Machine Learning from Data  
Assignment 1

Jae Park (661994900)

Exercise 1.3

(a)

From t = 0,1,2, … , if (xT, yT) is misclassified by w(t), we have y(t) ≠ sign(wT(t)x(t)). (pg. 7)

So, y(t)wT(t)x(t) = some positive num \* sone negative num = < 0

(b)

*y(t)wT(t+1)x(t) > y(t)wT(t)x(t)*

*y(t)wT(t+1)x(t) – y(t)wT(t)x(t) > 0*

*y(t) (wT(t+1) – wT(t)) x(t) > 0*

*y(t) (w(t+1) - w(t))T x(t) > 0*

*y(t) (w(t) + y(t)x(t) – w(t))T x(t) > 0 … (1.3)*

*y(t) (y(t)x(t))T x(t) > 0*

*y(t) y(t) (x(t))T x(t) > 0*

Since x = [x0, x1, x2, … ]T, xT(t)x(t) = ||*x(t)*||2 will yield some positive scalar value.

And, y(t)y(t) = square of any number > 0.

So the inequality y(t)y(t) (x(t))Tx(t) > 0, and the initial equality, hold true.

(c)

Regardless of what the value or sign of y(t)wT(t)x(t) was, now with the updated weight vector w(t+1), y(t)wT(t+1)x(t) is now greater than or “more positive” than y(t)wT(t)x(t), i.e., it moves in the right direction.

Also, what we did above in 1.3 (b) to show that the inequality holds true implies the answer for this question.

Exercise 1.5

(a) Learning approach

(b) Design

(c) Learning approach

(d) Design

(e) Learning approach

Exercise 1.6

(a)

Supervised learning. Using the user-specific log and cookie data, we recommend books based on interests in book categories, authors, etc.

(b)

Reinforcement learning or Supervised learning. For Reinforcement learning, you aggregate performance data based on each move a player/machine could make (generative adversarial network, AlphaGo); For Supervised learning, you take previous played game logs as inputs and train a model.

(c)

Unsupervised learning. Training data would be a list of bunch of movies.

(d)

Supervised learning (and Reinforcement learning). For Supervised learning, the training data will be the music note sheets. For Reinforcement learning, the training data will be the music generated from the previous Supervised learning methods and data based on which music humans prefer better (a measure of how good the output is).

(e)

Supervised learning. The training data would be the information on the past customers: assets, credit history, repayment rate, income, etc.

Exercise1.7

(a)

Hypothesis that always return •(+1): one function agrees on all three points (f8), three fn’s agree on two points (f4, f6, f7), three fn’s agree on one point (f2, f3, f5), one fn agrees on zero points (f1).

Hypothesis that always return (-1): one fn agrees on all three points (f1), three fn’s agree on two points (f2, f3, f5), three fn’s agree on one point (f4, f6, f7), one fn agrees on zero points (f8).

Performance-wise, the two hypotheses do the same (1:3:3:1).

(b)

Hypothesis that always return •(+1): one function agrees the least on all three points(f1 has the least agreement with g), three fn’s agree the least on two points (f4, f6, f7), three fn’s agree the least on one point (f2, f3, f5), one fn agree the least on zero points (f8).

Hypothesis that always return (-1): one fn agrees the least on all three points (f8), three fn’s agree on two points (f4, f6, f7), three fn’s agree on one point (f2, f3, f5), one fn agrees on zero points (f1).

Performance-wise, the two hypotheses do the same (1:3:3:1).

(c)

one function agrees on all three points (f2), three fn’s agree on two points (f1, f4, f6), three fn’s agree on one point (f3, f5, f8), one fn agrees on zero points (f7).

(d)

For all distinct boolean functions (there are 22 ^ 3 = 256 distinct Boolean functions on 3 Boolean inputs and one Boolean output function, pg. 16), they will have the same performances; one f will agree on all three points, three f’s will agree on two points, three f’s will agree on one point, and one f will agree on zero points. (1:3:3:1)

Problem 1.1

P(second ball from the same bag is black | first ball is black)

= P(second ball from the same bag is black AND first ball is black) / P(first ball is black)

P(second ball from the same bag is black AND first ball is black) = 0.5, because if you picked a black and white bag in the first place, it’s not possible that the second ball is black.

