Machine Learning From Data HW1

Shane O'Brien

September 2017

Exercise 1.3

a

We know that $y(t), \mathbf{x}(t)$ is misclassified. This tells us, be definition, that $y(t) \neq sign(\mathbf{w}^T\mathbf{x}(t))$. So if the signs are not matching, the result will be a negative number. This confirms that $y(t)\mathbf{w}^T(t)\mathbf{x}(t) < 0$.

b

We need to show that $y(t)\mathbf{w}^T(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^T(t)\mathbf{x}(t)$. As shown in the last problem, $y(t)\mathbf{w}^T(t)\mathbf{x}(t) < 0$. The learning algorithm, which be definition is always correct, gives us $y(t)\mathbf{w}^T(t+1)\mathbf{x}(t)$. Since $y(t)\mathbf{w}^T(t+1)\mathbf{x}(t)$ is correct, $y(t)$ and $\mathbf{w}^T(t+1)\mathbf{x}(t)$ are the same sign. This means $y(t)\mathbf{w}^T(t+1)\mathbf{x}(t)$ is positive, which concludes that $y(t)\mathbf{w}^T(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^T(t)\mathbf{x}(t)$.

c

We can argue that the move from $\mathbf{w}(t)$ to $\mathbf{w}(t+1)$ is a move in the 'right direction' by keeping in mind what $y(t)$ is. $y(t)$ is the correct output of the data inputs $\mathbf{x}(t)$. Since what we're doing is adding a vector to $\mathbf{w}(t)$, we just need to know if that vector is the 'right direction'. If the correct result, given by $y(t)$, is +1, the new vector $\mathbf{w}(t+1)$ will be made by adding a positive vector. If the correct result, given by $y(t)$, is -1, the new vector $\mathbf{w}(t+1)$ will be made by adding a negative vector.

Exercise 1.5

a

Learning approach. This is because you don't know the age that someone needs the test. Even though you know what the test searches for, the test wasn't designed for a certain age group.

b

Design approach. This is because it is known exactly what a prime and non-prime are. These specifications have already been given to you.

c
Learning approach. This is because there is no clear specification for what charges are going to be frauds.

d
Design approach. The specifications for gravity, air resistance, etc, are all already defined.

e
Leaning approach. There are no clear specifications that define when to turn a light green or red. This must be defined through learning.

Exercise 1.6

a
Supervised (Reinforcement Possible). This is similar the one of the first examples from class regarding movies on Netflix. The input data, x, is all the information about the book. The output data, y, is whether or not the movie is recommended. This could also be reinforcement learning if the output is now this: one part for whether or not the book is recommended, and another part if the user actually clicked on the recommended book

b
Reinforcement. This is similar to the backgammon example. The learning in this case would be picking a reasonable move, and then reporting how it went. Our data would be the previous moves and state of the game, our output would be a move, and our result would be if we won the game in the end.

c
Unsupervised. There are no defined groups or categories for the movies. Our task is to just separate them. This is very similar to the coin cluster example. Our data is a bunch of details about the movie, with no outputs.

d
Unsupervised. In this example, we don't have a defined goal or specifications to reach. The machine can listen to music, but there is no explicit meaning or 'correct answer' behind it. Our data is a mass of songs and notes.

e
Supervised. This is very similar to the problem used in class. Our data set is all the information about the bank customer, x, and whether or not the bank made money on the bank customer, y.

Exercise 1.7

a

The g picked is the function that always returns a black dot.
$f_1 = 0$
$f_2 = 1$
$f_3 = 1$
$f_4 = 2$
$f_5 = 1$
$f_6 = 2$
$f_7 = 2$
$f_8 = 3$

b

The g picked is the function that always returns a white dot.
$f_1 = 3$
$f_2 = 2$
$f_3 = 2$
$f_4 = 1$
$f_5 = 2$
$f_6 = 1$
$f_7 = 1$
$f_8 = 0$

