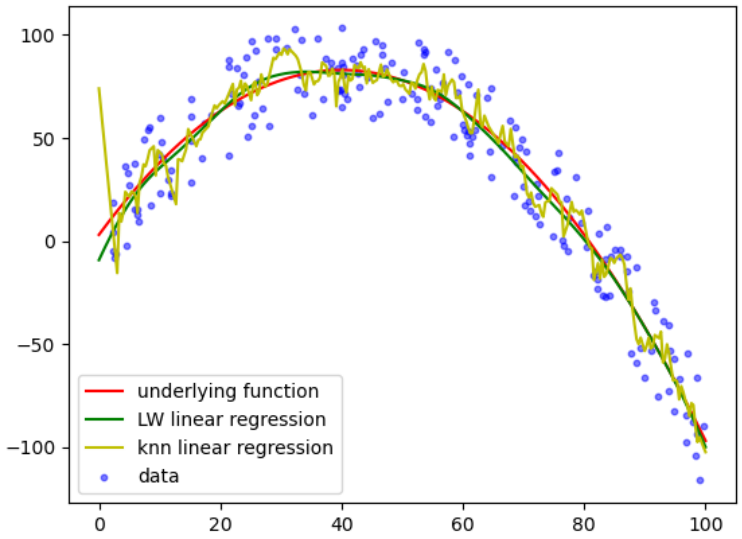
Problem 4:

a

See the code

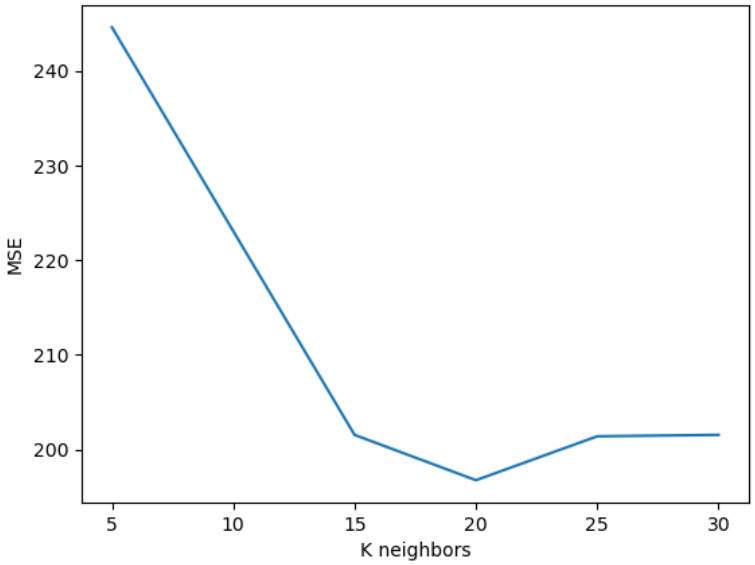
b

See the graph:

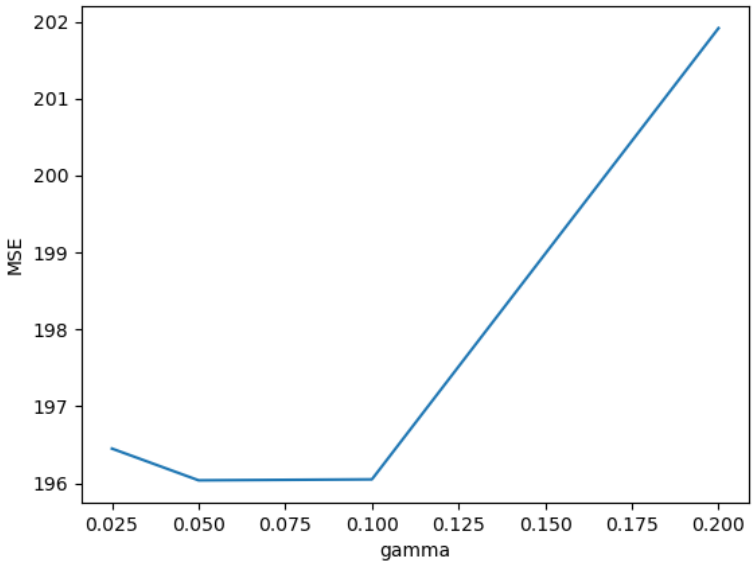


c

Tuning K:



Tuning gamma:



For tuning K: the best MSE is : 196.76573216829104 with neighbors number: 20

For tuning gamma: the best MSE is : 196.0381904934169 with gamma: 0.05

Since there is randomness in the dataset, it is hard to say which model we should use. In my case(seed = 1), the best model for these two have roughly the same MSE.