P(first ball is black) = pick the black and black bag, where all the balls are black (0.5) + pick the black and white bag then pick a black ball from within (0.5 \* 0.5 = 0.25) = 0.75

0.5 / 0.75 = 2/3

Problem 1.2

(a)

wTx = w0 + w1x1 + w2x2 = k for some arbitrary number k.

Now, w1x1 + w2x2 = k - w0, but we will reduce and make k – w0 = w0, as w0 itself is also some arbitrary number.

w2x2 = -w1x1 + w0

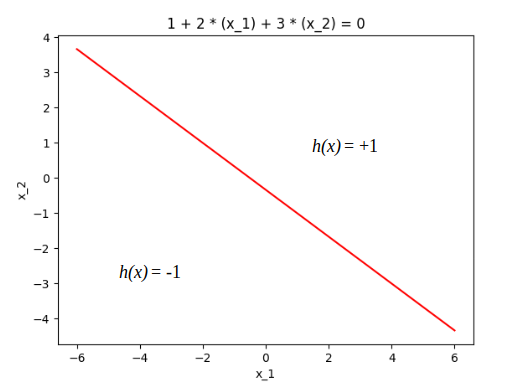
x2 = (-w1/w2)x1 + (w0/w2)

So, a = -w1/w2, b = w0/w2 (or, b= -w0/w2 if the replacement of k – w0 with w0 didn’t take place earlier as it maintains the sign of w0)

(b)

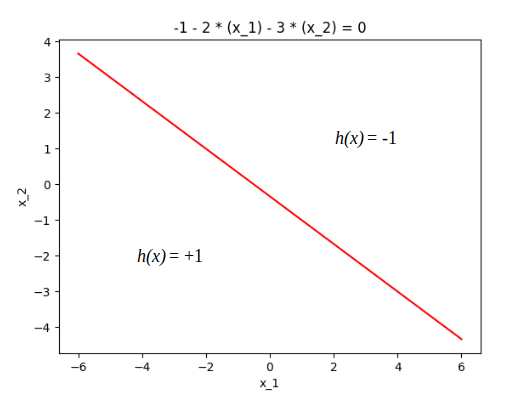
w = [1, 2, 3]T

wTx = 1 + 2x1 + 3x2 = k = 0 (assume k was 0 in this step of PLA iteration)



where *h*(x) = sign(wTx)

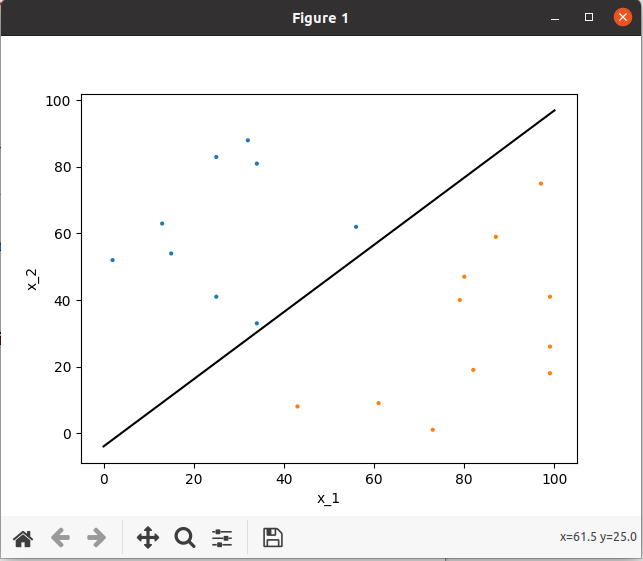
wTx = -1 - 2x1 - 3x2 = k = 0 (assume k was 0 in this step of PLA iteration)



where *h*(x) = sign(wTx)