c

$f_1 = 2$
$f_2 = 3$
$f_3 = 1$
$f_4 = 1$
$f_5 = 1$
$f_6 = 2$
$f_7 = 0$
$f_8 = 1$

d

$f_1 = 1$
$f_2 = 0$
$f_3 = 2$
$f_4 = 2$
$f_5 = 2$
$f_6 = 1$
$f_7 = 3$
$f_8 = 2$

Problem 1.1

P[pick BB bag] = 0.5
P[pick BW bag] = 0.5

P[pick black first|pick BB bag] = 1.0
P[pick black first|pick BW bag] = 0.5

P[pick black first & pick BB bag]
= P[pick black first|pick BB bag] * P[pick BB bag]
$= 1.0 * 0.5$
$= 0.5$

P[pick black first & pick BW bag]
= P[pick black first|pick BW bag] * P[pick BW bag]
$= 0.5 * 0.5$
$= 0.25$

P[pick black first] = 0.75

P[pick BB bag|pick black first]
$= \frac{P[pickblackfirst\&pickBBbag]}{P[pickblackfirst]}$
$= \frac{0.5}{0.75}$
$= 0.666$

Problem 1.2

a
To generate an equation in the form $x_2 = ax_1 + b$:

$\mathbf{w}^T\mathbf{x} = \sum_{i=0}^{2} w_i x_i = w_0x_0 + w_1x_1 + w_2x_2 = 0$

Because x_0 is 1, this becomes:

$w_1x_1 + w_2x_2 = -w_0$

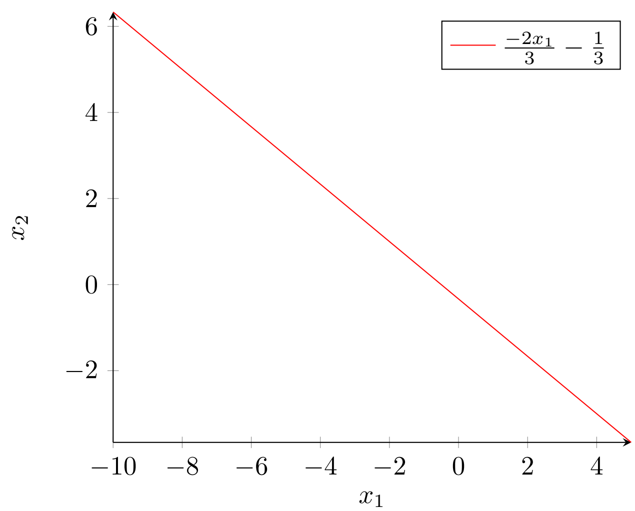
$w_2x_2 = -w_0 - w_1x_1$

$x_2 = \frac{-w_1}{w_2}x_1 - \frac{w_0}{w_2}$

b

There are two cases, one where $\mathbf{w}^T$ is $\begin{bmatrix}1 & 2 & 3\end{bmatrix}$ and one where $\mathbf{w}^T$ is $\begin{bmatrix}-1 & -2 & -3\end{bmatrix}$.

They both result in the same line, shown below.

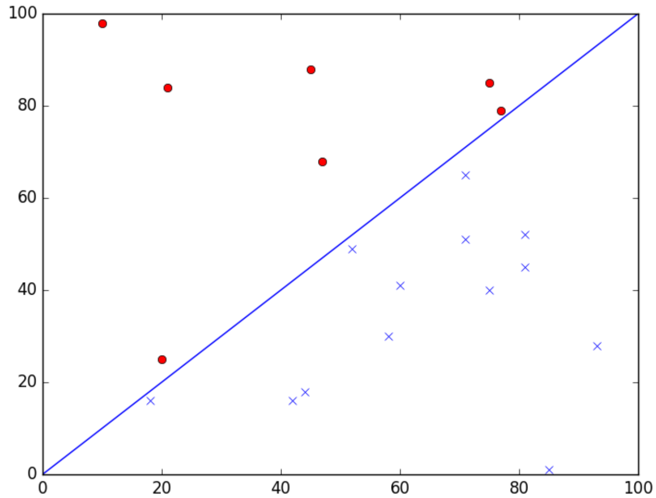


In the first $\mathbf{w}^T$, the correct region is above the line. In the second $\mathbf{w}^T$, the correct region is below the line.