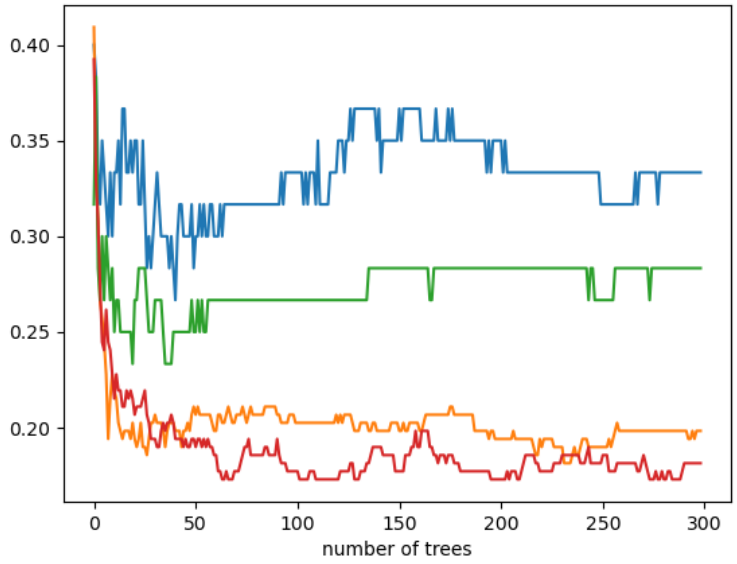
Problem 5:

It could be possible the error is greater than ϵ. For example, we have 3 hypothesis with each has error rate 2/5. And we use the voting method, and we only vote example 1 and 5 correctly. So the error rate is 4/5 which is greater than 3/5 here.

	example1	example2	example3	example4	example5
H1	T	T	F	F	T
H2	T	F	T	F	T
H3	T	F	F	T	T

Problem 6:

See the code and the attached graph:



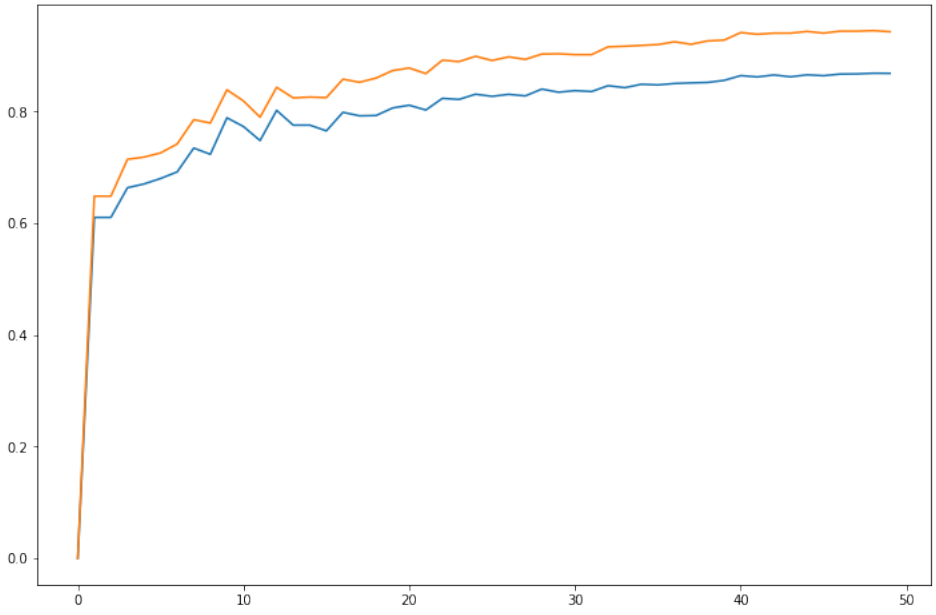
Problem 7:

a

See the code

b

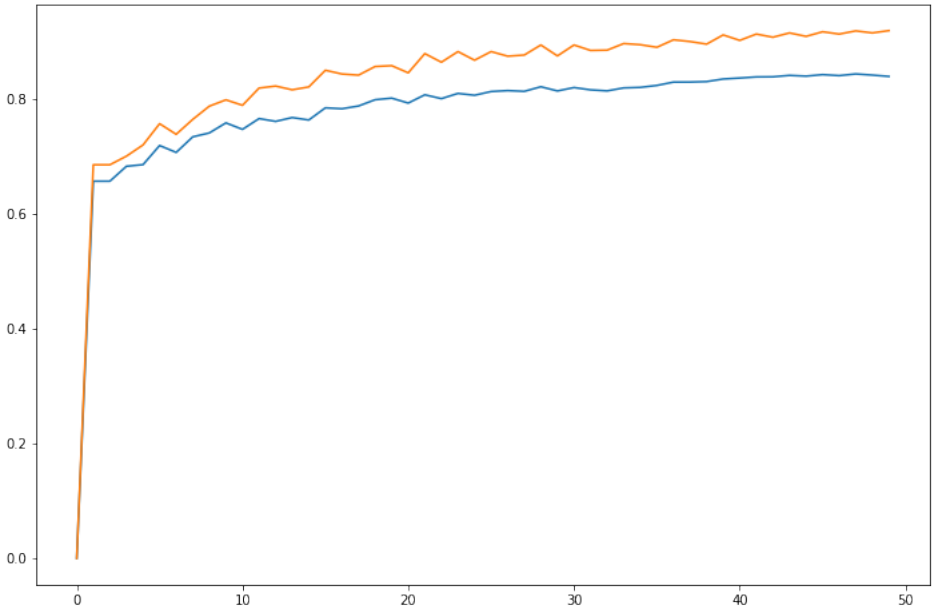
The graph is actually the accuracy of the AdaBoost:



c

I tried the number of iterations/trees from 100 to 700, with each tree of max depth of 3 and it seems the test error still don't increase(here is the accuracy won't decrease). It is hard to overfit with this algorithm especially when the weak learner is simple model.

d



The accuracy increase faster than the previous case as the number of trees increase.