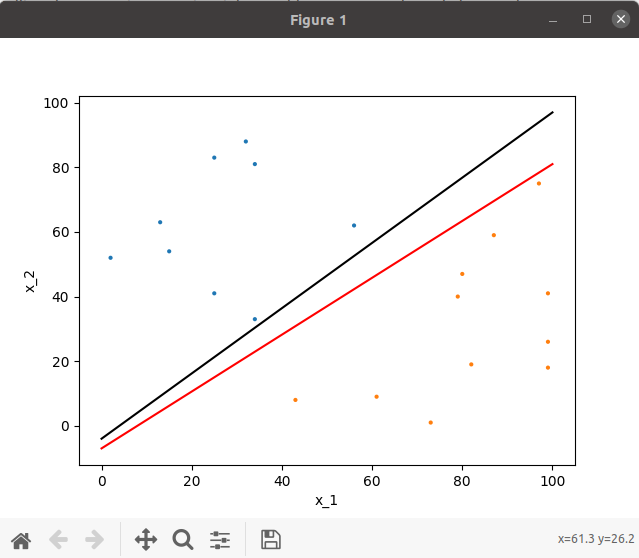
Problem 1.4

(a) data set of size 20 with an arbitrary target function in black.

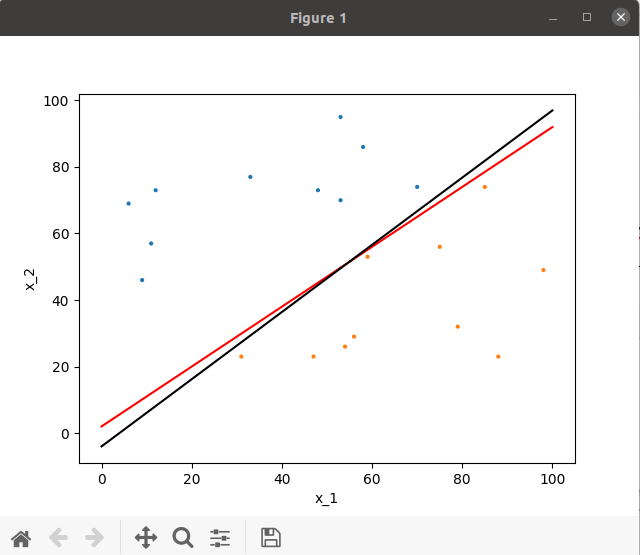


(b)

PLA took 17 iterations to converge. The target function is in black, and the hypothesis function is in red. f and g are pretty close.

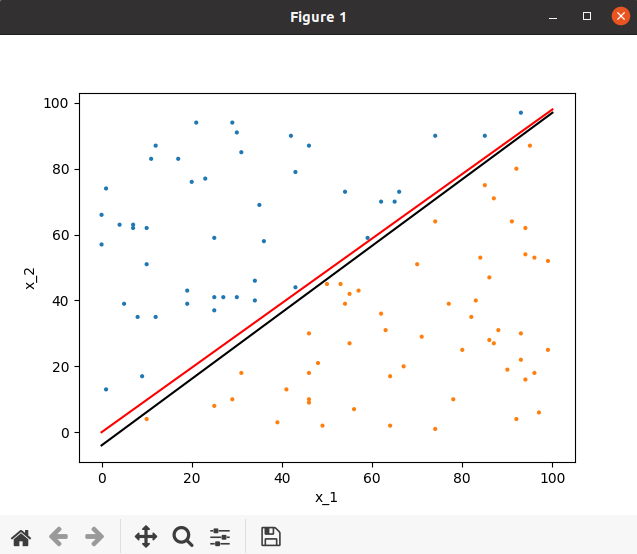


(c) PLA took 11 iterations to converge. The target function is in black, and the hypothesis function is in red. f and g are pretty close to each other, closer than before.



(d)

PLA took 192 interations. The functions seems a bit closer.



(e)

PLA took 1453 interations. With more data, now there is virtually no difference between f and g.