Problem 1.4

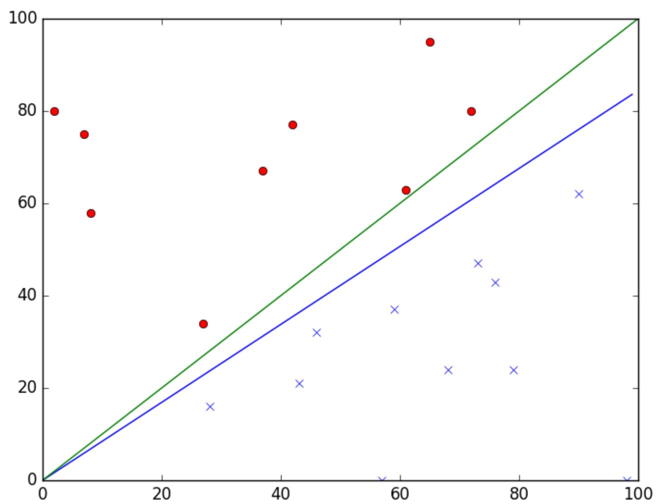
a

The f is the blue line



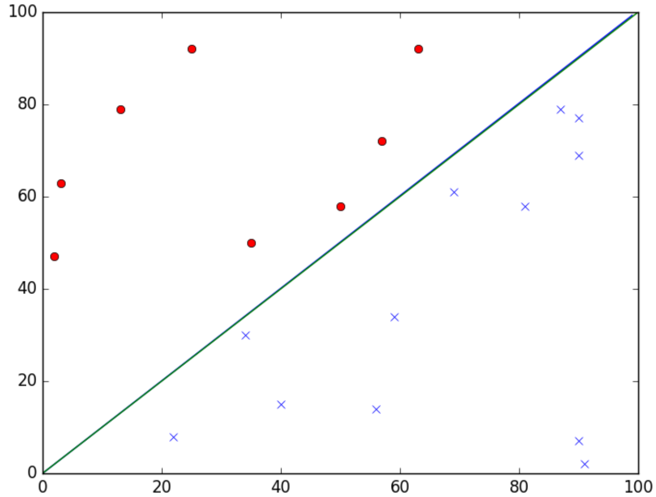
b

The f is the green line. The g that the algorithm found is the blue line. The algorithm found the g in 4 iterations. Although they are both correctly labeling the data, the g is clearly off from f



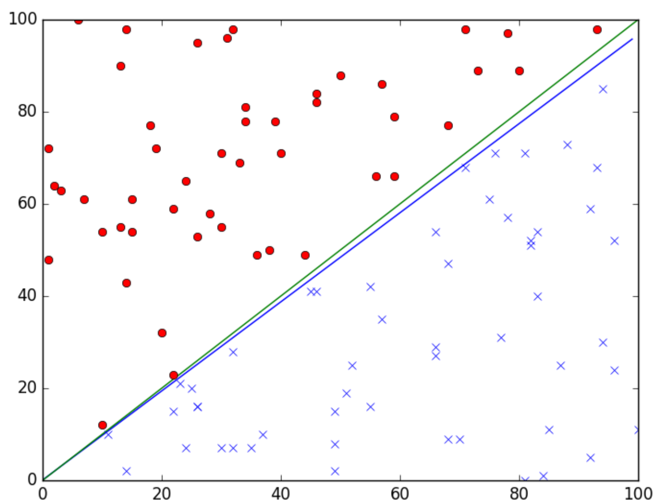
c

The f is the green line. The g that the algorithm found is the blue line. The algorithm found the g in 46 iterations. This time, the algorithm took much longer, but got a much closer result.



d

The f is the green line. The g that the algorithm found is the blue line. The algorithm found the g in 21 iterations. Surprisingly, this took fewer iterators than part c.



e

The f is the green line. The g that the algorithm found is the blue line. The algorithm found the g in 38 iterations. The amount of iterations doesn't seem to increase with the amount of data points.